UNIVERSITAT DE BARCELONA

Mastering the Triad of Data, Models and Tasks in Deep Learning for Image Understanding

Bhalaji Nagarajan

ADVERTIMENT. La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX (**www.tdx.cat**) i a través del Dipòsit Digital de la UB (**diposit.ub.edu**) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX ni al Dipòsit Digital de la UB. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX o al Dipòsit Digital de la UB (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

ADVERTENCIA. La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR (**www.tdx.cat**) y a través del Repositorio Digital de la UB (**diposit.ub.edu**) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR o al Repositorio Digital de la UB. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR o al Repositorio Digital de la UB (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

WARNING. On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX (**www.tdx.cat**) service and by the UB Digital Repository (**diposit.ub.edu**) has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized nor its spreading and availability from a site foreign to the TDX service or to the UB Digital Repository. Introducing its content in a window or frame foreign to the TDX service or to the UB Digital Repository is not authorized (framing). Those rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

UNIVERSITAT DE BARCELONA

DOCTORAL THESIS

Mastering the Triad of Data, Models and Tasks in Deep Learning for Image Understanding

Author:
Bhalaji NAGARAJAN

Supervisor:
Prof. Petia RADEVA
Co-Supervisor:
Dr. Eduardo AGUILAR

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Doctorat en Matemàtiques i Informàtica
Departament de Matemàtiques i Informàtica

UNIVERSITAT DE BARCELONA

May 3, 2024

Thesis by Compilation of Articles

UNIVERSITAT DE BARCELONA

Departament de Matemàtiques i Informàtica

Doctor of Philosophy

Mastering the Triad of Data, Models and Tasks in Deep Learning for Image Understanding

by Bhalaji NAGARAJAN

Abstract

Deep learning's rapid growth brings vast application potential across diverse domains. Achieving optimal performance hinges on a critical interplay between three key elements: powerful model architectures, vast amounts of data, and a deep understanding of the target domain. Each element presents unique challenges. This thesis tackles these challenges to unlock the full potential of models, exploring solutions for data, models, and task understanding.

The first part of this thesis tackles the fundamental challenges associated with data used in deep learning. Acquiring large-scale data is a significant challenge, often limited by factors like annotation costs and label errors. Data within a dataset frequently exhibits significant diversity. We address these challenges with a multifaceted approach. We investigate the development of noise-robust sample-selection-based deep learning models to handle the presence of label errors. To leverage the large volumes of unlabeled data available, we explore contrastive self-supervised learning strategies. To address the heterogeneity within datasets, we propose a sample importance strategy to prioritize samples that present learning challenges. These solutions address the various data-related challenges that hinder deep learning models. The second part of the thesis covers the critical role of understanding model behaviour. We use uncertainty quantification metrics to gain valuable insights into the capabilities of the models in making predictions. By understanding these metrics, we identify areas where the model's predictions might be less reliable. We extend our exploration by applying these uncertainty metrics across various tasks to improve the decision-making process of the models. The final part of this thesis explores the importance of task understanding. We utilize the challenging domain of food recognition as a case study. Food recognition presents unique challenges due to the visual complexity of food images. We address the domain-specific challenges of fine-grained and multi-label classification by strategically designing and modifying deep learning models to improve their performances.

Our research during this thesis yielded significant advancements in several key areas of model development. We achieved state-of-the-art results on several benchmarks across various tasks, demonstrating the effectiveness of our proposed solutions. This highlights the potential of our work to contribute to the broader field of deep learning.

UNIVERSITAT DE BARCELONA

Departament de Matemàtiques i Informàtica

Doctor of Philosophy

Mastering the Triad of Data, Models and Tasks in Deep Learning for Image Understanding

by Bhalaji NAGARAJAN

el Resumen

El rápido crecimiento del aprendizaje profundo ofrece un enorme potencial de aplicación en diversos ámbitos. El logro de un rendimiento óptimo depende de una interacción crítica entre tres elementos claves: potentes arquitecturas neuronales, grandes volúmenes de datos y un profundo conocimiento del dominio de destino. Cada elemento presenta retos únicos. Esta tesis aborda estos retos para liberar todo el potencial de los modelos de aprendizaje profundo, explorando soluciones que contemplen la comprensión de los datos, de los modelos y de las tareas.

La primera parte de esta tesis aborda los retos fundamentales asociados a los datos utilizados en el aprendizaje profundo. La adquisición de datos a gran escala es un reto importante, a menudo limitado por factores como los costes de anotación y los errores de etiquetado. Los datos dentro de un conjunto de datos a menudo presentan una diversidad significativa. Abordamos estos retos con un enfoque multifacético. Investigamos el desarrollo de modelos de aprendizaje profundo basados en la selección de muestras robustos al ruido para hacer frente a la presencia de errores en las etiquetas. Para aprovechar los grandes volúmenes de datos sin etiquetar disponibles, exploramos estrategias de aprendizaje autosupervisado contrastivo. Para abordar la heterogeneidad dentro de los conjuntos de datos, proponemos una estrategia de muestreo por importancia para priorizar las muestras que presentan desafíos de aprendizaje. Estas soluciones abordan los diversos retos relacionados con los datos que dificultan los modelos de aprendizaje profundo. La segunda parte de la tesis aborda el papel fundamental de la comprensión del comportamiento de los modelos. Utilizamos métricas de cuantificación de la incertidumbre para obtener información valiosa sobre las capacidades de los modelos a la hora de realizar predicciones. Al comprender estas métricas, identificamos las áreas en las que las predicciones del modelo podrían ser menos fiables. Ampliamos nuestra exploración aplicando estas métricas de incertidumbre a diversas tareas para mejorar el proceso de toma de decisiones de los modelos. La parte final de esta tesis explora la importancia de la comprensión de las tareas. Utilizamos el desafiante campo del reconocimiento de alimentos como caso de estudio. El reconocimiento de alimentos presenta retos únicos debido a la complejidad visual de las imágenes de alimentos. Abordamos los retos específicos del dominio de clasificación detallada y de clasificación de múltiples etiquetas, diseñando y modificando estratégicamente modelos de aprendizaje profundo para mejorar su rendimiento.

Nuestra investigación durante esta tesis ha dado lugar a avances significativos en varios aspectos claves del desarrollo de modelos. Hemos obtenido los mejores resultados del estado del arte en varias pruebas comparativas de diversas tareas, lo que demuestra la eficacia de las soluciones propuestas. Esto pone de relieve el potencial de nuestro trabajo para contribuir ampliamente al campo del aprendizaje profundo.

Acknowledgements

The life of PhD is akin to running a marathon !!!
I am grateful to have persevered through this journey, one filled with challenges and rewards with valuable knowledge and experiences. The success of this journey is both owed to and shared by the many hands that have supported me along the way.

First, my deepest gratitude to all my teachers, who have equipped me with the skills I possess today and will continue to shape my journey ahead. *I owe every syllable and every word of mine!!!!*

My sincere thanks to Petia for her unwavering support throughout this journey. Her guidance and expertise were instrumental in shaping my research and helping me achieve my academic goals. She is the source of incredible positive energy and she has nurtured every aspect of my well-being right from the day she incorporated me into the group. Proud and grateful to have her by my side!

A big thank you to Eduardo, who first started as my colleague, and now serves as my co-supervisor. Your support has been instrumental in developing my technical skills. Deeply appreciate your dedication, especially during those late-night sessions, pushing our progress forward.

I extend my sincere gratitude to Ricardo for our invaluable professional collaboration and all the personal guidance you have generously offered. Working alongside you has been a pleasure, and I eagerly anticipate the possibility of future endeavours together.

Thanks to Marc Bolaños for his unwavering support since my first day at this university. Enjoyed all our collaborations.

My deepest gratitude to my little ones, Jesus and Imanol, for always being there for me. I thoroughly enjoyed being part of your master's journey and I cherish every moment as you both embark on your PhD journeys. I am filled with pride and gratitude for having shared this time with you. Your trust in me has been a driving force. Very fortunate to have crossed paths with you both.

I also want to express my sincere appreciation to the previous graduates from our group. You have consistently served as a source of inspiration for me. A huge thanks to the OG members - Rupali, Mostafa, Ahmed, Waseem, Javi, Santi, Pau, Sana, Francisco, Giuseppe, Marcos, and Omar - as well as our current members - Usman, Changhui, Emily, Daniel, Umair, and Ahmad. You guys make my every day memorable.

Thanks to Albert, Andrés, Marta, Pritomrit, and Arnau for making me feel at home when I first arrived. Enjoyed every Japanese lunch !!!

I extend my heartfelt thanks to the TFG students, whom I co-supervised – Ling, Nil, Guillem, Albert, and Oscar - for their dedication and hard work throughout their thesis. It was a pleasure working with each one of you.

Sincere gratitude to the Department of Mathematics and Computer Science, Research Office, Administrations as well as the staff at the University of Barcelona for their invaluable help and unwavering support.

That covers my life at the University and now for THE PEOPLE BEHIND THE SCENES.

First and foremost, I owe an immense debt of gratitude to my parents for their unwavering support and countless sacrifices. Without their guidance and encouragement, I don't see myself here today! As a first-generation graduate, their efforts played an instrumental role in shaping my journey.

To my dear wife and friend, Priya. Thank you for your patience and understanding through all the challenges. Having you by my side is a constant source of strength and I am deeply grateful for your unconditional love and support.

To some remarkable individuals - Dr Ramanamurthy, Dr Rajathilagam, Ms Bindu, Sumod and Janani – who have each played a pivotal role in guiding me through different stages of my career. Your invaluable support and mentorship have been instrumental in shaping my path forward. Without your support, I doubt pursuing a PhD would have been possible for me.

Sending big hugs to Naveena and Bhuvee for being there and sorting out every little thing for me. I am grateful for every iota of support I receive from both of you. Big thanks to my

best friends – Richy, Swarna, and Dinesh – for sharing all the joyful moments and supporting me through the difficult times. Those long phone conversations helped to keep me upbeat.

For my cherished family away from home – Deepthi, Ishu, Amrutha, and Gargie. And to my friends who have lightened my life in Barcelona – Marc and Carlotta, Giulia and Alessandro, Tini and Tornike. You all made me feel truly special.

To my fellow friends who embarked on their PhD journey alongside me, albeit in different locations - Radhu, Devu, Priyanka, and Sruti - Your journeys have been a perpetual wellspring of inspiration. To Rajsree, Gokul, Guna, and Shobana - You all kept me motivated. To Raji, Balaji, Pradeep, Vibha, Hema, and Sarath - Your invaluable support has been my anchor. Every one of you has a vital role in shaping who I am today.

Thanks to my extended family for their constant encouragement.

A heartfelt thank you to my doctor, Dr. Elisenda Climent, for ensuring my health remained intact throughout this time.

Big thanks to the Indian Diaspora, especially the Tamil community and the Barcelona Tamil Sangam, for their invaluable help and support during my time in Barcelona.

Every day brings new experiences. I am grateful for the beautiful souls I have met along the way. You bring vibrancy to my life.

Importantly, I express my PRANAMS TO THE ALMIGHTY for guiding me through this remarkable journey, enriching not only my doctoral studies but for every aspect that has made my life truly wonderful.

Bhalaji NAGARAJAN
May 3, 2024

Contents

Abstract **v**

el Resumen **vii**

Acknowledgements **ix**

1 Introduction **1**
1.1 Motivation . . . 2
1.1.1 Bridging the Gaps in Deep Learning . . . 2
Role of Data . . . 2
Rise of Model Architectures . . . 3
Versatility of Tasks . . . 3
1.1.2 Thesis Relevance . . . 4
1.2 Background . . . 4
1.2.1 Deep Learning . . . 4
1.2.2 Importance of Data . . . 5
Learning with Noisy Labels . . . 6
Learning without Labels . . . 7
Learning with Hard Samples . . . 9
1.2.3 Understanding Model Behaviour . . . 10
Uncertainty in Deep Learning Models . . . 10
1.2.4 Task Understanding: Application to Food Analysis . . . 11
1.3 Objectives . . . 12
1.4 Research Contributions . . . 13
1.5 Thesis Organization . . . 15

2 Learning with Noisy Labels **17**
2.1 Class-conditional Importance Weighting for Deep Learning with Noisy Labels . . . 18
2.2 CCLM: Class- conditional Label Noise Modelling . . . 26
2.3 Decoding Class Dynamics in Learning with Noisy Labels . . . 38
2.4 Bayesian DivideMix++ for Enhanced Learning with Noisy Labels . . . 45

3 Self-Supervised Learning **61**
3.1 Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition 62

3.2 All4One: Symbiotic Neighbour Contrastive Learning via Self-Attention and Redundancy Reduction . . . 74
3.3 Good Fences Makes Good Neighbours . . . 85

4 Quantizing Uncertainty **97**
4.1 Bayesian Deep Learning for Semantic Segmentation of Food Images . 98
4.2 Uncertainty-Aware Selecting for an Ensemble of Deep Food Recognition Models . . . 111
4.3 Deep Ensemble-Based Hard Sample Mining for Food Recognition . . . 123

5 Task Understanding: Applications to Food Recognition **135**
5.1 Learning Multi-Subset of Classes for Fine-Grained Recognition . . . 136
5.2 Dining on Details: LLM-Guided Expert Networks for Fine-Grained Food Recognition . . . 146
5.3 S2ML-TL Framework for Multi-label Food Recognition . . . 156

6 Results and Discussions **175**
6.1 Data . . . 175
6.1.1 Learning with Noisy Labels . . . 175
6.1.2 Self-Supervised Learning . . . 178
6.1.3 Learning with Hard Samples . . . 180
6.2 Models . . . 181
6.3 Tasks . . . 183

7 Conclusions **185**
7.1 Conclusions . . . 185
7.2 Thoughts for the Future . . . 186

A Research Activities **189**

Bibliography **193**

CHAPTER 1

Introduction

ARTIFICIAL INTELLIGENCE has permeated every facet of our lives, evolving into a mainstream phenomenon that holds tremendous value for all. No longer confined within the research boundaries, it has seamlessly integrated into our routines. Today, Artificial Intelligence (AI) revolutionizes countless aspects of human experience. The democratization of AI has allowed people from all walks of life to harness its benefit and enhance their daily activities. AI is regarded as an enhancement to our lives, paving the way for individual prosperity and collective advancement.

Deep Learning (DL) algorithms, a subset of AI, have revolutionized how machines perceive and interpret data, placing them at the forefront of AI innovation. It enables AI systems to learn and improve by extracting valuable insights from vast amounts of data. The recent "AI Summer" has catalyzed the rapid development of DL algorithms by leveraging increasingly large datasets and more sophisticated architectures. The remarkable ability of DL models to handle complex data and discern intricate patterns has paved the way for several advanced models. DL has extensive applications in numerous domains and thrives across various data modalities, with image data being one of the most common. **Image Understanding** focuses on interpreting the content of an image and extracting meaningful information from it. Image Understanding endeavours to get a deeper understanding, mirroring the way humans perceive and interpret the visual world. The success of DL models hinges on the availability of large datasets, increasingly complex model architectures and broad applicability across diverse domains and tasks. We call this the "TRIAD of Deep Learning". While each element is crucial for a successful DL system, they also present unique challenges.

The **theme of this thesis** centres around the challenges in "image understanding", due to inconsistencies in data, the role of understanding model behaviour, and the importance of incorporating domain information into the training process of DL models. We explore innovative strategies to overcome them and unleash the full potential of DL models. This chapter lays the groundwork for the thesis by outlining the motivation that drives this research, providing essential background knowledge on core concepts, delineating the specific objectives, and summarizing the novel scientific contributions of this thesis.

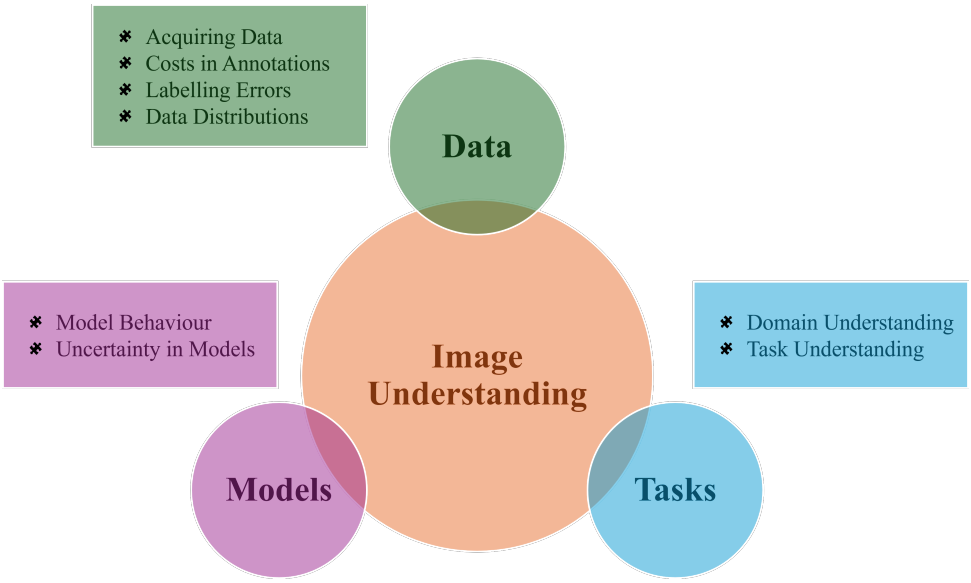


FIGURE 1.1: Deep Learning Triad with Challenges. The interplay between the three makes a successful model.

1.1 Motivation

Deep Learning is rapidly evolving. Recent breakthroughs, such as Large Language Models (LLMs), highlight the ability to process massive amounts of text data, generating human-quality comprehensions [Tou+23; Alm+23]. Large-scale transformers, boasting hundreds of millions to billions of parameters, excel at disentangling patterns from large complex data [Zha+22a]. Self-supervised Learning (SSL) paradigm [Bal+23] enables models to extract representations from input data, bypassing the need for external labels, thereby leveraging the abundance of unlabeled data to enhance performance and applicability further. The rapid emergence of even more powerful techniques underscores the transformative potential of DL. With DL penetrating various domains and reaching a wider audience, it becomes imperative to tackle numerous pivotal challenges. Addressing these challenges is crucial to ensure DL becomes even more robust and successful.

1.1.1 Bridging the Gaps in Deep Learning

The success of DL thrives on the intricate interplay between three core components: data, models and tasks. We delve into several critical issues within each component, demonstrating how addressing them fosters a more effective DL model for image understanding. We visualize this interplay and illustrate the challenges encountered by the data-model-task triad in Figure 1.1.

Role of Data

Traditional DL primarily concentrated on creating and training the best models tailored for specific tasks. Data collection and curation of data were merely used as a tool to enhance model performances. Data has always been considered indispensable in deep learning. However, it is only recently, with the emergence of powerful models, that there has been a shift in focus of the research community towards the data itself.

Understanding the intricacies of the data has had a profound impact on enhancing model performances.

Regardless of the learning paradigm, data remains a critical element in the realm of DL. Acquiring high-quality data, with sufficient volume and accurate annotations, is a cornerstone in developing effective DL models. However, the pursuit of such data is fraught with several challenges. The process of data discovery and data generation, encompassing both real and synthetic data, plays a pivotal role in generating substantial volumes of data. The data used for training DL models should reflect real-world scenarios. Distribution differences within the dataset can lead to models performing poorly in unseen scenarios. Creating accurate labels for training models, especially for complex tasks, often requires significant human effort. This expensive process is susceptible to errors, adding to the complexity of data acquisition. Overcoming these challenges becomes crucial for achieving optimal DL performance.

Rise of Model Architectures

Over the years, one key trend in DL is the development of increasingly larger and more complex models, resulting in significant breakthroughs in various tasks. Sequential models such as Recurrent Neural Networks (RNNs) [SP97] and Long Short-Term Memory (LSTMs) [HS97] excel at natural language processing [SMH11; Zhu+24]. Convolutional Neural Networks (CNNs) has been at the forefront of image processing tasks. Generative models such as Generative Adversarial Networks (GANs) [Goo+14] and diffusion models [Cro+23] can generate data, that closely resemble real-world samples [Ram+21; Ess+24]. Introduction of self-attention [Vas+17] in transformers [Dos+20] sparked a paradigm shift. Autoencoders [HS06] learn efficient representations of data from unlabeled sources. The success of pre-trained models [Wan+23], trained on massive datasets, and transfer learning strategies allow us to leverage the knowledge learned from a large, general-purpose model and easily adapt it to any specific domain or task.

DL models offer immense potential. However, they also present several technical challenges such as overfitting, underfitting, and finding the best hyperparameters for the particular task. Additionally, DL models inherit biases from the training data, leading to unfair outcomes. One of the crucial challenges is the interpretation of model behaviour. DL models are often criticized as "black box" models as the internal decision-making process is difficult to understand. Lack of interpretability can be problematic in tasks, where explaining the model's reasons is crucial. Providing avenues for understanding model behaviour would unlock the true potential of DL models. This is also critical for building trust and ensuring responsible DL application.

Versatility of Tasks

Transfer Learning [WKW16] has revolutionized DL by leveraging pre-trained models for new tasks [Ray+23; The+23]. However, it is not a one-size-fits-all solution [Zha+22b; Wan+19b]. Differences in tasks and the underlying data distribution necessitate adjustments to the model. Effectively applying DL in various application areas requires incorporating domain knowledge into the models. Domain knowledge encompasses specific characteristics, challenges and nuances of the particular task and the associated data. This knowledge enables us to effectively adapt DL models to various real-world problems. The consequence of neglecting domain and task knowledge can be significant. DL models developed without using the domain knowledge

often result in poor performance, misclassification and missing important features. Understanding the domain and task allows us to tailor data pre-processing steps to extract the most relevant features and to make model architecture changes. The task knowledge helps in DL model design that provides improved performance compared to conventional models, that are created by fine-tuning on ImageNet weights.

1.1.2 Thesis Relevance

Having established the critical challenges within each facet of DL models, we set to establish the main goal of this thesis. To achieve optimal performance, effectively addressing the above listed challenges is paramount. On that front, this research focuses on distinct, yet interconnected, challenges of data inconsistencies, limited labelled data, understanding DL model decision-making, and tackling domain-specific problems through architectural changes. *We will now get deeper into understanding how to tackle these challenges.*

1.2 Background

This section lays the groundwork for understanding the core concepts referenced throughout this thesis. We will introduce the key frameworks that provide a foundation for describing and addressing the challenges this research aims to tackle.

1.2.1 Deep Learning

Deep Learning is a subset of machine learning that leverages **Artificial Neural Networks (ANNs)**, inspired by the human brain. ANNs consist of interconnected layers of artificial neurons that mimic how biological neurons process information. DL excels at automatically uncovering hidden meaningful relationships and patterns within data by learning feature representations through these layers. DL models can work with diverse data types including images, text, audio, video, and complex data such as electronic health records and sensory data. This capacity empowers DL models to achieve human-level performances and even surpass them across tasks such as image understanding [Guo+16], natural language processing [LLA22], and speech recognition [Pra+23]. Recent DL models showcase an extraordinary ability to generate realistic images [Cho+24; Rom+22] and produce musical compositions [Don+18; Vya+23].

While ANNs have been around since the 1940s, their potential remained limited due to computational constraints. The recent explosion in computing resources, particularly the advent of Graphics Processing Unit (GPUs), have revolutionized the field. GPUs have accelerated the computation speeds by 1000 times over 10 years [Foo22]. Landmark progress was made around in the early 2010s with AlexNet [KSH12] winning the ImageNet 2012 challenge [Rus+15]. This breakthrough flared rapid acceleration in DL applications across various fields. The immense speed boost provided by GPUs allowed researchers to design much larger and more intricate neural networks. Powerful computation allowed for bigger models, which in turn necessitated large datasets [Rus+15]. The confluence of these factors – access to large datasets, powerful algorithms, robust frameworks, and immense computational power – has propelled the growth of DL algorithms, leading to significant breakthroughs in various tasks and establishing them as a leading force in AI.

The evolution of DL model development can be characterized by distinct periods, each focusing on a specific aspect. Early research primarily explored applying these models to various tasks, demonstrating their versatility. DL found its way into natural language processing, different visual tasks, and audio processing. This period was followed by an intense focus on model architectures, with researchers striving to create ever-larger and more complex models. Classic examples include the vast parameter count of VGGNet [SZ14] (VGG19 has 143.7M parameters and achieves 71.3% Top-1 accuracy on ImageNet) compared to the more efficient EfficientNet [TL19; TL21] architecture (EfficientNetV2L has 119.0M parameters and achieves 85.7% Top-1 accuracy on ImageNet). The focus extended to entirely new architectures, such as Transformers [Dos+20], with their fundamentally different approach to processing data. Alongside architecture advancements, significant progress has been made in optimization algorithms and hyperparameter tuning, further improving model performance. The recent years have witnessed a significant shift in focus. While model architecture remains important, researchers increasingly recognize the importance of data [She+23; Zha+23]. The adage "garbage in, garbage out" holds for DL. High-quality, well-curated datasets are now recognized as crucial for achieving optimal model performance. As we move forward, success in DL will hinge on a well-balanced approach that addresses all three aspects – model architectures, domain and task understanding, and the quality and quantity of data used for training.

Understanding the triad of data, models and tasks is crucial for effectively designing, training, evaluating, deploying, and maintaining DL models in real-world applications. While DL can be applied to various data types, this thesis will focus primarily on ***image data*** *and the challenges associated with developing* ***visual DL models****. To maintain focus and consistency, we will primarily frame the context of this thesis around classification tasks, with exceptions explicitly highlighted.*

1.2.2 Importance of Data

DL models are data-driven algorithms. They rely heavily on the quality, diversity, and quantity of training data. Unlike traditional approaches, DL models learn by identifying patterns within the vast volumes of data. *The bigger the model, the bigger the data appetite is.* For instance, EfficientNetV2 [TL21] is trained on ImageNet-21K dataset [Rid+21] boasting 14M labelled images across 21,841 possible classes. Similarly, the recent ViT-G [Zha+22a] leverages an even larger JFT-3B dataset, containing 3 billion images across 30k classes. These examples highlight the growing trend in increasingly colossal datasets to train ever-more sophisticated DL models.

In essence, data serves as the foundation for DL models. The quality of this foundation directly influences the model's ability to learn effectively and perform well in real-world scenarios. High-quality data ensures that the model learns accurate and meaningful representations. A diverse dataset exposes the model with a wide range of variations. Conversely, insufficient data can lead to overfitting, limiting their generalizability. While the importance of high-quality data is undeniable, curating such datasets presents significant challenges. It requires meticulous efforts to ensure data accuracy, consistency, and relevance to the model's intended purpose.

Effectively harnessing the vast volume of data comes with its unique challenges. One major hurdle lies in the cost and time associated with the labelling data. Approaches such as Self-supervised Learning enable models to learn meaningful representations from unlabeled data, reducing this reliance on the labelling process. Another challenge is the presence of errors in labelled data. Learning with Noisy Labels (LNL) methods identify and mitigate the impact of these errors, making

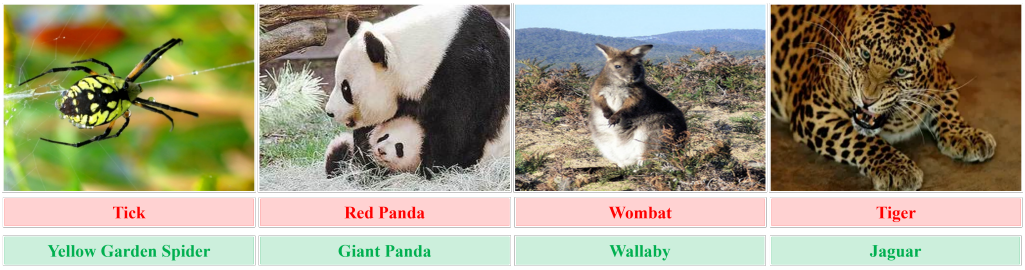


FIGURE 1.2: Examples of Label Noise present in the ImageNet [Rus+15] dataset. A gallery of label errors across 10 different benchmarks is available in `https://labelerrors.com`. *(The figure shows ImageNet Label and Human-Validated Corrected Label)*

these models more robust. Beyond labelling concerns, real-world data often exhibits skewed distributions, where certain samples are more prevalent. Sample importance addresses this by assigning higher importance to under-represented samples during training. We provide a more in-depth discussion of each approach below.

Learning with Noisy Labels

Generating high-quality large-scale training datasets for DL often involves manual annotations, which are tedious, cumbersome and expensive. It requires domain expertise to accurately label the data, which can be difficult to obtain and costly. Additionally, human perception and limitations in identifying specific samples can introduce annotation errors. While alternate methods like web crawling [SKL19], crowdsourcing [Yan+14], and leveraging pre-training models [FBQSM16] have emerged to address these cost concerns, they hold certain drawbacks. These approaches can still be susceptible to annotation errors [NAM21]. We showcase some examples of label noise in the popular ImageNet [Rus+15] benchmark in Figure 1.2. Even with error correction controls, some inaccuracies can inevitably slip through the creation of training datasets.

Several projects have been dedicated to unravelling the intricacies of label noise, seeking to comprehend its presence and impact within benchmark datasets [CS21]. Real-world datasets often suffer from label noise, estimated to be between 8.0% and 38.5% [Son+22]. This label noise significantly hinders the performance of DL models. Learning with label noise poses two significant challenges to training DL models. First, Deep Neural Networks (DNNs) have a large number of model parameters. This can lead to overfitting on corrupted labels because DNNs have the learning capability to intricate, yet potentially irrelevant, patterns from the noisy data [AZLL19]. Second, DNNs also possess a strong memorization capacity, that can easily fit an entire corrupted training set, resulting in poor generalizability [Zha+21a]. This means that the DL models perform well on the training data with noisy labels but struggle to adapt and perform accurately on unseen test data. Since correcting the label errors can be near-impossible, developing DL methods that learn in the presence of label noise becomes crucial. Consequently, robust training methods are essential to build DL models that can withstand such noise and achieve optimal performance.

LNL, introduced in the late 1980s [AL88], has become a prominent area of research within the machine learning community due to its importance in creating noise-robust models. Different LNL algorithms have been developed to address this challenge. A comprehensive summary of different LNL algorithms was presented by Song et al [Son+22]. LNL algorithms can be broadly categorized into four approaches:

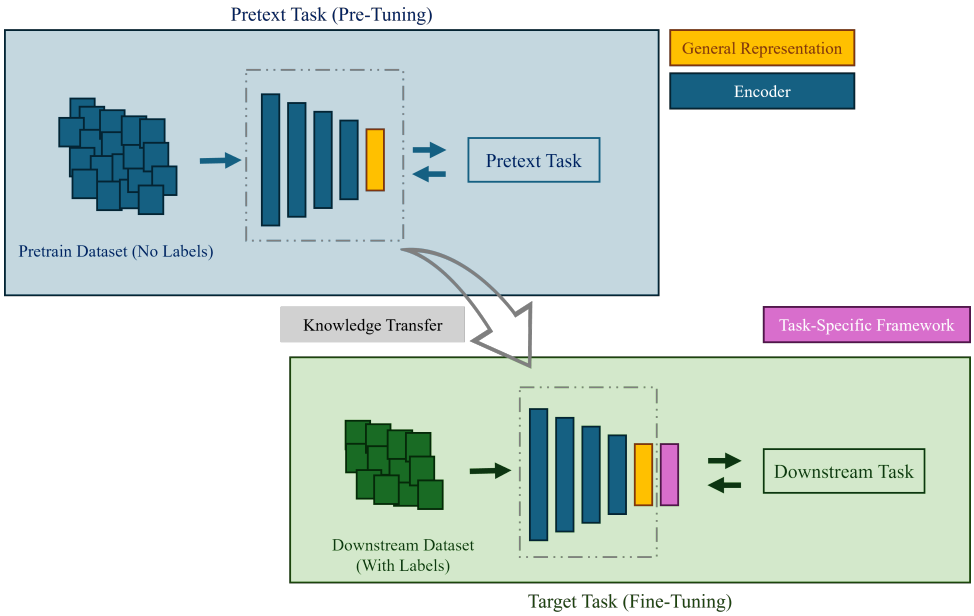


FIGURE 1.3: Self-supervised Learning tasks. Pre-text tasks allow the encoder to learn representations with data. Downstream tasks allow the model to be adapted to the specific task.

» Loss correction methods [ZNS21; Wu+21; Yao+20]: Modify the loss function used during the training to account for the presence of label noise using noise transition matrices.

» Regularization-based methods [Zha+17; Che+22; WSF22]: Making changes to the model architecture and the learning process, aimed at limiting the memorization of labels.

» Noise-robust architectures and loss functions [Wan+19a; Ma+20; Li+22]: Focus on designing model architectures and loss functions specifically tailored to handle label noise.

» Sample-selection methods [LSH20; Kar+22; Cai+23]: Focus on selecting a subset of samples (clean or noisy) and device training steps to learn from the subset of the training set.

Sample-selection methods represent a recent and increasingly popular LNL approach. While these methods offer advantages, they come with open challenges. A key challenge lies in determining the optimal strategies for selecting a clean subset of samples from the potentially noisy dataset. Additionally, challenges remain regarding the effective utilization of this subset to learn patterns representative of the entire dataset. Part of this thesis delves specifically into these challenges associated with sample-selection methods within the LNL framework.

Learning without Labels

The ever-growing volume of data generated daily in this digital age amplifies the challenges in managing and extracting training datasets for DL models. Relying solely on labelled data for DL model development may leave us at risk of falling behind in this data deluge. Alternative approaches have emerged recently that alleviate the need for extensive manual labelling. SSL [Bal+23] deals with training models on tasks

that do not rely on labels to learn meaningful representations. This line of work does not require human-annotated labels and aims at learning better representations for any downstream task.

SSL training and evaluation can be realized in two steps: pretext tasks and downstream tasks. Figure 1.3 highlights the two tasks in a typical SSL setting. Pretext tasks train the DL models to learn meaningful intermediate representations from unstructured data through self-supervised learning objectives. The model learns to solve a different task, allowing it to extract useful generic features from the data. These learned "general" representations are subsequently transferred to the downstream tasks such as classification, detection, and retrieval. SSL leverages a variety of pretext tasks, including instance-instance contrast (contrasting between views), image colourization (predicting missing colour), inpainting (filling missing region), puzzle solvers (rearranging image parts), masked image modelling (predicting masked portions of the image), and context prediction (inferring missing information based on surrounding data) [Ozb+23]. Downstream tasks are the same as traditional supervised learning tasks. The learned representations are fine-tuned on the downstream task using a smaller amount of labelled data, significantly improving its performance compared to training from scratch. This also reduces the training time required for the downstream task as the SSL model is equipped with "learned" representations.

SSL pretext task design can be generative (self-predictive learning) or discriminative (contrastive learning) [Liu+21]. The generative tasks involve training a model to predict a part of an individual data sample while operating in the pixel space [Bao+21; He+21; Li+23]. For example, reconstruct an image after corrupting parts of it. Discriminative tasks focus on learning relationships between multiple data samples [Che+20; Zbo+21; Dwi+21]. The model learns to distinguish between similar and dissimilar data points. This can involve tasks like comparing different augmentations of the same image.

Contrastive loss [Bro+93] underpins discriminative tasks, where a model learns representations by contrasting similar (or dissimilar) inputs moving their embedding close (or far from each other). In SSL, the contrasting inputs are often generated by applying augmentations to a single image. The introduction of InfoNCE loss [OS+16] played a pivotal role in SSL research advancements. Dual-backbone networks are the dominant architectural choice for contrastive learning methods. It utilizes a dual-backbone "encoder" network where the model weights are shared [Bro+93]. In the context of SSL, these encoders process two augmented instances of the same image, generating representations that can be compared with each other through a contrastive loss function [Ozb+23]. Recent advancements have shown promising results even with non-weight shared dual-backbone architectures. Most SSL frameworks rely on variants of ResNet as backbones. With the recent surge in vision transformers, they are also being explored as backbones.

Recent advancements in contrastive SSL have explored various techniques to improve SSL performance. MoCo [He+20] used memory banks with delayed weight updates and SimCLR [Che+20], which focused on projection heads and strong augmentations, sparked significant interest in SSL research. BYOL [Gri+20] presents an asymmetric framework where a model is trained to predict representations of another (teacher) model. BYOL introduced the concept of "predictors". SimSiam [CH21], another popular model used the concept of "stop-grad" and "projection/prediction multi-layer perceptions". Barlow twins [Zbo+21] uses a unique approach, utilizing an objective function based on the redundancy-reduction principle instead of the usual InfoNCE loss. The proposed advancements in each framework brought a diverse set

FIGURE 1.4: Examples of "easy" (Top row) and "hard" (Bottom row) samples in the same class. Some of the "hard" samples are difficult to be identified even by humans.

of unique components to the SSL architecture, further enriching the domain. Importantly, projectors are added to *project* the backbone features to a different dimensional space before applying the contrastive loss. Predictors and stop-grad operations are added to avoid model collapse in asymmetric contrastive methods.

While SimCLR and the other previously listed approaches effectively utilizes contrasting augmentations of the same image for contrastive SSL, it can struggle to provide sufficient diversity among positive samples. Neighbour contrasting approaches used neighbours instead to increase the diversity. NNCLR [Dwi+21] proposes leveraging the first nearest neighbours of an image instead of directly using the data augmentation itself. This injects more diversity into the positive pairs (pair of views from the same image) used for contrastive learning. MeanShift [KTP21] takes this concept further by utilizing k-nearest neighbours, which enhances the model's generalization capabilities. Neighbour contrastive learning offers a promising approach for SSL. However, key challenges remain:

» How to generate effective contrast information?

» How can we efficiently utilize the neighbour information?

The thesis delves into these challenges in creating an efficient neighbour contrasting strategy.

Learning with Hard Samples

DL models have revolutionized various tasks due to the theoretical ability of neural networks to learn any prediction function, given enough neurons, layers, and data [HSW89]. However, a significant hurdle in achieving optimal performance is the requirement for vast amounts of labelled training data [Sun+17]. SSL and LNL techniques are powerful tools to address issues of annotations. However, even within datasets with existing annotations, challenges persist. Real-world datasets are not uniform. They are often curated with a specific problem in mind, resulting in a distribution of samples. Most samples within the distribution are well-defined, meaning they are high-quality and numerically well-represented. However, some samples deviate significantly from this. Figure 1.4 illustrates examples of "hard" samples - on the top are the well-represented images, while on the bottom are those that are less well-represented but remain relevant to the class. These samples pose a challenge for DL model training, as they can struggle to learn from them effectively. The overall ability of the model to generalize on unseen data can be improved by paying special attention to these samples. This would ensure that the model does

not solely focus on the "easy" samples, leading to better overall performance and robustness.

Example difficulty-based learning [Jia+20; ADH22] is an interesting field of research that tackles the issue of varying learning difficulty within datasets. This difficulty can stem from statistical properties or the learning complexity of the samples. The challenges associated with the data collection often lead to the presence of "hard" samples. With the ever-increasing dataset size and complexity, particularly those obtained through web crawling, the likelihood of encountering significant amounts of these challenging "hard" samples rises. Understanding the learning behaviour of samples becomes crucial for efficient model training [Agu+21]. The thesis tackles the challenges of identifying these "hard" samples within datasets and developing strategies to optimize their learning.

1.2.3 Understanding Model Behaviour

DL models are powerful tools, however, their effectiveness hinges on one crucial factor: understanding how they behave. This transparency is even more critical as DL models are increasingly integrated into our daily lives [LPK20]. By unveiling the model's inner workings, we foster user trust and pave the way for responsible applications [Pan+23]. Explainable AI (XAI) techniques [SO23] reveal the model's expected impact and potential biases. Tools such as sensitivity analysis are used to remove bias, drift, and other gaps in both data and models [DBH18]. Interpretability techniques allow us to understand the cause of why a model makes certain decisions [Zha+21b]. Feature importance analysis, partial dependence plots, LIME and SHAP are widely used to analyse the key factors influencing the model's predictions. Finally, uncertainty quantification [Gal+16] allows the DL model to express its limitations. The models can indicate scenarios where it might not have a definitive answer. This section of the thesis centres on understanding and mitigating uncertainty within DL models.

Uncertainty in Deep Learning Models

Uncertainty plays a crucial role in making informed decisions. It refers to the model's degree of confidence in its predictions. There are two types of uncertainty: epistemic and aleatoric [DKD09]. Epistemic uncertainty, also called model uncertainty, arises due to limitations in the model's knowledge or the training data. It is a reducible form of uncertainty and can be reduced by acquiring more data. Aleatoric uncertainty, also called data uncertainty, stems from the inherent variability within the data, such as labelling noise, measurement noise, or missing data. This type of uncertainty is irreducible and persists even in the limit of infinite data. **Uncertainty Quantification** [Soi17] refers to estimating and quantifying these uncertainties. Uncertainty quantification is a powerful tool for building reliable and trustworthy DL systems. It fosters informed decision-making by acknowledging the limitations of the model's knowledge. An in-depth exploration of different uncertainty quantification techniques is presented by different works [Abd+21; Gaw+23]. In this thesis, we delve deeper into understanding the model uncertainty. By quantifying and analyzing this uncertainty, we aim to improve the robustness and confidence of models and enable them to make more informed decisions.

Epistemic Uncertainty using Monte-Carlo Dropouts Bayesian Deep Learning approaches [Mac92; Nea12], where Bayesian statistics are incorporated into the models

FIGURE 1.5: Unique Challenges Presented by Food Images.

are popular methods to compute uncertainty. Bayesian neural networks estimate a probability distribution over the weights, representing the model's uncertainty. A prior distribution is placed over the weights, reflecting our initial beliefs about the model's parameters. Bayesian inference is used to update the prior distribution based on observed data, which is the required posterior distribution over the weights. Bayesian neural networks are easy to formulate but are difficult to perform inference [KG17]. Several approximations exist for making Bayesian inference [Gri11]. Monte-Carlo Dropouts [GG16] is a widely popular approximation technique. Dropouts [Sri+14] are regularization techniques used in avoiding overfitting, where a certain percentage of neurons are dropped during training. Monte-Carlo Dropouts leverages using dropouts during training and during inference to introduce randomness into the model. It follows the law of large numbers and approximates the expected value by the mean of N stochastic networks. To compute uncertainty, a model is trained with a dropout before every weight layer and the dropout is kept active at the inference. With N inferences over the same sample, we compute the approximate posterior.

1.2.4 Task Understanding: Application to Food Analysis

DL models excel when tailored to the specific characteristics of a domain. Each domain presents unique challenges, and leveraging domain knowledge is crucial for designing effective models. This knowledge is often used to design the data pipeline and adapt the model architecture. We will highlight the importance of understanding the domain through visual food recognition. Visual food understanding is a complex domain presenting unique challenges apart from typical computer vision challenges, necessitating specialized approaches. Visual food recognition [Sal+21; KPM20] focuses on automatically identifying the food category depicted in a given image. Here, we will delve into the specific obstacles encountered in this domain and explore strategies for overcoming them, highlighting the importance of domain-specific knowledge for successful DL applications.

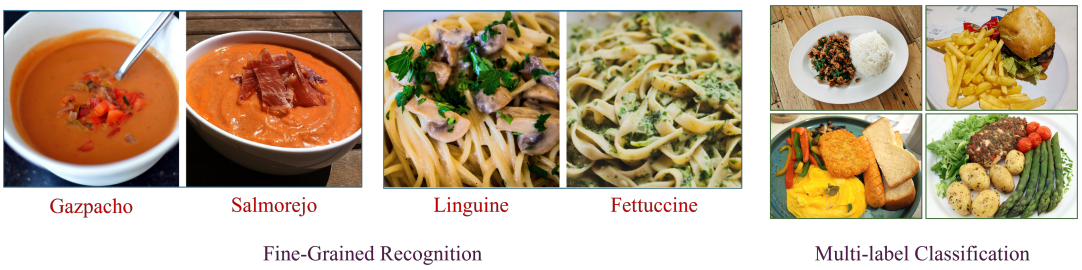


FIGURE 1.6: Challenges in Food Tasks.

Food recognition plays a vital role in various food computing tasks [Min+19]. It finds its applications in dietary management and nutritional analysis [Yun+18]. Several popular food apps such as MyFitnessPal[*], LogMeal[†], and, GreenHabit[‡] leverage food recognition models to empower users in tracking their food intake. Food recognition holds immense potential, however, it presents challenges beyond issues like occlusion and blurry images, making it a difficult but interesting computer vision task.

We highlight different vision-based challenges particular to food images in Figure 1.5. Food images show high intra-class and low inter-class variability. Since food is so diverse, we can encounter food categories that are very similar to each other *(Carpaccio and Tomato Salad)*. Due to the cooking variations, it is possible to find samples within the same category that are very different *(Babi Guling)*. It is difficult for models to distinguish between them compared to classes with more distinct characteristics. The visual complexity of food dishes is high due to the lack of distinctive spatial layouts, rigid structures, and the presence of randomly distributed ingredients. This makes it challenging for the DL models to extract features for accurate classification. Training datasets should be comprehensive and carry the diversity and variety of data points that reflect the real-world characteristics of the domain. However, creating such representative datasets can pose significant challenges in a domain such as food [Min+23].

Along with the challenges in food images, there are challenges particular to the food recognition task. Unlike traditional single-label tasks (identifying one object in an image), real-world food recognition often involves multiple items on a single plate, necessitating multi-label classification. It is a common trait in almost all cuisines across the globe. Food is served in combinations. The fine-grained nature of food classes poses a significant challenge. Common DL models struggle to differentiate between highly similar categories, hindering accurate recognition of food items with subtle variations. A deep understanding of these domain-specific and task-specific challenges is essential for designing effective training pipelines for DL models in food recognition applications.

1.3 Objectives

As established, the success of DL models hinges on the delicate interplay between data, models and tasks. Building efficient DL models requires understanding the intricacies of each component and the potential challenges it may pose. The main goal of this thesis is to delve into the open challenges and identifying optimal solutions to

[*] https://www.myfitnesspal.com/
[†] https://www.logmeal.es/
[‡] https://greenhabit.nl/

overcome them. Focusing on the *deeper understanding of the deep learning triad and its associated challenges*, we delineate the specific objectives of the thesis as follows.

» **Learning with label noise:** We aim to develop noise-robust DL models, focusing on sample-selection-based algorithms capable of learning in the presence of label noise, overcoming the issues of generalization and memorization.

» **Learning from unlabeled data:** We aim to enhance contrastive SSL methods, particularly neighbour contrast by improving the design of contrast tasks and achieving better *learned* representations.

» **Making informed decisions:** Our objective is to gain insights into the model's confidence in making decisions using Monte-Carlo Dropouts and to use this knowledge of model uncertainty to improve the overall decision-making process.

» **Task Understanding through Visual Food Analysis:** We aim to address the domain-specific challenges of fine-grained and multi-labelled classification by tailoring model architectures for enhanced performance in visual food analysis.

1.4 Research Contributions

This thesis delves into the characteristics and challenges of each element of the triad in building efficient DL models, progressively shifting focus from a model-centric approach to a data-centric one. Specifically, the contributions of this thesis can be summarized as follows. *Each contribution is linked with publications, ordered by the most recent first. (* denotes "Equal first author" contributions).*

(Contribution #1) Learning with noisy labels (see Chapter 2): Label noise in the training data can significantly impact the performance and generalization of DL models. Developing robust models that can handle and mitigate the effect of label noise is crucial. First, we proposed a loss correction method that utilizes dynamic weights during model training to assign greater importance to the less-learned classes. In our subsequent works, we highlighted the limitations of the existing noise modelling techniques. We proposed a class-conditional approach to split the clean and noisy samples for subsequent training. Finally, we proposed a novel LNL framework that tackles two critical challenges of model memorization and label uncertainty.

» **Bhalaji Nagarajan,** Ricardo Marques, Eduardo Aguilar, and Petia Radeva. "Bayesian DivideMix++ for Enhanced Learning with Noisy Labels." In: Neural Networks, 2024. **(JCR: Q1, IF: 7.8).**

» Albert Tatjer*, **Bhalaji Nagarajan*,** Ricardo Marques, and Petia Radeva. "Decoding Class Dynamics in Learning with Noisy Labels." In: Pattern Recognition Letters, 2024. **(JCR: Q2, IF: 5.1).**

» Albert Tatjer*, **Bhalaji Nagarajan*,** Ricardo Marques, and Petia Radeva. "CCLM: Class-Conditional Label Noise Modelling." In: Iberian Conference on Pattern Recognition and Image Analysis, 2023. **(Best Paper Award).**

» **Bhalaji Nagarajan*,** Ricardo Marques*, Marcos Mejia, and Petia Radeva. "Class-conditional Importance Weighting for Deep Learning with Noisy Labels." In: 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2022.

(Contribution #2) Learning from unlabeled data (see Chapter 3): SSL offers a method to train DL models that learn inherent structures and relationships from raw and unlabeled data. This is particularly useful where the amount of labelled training data is limited. Our initial work focused on an enhanced SSL framework, that combines image contrasting and feature contrasting techniques. Subsequently, we introduced a novel contrastive SSL method, that utilizes "centroids" created through a self-attention mechanism. Finally, we extended our work on contrastive methods to propose a technique to identify and address "bad" neighbours using generated novel representations in the latent space.

- Imanol G. Estepa, Ignacio Sarasúa, **Bhalaji Nagarajan,** and Petia Radeva. "All4one: Symbiotic neighbour contrastive learning via self-attention and redundancy reduction." In: IEEE/CVF International Conference on Computer Vision, 2023. **(GGS Rating: A++).**
- Imanol G. Estepa, Jesús M. Rodríguez-de-Vera, **Bhalaji Nagarajan,** and Petia Radeva. "Good Fences Make Good Neighbours." In: IEEE/CVF International Conference on Computer Vision Workshops, 2023.
- Nil Ballús, **Bhalaji Nagarajan,** and Petia Radeva. "Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition." In: Iberian Conference on Pattern Recognition and Image Analysis, 2022.

(Contribution #3) Hard sample mining (see Chapter 4): Datasets often encompass samples with drastically different representative features and those that are poorly represented numerically. These "hard" samples are important in providing the necessary diversity in the dataset. Information on *easy* and *hard* samples benefits the DL model training. We proposed a new measure called "coincidence score", to help better understand the learning behaviour of DL ensemble models concerning the nature of samples. Furthermore, we introduced a controlled over-sampling method that leverages this score to guide the model's learning process.

- **Bhalaji Nagarajan,** Marc Bolaños, Eduardo Aguilar, and Petia Radeva. "Deep ensemble-based hard sample mining for food recognition." In: Journal of Visual Communication and Image Representation, 2023. **(JCR: Q2, IF: 2.6)**

(Contribution #4) Modeling uncertainty to enhance visual food tasks (see Chapter 4): Quantifying and analyzing uncertainty in DL models aims to assess the reliability of the model predictions. We emphasize the importance of epistemic uncertainty, which reflects the inherent limitations of the model's knowledge. First, we proposed an uncertainty-aware ensemble selection method, which allows us to select the most effective ensemble of base classifiers. In the subsequent work, we extended benchmark food segmentation methods by incorporating a Bayesian framework to provide uncertainty associated with each prediction.

- Eduardo Aguilar, **Bhalaji Nagarajan,** and Petia Radeva. "Uncertainty-aware selecting for an ensemble of deep food recognition models." In: Computers in Biology and Medicine, 2022. **(JCR: Q1, IF: 7.7).**
- Eduardo Aguilar, **Bhalaji Nagarajan,** Beatriz Remeseiro, and Petia Radeva. "Bayesian deep learning for semantic segmentation of food images." In: Computers and Electrical Engineering, 2022. **(JCR: Q2, IF: 4.3).**

(Contribution #5) Handling fine-grained food classes (see Chapter 5): Fine-grained food recognition presents significant challenges due to the prevalence of classes with subtle variations. These classes encompass subtle variations in ingredients

and cooking methods, which can lead to significant semantic differences between visually similar food classes. We propose a novel approach utilizing subset learning techniques to tackle the complexities of fine-grained food classes. In this method, we leverage expert models, each trained on a specifically curated subset of the training data. In our initial work, we focused on creating these subsets by identifying the most confusing classes, aiming to improve differentiation between them. Building upon this success, in our second work, we explored using LLMs and multi-modal models to generate the subsets, further enhancing the overall recognition accuracy.

» Jesús M. Rodríguez-de-Vera, Pablo Villacorta, Imanol G. Estepa, Marc Bolaños, Ignacio Sarasúa, **Bhalaji Nagarajan,** and Petia Radeva. "Dining on Details: LLM-Guided Expert Networks for Fine-Grained Food Recognition." In: ACM Multimedia Workshops, 2023.

» Javier Ródenas*, **Bhalaji Nagarajan*,** Marc Bolaños, and Petia Radeva. "Learning Multi-Subset of Classes for Fine-Grained Food Recognition." In: ACM Multimedia Workshops, 2022.

(Contribution #6) Working with multi-label classes (see Chapter 5): Real-world food images often contain multiple ingredients and dishes, resulting in a multi-label classification problem. This multi-label approach more accurately reflects real-world scenarios. To address this challenge, we proposed a novel multi-stage transfer learning framework, which leverages the learning capabilities of single-label classifiers to tackle the complex task of multi-label food recognition.

» **Bhalaji Nagarajan,** Eduardo Aguilar, and Petia Radeva. "S2ML-TL framework for multi-label food recognition." In: International Conference on Pattern Recognition Workshops, 2021.

1.5 Thesis Organization

The current chapter serves as a comprehensive introduction to the core themes and specific areas of investigation explored throughout this thesis. The following chapters build upon this foundation, each focusing on a distinct aspect of the objectives of the thesis. *(The subsequent chapters group various papers presented throughout this thesis).* Chapter 2 engages with papers exploring noisy labels and robust methods developed to facilitate learning in the presence of noisy labels. Chapter 3 includes papers centred around learning methods that do not rely on labelled data. Chapter 4 categorizes papers addressing the challenge of model uncertainty across diverse food-related tasks. Chapter 5 demonstrates the importance of task understanding, exemplified through different papers in the visual food analysis domain. Chapter 6 summarizes the results from distinct studies, highlighting the outcomes of diverse evaluations. Finally, Chapter 7 concludes this thesis by providing a comprehensive summary and outlining discussions about the avenues for further research and exploration.

CHAPTER 2

Learning with Noisy Labels

Introduction

Deep Learning models thrive on large, high-quality datasets, but creating them is expensive and time-consuming. Human labelling is error-prone. Alternative labelling methods such as web crawling, crowdsourcing, and creating annotations using semi-supervised models can introduce label noise. Regardless of the labelling method, label noise is inevitable in real-world data. To address this, Learning with Noisy Labels techniques are crucial. Our research focuses on sample selection, a popular LNL approach that strategically selects clean (or noisy) samples to improve the model learning in the presence of label noise. We delve into the open challenges of creating robust sample-selection methods and aim to develop more effective algorithms for handling label noise.

The following questions guide our investigation in sample-selection-based LNL algorithms (Numbers in brackets represent the corresponding paper).

- *How to effectively determine the clean and noisy samples? (#2, #3)*
- *Should all samples in a LNL algorithm be treated uniformly? (#1)*
- *How to mitigate the impact of DNNs memorization? (#4)*
- *How does label uncertainty influence the learning process in LNL? (#4)*

We aim to unravel deeper insights within the LNL landscape, by directly addressing the critical questions we outlined earlier.

Works Published Within This Chapter

1. ***Bhalaji Nagarajan****, Ricardo Marques*, Marcos Mejia, and Petia Radeva. "Class-conditional Importance Weighting for Deep Learning with Noisy Labels." VISAPP, 2022.*
2. *Albert Tatjer*, **Bhalaji Nagarajan***, Ricardo Marques, and Petia Radeva. "CCLM: Class-conditional Label Noise Modelling." IbPRIA, 2023.*
3. *Albert Tatjer*, **Bhalaji Nagarajan***, Ricardo Marques, and Petia Radeva. "Decoding Class Dynamics in Learning with Noisy Labels." Pattern Recognition Letters, 2024.*
   (JCR: Q2, IF: 5.1).
4. ***Bhalaji Nagarajan**, Ricardo Marques, Eduardo Aguilar, and Petia Radeva. "Bayesian DivideMix++ for Enhanced Learning with Noisy Labels." Neural Networks, 2024.*
   (JCR: Q1, IF: 7.8).

Class-conditional Importance Weighting for Deep Learning with Noisy Labels

Bhalaji Nagarajan[1,†][a], Ricardo Marques[1,†][b], Marcos Mejia[1][c] and Petia Radeva[1,2,*][d]
[1]*Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona, Spain*
[2]*Computer Vision Center, Cerdanyola (Barcelona), Spain*

Keywords: Noisy Labeling, Loss Correction, Class-conditional Importance Weighting, Learning with Noisy Labels.

Abstract: Large-scale accurate labels are very important to the Deep Neural Networks to train them and assure high performance. However, it is very expensive to create a clean dataset since usually it relies on human interaction. To this purpose, the labelling process is made cheap with a trade-off of having noisy labels. Learning with Noisy Labels is an active area of research being at the same time very challenging. The recent advances in Self-supervised learning and robust loss functions have helped in advancing noisy label research. In this paper, we propose a loss correction method that relies on dynamic weights computed based on the model training. We extend the existing Contrast to Divide algorithm coupled with DivideMix using a new class-conditional weighted scheme. We validate the method using the standard noise experiments and achieved encouraging results.

1 INTRODUCTION

Deep Neural Networks (DNNs) tend to show an incredible upshot in performance when trained with large-scale labeled data under supervised environments (Krizhevsky et al., 2012). The strong and implicit assumption in training any DNN is that the dataset is clean and reliable. However, in real-world it is difficult to meet this assumption owing to the expensive cost and the time required to create such large high-quality datasets (Liao et al., 2021). The labelling cost is reduced substantially by crowd-sourcing the labelling process or by using an automated labelling system. However, this inherently leads to having errors in the labels.

Recent advances in DNNs show that it is possible to create learning algorithms that abide to less accurate training data (Sun et al., 2017; Pham et al., 2021; Ghiasi et al., 2021). However the DNNs have a tendency to overfit on the label noise (Zhang et al., 2021a). There are two common approaches to tackle the problem of overfitting on noisy labels - Semi-Supervised Learning (SSL) and Learning with Noisy Labels (LNL) (Zheltonozhskii et al., 2021). SSL uses scarce high-quality labelled data to learn representations of large amount of unlabelled data (Hendrycks et al., 2019). LNL approach uses less expensive annotations, but uses noisy labels as a trade-off (Natarajan et al., 2013). Both approaches are closely related to each other and are often used in combination to help DNNs learn from less accurate samples (Zheltonozhskii et al., 2021; Li et al., 2020; Chen et al., 2021).

LNL has been already studied both in machine and deep learning (Frénay and Verleysen, 2013; Frénay et al., 2014; Nigam et al., 2020; Cordeiro and Carneiro, 2020). The objective of any LNL algorithm is to find the best estimator for a dataset distribution learnt from the original distribution with noise. It is necessary for the DNN to learn the noise structure and estimate the parameters accordingly. In many LNL approaches, there is short 'warm-up' phase where supervised learning or self-learning is used before dealing with the label noise. By using the warm up, it is possible to model the loss into a Mixture Model (Arazo et al., 2019). The main reasoning for using this phase is based on the behaviour of DNNs to learn the clean samples faster than the noisy samples (Arpit et al., 2017).

The next phase of LNL deals with adapting the noise of the distribution and achieve robust classi-

[a] https://orcid.org/0000-0003-2473-2057
[b] https://orcid.org/0000-0001-8261-4409
[c] https://orcid.org/0000-0002-6839-8436
[d] https://orcid.org/0000-0003-0047-5172
[*] IAPR Fellow
[†] Joint first authors

Nagarajan, B., Marques, R., Mejia, M. and Radeva, P.
Class-conditional Importance Weighting for Deep Learning with Noisy Labels.
DOI: 10.5220/0010996400003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 679-686
ISBN: 978-989-758-555-5; ISSN: 2184-4321
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

fiers. Several strategies have been proposed to make the LNL network learn the dataset distribution without the noise (Algan and Ulusoy, 2021). The commonly used Cross-Entropy Loss and Mean Absolute Error are not robust to the underlying noise (Ma et al., 2020) and it is important for the objective functions to be robust to the noise of the underlying distribution. Loss correction methods help in increasing the robustness of losses by modifying the loss functions based on the weights of the labels. In this paper, we propose a class-conditional loss correction method based on the importance of classes. The loss is adapted during each step of the training using weights computed from the classifier scores. This adjustment is carried out such that the classes that are weakly learned are emphasized better during the learning process. To validate the proposed method, we use the Contrast to Divide framework (Zheltonozhskii et al., 2021) and correct the loss during the training phase. Below, we outline the main contributions of this work.

- First, we propose the weighed version of the loss function for LNL. By weighting the unlabeled part of the training data, it is made possible to give more importance to the less learnt or hard to learn classes.
- Second, we do an extensive analysis of various components in the loss function and study the progression of the LNL framework. Moreover, we show improvement with respect to the state of art on LNL on a public dataset.

The rest of the paper is organized as follows. In Section 2, we briefly discuss the related work. We present the details of the proposed technique in Section 3. The experiments and evaluations used to validate the proposed method is explained in Section 4 followed by conclusion in Section 5.

2 RELATED WORK

There are several works in the literature on learning with noisy labels. In this section, we briefly review the recent literature that are relevant to our proposed method.

2.1 Learning with Noisy Labels

There are several classes of LNL algorithms, broadly falling into loss modifications and noise detection schemes. Some methods use label correction (Xiao et al., 2015; Li et al., 2017), where the noisy labels are corrected using inferences made by DNNs, which are in-turn trained only on clean labels, while other methods use loss correction schemes. In this class of algorithms, the network aims at increasing its robustness towards noise by modifying the loss function (Han et al., 2018; Ma et al., 2020). A computationally efficient method based on noise similarity labels was used instead of learning from noisy class labels and was able to reduce the noise rate (Wu et al., 2021). In general, similarity-based approaches have been effective in many LNL algorithms where using a noise transition matrix serves as a bridge between the clean and noisy samples (Hsu and Kira, 2015; Hsu et al., 2019; Wu et al., 2020).

The loss correction methods are based on modifying the loss function with weights during the training of DNNs. Common problems with the existing loss functions are over-fitting of noise and under-learning. Importance weighting schemes have been effective in making the losses more robust to noise (Liu and Tao, 2015; Zhang and Sabuncu, 2018; Yu et al., 2019; Zhang and Pfister, 2021). The Symmetric Cross Entropy loss was created using a Reverse Cross Entropy term along with the Cross Entropy term to make the loss more robust to noise and achieve better learning of the samples (Wang et al., 2019). Normalization techniques proved to make the commonly used loss functions more robust to noise and also by using two robust loss functions to create an Active Passive Loss helped in boosting each other's performance (Ma et al., 2020). Backward and forward noise transition matrices, which are based on matrix inversion and multiplication were pre-computed and shew to increase the robustness of the loss function (Patrini et al., 2017). In the above discussed methods, the basic assumption in a relabeling approach is to have clean labels, which is also a limitation of these algorithms.

Another variation of LNL algorithms focuses on new learning schemes adapted to noisy labels (Malach and Shalev-Shwartz, 2017; Yu et al., 2019). DivideMix (Li et al., 2020) uses a co-teaching strategy to learn two networks simultaneously, so that one network learns from the other networks' confident samples. This algorithm uses a loss to fit a Gaussian Mixture Model in order to divide the samples into labeled and unlabeled set. A Beta-mixture model was also used to model the losses for learning the noise in an unsupervised manner (Arazo et al., 2019). Selective Negative Learning and Positive Learning were used to selectively apply positive learning on expected-to-be-clean data, which is obtained by Negative Learning, where complimentary labels were used instead of the actual labels (Kim et al., 2019). This approach proved to be very effective compared to the normal positive selection of samples. Early

Learning Regularization (Liu et al., 2020) learned the clean samples first, followed by noisy samples in later epochs. This method was beneficial as it prevented the network from memorization of the noisy samples. Data augmentation is also an effective means to combat the noisy label problem (Berthelot et al., 2019b; Li et al., 2020; Berthelot et al., 2019a; Sohn et al., 2020). AugDesc (Nishi et al., 2021) used weak augmentations to learn the loss and strong augmentations to improve the generalizations.

Most of the literature presented above, uses a combination of different LNL schemes to make the network robust to noise. In this paper, we propose a loss correction scheme on top of the already effective DivideMix and Contrast to Divide learning schemes to enhance the learning of models.

2.2 Self-supervised and Semi-supervised Learning

Semi-supervised learning algorithms utilize the unlabeled data by performing providing pseudo-labels to the unlabeled data and adding constraints to the objective functions. Regularization could be consistency regularization or entropy minimization. MixMatch (Berthelot et al., 2019b) combined both the regularization methods to produce labels to the unlabeled classes. ReMixMatch (Berthelot et al., 2019a) and FixMatch (Sohn et al., 2020) were adaptations of MixMatch, which used weakly augmented images to produce labels and predict against the strongly augmented images. It is also beneficial to remove wrong labels that have high levels of noise. By using only a portion of the training set which is correct, the same performance could be achieved (Ding et al., 2018; Kong et al., 2019).

Self-Supervised Learning (SSL) algorithms learn representations in a task-agnostic environment so that the representations are meaningful irrespectively of the labels. Contrastive loss has been vital in the recent success of SSL algorithms, which clusters data points based on the (dis-)similarity of classes (Wang and Liu, 2021). By using these representations, any downstream task could be well learned by the DNNs. SSL algorithms have been widely used in solving the noisy label problems. Since the networks are learned without labels, they are able to produce features that are robust to noise (Cheng et al., 2021). Data re-labeling helps in increasing the effectiveness of DNNs. The performance was boosted by using a parallel network to learn the portion of clean labels (Mandal et al., 2020). Supervised learning and self-supervised learning can also be used together as a co-learning scheme as this could maximize the learning behaviour using both the constraints (Tan et al., 2021; Huang et al., 2021). Contrastive DivideMix (Zhang et al., 2021b) fuses the contrastive and semi-supervised learning algorithms.

DivideMix (Li et al., 2020) uses a semi-supervised training phase. It uses the MixMatch algorithm to perform label co-refinement and co-guessing on labeled and unlabeled samples. This works on per-sample loss behaviour and has been an effective technique to model the noise. One of the bottleneck in this method is the warm-up phase. This was overcome using the Contrast to Divide (Zheltonozhskii et al., 2021) method. Instead of using a supervised learning in DivideMix, this algorithm used a self-supervised learning method. In our proposed approach, we add an importance weighting scheme that would enable the algorithms to focus selectively on the classes.

3 IMPORTANCE WEIGHTING

In this section, we first brief the rationale behind the approach. We provide background information followed by the proposed weighted scheme.

3.1 Rationale

Our approach is motivated by the observation that learning is unbalanced across classes, that is, after a given number of epochs, the accuracy of the model tends to vary significantly over different classes. Our hypothesis is that, by focusing the learning effort in those classes for which the model is currently less efficient, the overall accuracy of the model can be improved. To test this hypothesis, we propose to enhance the DivideMix algorithm (Li et al., 2020) with a class-conditional importance weighting scheme which assigns a larger weight to the classes for which the model has a poorer performance.

3.2 Background

At each epoch, the DivideMix algorithm, on which we build, separates the training set into two disjoint sets: a set $\mathcal{X}$ containing potentially clean data, and a set $\mathcal{U}$ containing potentially noisy data. This separation between clean and noisy data is made by fitting a Gaussian mixture model to the softmax output of a pretrained network (Li et al., 2020). The loss function used for training thus combines the losses on both the potentially clean and noisy sets, and is given by (Li et al., 2020):

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_u \mathcal{L}_{\mathcal{U}} + \lambda_r \mathcal{L}_{\text{reg}}\,, \tag{1}$$

where $\mathcal{L}_{\mathcal{X}}$ is the cross-entropy loss over the augmented and mixed *clean* data $\mathcal{X}'$; $\mathcal{L}_{\mathcal{U}}$ is a mean squared error loss over the augmented and mixed *noisy* data $\mathcal{U}'$; and finally, $\mathcal{L}_{\text{reg}}$ is a regularization term used to encourage the model to evenly distribute its predictions across all classes.

The loss $\mathcal{L}_{\mathcal{U}}$ of the noisy data is defined as:

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{|\mathcal{U}'|} \sum_{(x,p)\in\mathcal{U}'} ||p - \mathrm{p}_\theta(x)||_2^2 \,,$$

where $|\mathcal{U}'|$ is the number of noisy samples at the current epoch, p is the label assigned to each noisy sample x through co-guessing (Li et al., 2020), and $\mathrm{p}_\theta(x)$ is the model prediction for x given the current model parameters θ. The regularization term $\mathcal{L}_{\text{reg}}$, in its turn, is given by:

$$\mathcal{L}_{\text{reg}} = \sum_c \pi_c \log\left(\pi_c \left(\frac{1}{|S|}\sum_{x\in\mathcal{S}} \mathrm{p}_\theta^c(x)\right)^{-1}\right), \quad (2)$$

where $\mathcal{S} = \mathcal{X}' + \mathcal{U}'$, and $\pi_c = 1/C$ is a uniform prior distribution over the probability of each class in $\mathcal{S}$. Providing a uniform prior distribution $\pi_c = 1/C$ in Equation (2) causes the loss to be minimal when the model yields exactly the same number of predictions for all classes in the data set.

3.3 Class-conditional Importance Weighted Loss

We now describe the approach taken to assign a weight for each class. Let $\mathbf{f}$ be a vector of C elements, C being the number of classes in the data set. Each element $f_c \in \mathbf{f}$ is given by $f_c = 1 - F_1^c$, where F_1^c represents the F_1 score for a particular class c at the current epoch. The vector $\mathbf{f}$ is then smoothed over a window of n_e epochs, yielding $\mathbf{f}'$. Then, the weight vector $\mathbf{w}$ is computed as:

$$\mathbf{w} = \frac{\max(\lambda_w, \mathbf{f}')}{|\max(\lambda_w, \mathbf{f}')|} \times C \,, \quad (3)$$

where λ_w is a hyperparameter which has the role of limiting how far the resulting weights can deviate from the value 1. Hence, the resulting weights $\mathbf{w} = \{w_1, \ldots, w_C\}$ of Equation (3) take a larger value for those classes for which the F_1 score is smaller.

Furthermore, to account for the importance of each class in the learning phase, we introduce the weights $\mathbf{w}$ (Equation (3)) in the loss function $\mathcal{L}_{\mathcal{U}}$ of the unlabeled set (Equation (3.2)), yielding:

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{|\mathcal{U}'|} \sum_{(x,p)\in\mathcal{U}'} w_c ||p - \mathrm{p}_\theta(x)||_2^2 \,, \quad (4)$$

where w_c is the weight of class c, and c is the class of the image x. Finally, to apply the weights to the regularization loss $\mathcal{L}_{\text{reg}}$, we simply replace the uniform prior distribution $\pi_c = 1/C$ used in DivideMix (Equation (2)) by a non-uniform prior based on the weights $\mathbf{w}$, such that:

$$\pi_c = \frac{w_c}{C} \,, \quad (5)$$

where w_c is the class weight according to Equation (3), and C is the number of classes in the dataset. This way, when the model performs poorly for a given class due to not choosing that same class as many times as it should, using the prior specified in Equation (5) will encourage the model to increase the number of prediction for this same class. Figure 1 shows the pipeline of our proposed approach.

4 EXPERIMENTS

We evaluate our proposed framework following the common methodology in synthetic noise benchmarks. We use CIFAR-10 (Krizhevsky and Hinton, 2009) to validate the method, varying the amount of injected noise. We measure the performance of the networks using accuracy as an evaluation metric. We provide the accuracy over five runs for each noise ratio, following the results presented in Contrast to Divide (Zheltonozhskii et al., 2021).

4.1 Implementation Details

Similarly to (Li et al., 2020), we used a PreAct ResNet-18 architecture (He et al., 2016) for DivideMix. Moreover, we coupled DivideMix with *Contrast to Divide* (C2D), which has been shown to considerably boost the original DivideMix algorithm (Zheltonozhskii et al., 2021). As regards the injected noise in the CIFAR-10 data set, we used two types of label noise: symmetric and asymmetric. Given a target noise ratio, the symmetric noise is generated by randomly substituting the original label by a randomly selected new label chosen with a uniform probability over the rest of the class labels. Regarding the asymmetric noise, we follow (Zheltonozhskii et al., 2021) who designed the noise so as to mimic the structure of real-world noise labels substituting the labels with those of the most similar classes. In each experiment, the networks are optimized during 360 epochs.

Regarding the λ_r hyperparameter, we follow (Li et al., 2020) and set its value to 1. To compute the weights per class, we set $\lambda_w = 0.1$ in Equation (3), and use a smoothing window of 5 epochs (i.e., $n_e = 5$). The problem of selecting λ_u is discussed below.

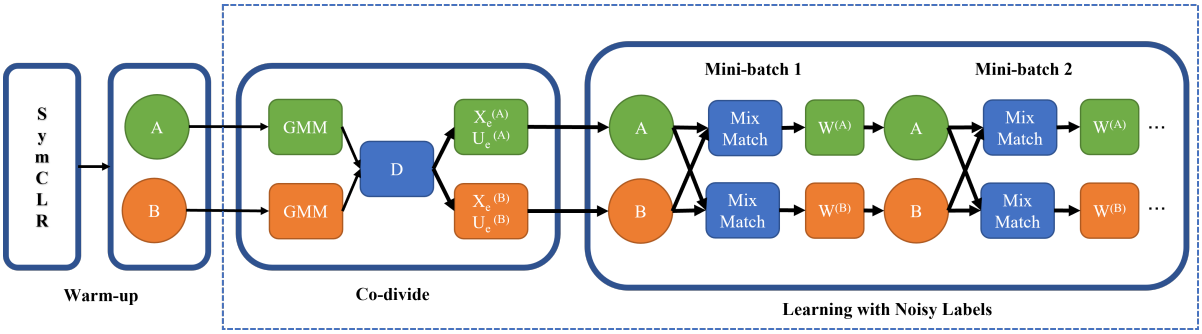


Figure 1: Pipeline of our proposed approach. The symCLR component pre-trains networks A and B. Then, these undergo the DivideMix warm-up phase where the two networks are trained on all the data set during a small number of epochs using standard cross-entropy loss. Then, at each epoch, the co-divide is applied to divide the data set in two disjoint sets, yielding the set of the clean and of the noisy labels ($\mathcal{X}$ and $\mathcal{U}$, respectively). For each mini-batch, networks A and B are then trained separately using MixMatch and our proposed weighting scheme.

Table 1: Study of the optimal λ_u value. The table shows the peak and final accuracy on CIFAR-10 for $5 \leq \lambda_u \leq 50$.

Method		5	10	15	20	25	30	35	40	45	50
Weighted C2D	Peak	91.73	92.45	92.92	93.50	93.50	93.60	93.58	**93.69**	93.52	93.66
+DM (90%)	Final	91.45	92.35	92.73	93.33	93.48	93.53	93.47	**93.28**	93.42	93.57

4.1.1 Selection of λ_u

As in (Li et al., 2020) and (Zheltonozhskii et al., 2021), the performance of our proposed approach can vary significantly depending on the used parameter λ_u, i.e., the hyperparameter specifying the weight of the unsupervised loss $\mathcal{L}_{\mathcal{U}}$ in the final loss (see Equation (1)). Therefore, in Table 1, we provide a detailed analysis of the effect of this hyperparameter on the final accuracy reached by our method when considering 90% of symmetric noise. The results show that the optimal λ_u for our method with a noise ratio of 90% is of 40. This value is roughly in-line with the one (50) reported by (Li et al., 2020).

Following a similar approach, we collected a set of selected λ_u values for each considered noise ratio value. We can observe that the optimal λ_u value found seems to decrease with the noise ratio present in the dataset. This seems to indicate that, the larger the amount of noisy labels present in the used dataset, the more relevant the loss $\mathcal{L}_{\mathcal{U}}$ of the unlabeled data becomes in the learning process. The set of selected λ_u values for each considered noise ratio value are shown in Table 2. These values are used henceforth in all experiments for each corresponding noise value.

Table 2: Table with the selected λ_u values for each noise ratio.

Noise Ratio	20%	50%	80%	90%	40% (asym)
λ_u	0	25	30	40	0

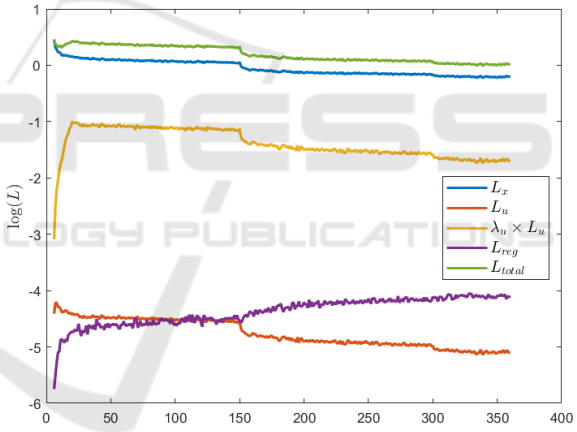


Figure 2: Different terms of the loss function as a function of the epoch. The plot was generated for CIFAR-10 with 80% of symmetric noise and $\lambda_u = 30$.

At this stage, it is interesting to analyze the role of the λ_u hyperparameter in the total loss value of Equation (1). Figure 2 depicts the different terms of the used loss function. We can observe that the hyperparameter λ_u acts as a scaling factor which brings $\mathcal{L}_{\mathcal{U}}$ (in red) up to a magnitude in which it can actually influence the final loss function shape (in orange the product $\lambda_u \times \mathcal{L}_{\mathcal{U}}$, and in green the final loss, denoted by L_{total} in Figure 2). It is also apparent from the curves that the loss of the labeled data ($\mathcal{L}_{\mathcal{X}}$, in blue) is the one that dominates the total loss shape. Finally, the regularization loss $\mathcal{L}_{\text{reg}}$ seems to have a rather marginal role on the overall optimization process.

Table 3: Peak and final accuracy (%, mean ± std over five runs) on CIFAR-10. DivideMix and C2D+DM results are obtained from literature.

Method		20%	50%	80%	90%	40% (asym)
DivideMix	Peak	96.1	94.6	93.2	76.0	-
	Final	95.7	94.4	92.9	75.4	-
C2D+DM	Peak	96.43±0.07	95.32±0.12	**94.40±0.04**	93.57±0.09	93.45±0.07
	Final	96.23±0.09	95.15±0.16	**94.30±0.12**	93.42±0.04	90.75±0.35
Weighted C2D+DM (ours)	Peak	**96.50±0.07**	**95.79±0.06**	**94.40±0.05**	**93.70±0.16**	**93.62±0.09**
	Final	**96.40±0.21**	**95.56±0.07**	94.24±0.09	**93.54±0.13**	**92.83±0.21**

4.2 Results and Analysis

The results for the application of our proposed method to the CIFAR-10 dataset are shown in Table 3, where a comparison with the results of the original method is provided. The results show that, when using our importance weighting scheme, the accuracy results generally improve over that of C2D+DM, and it never performs worse. Indeed, except for a noise level of 80%, our method delivers consistent improvements over its non-weighted counterpart. This confirms our hypothesis that the overall efficiency of the algorithm can be improved by focusing the learning effort in those classes that the model is having more difficulty to learn. Moreover, it also validates our weighting strategy based on the F_1 score proposed in Equation (3).

A detailed illustration of the weights values throughout the learning process is provided in Figure 3. It shows that the weights for a given class remain coherent through the learning phase, since we are able to clearly identify each class (corresponding to a particular color) through the weights plot. They also show that the weights for each class converge to a particular value, which is determined by the F_1 score that the model is able to get for each particular class as the learning progresses.

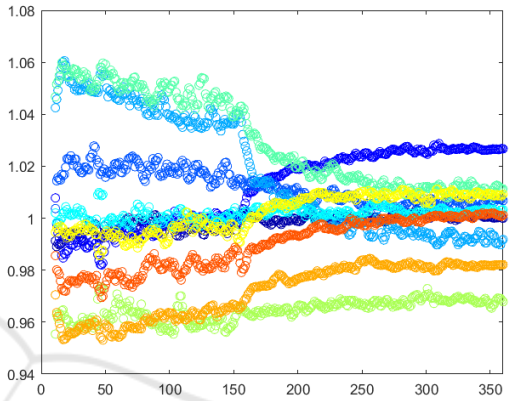


Figure 3: Illustration of the weights assigned to each class (y-axis) during 355 training epochs (x-axis, 5 warm-up epochs + 355 of DivideMix). The results are generated using the weights of a single network. Each color corresponds to a different class in the dataset (total of 10).

4.3 Ablation Study

In this section, we study the effect of weights in the $\mathcal{L}_{\mathcal{U}}$ and $\mathcal{L}_{\text{reg}}$ terms individually. We show the results of this ablation study in Table 4. They show that when the weights are applied to only one of the two considered terms ($\mathcal{L}_{\mathcal{U}}$ and $\mathcal{L}_{\text{reg}}$), the accuracy is inferior to the case in which the weights are included in both losses.

Table 4: Ablation study. The entries for C2D+DM and weighted C2D+DM is mean over five runs, whereas the other two are mean over two runs.

Method		80% ($\lambda_u = 30$)	90% ($\lambda_u = 40$)
C2D+DM	Peak	**94.40±0.04**	93.57±0.09
	Final	**94.30±0.12**	93.42±0.04
Weights in $\mathcal{L}_{\mathcal{U}}$ Only	Peak	94.23±0.12	93.68±0.09
	Final	94.11±0.16	93.51±0.16
Weights in $\mathcal{L}_{\text{reg}}$ Only	Peak	94.28±0.06	93.58±0.09
	Final	94.11±0.05	93.37±0.11
Weighted C2D+DM	Peak	**94.40±0.05**	**93.70±0.16**
	Final	94.24±0.09	**93.54±0.13**

5 CONCLUSIONS

In this paper, we propose a class-conditional dynamically weighted Contrast to Divide algorithm, where the weights emphasize the learning behaviour of individual classes. Here, we use a per-class importance weighting scheme based on F1-score obtained in each epoch. Our importance weighting approach proved to outperform the state of the art for the CIFAR-10 data set in all the noise rates. We studied the behavior of λ_u in different noise rates and also analysed the weights throughout the learning process. The results prove the effectiveness of the proposed scheme on an existing state of the art LNL approach.

Although, the algorithm has shown performance improvements, it is important to study the behaviour in more complex data sets such as CIFAR-100, Clothing 1M and WebVision. In this paper, we have used F1-score to create the weights, however, other methods have to be studied to compute the weights per class, which can eventually improve the results presented here. This information regarding the performance per class (i.e., F_1 score or other) can be used to improve other stages of the original DivideMix algorithm, such as, for example, the division between clean and noisy data.

ACKNOWLEDGEMENTS

This work was partially funded by TIN2018-095232-B-C21, SGR-2017 1742, Greenhabit EIT Digital program and CERCA Programme / Generalitat de Catalunya. Bhalaji Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

REFERENCES

Algan, G. and Ulusoy, I. (2021). Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771.

Arazo, E., Ortego, D., Albert, P., O'Connor, N., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. (2019a). Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019b). Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32.

Chen, Y., Shen, X., Hu, S. X., and Suykens, J. A. (2021). Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2688–2692.

Cheng, H., Zhu, Z., Sun, X., and Liu, Y. (2021). Demystifying how self-supervised features improve training from noisy labels. *arXiv preprint arXiv:2110.09022*.

Cordeiro, F. R. and Carneiro, G. (2020). A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations? In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 9–16. IEEE.

Ding, Y., Wang, L., Fan, D., and Gong, B. (2018). A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1215–1224. IEEE.

Frénay, B., Kabán, A., et al. (2014). A comprehensive introduction to label noise. In *ESANN*. Citeseer.

Frénay, B. and Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Ghiasi, G., Zoph, B., Cubuk, E. D., Le, Q. V., and Lin, T.-Y. (2021). Multi-task self-training for learning general representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8856–8865.

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. (2018). Masking: A new perspective of noisy supervision. *Advances in Neural Information Processing Systems*, 31:5836–5846.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32:15663–15674.

Hsu, Y.-C. and Kira, Z. (2015). Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*.

Hsu, Y.-C., Lv, Z., Schlosser, J., Odom, P., and Kira, Z. (2019). Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*.

Huang, L., Zhang, C., and Zhang, H. (2021). Self-adaptive training: Bridging the supervised and self-supervised learning. *arXiv preprint arXiv:2101.08732*.

Kim, Y., Yim, J., Yun, J., and Kim, J. (2019). Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 101–110.

Kong, K., Lee, J., Kwak, Y., Kang, M., Kim, S. G., and Song, W.-J. (2019). Recycling: Semi-supervised learning with noisy labels in deep neural networks. *IEEE Access*, 7:66998–67005.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918.

Liao, Y.-H., Kar, A., and Fidler, S. (2021). Towards good practices for efficiently annotating large-scale image classification datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4359.

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33.

Liu, T. and Tao, D. (2015). Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., and Bailey, J. (2020). Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR.

Malach, E. and Shalev-Shwartz, S. (2017). Decoupling" when to update" from" how to update". *Advances in Neural Information Processing Systems*, 30:960–970.

Mandal, D., Bharadwaj, S., and Biswas, S. (2020). A novel self-supervised re-labeling approach for training with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. *Advances in neural information processing systems*, 26:1196–1204.

Nigam, N., Dutta, T., and Gupta, H. P. (2020). Impact of noisy labels in learning techniques: a survey. In *Advances in data and information sciences*, pages 403–411. Springer.

Nishi, K., Ding, Y., Rich, A., and Hollerer, T. (2021). Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8022–8031.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. (2021). Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

Tan, C., Xia, J., Wu, L., and Li, S. Z. (2021). Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1405–1413.

Wang, F. and Liu, H. (2021). Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330.

Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., Liu, H., and Niu, G. (2020). Multi-class classification from noisy-similarity-labeled data. *arXiv preprint arXiv:2002.06508*.

Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., Liu, H., and Niu, G. (2021). Class2simi: A noise reduction perspective on learning with noisy labels. In *International Conference on Machine Learning*, pages 11285–11295. PMLR.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021a). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Zhang, X., Liu, Z., Xiao, K., Shen, T., Huang, J., Yang, W., Samaras, D., and Han, X. (2021b). Codim: Learning with noisy labels via contrastive semi-supervised learning. *arXiv preprint arXiv:2111.11652*.

Zhang, Z. and Pfister, T. (2021). Learning fast sample reweighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 725–734.

Zhang, Z. and Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*.

Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M., and Litany, O. (2021). Contrast to divide: Self-supervised pre-training for learning with noisy labels. *arXiv preprint arXiv:2103.13646*.

CCLM: Class-Conditional Label Noise Modelling

Albert Tatjer[1], Bhalaji Nagarajan[1(✉)], Ricardo Marques[1], and Petia Radeva[1,2]

[1] Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona, Spain
{acatalta11@alumnes.ub.edu,
bhalaji.nagarajan,ricardo.marques,petia.ivanova}@ub.edu
[2] Computer Vision Center, Cerdanyola, Barcelona, Spain

Abstract. The performance of deep neural networks highly depends on the quality and volume of the training data. However, cost-effective labelling processes such as crowdsourcing and web crawling often lead to data with noisy (i.e., wrong) labels. Making models robust to this label noise is thus of prime importance. A common approach is using loss distributions to model the label noise. However, the robustness of these methods highly depends on the accuracy of the division of training set into clean and noisy samples. In this work, we dive in this research direction highlighting the existing problem of treating this distribution globally and propose a class-conditional approach to split the clean and noisy samples. We apply our approach to the popular DivideMix algorithm and show how the local treatment fares better with respect to the global treatment of loss distribution. We validate our hypothesis on two popular benchmark datasets and show substantial improvements over the baseline experiments. We further analyze the effectiveness of the proposal using two different metrics - Noise Division Accuracy and Classiness.

Keywords: Learning with Noisy Labels · Label Noise Modelling · Class-conditional data splitting

1 Introduction

Deep Neural Networks (DNNs) have gained immense attention in the research community due to their success in various challenging domains. High-performing models in all verticals require good quality data of huge volume. However, it is difficult to create such large datasets with high precision as it is labour-intensive, both in collecting and labelling the samples [16]. Crowdsourcing and using semi-automatic labelling pipelines on web-scrapped data reduce the cost of creating datasets. However, they lead to the introduction of noise in the assigned labels [23,24,30]. As a result, real-world datasets tend to have significant label noise,

A. Tatjer and B. Nagarajan—Joint First Authors.
P. Radeva—IAPR Fellow.
R. Marques—Serra Húnter Fellow.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
A. Pertusa et al. (Eds.): IbPRIA 2023, LNCS 14062, pp. 3–14, 2023.
https://doi.org/10.1007/978-3-031-36616-1_1

estimated to be in the range of 8.0% to 38.5% [26]. Modern DNNs have a high number of learnable parameters compared to the size of the dataset and often result in overfitting to the label noise [32].

Learning with Noisy Labels (LNL) was introduced in the late 1980s s [1] and has been a long-studied problem with a focus on making the models robust to the label noise. Loss modification-based methods use noise distribution to create more robust loss functions [19], replacing the cross entropy loss or using a noise transition matrix [25]. Correctly estimating the noise and creating methods that are robust to high noise levels are very challenging tasks. Other methods that use sample selection [18] or reweighting [33] use *ad-hoc* criteria to select noisy samples and reduce the impact on the learning process. The challenge here is the selection of a good criterion to split the clean and noise samples.

Semi-supervised learning (SSL) is a helpful strategy for reducing the cost of annotating datasets. SSL-based frameworks use a limited subset of the original dataset being carefully labelled and learn the unlabelled data, which is multi-fold larger. Recent LNL methods aimed at combining SSL methods and robust loss methods. DivideMix [15] is a popular benchmark that uses two networks to learn the noisy data. At each iteration, one of the networks divides the data into a clean and a noisy datasets, which are then used to train the other network. DivideMix formed the basis for a new family of algorithms [13,27,34]. Several methods based on DivideMix have improved various stages of the pipeline. One of the important questions in most of these methods is *How to effectively demarcate the clean and the noisy samples?* As can be seen with all the works, the selection of clean samples plays an important role in the subsequent sub-tasks. The usage of loss distribution to split the samples has been a well-documented technique. However, not all samples in the training set are learned in the same rate, which hinders noise detection using this technique [21]. Here, we explore this global loss conundrum. We study the division in detail and propose a per-class label noise modelling. We adapt the noise model for each class and create a local threshold used to separate the clean and noisy samples. In order to further study this effect, we propose new metrics to identify the class behaviour of the LNL algorithm. The key contributions of this paper are as follows:

- First, we highlight and study the noise modelling obstacle present in the existing LNL algorithms.
- Second, we propose a class-conditional label noise modelling approach. By replacing the existing global-noise modelling in the popular DivideMix algorithm, we show how the class-conditional approach benefits the algorithm.
- Finally, we introduce new metrics to understand the importance of learning with class importance and show the improvement using two public datasets.

2 Related Works

Typical approaches in LNL algorithms include the use of advanced techniques such as sample selection, loss correction, label correction, sample reweighting or using robust loss methods [8]. Sample selection methods deal with finding clean

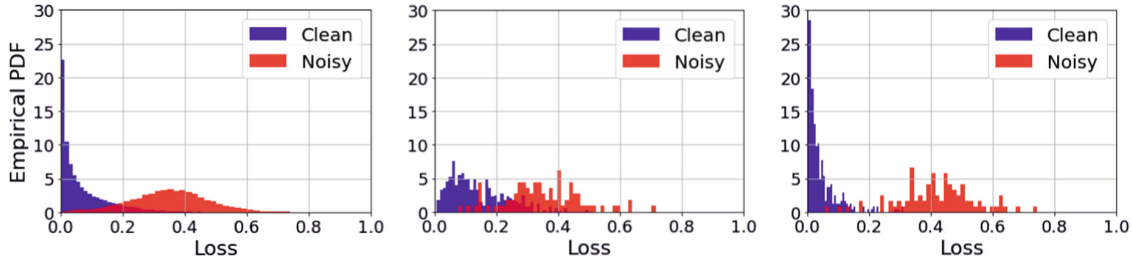
samples using different strategies such as small loss [9] and topological information [31]. Loss correction is achieved by reweighting the loss to avoid overfitting on noisy samples, which helps mitigate the errors introduced by wrongly labelled samples. Similarly, in terms of sample reweighting, the samples that are noisiest are identified and weighted less compared to the clean samples [18]. Another family of LNL algorithms attempt to correct the noise (i.e., errors) in the labels, typically resorting to the notion of *transition matrix*. [7,11,25]. Most of these approaches are based on the assumption that there is a single transition probability between the noisy label and the ground-truth label [10]. Several noise tolerance loss functions have also been proposed to replace the conventional cross-entropy loss [19,20,35]. Recent studies on LNL have shown that hybrid approaches can further boost the algorithms and make them more robust towards label noise [6]. A comprehensive survey on the various LNL algorithms is presented in [26].

Co-teaching [9] of two networks by iteratively selecting clean samples (training samples having small loss) from the other network proved to be very successful. DivideMix [15] used two independent networks for sample selection and then adopted MixMatch [3] to further boost the label correction. DivideMix is a popular benchmarking algorithm in LNL problems. Augmentation plays an important role in LNL problems and was studied over DivideMix [22]. C2D further explored the warmup stage of DivideMix [34]. Probabilistic Noise Prediction [27] used two networks, one to predict the category label and the other to predict the noise type. ProMix [29] used an iterative selection process to select the clean samples using a high-confidence selection technique. SplitNet [13] used an additional learnable module to assist in splitting the clean and noisy samples.

Importance-weighted risk minimization [4] has been studied and employed in different DNNs over the years. A common scheme of importance weighting is weighting the loss terms [12] between different learning tasks. Class-conditional approaches were adapted to LNL algorithms. Generalized Data Weighting via Class-level Gradient Manipulation [5] converted the loss gradient into class-level gradients by applying the chain rule and reweighting the gradients separately. Noise transition matrix [33] was used to capture the class conditional label corruption and used total variation regularization to create more distinguishable predictions. Class-dependent softmax classifier [17] was used to entangle the classification of multi-class features. Our proposed approach is motivated by the LNL methods that deal with the separation of clean and noisy data using different means. As we observe during our study, using a global approach to all the samples is suboptimal and using a class-conditional split at the early stages of the algorithm makes the models more robust towards label noise.

3 The Label Noise Modelling Obstacle

Algorithms which allow efficient learning in the presence of noisy labels often require the division of the training dataset between clean and noisy samples [15]. The samples are then treated differently by the learning algorithm, to extract



(a) All classes. (b) Labelled *otter*. (c) Labelled *road*.

Fig. 1. Empirical per-sample loss distribution between the clean samples and the noisy samples after the warm-up phase (CIFAR-100 20% sym. noise).

the maximum amount of information possible from each of the partitions of the data via semi-supervised learning [3]. The correct division between clean and noisy data is thus crucial for effective learning in the presence of noisy labels. Indeed, if too many noisy labels are classified as clean, then the algorithm risks trusting wrongly labelled data, eventually introducing important biases in the training process and leading to poor results. Conversely, if too many clean samples are deemed noisy, the algorithm would risk disposing of important information, hence potentially impairing the learning outcome. A common approach to splitting the dataset into clean and noisy samples is to create a noise model of the *whole* training data, allowing to characterize the probability of a given sample being clean (i.e., correctly labelled). Given the obvious limitations, it is common to rely on some proxy measure such as loss, which can be used to estimate to which extent a sample is likely to be clean or noisy [2]. We provide a more detailed analysis of this approach and show how the global noise model is suboptimal for some particular classes.

Global Label Noise Modelling. The typical approach to the label noise detection problem relies on the samples' loss to determine whether a given sample is deemed noisy or clean. The sample loss is obtained using a NN model, and the underlying principle is that the NN will tend to yield larger loss values to noisy samples because wrongly labelled data are a more complex pattern and therefore are more difficult to learn [28]. Given a sample x in dataset D, its corresponding one-hot encoded label y and a model p_θ parameterized with the set of parameters θ, the individual loss l of sample x is defined using the cross-entropy loss $l\left(p_\theta(x), y\right)$. The distribution of the cross-entropy loss is shown in Fig. 1a. The distribution appears to follow two modes, with the clean samples (in blue) exhibiting a characteristic loss generally lower than the noisy samples (in red). To turn the cross-entropy loss into a probability of a given sample being correctly (or wrongly) labelled, Li et al. [15] fit a two-component 1-D Gaussian Mixture model (GMM) to the loss distribution l. They select the GMM component g with lower mean and use it to model the probability ω_i of a given sample x_i with loss l_i to be clean. Finally, they use a threshold τ on ω_i to partition the dataset into a clean set $\mathcal{C}_D$ and a noisy set $\mathcal{N}_D$, given by:

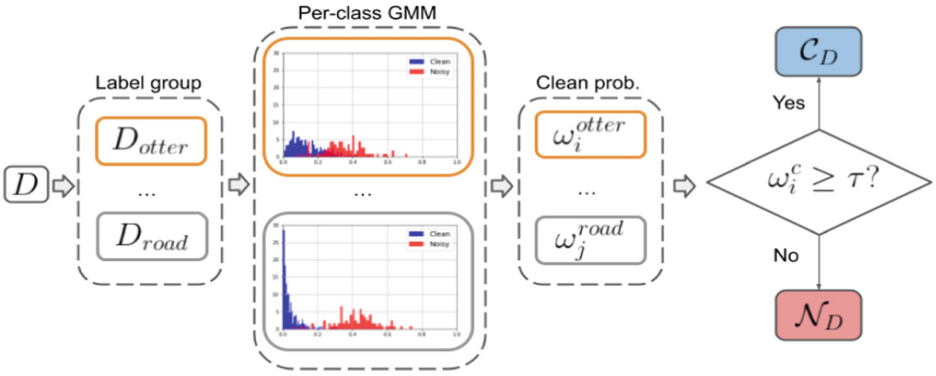


Fig. 2. CCLM splits the dataset according to labels, fitting a GMM to each independent set and yielding the Clean-Noise partition from a decision boundary on each label group's probability distribution.

$$\begin{aligned} \mathcal{C}_D &= \{(x_i, y_i) | \, \omega_i \geq \tau\}_{(x_i,y_i)\in D} \\ \mathcal{N}_D &= \{(x_i, y_i) \, | \, \omega_i < \tau\}_{(x_i,y_i)\in D} \, , \end{aligned} \tag{1}$$

where $\omega_i = p(g|l_i)$ and $l_i = l(p_\theta(x_i), y_i)$.

Limitations of Global Label Noise Modelling: The global noise modelling approach described above assumes that all samples are i.i.d and, therefore, that their per-sample loss tends to behave consistently. However, with the presence of noisy labels, it is obvious that some classes might be harder to learn than others, leading to potentially different characteristic losses across different classes. This situation can be appreciated in Fig. 1, where the per-sample loss distribution for the 'otter' class (Fig. 1b) differs from that of the 'road' class (Fig. 1c) and, more importantly, from the global per-sample loss distribution of the whole dataset (Fig. 1a). We identify this difference between the loss distribution of a particular class and the global loss distribution as an obstacle towards the correct partition of the data into a clean set, $\mathcal{C}_D$ and a noisy set, $\mathcal{N}_D$ which, in its turn, might hinder the training of NN models in the presence of noisy labels. Moreover, we conjecture that taking a class-conditional approach to the problem of label noise detection can lead to better dataset partitions, and thus to superior model accuracy after training in the presence of noisy labels. In the following, we propose a class-conditional approach to the problem of noise label detection, aimed at overcoming the aforementioned limitations of global label noise modelling.

4 Proposed Methodology

Typical approaches in LNL assume all samples to be i.i.d. Consequentially, label noise modelling is performed leveraging solely per-sample loss information, which can impair the accuracy of the trained model. We propose an alternative label-noise model we call Class-Conditional Local Noise Model (CCLM) that allows

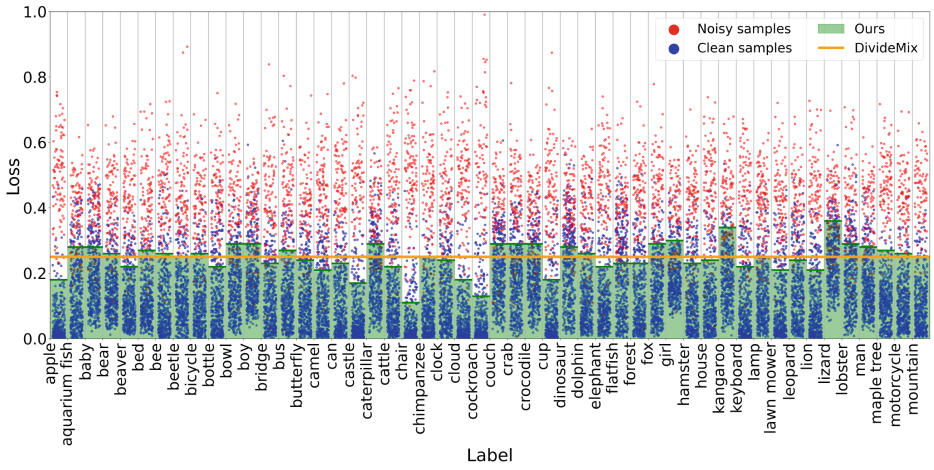


Fig. 3. Per-sample loss distribution of 50 of the CIFAR-100 label groups (20% sym.). Label groups are formed according to the potential noisy labels.

to locally adapt the noise model to the specificity of each considered class, hence improving the overall accuracy of the noisy label detection task. To achieve this, we explicitly include the label class information in the noise modelling process. Our proposed approach is depicted in Fig. 2. We take as input the training set D composed of pairs $(x_i, y_i)_{i=1}^{N}$, N being the total number of samples in D. Then, the dataset is split by sample label y, yielding C disjoint subsets of D such that $D = \cup_{c=1}^{C} D_c$, with C being the total number of possible classes. Then, for each subset D_c, we fit a two-component GMM to the per-sample loss and use the GMM component with a lower mean to model the probability of the samples being clean. We use the threshold τ over the probability of a sample being clean to split the data into noisy and clean within each class. In other words, the probability of a sample being clean is conditional on its loss and its label, $\omega_i^{y_i} = p(g|l_i, y_i)$. Finally, the Clean-Noisy partition is given by:

$$\begin{aligned} \mathcal{C}_D &= \cup_{c=1}^{C} \mathcal{C}_{D_c} = \cup_{c=1}^{C} \{(x_i, y_i) \,|\, \omega_i^c \geq \tau\}_{(x_i,y_i)\in D_c} \\ \mathcal{N}_D &= \cup_{c=1}^{C} \mathcal{N}_{D_c} = \cup_{c=1}^{C} \{(x_i, y_i) \,|\, \omega_i^c < \tau\}_{(x_i,y_i)\in D_c}, \end{aligned} \tag{2}$$

Figure 3 illustrates the result of our noise detection model on 20% noise of the CIFAR-100 dataset. The y-axis represents the per-sample loss associated with each sample, with noisy samples being represented in red, and clean samples in blue. The orange horizontal line depicts the split between noisy and clean labels using the typical global noise modelling approach. The green area depicts the per-class split between noisy and clean labels using our CCLM. We clearly see that the shape of the per-sample loss distribution varies across classes (e.g., 'apple' vs. 'aquarium fish'). Moreover, it can be seen that the global noise approach does not distinguish between classes, yielding a single decision boundary for all classes independently of their particular loss features. In contrast, our CCLM provides a locally adapted division between clean and noisy data which allows improving both the split accuracy and the NN accuracy when trained in the presence of noisy labels.

Table 1. Best and Last accuracy on CIFAR-10 and CIFAR-100.

Method		CIFAR-10					CIFAR-100				
		20%	50%	80%	90%	Asym. 40%	20%	50%	80%	90%	Asym. 40%
DivideMix [15]	Best	96.1	94.6	93.2	76	**93.4**	77.3	74.6	60.2	31.5	72.2**
	Last	95.7	94.4	92.9	75.4	92.1	76.9	74.2	59.6	31.0	72.4**
CCLM (ours)	Best	**96.5**	**95.6**	**93.7**	**83.6**	92.6	**79.5**	**76.4**	**61.1**	**33.5**	**75.4**
	Last	96.3	95.3	93.6	82.4	91.5	79.1	75.9	60.9	33.0	75.1

** Results reported from Contrast2Divide [34]. DivideMix does not report this setting.

5 Experiments and Results

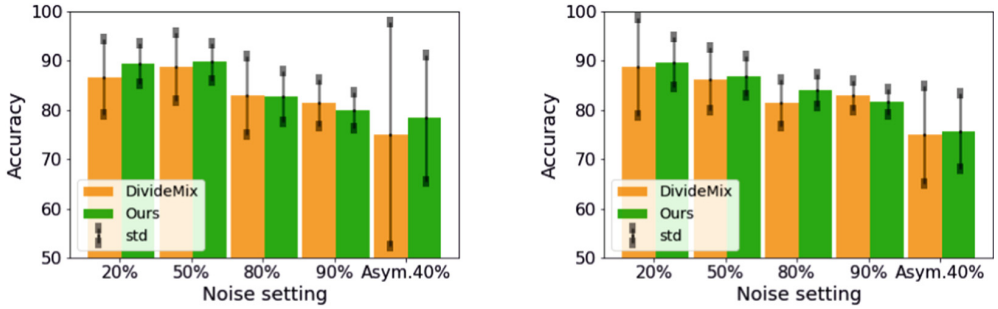
In this section, we present the datasets and the implementation details for evaluating our proposal. We show the performance with two types of label noise - symmetric and asymmetric and compare them against the baseline DivideMix.

5.1 Datasets and Implementation Details

We conduct our evaluations using two benchmark datasets - CIFAR-10 and CIFAR-100 [14]. Both CIFAR datasets contain images of 32×32 RGB pixels, with 50k training samples and 10k test samples. To maintain consistency across the benchmark methods, we follow the same train/test split [15,34]. We conduct our experiments on two different noise types. *Symmetric* (sym.) noise is generated by replacing the labels for a percentage of the training data with a uniform distribution over all the possible labels. *Asymmetric* (asym.) noise is injected by replacing labels with similar classes (e.g. deer → horse, dog ↔ cat). For both datasets, we use 18-layer PreAct Resnet following the settings of DivideMix [15]. We keep all the hyperparameters the same as in DivideMix. We use a batch size of 128 and train the models for 300 epochs using SGD optimizer with a momentum of 0.9 and weight decay of 0.0005. We set an initial learning rate of 0.02 and reduce it by a factor of 10 after 150 epochs. For CIFAR-10, we use a warmup of 10 epochs and for CIFAR-100, we use a warmup of 30 epochs. We perform all experiments using PyTorch on NVIDIA RTX2080Ti GPU. Following previous works [15,34], we report the best test accuracy across all epochs and the average test accuracy over the last 10 epochs (identified as Last).

5.2 Results

First, we show the overall performance of the CCLM against DivideMix, which follows a global noise model approach. We compare it against DivideMix as it is considered a benchmark in LNL. The improvements brought by our proposed method in terms of label noise detection are likely to have a positive impact on algorithms which rely on the partition of the data between clean and noisy.

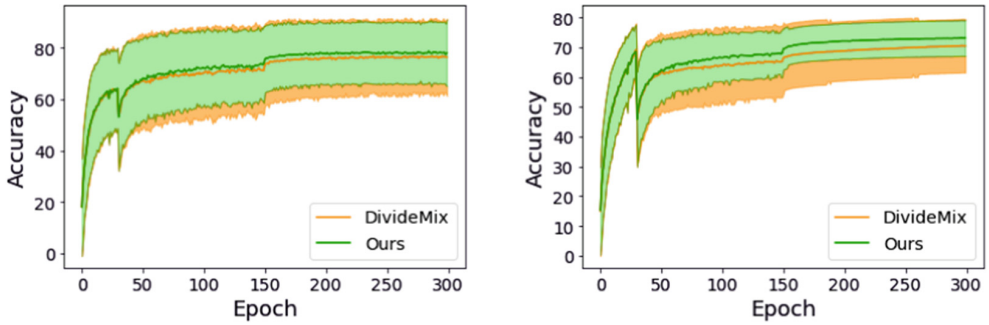


(a) CIFAR-10. (b) CIFAR-100.

Fig. 4. Clean-Noise split accuracy and std. dev. at the first epoch after warm-up.

Performance. Table 1 shows the test accuracy on CIFAR-10 and CIFAR-100 with different levels of label noise. Our CCLM outperforms the global noise label modelling approach of DivideMix in all the noise settings on CIFAR-100. On CIFAR-10, our approach fares better in all the noise settings except for the asym. noise setting. This superior performance is explained by the local adjustment (i.e., per class) of the label noise model, yielding a more accurate split between clean and noisy samples, hence improving the final model accuracy. Regarding CIFAR-10 asym. noise, where label groups are characterized by a limited number of classes, adopting a class conditional strategy may lead to overly confident discrimination between clean and noisy samples. This phenomenon could be attributed to a lack of related classes, resulting in insignificant "help" on the label group. The results in Table 1 show that our CCLM consistently outperforms the original global noise modelling used in DivideMix, being also more stable at low-noise settings. For both approaches, the accuracy of the models decreases considerably as the noise ratio increases. One of the reasons for it is the overfitting of the models in large noise settings which might make loss-based noise detection models less efficient. Hence, we further analyse the behaviour of CCLM using two different metrics - Noise Division Accuracy and Classiness.

Noise Division Accuracy. We evaluate the noise detection accuracy at the first epoch after the warmup. To this end, we first collect the per-sample loss provided by the pre-trained DL model. Then, we feed the collected loss values to both the global (DivideMix, using the threshold τ proposed for each noise level) and our CCLM noise detection approaches, yielding two distinct clean-noisy splits. Finally, using the reference solution, we evaluate the accuracy of the splits. The result is shown in Fig. 4, where the average split accuracy and the corresponding standard deviation for CIFAR-10 (Fig. 4a) and CIFAR-100 (Fig. 4b) are depicted, averaged over two runs. Our proposed method achieves a better average accuracy in all noise levels except for 90% sym. noise. It should be noted that a more accurate split after the warm-up does not guarantee better overall model accuracy. In the case of 90% noise, predicting all the samples as noisy would be very bad. The results also show that our method consistently yields a smaller standard deviation than the original one, which indicates that



(a) Test set. (b) Train set.

Fig. 5. Model accuracy and *Classiness* on CIFAR-100 with 20% sym. noise. The accuracy drop is a result of the ending of the warm-up phase (at epoch 30).

Table 2. Training and Testing Classiness of CIFAR-100.

		Train				Test			
Noise Rate		20%	50%	80%	90%	20%	50%	80%	90%
DivideMix [15]	Average	11.0	14.3	18.2	18.1	16.0	16.7	20.0	19.8
	Best	9	12.7	17.6	18.8	14	15	19	**21**
Ours	Average	7.7	10.9	17.5	17.1	13.6	14.9	19.4	20.2
	Best	**6**	**8.8**	**17.1**	**17.1**	**12**	**13**	**19**	22

our proposal yields a clean-noisy split that performs more evenly over the different classes. This results in a reduced bias towards the easier classes.

Classiness. In the classification task, it is clear that some classes can be more robust to label noise than others. A desired property of any classification model is to achieve similar accuracy in all the classes. In order to capture this accuracy distribution over classes, we introduce the notion of *classiness*, measured as the standard deviation of the model accuracy over the different latent classes. A lower classiness value thus represents a more evenly accurate network. We study classiness during training and testing. Training classiness allows us to inspect whether the training process is biased towards some of the classes. With class-agnostic label-noise modelling, *difficult* classes would be treated as mostly noise. Testing classiness is the desired model property informing about the achieved evenness. We show the training and testing model accuracy and classiness evolution during the training process using 20% sym. noise of CIFAR-100 in Fig. 5, where the first 30 epochs correspond to the warm-up phase. The training classiness (Fig. 5b (painted area)) is smaller on average for our proposed CCLM. This indicates that the model is less biased during training, allowing for better learning in the posterior epochs. The average accuracy of the baseline model and our proposal are similar, but the smaller classiness tends towards better learning seen in the later epochs. Table 2 shows the average training classiness

Table 3. Impact of the clean noise split threshold on the final model accuracy.

τ	0.5	0.6	0.7	0.8	0.9
DivideMix [15]	30.3	**31.5**	31.6	27.6	27.6
Ours	**30.6**	31.1	**32.6**	**33.5**	**30.8**

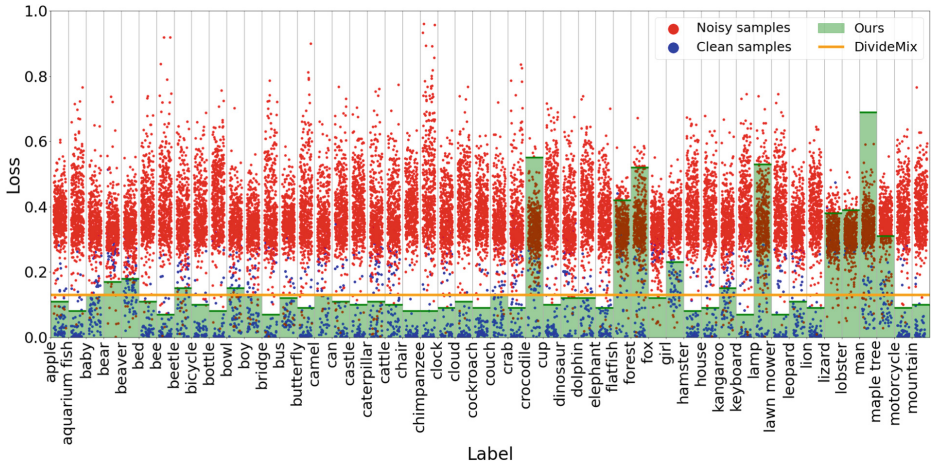


Fig. 6. Per-sample loss distribution of 50 classes of CIFAR-100 (90% sym.).

on CIFAR-100 with all different noise levels and the test classiness of the best epoch. The best epoch is chosen as the one with the better test accuracy. We see that our method has reasonably better test and train classiness on low noise settings and comparable test and train classiness on higher noise settings.

Effect of Clean-Noise Splitting Threshold. In current LNL practice, the label noise modelling step is most commonly used to perform the Clean-Noise split. However, the resulting split also depends on the threshold used as a decision boundary. A critical aspect of the analysis entails examining the impact of the selected threshold on the split outcome. To this end, we present Fig. 6 and Fig. 3, which illustrate the feature space we aim to split. The label-noise modelling procedure yields the probability of a sample belonging to the clean set, $\omega_i = p(g|l_i)$. Following Eq. (1) and Eq. (2), the Clean-Noisy partition is obtained in a sample level by $\omega_i > \tau$. In Table 3, we compare the model accuracy between the global LNL model, and our CCLM, across different split thresholds. Upon inspecting Fig. 6, we observe that a class-agnostic label noise modelling procedure, with a high threshold, disregards harder classes entirely. In contrast, our CCLM is not susceptible to this particular pitfall and is therefore capable of managing higher thresholds.

6 Conclusions

Learning with Noisy Labels is a very important data-centric Machine learning problem. Modern algorithms rely on the division of clean and noisy samples to

make further learning decisions. In this paper, we study this division of training samples and show how the global division is sub-optimal. On this front, we propose a class-conditional label noise model, which uses a local division of clean and noisy samples. We validate our approach comparing it with the baseline on two benchmarking datasets. We further introduce two metrics - noise division accuracy and classiness to show the improvements. Our future work focuses on studying the impact of global vs. local threshold in other parts of the algorithm, using other metrics to understand this division and validating the method on other large LNL datasets.

Acknowledgements. This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC-2022-133642-I00) and CERCA Programme/Generalitat de Catalunya. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs. As Serra Húnter Fellow, Ricardo Marques acknowledges the support of the Serra Húnter Programme.

References

1. Angluin, D., Laird, P.: Learning from noisy examples. Mach. Learn. **2**, 343–370 (1988)
2. Arazo, E., Ortego, D., Albert, P., O'Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: International Conference on Machine Learning, pp. 312–321. PMLR (2019)
3. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: a holistic approach to semi-supervised learning. In: NIPS, vol. 32 (2019)
4. Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: International Conference on Machine Learning, pp. 872–881. PMLR (2019)
5. Chen, C., et al.: Generalized data weighting via class-level gradient manipulation. In: NIPS, vol. 34, pp. 14097–14109 (2021)
6. Chen, Z., Song, A., Wang, Y., Huang, X., Kong, Y.: A noise rate estimation method for image classification with label noise. In: Journal of Physics: Conference Series, vol. 2433, p. 012039. IOP Publishing (2023)
7. Cheng, D., et al.: Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In: CVPR, pp. 16630–16639 (2022)
8. Ding, K., Shu, J., Meng, D., Xu, Z.: Improve noise tolerance of robust loss via noise-awareness. arXiv preprint arXiv:2301.07306 (2023)
9. Han, B., et al.: Co-teaching: robust training of deep neural networks with extremely noisy labels. In: NIPS, vol. 31 (2018)
10. Han, J., Luo, P., Wang, X.: Deep self-learning from noisy labels. In: ICCV, pp. 5138–5147 (2019)
11. Hendrycks, D., Mazeika, M., Wilson, D., Gimpel, K.: Using trusted data to train deep networks on labels corrupted by severe noise. In: NIPS, vol. 31 (2018)
12. Khetan, A., Lipton, Z.C., Anandkumar, A.: Learning from noisy singly-labeled data. arXiv preprint arXiv:1712.04577 (2017)
13. Kim, D., Ryoo, K., Cho, H., Kim, S.: SplitNet: learnable clean-noisy label splitting for learning with noisy labels. arXiv preprint arXiv:2211.11753 (2022)

14. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
15. Li, J., Socher, R., Hoi, S.C.: DivideMix: learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394 (2020)
16. Liao, Y.H., Kar, A., Fidler, S.: Towards good practices for efficiently annotating large-scale image classification datasets. In: CVPR, pp. 4350–4359 (2021)
17. Liu, S., Zhu, Z., Qu, Q., You, C.: Robust training under label noise by over-parameterization. In: ICML, pp. 14153–14172. PMLR (2022)
18. Liu, X., Luo, S., Pan, L.: Robust boosting via self-sampling. Knowl.-Based Syst. **193**, 105424 (2020)
19. Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., Bailey, J.: Normalized loss functions for deep learning with noisy labels. In: ICML, pp. 6543–6553 (2020)
20. Miyamoto, H.K., Meneghetti, F.C., Costa, S.I.: The Fisher-Rao loss for learning under label noise. Inf. Geometry 1–20 (2022)
21. Nagarajan, B., Marques, R., Mejia, M., Radeva, P.: Class-conditional importance weighting for deep learning with noisy labels. In: VISIGRAPP (5: VISAPP), pp. 679–686 (2022)
22. Nishi, K., Ding, Y., Rich, A., Hollerer, T.: Augmentation strategies for learning with noisy labels. In: CVPR, pp. 8022–8031 (2021)
23. Northcutt, C., Jiang, L., Chuang, I.: Confident learning: estimating uncertainty in dataset labels. J. Artif. Intell. Res. **70**, 1373–1411 (2021)
24. Oyen, D., Kucer, M., Hengartner, N., Singh, H.S.: Robustness to label noise depends on the shape of the noise distribution in feature space. arXiv preprint arXiv:2206.01106 (2022)
25. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: a loss correction approach. In: CVPR, pp. 1944–1952 (2017)
26. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: a survey. IEEE Tran. NNLS (2022)
27. Sun, Z., et al.: PNP: robust learning from noisy labels by probabilistic noise prediction. In: CVPR, pp. 5311–5320 (2022)
28. Valle-Pérez, G., Camargo, C.Q., Louis, A.A.: Deep learning generalizes because the parameter-function map is biased towards simple functions. arXiv e-prints arXiv:1805.08522 (2018)
29. Wang, H., Xiao, R., Dong, Y., Feng, L., Zhao, J.: ProMix: combating label noise via maximizing clean sample utility. arXiv preprint arXiv:2207.10276 (2022)
30. Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., Liu, Y.: Learning with noisy labels revisited: a study using real-world human annotations. arXiv preprint arXiv:2110.12088 (2021)
31. Wu, P., Zheng, S., Goswami, M., Metaxas, D., Chen, C.: A topological filter for learning with label noise. In: NIPS, vol. 33, pp. 21382–21393 (2020)
32. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. ACM **64**(3), 107–115 (2021)
33. Zhang, Y., Niu, G., Sugiyama, M.: Learning noise transition matrix from only noisy labels via total variation regularization. In: ICML, pp. 12501–12512 (2021)
34. Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A.M., Litany, O.: Contrast to divide: self-supervised pre-training for learning with noisy labels. In: WACV, pp. 1657–1667 (2022)
35. Zhou, X., Liu, X., Zhai, D., Jiang, J., Ji, X.: Asymmetric loss functions for noise-tolerant learning: theory and applications. IEEE Trans. PAMI (2023)

Pattern Recognition Letters xxx (xxxx) xxx

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Decoding class dynamics in learning with noisy labels

Albert Tatjer [a,1], Bhalaji Nagarajan [a,*,1], Ricardo Marques [a,b,2], Petia Radeva [a,2]

[a] Department de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes, 585, 08007, Barcelona, Spain
[b] Computer Vision Center (CVC), Campus UAB, Edifici O, Cerdanyola del Vallès, 08193, Barcelona, Spain

ARTICLE INFO

Editor: Maria De Marsico

MSC:
41A05
41A10
65D05
65D17

Keywords:
Learning with Noisy Labels
Label Noise Modelling
Class Dynamics

ABSTRACT

The creation of large-scale datasets annotated by humans inevitably introduces noisy labels, leading to reduced generalization in deep-learning models. Sample selection-based learning with noisy labels is a recent approach that exhibits promising upbeat performance improvements. The selection of clean samples amongst the noisy samples is an important criterion in the learning process of these models. In this work, we delve deeper into the clean-noise split decision and highlight the aspect that effective demarcation of samples would lead to better performance. We identify the Global Noise Conundrum in the existing models, where the distribution of samples is treated globally. We propose a per-class-based local distribution of samples and demonstrate the effectiveness of this approach in having a better clean-noise split. We validate our proposal on several benchmarks — both real and synthetic, and show substantial improvements over different state-of-the-art algorithms. We further propose a new metric, classiness to extend our analysis and highlight the effectiveness of the proposed method. Source code and instructions to reproduce this paper are available at https://github.com/aldakata/CCLM/

1. Introduction

Recent breakthroughs in Deep Learning (DL) have led to remarkable achievements across various intricate tasks and have surpassed human-level performances. The high performance of these models hinges on high-quality large-scale datasets, which, however, involves substantial cost and labour-intensive efforts [1]. Despite the substantial efforts, mislabelling remains unavoidable due to the data complexity and potential human annotation errors [2]. Weakly supervised approaches such as crowdsourcing [3] and web-crawling [4] are particularly susceptible to label noise due to human participation. Introduction of noise in the assigned labels thus becomes inevitable [5,6]. Training Deep Neural Networks (DNNs) with noisy labels leads to a significant decline in their generalization performance [7]. Over-parameterization [8] and DNN's strong memorization capability [9] often lead to over-fitting to this label noise. Traditional regularization techniques, such as dropouts or weight decay, fall short of effectively mitigating this problem [9].

Learning with Noisy Labels (LNL) has been a long-studied problem ever since its inception in the last 1980s [10]. It holds a crucial position within the DL community due to its critical role in developing robust models capable of handling label noise. Several strategies have been proposed to create noise-robust DNNs [11]. Making loss function modifications [12,13], creating new robust loss functions [14,15], adding regularization [16,17], reweighting samples [18,19], label correction [20,21], label aggregation [22], learning jigsaw puzzles [23], and using topological structure information [24] have contributed to alleviating the impact of label noise.

Sample selection-based methods are the most straightforward and widely adopted LNL method. They work on identifying and selecting clean samples from the training samples and imposing different schemes in order to reduce the interference of noisy samples. Different criteria are used in sample selection-based methods, of which, the small-loss criterion is a popular scheme [25,26]. In this criterion, the small loss samples are treated as the clean samples [27]. The success of sample selection methods depends on the quality of the clean-noise separations and the criteria designed to configure the refined set [28].

Recent LNL methods have enhanced robustness by combining diverse semi-supervised learning strategies. **DivideMix** (DM) [26] is a prominent LNL benchmark algorithm based on the modelling of sample loss distribution. DM trained two 'peer' DNNs and modelled the loss distribution of each model using a Gaussian Mixture Model (GMM) in order to dynamically divide the clean and noisy samples. DM uses the

* Corresponding author.
E-mail addresses: albert.catalan-tatjer@student.uni-tuebingen.edu (A. Tatjer), bhalaji.nagarajan@ub.edu (B. Nagarajan), ricardo.marques@ub.edu (R. Marques), petia.ivanova@ub.edu (P. Radeva).
[1] Equal contribution.
[2] Equal supervision.

https://doi.org/10.1016/j.patrec.2024.04.012
Received 31 October 2023; Received in revised form 25 January 2024; Accepted 15 April 2024
Available online 19 April 2024
0167-8655/© 2024 Elsevier B.V. All rights reserved.

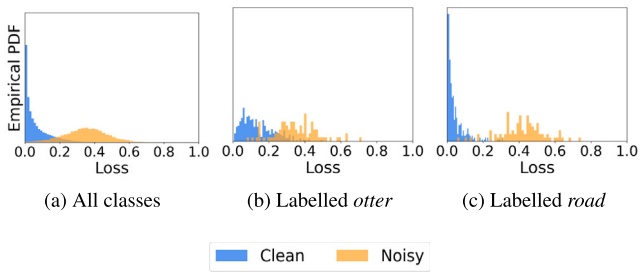


Fig. 1. Empirical loss distribution of the clean samples and the noisy samples after the warm-up phase (CIFAR-100 20% Sym. noise).

clean samples as a labelled set and noisy samples as an unlabelled set and employs MixMatch [29] to learn the samples in a semi-supervised manner. **Contrast to Divide** (C2D) [30], improved DM by applying self-supervised pre-training. One of the important design questions revolves around *the effective selection of clean and noisy samples* playing an important role as it directly affects the subsequent tasks in the training pipeline.

DNNs tend to learn the underlying patterns of the data first before gradually memorizing the samples [7]. Small-loss-based methods assume that the smaller loss samples have a lower likelihood of being affected by label noise. However, not all samples are learned in this fashion [31], which hinders noise detection in these methods, leading to *Global loss conundrum*. In our previous work published in IbPRIA 2023, **Class-Conditional Local noise Model** (CCLM) [32], we studied this global noise modelling and proposed a per-class label noise model. In this work, we extend our work on CCLM in several dimensions: (1) We provide an extended comprehensive review of the latest SoTA works that are related to our proposal (Section 2). (2) We extend CCLM to other SoTA LNL algorithms, namely, Contrast to Divide (Section 4). (3) We further expand the evaluations to two real noise benchmarks [6] along with the two synthetic noise benchmarks used in CCLM. (4) We provide a deeper analysis by including new metrics and extend the analysis of CCLM applied to both DM and C2D (Section 5.2). The obtained results highlight the effectiveness of CCLM in label noise modelling.

2. Related work

LNL is of significant research interest and several methods have been proposed to tackle the problem of label noise [11]. Using noise transition matrix [12,13], employing regularization methods such adding dropouts [17], gradient clipping [16], progressive early stopping [33], SANM [34] have alleviated the effect of label noise. Robust loss functions such as Symmetric cross-entropy learning [35] and Active Passive Loss [36] have increased the performance of models in the presence of label noise. Recently, contrastive learning methods have been used in the context of LNL algorithms. ChiMera [37], UNICON [38] and TCL [39] have been successful in LNL training. Additional knowledge, such as using uncertainty, help complement the learning process [40,41].

Sample selection methods work on selecting clean samples amongst the noisy samples and devise different strategies to reduce the impact of the label noise. They work on the intuition that less noisy data leads to more robust DNNs [7]. Different strategies exist in demarcating clean and noisy samples of which small-loss selection [25,27] is the most widely adopted. In this selection criterion, the small loss samples are treated as clean samples [27]. A prevalent issue of sample selection methods is the accumulation of errors by incorrect selection of samples. 'Peer' networks or multiple DNNs, such as Co-teaching [42], Co-teaching+ [43] and JoCoR [44] were used to reduce this confirmation bias.

Different **loss modelling strategies** exist in the literature, such as using a beta-distribution model [25] and using GMMs [25,26] to split the training samples into clean and noisy samples. UNICON [38] used a Jensen–Shannon divergence based criterion to maintain class-balancing in selecting the clean samples. SSS-Net [45] combined shadowed-sets theory with clustering based on loss-similarity to select the clean samples. Recent sample selection methods incorporated advanced strategies based on the confidence of the models. DISC [46] incorporated confidence of samples to devise a dynamic thresholding strategy. Gradient Switching Strategy [47] used gradient direction as a weighting mechanism for the high-confidence and uncertain samples. CoDis [48] used high discrepancy data to maintain the divergence of two networks so as to enable the models to mine hard clean samples. Label Confidence Incorporation framework [49] incorporated label confidence as a measure of label noise and in turn prioritized the training samples.

One of the most widely recognized LNL multi-network benchmark algorithms in the context of sample selection methods is **DivideMix** [26]. DM trains two 'peer' networks to train each other by modelling the per-sample loss distribution using GMMs. It uses an improved MixMatch [29] algorithm to train the models using the clean and noisy samples as labelled and unlabelled sets. Several methods have been since developed highlighting the potential pitfalls in its training pipeline and improving the potential drawbacks. Augment Descent [50] used two different augmentation schemes instead of a global data augmentation pipeline. Contrast to Divide [30] proposed a self-supervised pre-training to create better feature extractors during the warm-up phase. ProMix [51] improved the small-loss sample selection by also considering the confidence of the samples. SplitNet [52] used an additional network to predict the samples being clean or noisy. PNP [53] used different optimizations based on the type of the samples. Manifold DivideMix [54] employed the k-nearest algorithm to remove out-of-distribution samples and used an iterative method to find the clean and noisy samples. Bayesian DivideMix++ [55] added several components to the training pipeline to improve the robustness against DNN memorization and utilized uncertainty metrics to further enhance its effectiveness. ULC [56] used different uncertainty measurements in label correction for imbalanced LNL problems.

One of the patterns observed during the training of DNNs is that not all samples are learned in the same way. **Class-based conditional** strategies have been used to better model the training data. CMW-Net [57] used a metamodel to learn a weighting scheme based on the training class. A dynamic class-conditional weighting was used to improve the balance of losses in DivideMix [31]. CPC [58] considered the loss distribution in a heterogeneous fashion and used class-aware modulation to partition the clean and noisy data. LCCN [59] used a Bayesian framework to parameterize the noise transition. SNSCL [60] employed a stochastic noise-tolerated supervised contrastive learning framework to learn representations in fine-grained classes. On this front, CCLM [32] proposed a class-conditional label noise modelling instead of the global noise modelling used in DivideMix. CCLM was able to show how the local treatment of samples fared better with respect to the global treatment of loss distribution. Following this, we extend our findings of CCLM in this manuscript.

3. Rationale

The typical approach taken by the most performing algorithms in LNL is to actively detect the potentially noisy samples and treat these samples differently from the samples which are deemed clean [26]. A common procedure is to apply semi-supervised learning techniques [1] to make the most out of the information provided by samples (either clean or noisy). Given the differentiated treatment applied to clean and noisy samples, an accurate label noise detection is thus essential for a successful learning procedure in the presence of noisy labels. On the one hand, treating noisy samples as clean samples often leads the model to learn (and thus, to memorize) wrong information, which

can significantly hinder the final accuracy of the model. On the other hand, treating clean samples as noisy can lead the model to discard relevant (and correct) information, eventually introducing mistakes in the learning process which degrade the learning outcome. To deal with this problem, LNL-based algorithms usually rely on a *noise model*, the goal of which is to capture a *global* statistical characterization of the samples' label noise. During training, this model is used to assign a probability of being clean (or noisy) to each sample in the training set, based on which the separation between clean and noisy samples is made. Typically, the noise model relies on a noisy proxy metric (i.e. the loss), leveraging the observation that noisy samples are harder to learn and thus tend to have larger loss values [25]. We provide an in-depth analysis of the *global noise model* approach and show that such an approach is locally suboptimal.

3.1. Global noise modelling

When tackling the problem of label noise detection, many existing approaches rely on the principle that noisy samples are more difficult to learn [61]. Therefore, when a neural network is used to evaluate the loss associated with each sample in the training set, noisy samples are statistically expected to yield larger losses, thus enabling the separation between clean and noisy samples. To formalize this process, let us consider a dataset $\mathcal{D}$ of n labelled samples such that $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, with x_i being the ith sample image and y_i its corresponding one-hot encoded label. Let us also consider a model m_Θ, where Θ are the model parameters, and a loss l function such that the set $\mathcal{L}$ of individual losses l_i for each sample $x_i \in \mathcal{D}$ is given by $\mathcal{L} = \{l_i = l\left(m_\Theta(x_i), y_i\right)\}_{i=1}^{n}$. An example of a histogram of $\mathcal{L}$ considering the cross-entropy loss is shown in Fig. 1(a). We can appreciate a clearly bimodal distribution, where the noisy samples, shown in orange, exhibit a characteristic loss generally larger than that of the clean samples, shown in cyan. Once the set $\mathcal{L}$ of all losses is computed, it is fed to the global noise model to produce a probability of a given sample being correctly (or wrongly) labelled. To this end, Li et al. [26] fit a 2-component Gaussian Mixture Model (GMM) to $\mathcal{L}$. Then, they retain the GMM component g with smaller mode, and use it to compute the clean probability of the ith given by $\omega_i := p(g|l_i)$. Lastly, the dataset D is partitioned into a clean D_C and a noisy set D_N given by:

$$\begin{aligned} D_C &= \{(x_i, y_i)|\omega_i \geq \tau\}_{(x_i,y_i)\in D}, \\ D_N &= \{(x_i, y_i)|\omega_i < \tau\}_{(x_i,y_i)\in D}, \end{aligned} \tag{1}$$

with τ being a threshold over the clean probability ω_i, and $l_i = l(m_\theta(x_i), y_i)$ as previously stated.

Drawbacks of global noise modelling. Modelling noise as a global process relies on the hypothesis that the per-sample loss behaves consistently across classes. Indeed, in this approach, the samples' losses are assumed to be i.i.d. realizations of a bi-modal loss distribution with 2 populations, where the population with a smaller mean corresponds to the clean samples, and the other corresponds to the noisy samples. Nevertheless, since different classes can have different characteristic losses, noisy samples of a given class might have a characteristic loss distributed differently from noisy samples of another class. This can potentially cause noisy samples to have a loss that falls within the limits of the low mean population, or, conversely, clean samples have a loss value around the mean of the high-loss values population. This observation is confirmed by comparing the per-sample loss distribution for the 'otter' class (Fig. 1(b)) with that of the 'road' class (Fig. 1(c)). Indeed, the loss distribution of both classes differs significantly from the global loss distribution (Fig. 1(a)). Consequently, the global noise model, although globally optimal, is clearly locally suboptimal regarding each individual class. We identify this situation as an obstacle to an accurate division between clean and noisy samples, which, in turn, can have significant negative effects on the training outcome. We

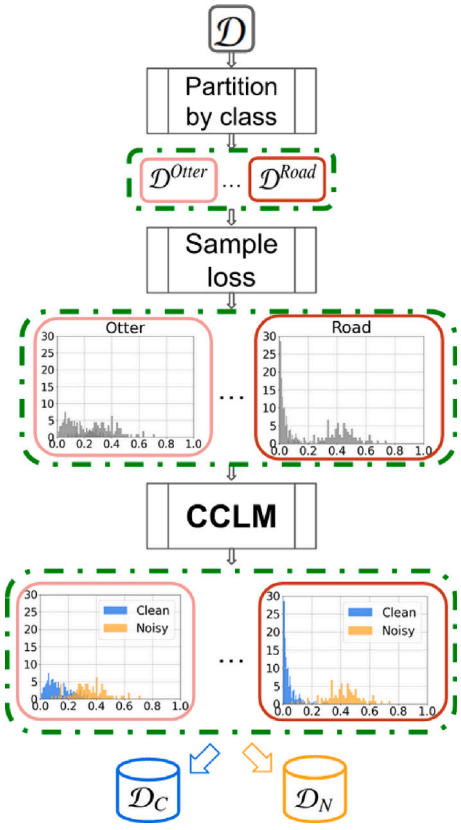


Fig. 2. CCLM partitions the dataset by the class labels and then fits a GMM on the sample loss of each subset. Subsequently, the resulting distribution combined with an adjustable threshold determines the Clean-Noisy partition.

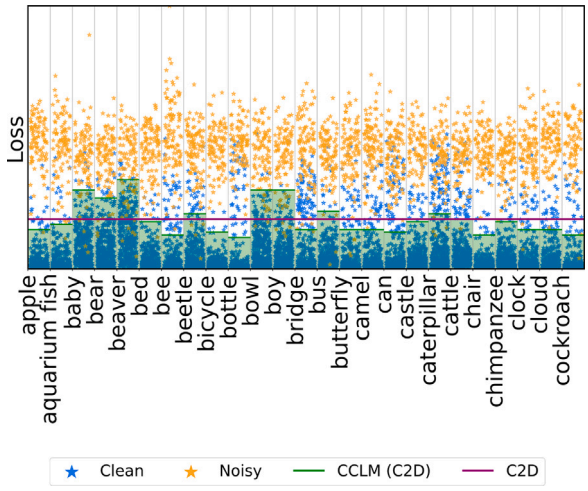


Fig. 3. Empirical sample loss distribution for 25 random classes of CIFAR-100 with Sym. 20% noise. The loss was computed at epoch 30 of the training procedure, using the C2D algorithm.

hypothesize that better clean/noisy data partitions and increased model accuracy can be achieved using a class-conditional approach to the problem of noise label detection, therefore relaxing the aforementioned strong i.i.d. assumption on the samples' loss distribution. To test our hypothesis, we propose a class-conditional noise label model, aimed at overcoming the aforementioned limitations of global label noise modelling.

4. Proposed methodology

In this section, we propose a new label noise model which, as opposed to the typical approaches in LNL, is able to locally adapt to the particularities of the noise features of each class present in the dataset. This is achieved by explicitly including the information brought by the label class in the noise modelling process. As shown by our results, our method, which we label *Class-Conditional Local Noise Model* (CCLM), leads to a generalized improvement of the label noise detection task. Furthermore, CCLM can be coupled with widely used LNL algorithms such as DivideMix [26] or C2D [30]. Fig. 2 provides a global view of our proposed approach. The input noisy training set $\mathcal{D}$ is defined as $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, with x_i being the ith sample image and y_i its corresponding one-hot encoded label. The dataset $\mathcal{D}$ is then partitioned into J disjoint subsets, where J is the total number of classes, such that $\mathcal{D} = \cup_{j=1}^{J} \mathcal{D}^j$. Once the data is divided into classes, a 2-component GMM is fit to the per-sample loss of each subset $\mathcal{D}^j$ and, the GMM component with a lower mode is used to model the clean probability of the samples. Note that these results in a different distribution for each class, tailored to its specificity. Within each class, we then split between clean and noisy samples by thresholding the clean probability such that:

$$\begin{aligned} \mathcal{D}_C &= \cup_{j=1}^{J} \mathcal{D}_C^j = \cup_{j=1}^{J} \{(x_i, y_i) \mid \omega_i^j \geq \tau\}_{(x_i,y_i)\in D^j} \\ \mathcal{D}_N &= \cup_{j=1}^{J} \mathcal{D}_N^j = \cup_{j=1}^{J} \{(x_i, y_i) \mid \omega_i^j < \tau\}_{(x_i,y_i)\in D^j}, \end{aligned} \tag{2}$$

where τ is the same threshold used for all classes.

A comparison between our approach and the global noise modelling strategy is shown in Fig. 3. The horizontal axis represents a subset of 25 classes randomly chosen, while the vertical axis measures the per-sample loss. Clean samples are depicted as cyan dots, whereas noisy samples are depicted in orange. The split between clean and noisy samples for the global noise model is given by the magenta horizontal line (baseline): samples above the line are deemed noisy, whereas samples below the line are considered clean. This results in a division with a single decision boundary for all classes, effectively ignoring each class' particular noise features. The split between clean and noisy samples for our model is given by the green vs white areas corresponding to each class: samples within the green area are deemed clean, whereas samples within the white area are considered noisy. Note that, as opposed to the global noise model, our model provides a significantly different loss threshold for each class, even if the same probability threshold τ is used for all classes. This is because each class has its own clean probability function, so applying the same probability threshold for all classes results in a different loss threshold in practice. As shown next, our locally adapted split yields improvements both in the noise detection accuracy and in the final model classification accuracy when trained in the presence of noisy labels.

5. Experiments

Datasets. We evaluate the performance of CCLM on two synthetic noise benchmarks – CIFAR-10 [62] and CIFAR-100 [62] at different noise rates, as well as on two recent real noise benchmarks – CIFAR-10N [6] and CIFAR-100N [6]. All four benchmarks consist of 50,000 training samples and 10,000 test samples, where each image is of size 32×32. Following previous works [26,30], for the synthetic label noise, we validate our method on two types of noise — Symmetric (Sym) and Asymmetric (Asym). Symmetric noise is introduced by substituting the labels of a percentage of samples with random labels (all possible classes) drawn from a uniform distribution. Asymmetric noise is introduced by replacing the labels with "similar" classes (For example, 'Bird' and 'Airplane' in CIFAR-10). We vary the amount of injected noise (Sym. 20%, 50%, 80%, 90%, and Asym. 40%), following Patrini et al. [63] that is used as a standard experimental setting [25,26,30]. CIFAR-10N and CIFAR-100N use the training data of the corresponding CIFAR datasets with human-annotated real-world noisy labels. CIFAR-10N has five noisy label sets,[3] whereas CIFAR-100N has fine and coarse labels. We use the Noisy-Fine setting for our experiments. We evaluate all benchmarks using an 18-layer PreAct ResNet [64] as the backbone architecture. The experiment settings are presented in the Supplement.

Comparison methods. We compare the proposed CCLM against small-loss, sample-selection multi-network benchmark algorithms, DivideMix [26] and C2D [30]. Both methods follow a global noise model approach. CCLM (DM) corresponds to CCLM on DivideMix, whereas CCLM (C2D) corresponds to CCLM on C2D. We compare CCLM against DivideMix and C2D, as they are baseline frameworks for several algorithms [50,51,53,55].

5.1. Results

Following the experiments of DivideMix [26] and C2D [30], we report the best test accuracy (%) over all epochs (Best) and the average of the last 10 epochs (Last). The test set is used to provide insights into the generalization capabilities of the method. Long-time training may lead to performance degradation [65] and therefore, "last" is used as a measure of robustness [66,67]. Larger gaps between "best" and "last" correspond to overfitting.

Performance on synthetic noise benchmarks. Table 1 shows the test accuracy of CIFAR-10 and CIFAR-100 with different noise ratios. On both CIFAR datasets, CCLM (DM) outperforms DivideMix in all symmetric noise ratios. In the case of C2D, our method CCLM (C2D) works better on low-noise settings. However, the difference is marginal in high-noise settings. One of the possible reasons could be that the self-supervised pre-training induces a lot of class biases. In the high noise levels, it is hard to overcome such a strong trend with very few clean samples.

Performance on real noise benchmarks. We evaluate the performance of CCLM using two real-noise benchmarks, CIFAR-10N and CIFAR100-N (Table 2). In most noise levels of CIFAR-10N, CCLM fares better than DivideMix, achieving a significant performance gain of 1.3% in CIFAR-100N. However, the difference between CCLM (C2D) and C2D is marginal. Similar to the case of synthetic noise benchmarks, the small impact of CCLM on C2D can be explained by the effect of the pre-training phase used in C2D.

5.2. Analysis

We further analyse the behaviour of CCLM using two different metrics — Classiness and Noise Division Accuracy.

Classiness. Historically, statistical learning has been concerned with evaluation metrics and their potential for misinformation. For instance, model accuracy does not fare well in large class imbalances. We argue that LNL is one such vulnerable setting. As we can see in Table 1, on CIFAR-100 with 20% symmetric noise, DivideMix achieves a model accuracy of 77.3%. However, it is interesting to also understand the behaviour of LNL algorithms on different individual classes, as, not all classes are uniformly represented in a dataset (varying sample counts, sample difficulties, etc.).

In order to track this phenomenon, we introduce a new metric, called *classiness*. Classiness is defined as the standard deviation of the model accuracy over the classes, with a lower bound of zero and an unbounded upper bound. Let m and $\mathcal{D}$ be the model and dataset under evaluation, respectively. The accuracy μ of model m on the dataset $\mathcal{D}$ is given by $\mu = \text{Acc}(m, \mathcal{D})$. Then the classiness of m on $\mathcal{D}$ can be defined as:

$$\text{Classiness}(m, \mathcal{D}) = \sqrt{\frac{1}{C} \sum_{j=1}^{C} \frac{N^j}{N} \left(\text{Acc}(m, \mathcal{D}^j) - \mu\right)^2}, \tag{3}$$

[3] Construction of label sets are detailed in CIFAR-N website.

Table 1
Synthetic noise performance comparison (test accuracy (%) on CIFAR datasets under different noise types). Bold indicates best performance.

Method		CIFAR-10					CIFAR-100				
		Symmetric				Asym.	Symmetric				Asym.
		20%	50%	80%	90%	40%	20%	50%	80%	90%	40%
DivideMix [26]	Best	96.1	94.6	93.2	76.0	**93.4**	77.3	74.6	60.2	31.5	72.2
	Last	95.7	94.4	92.9	75.4	92.1	76.9	74.2	59.6	31.0	72.4
CCLM (DM)	Best	**96.5**	**95.6**	**93.7**	**83.6**	92.6	**79.5**	**76.4**	**61.1**	**33.5**	**75.4**
	Last	96.3	95.3	93.6	82.4	91.5	79.1	75.9	60.9	33.0	75.1
Contrast2Divide [30]	Best	96.4	95.3	**94.4**	**93.6**	**93.5**	78.7	76.4	**67.8**	**58.7**	**75.5**
	Last	96.2	95.2	94.3	93.4	90.8	78.3	76.1	67.4	58.5	75.1
CCLM (C2D)	Best	**96.5**	**95.4**	94.4	93.6	92.2	**79.4**	**76.9**	67.6	55.2	75.5
	Last	96.4	95.4	94.1	93.2	90.8	79.2	76.6	67.1	55.0	73.2

Table 2
Real noise performance comparison (test accuracy (%) on CIFAR-N datasets under different noise types). Bold indicates best performance. *Baseline results are reproduced*.

Method		CIFAR-10N					CIFAR-100N
		Aggregate	Random 1	Random 2	Random 3	Worst	Noisy fine
DivideMix [26]	Best	95.3	95.6	**95.6**	95.6	**93.2**	69.4
	Last	95.3	95.6	95.0	95.2	92.9	68.8
CCLM (DM)	Best	**95.6**	**95.6**	95.4	**95.6**	93.0	**70.7**
	Last	95.4	95.5	95.2	95.4	92.9	69.9
Contrast2Divide [30]	Best	95.7	**95.9**	**95.8**	93.8	**92.8**	**70.8**
	Last	95.6	95.6	95.8	93.0	92.0	70.3
CCLM (C2D)	Best	**95.8**	95.7	95.7	**95.9**	92.4	70.6
	Last	95.5	95.6	95.3	95.7	92.0	69.7

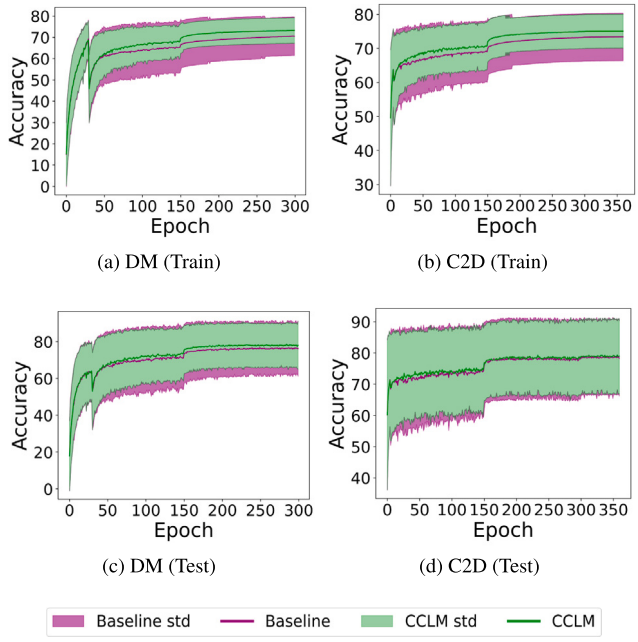


Fig. 4. Model accuracy and Classiness on CIFAR-100 20% Sym. noise. The accuracy drop is a result of the ending of the warm-up phase (at epoch 30 for DivideMix and 5 for C2D).

where $\text{Acc}(m, D^j)$ is the accuracy of m on the jth class, N is the total number of samples and N^j is the number of samples in the jth class. Moreover, the term $\frac{N^j}{N}$ is a weight that extends the definition of classiness to class-imbalanced datasets. Classiness measures the spread of the model accuracy over the classes. Hence, low classiness is a desired property.

The evolution of model accuracy and classiness using 20% sym. noise of CIFAR-100 is shown in Fig. 4. The green and magenta lines show the accuracy of our method and of the concurrent method, respectively. For our method (concurrent method) the classiness is depicted as a green (magenta) area around the accuracy. We can observe that the training classiness is consistently smaller for the proposed CCLM (Figs. 4(a) and 4(b)). This shows that CCLM is less class-biased during training, yielding a better learning outcome. Finally, Table 3 shows the average and the best training and testing classiness on CIFAR-100 for the different noise levels, where the *best* epoch is chosen as the one with the best test accuracy. We differentiate between train and test classiness scores as the train classiness measures how much noise is being learned, and the test classiness measures the spread of the model accuracy on unseen data. It can be seen that CCLM has lower classiness in low-noise settings and comparable classiness in high-noise settings for both training and testing.

Table 3
Model classiness (Eq. (3)) on CIFAR-100. *Best* reports the classiness of the model at the epoch with the highest test accuracy. *Average* reports the average classiness of the model across all the epochs.

Noise			Method			
			DivideMix		CCLM (DM)	
			Average	Best	Average	Best
Train	Sym.	20%	11.0	9.0	7.7	**6.0**
		50%	14.3	12.7	10.9	**8.8**
		80%	18.2	17.6	17.5	**17.1**
		90%	18.1	18.8	17.1	**17.1**
	Asym. 40%		17.3	14.9	14.1	**11.2**
Test	Sym.	20%	16.0	14.0	13.6	**12.0**
		50%	16.7	15.0	14.9	**13.0**
		80%	20.0	19.0	19.4	**19.0**
		90%	19.8	**21.0**	20.2	22.0
	Asym. 40%		18.7	17.5	17.2	**14.0**

Noise division accuracy. To evaluate our label noise model, we check the performance on the Clean-Noisy split, at the first epoch after warm-up. We show the average (over two runs) division accuracy and the class standard deviation in Fig. 5 on CIFAR-10 and CIFAR-100 for

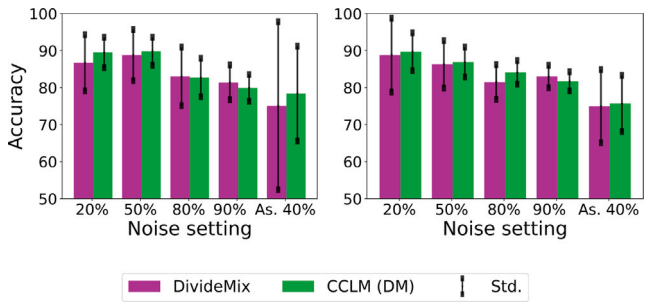


Fig. 5. Clean/Noisy split accuracy on CIFAR-10 (left) and CIFAR-100 (right) for all the synthetic noise settings.

Table 4
Impact of clean/noise split threshold on final model accuracy using CIFAR-100 with Sym. 80% and 90% noise. Bold indicates best performance.

Noise	GMM thres. (τ)	0.5	0.6	0.7	0.8	0.9
80%	DivideMix	**60.2**	**59.7**	59.1	58.7	54.7
	CCLM (DM)	59.0	58.4	**60.6**	**61.1**	**61.0**
90%	DivideMix	30.3	**31.5**	31.6	27.6	27.6
	CCLM (DM)	**30.6**	31.1	**32.6**	**33.5**	**30.8**

different noise settings, right after the warm-up. Our proposed method achieves a better accuracy on low noise levels, and a comparable accuracy in high noise levels. More importantly, we show how our method consistently achieves a smaller standard deviation, which results in a reduced bias towards easier classes.

Effect of clean-noise split threshold. We have highlighted the importance of the Clean-Noise split thoroughly, especially focusing on how to model label noise more sensibly. However, the Clean-Noise partition is ultimately decided by a threshold parameter. Therefore, an essential component of the analysis involves evaluating the influence of the chosen threshold. To understand the effect of the split threshold, we vary the GMM Threshold (τ) in DivideMix-based experiments. The SSL pre-training of C2D enables the models to determine the noisy examples with high confidence, making this study less relevant. As shown in Table 4 on CIFAR-100 with 80% and 90% symmetric noise, we observe that the model accuracy is consistent through the different values of τ. Fig. 3 presents the feature space that we aim to split. We argue that a class-agnostic label noise modelling, with a high threshold, would include less clean samples, whereas, with a low threshold, would include more noisy samples. In contrast, CCLM is not susceptible to this pitfall and is capable of handling higher thresholds.

6. Conclusions and future directions

Learning with Noisy Labels plays a pivotal role in data-centric deep learning due to its wide applications. In this paper, we study the division of clean-noisy samples in detail. Typical global noise modelling relies on the assumption that data is independent and normally distributed. We show that such a treatment is not always reasonable, and propose Class-Conditional Local Noise Model (CCLM) that relaxes these assumptions. In addition, we propose a new metric, *classiness*, that is important both for its obvious desirability and as a tool to understand training dynamics. Our future work focuses on analysing the impact of these relaxed assumptions on other parts of the tested algorithms. Tracking *classiness* for other datasets and, in particular, for class-imbalanced datasets is also an interesting research avenue since it can potentially open the path for further understanding of the training dynamics with this type of dataset. Finally, studying how other algorithms could benefit from this new label noise model is also envisaged.

CRediT authorship contribution statement

Albert Tatjer: Investigation, Methodology, Visualization, Writing – original draft. **Bhalaji Nagarajan:** Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Ricardo Marques:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Supervision, Writing – review & editing. **Petia Radeva:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was partially funded by the EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), CERCA Programme/Generalitat de Catalunya, and Grants PID2022-141566NB-I00 (IDEATE), PDC2022-133642-I00 (DeepFoodVol), and CNS2022-135480 (A-BMC) funded by MICIU/AEI/10.13039/501100011033, by FEDER (UE), and by European Union NextGenerationEU/PRTR. R. Marques acknowledges the support of the Serra Húnter Programme. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2024.04.012.

References

[1] Y.-H. Liao, A. Kar, S. Fidler, Towards good practices for efficiently annotating large-scale image classification datasets, in: CVPR, 2021, pp. 4350–4359.
[2] G. Algan, I. Ulusoy, Image classification with deep learning in the presence of noisy labels: A survey, Knowl.-Based Syst. 215 (2021) 106771.
[3] S. Li, X. Xia, J. Deng, S. Ge, T. Liu, Transferring annotator-and instance-dependent transition matrix for learning from crowds, 2023, arXiv preprint arXiv:2306.03116.
[4] C. Tan, J. Xia, L. Wu, S.Z. Li, Co-learning: Learning from noisy labels with self-supervision, in: ACM ICMM, 2021, pp. 1405–1413.
[5] C. Northcutt, L. Jiang, I. Chuang, Confident learning: Estimating uncertainty in dataset labels, J. Artif. Intell. Res. 70 (2021) 1373–1411.
[6] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, Y. Liu, Learning with noisy labels revisited: A study using real-world human annotations, in: ICLR, 2021.
[7] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al., A closer look at memorization in deep networks, in: ICML, 2017, pp. 233–242.
[8] Z. Allen-Zhu, Y. Li, Y. Liang, Learning and generalization in overparameterized neural networks, going beyond two layers, Neural Inf. Process. Syst. 32 (2019).
[9] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Commun. ACM 64 (3) (2021) 107–115.
[10] D. Angluin, P. Laird, Learning from noisy examples, Mach. Learn. 2 (1988) 343–370.
[11] H. Song, M. Kim, D. Park, Y. Shin, J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[12] Y. Zhang, G. Niu, M. Sugiyama, Learning noise transition matrix from only noisy labels via total variation regularization, in: ICML, PMLR, 2021, pp. 12501–12512.
[13] S. Wu, X. Xia, T. Liu, B. Han, M. Gong, N. Wang, H. Liu, G. Niu, Class2simi: A noise reduction perspective on learning with noisy labels, in: ICML, PMLR, 2021, pp. 11285–11295.

[14] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, Neural Inf. Process. Syst. 31 (2018).
[15] Y. Kim, J. Yun, H. Shon, J. Kim, Joint negative and positive learning for noisy labels, in: CVPR, 2021, pp. 9442–9451.
[16] A.K. Menon, A.S. Rawat, S.J. Reddi, S. Kumar, Can gradient clipping mitigate label noise? in: ICLR, 2020.
[17] Y. Chen, S.X. Hu, X. Shen, C. Ai, J.A. Suykens, Compressing features for learning with noisy labels, IEEE Trans. NNLS (2022).
[18] M. Ren, W. Zeng, B. Yang, R. Urtasun, Learning to reweight examples for robust deep learning, in: ICML, 2018, pp. 4334–4343.
[19] M. Xu, Z. Lian, L. Feng, B. Liu, J. Tao, DALI:Dynamically adjusted label importance for noisy partial label learning, 2023, arXiv:2301.12077.
[20] D. Ortego, E. Arazo, P. Albert, N.E. O'Connor, K. McGuinness, Multi-objective interpolation training for robustness to label noise, in: Proceedings of the IEEE/CVF CVPR, 2021, pp. 6606–6615.
[21] X. Xia, J. Deng, W. Bao, Y. Du, B. Han, S. Shan, T. Liu, Holistic label correction for noisy multi-label classification, in: ICCV, 2023, pp. 1483–1493.
[22] M. Wu, Q. Li, F. Yang, J. Zhang, V.S. Sheng, J. Hou, Learning from biased crowdsourced labeling with deep clustering, Expert Syst. Appl. 211 (2023) 118608.
[23] Y. Chen, X. Shen, Y. Liu, Q. Tao, J.A. Suykens, Jigsaw-ViT: Learning Jigsaw puzzles in vision transformer, Pattern Recognit. Lett. 166 (2023) 53–60.
[24] M. Zhang, N. Xu, X. Geng, Feature-induced label distribution for learning with noisy labels, Pattern Recognit. Lett. 155 (2022) 107–113.
[25] E. Arazo, D. Ortego, P. Albert, N. O'Connor, McGuinness, Unsupervised label noise modeling and loss correction, in: ICML, 2019, pp. 312–321.
[26] J. Li, R. Socher, S.C. Hoi, Dividemix: Learning with noisy labels as semi-supervised learning, 2020, arXiv preprint arXiv:2002.07394.
[27] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, L. Fei-Fei, Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels, in: ICML, 2018, pp. 2304–2313.
[28] Y. Yao, Z. Sun, C. Zhang, F. Shen, Q. Wu, J. Zhang, Z. Tang, Jo-src: A contrastive approach for combating noisy labels, in: Proceedings of the IEEE/CVF CVPR, 2021, pp. 5192–5201.
[29] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, Mixmatch: A holistic approach to semi-supervised learning, Adv. Neural Inf. Process. Syst. 32 (2019).
[30] E. Zheltonozhskii, C. Baskin, A. Mendelson, A.M. Bronstein, O. Litany, Contrast to divide: Self-supervised pre-training for learning with noisy labels, in: WACV, 2022, pp. 1657–1667.
[31] B. Nagarajan, R. Marques, M. Mejia, P. Radeva, Class-conditional importance weighting for deep learning with noisy labels, in: VISIGRAPP (5: VISAPP), 2022, pp. 679–686.
[32] A. Tatjer, B. Nagarajan, R. Marques, P. Radeva, CCLM: class-conditional label noise modelling, in: IBPRIA, Springer, 2023, pp. 3–14.
[33] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, T. Liu, Understanding and improving early stopping for learning with noisy labels, Adv. Neural Inf. Process. Syst. 34 (2021) 24392–24403.
[34] Y. Tu, B. Zhang, Y. Li, L. Liu, J. Li, J. Zhang, Y. Wang, C. Wang, C.R. Zhao, Learning with noisy labels via self-supervised adversarial noisy masking, in: Proceedings of the IEEE/CVF CVPR, 2023, pp. 16186–16195.
[35] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, J. Bailey, Symmetric cross entropy for robust learning with noisy labels, in: ICCV, 2019, pp. 322–330.
[36] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, J. Bailey, Normalized loss functions for deep learning with noisy labels, in: ICML, 2020, pp. 6543–6553.
[37] Z. Liu, X. Zhang, J. He, D. Fu, D. Samaras, R. Tan, X. Wang, S. Wang, ChiMera: Learning with noisy labels by contrasting mixed-up augmentations, 2023, arXiv preprint arXiv:2310.05183.
[38] N. Karim, M.N. Rizve, N. Rahnavard, A. Mian, M. Shah, Unicon: Combating label noise through uniform selection and contrastive learning, in: Proceedings of the IEEE/CVF CVPR, 2022, pp. 9676–9686.
[39] Z. Huang, J. Zhang, H. Shan, Twin contrastive learning with noisy labels, in: Proceedings of the IEEE/CVF CVPR, 2023, pp. 11661–11670.
[40] E. Kazemi, F. Taherkhani, L. Wang, On complementing unsupervised learning with uncertainty quantification, Pattern Recognit. Lett. 176 (2023) 69–75.
[41] K. Wang, C. Zhang, Y. Geng, H. Ma, Evidential pseudo-label ensemble for semi-supervised classification, Pattern Recognit. Lett. 177 (2024) 135–141.
[42] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, M. Sugiyama, Co-teaching: Robust training of deep neural networks with extremely noisy labels, Neural Inf. Process. Syst. 31 (2018).
[43] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, M. Sugiyama, How does disagreement help generalization against label corruption? in: ICML, PMLR, 2019, pp. 7164–7173.
[44] H. Wei, L. Feng, X. Chen, B. An, Combating noisy labels by agreement: A joint training method with co-regularization, in: Proceedings of the IEEE/CVF CVPR, 2020, pp. 13726–13735.
[45] K. Cai, H. Zhang, W. Pedrycz, D. Miao, SSS-net: A shadowed-sets-based semi-supervised sample selection network for classification on noise labeled images, Knowl.-Based Syst. (2023) 110732.
[46] Y. Li, H. Han, S. Shan, X. Chen, DISC: Learning from noisy labels via dynamic instance-specific selection and correction, in: CVPR, 2023, pp. 24070–24079.
[47] X. Yu, Y. Jiang, T. Shi, Z. Feng, Y. Wang, M. Song, L. Sun, How to prevent the continuous damage of noises to model training? in: Proceedings of the IEEE/CVF CVPR, 2023, pp. 12054–12063.
[48] X. Xia, B. Han, Y. Zhan, J. Yu, M. Gong, C. Gong, T. Liu, Combating noisy labels with sample selection by mining high-discrepancy examples, in: Proceedings of the IEEE/CVF ICCV, 2023, pp. 1833–1843.
[49] C. Ahn, K. Kim, J. Baek, J. Lim, Han, Sample-wise label confidence incorporation for learning with noisy labels, in: ICCV, 2023, pp. 1823–1832.
[50] K. Nishi, Y. Ding, A. Rich, T. Hollerer, Augmentation strategies for learning with noisy labels, in: CVPR, 2021, pp. 8022–8031.
[51] H. Wang, R. Xiao, Y. Dong, L. Feng, J. Zhao, ProMix: Combating label noise via maximizing clean sample utility, 2022, arXiv:2207.10276.
[52] D. Kim, K. Ryoo, H. Cho, S. Kim, SplitNet: Learnable clean-noisy label splitting for learning with noisy labels, 2022, arXiv:2211.11753.
[53] Z. Sun, F. Shen, D. Huang, Q. Wang, X. Shu, Y. Yao, J. Tang, Pnp: Robust learning from noisy labels by probabilistic noise prediction, in: Proceedings of the IEEE/CVF CVPR, 2022, pp. 5311–5320.
[54] F. Fooladgar, M.N.N. To, P. Mousavi, P. Abolmaesumi, Manifold DivideMix: A semi-supervised contrastive learning framework for severe label noise, 2023, arXiv preprint arXiv:2308.06861.
[55] B. Nagarajan, R. Marques, E. Aguilar, P. Radeva, Bayesian DivideMix++ for enhanced learning with noisy labels, Neural Netw. (2024) 106122.
[56] Y. Huang, B. Bai, S. Zhao, K. Bai, F. Wang, Uncertainty-aware learning against label noise on imbalanced datasets, in: ICAI, 2022, pp. 6960–6969.
[57] J. Shu, X. Yuan, D. Meng, Z. Xu, Cmw-net: Learning a class-aware sample weighting mapping for robust deep learning, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[58] J. Huang, Y. Chen, J. Feng, X. Wu, Class prototype-based cleaner for label noise learning, 2022, arXiv preprint arXiv:2212.10766.
[59] J. Yao, B. Han, Z. Zhou, Y. Zhang, I.W. Tsang, Latent class-conditional noise model, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[60] Q. Wei, L. Feng, H. Sun, R. Wang, C. Guo, Y. Yin, Fine-grained classification with noisy labels, in: CVPR, 2023, pp. 11651–11660.
[61] G. Valle-Perez, C.Q. Camargo, A.A. Louis, Deep learning generalizes because the parameter-function map is biased towards simple functions, 2018, arXiv preprint arXiv:1805.08522.
[62] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Toronto, ON, Canada, 2009.
[63] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, L. Qu, Making deep neural networks robust to label noise: A loss correction approach, in: Proceedings of the IEEE CVPR, 2017, pp. 1944–1952.
[64] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer, 2016, pp. 630–645.
[65] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, 2017, arXiv preprint arXiv:1710.09412.
[66] D. Tanaka, D. Ikami, T. Yamasaki, K. Aizawa, Joint optimization framework for learning with noisy labels, in: Proceedings of the IEEE CVPR, 2018, pp. 5552–5560.
[67] K. Yi, J. Wu, Probabilistic end-to-end noise correction for learning with noisy labels, in: Proceedings of the IEEE/CVF CVPR, 2019, pp. 7017–7025.

Neural Networks 172 (2024) 106122

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Bayesian DivideMix++ for Enhanced Learning with Noisy Labels

Bhalaji Nagarajan [a,*], Ricardo Marques [a,c], Eduardo Aguilar [a,b,c], Petia Radeva [a,c]

[a] Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
[b] Dept. de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Avenida Angamos 0610, 1270709, Antofagasta, Chile
[c] Computer Vision Center, Cerdanyola (Barcelona), Spain

ARTICLE INFO

Keywords:
Learning with noisy labels
Neural network memorization
Data augmentation
Self-supervised pre-training
Label uncertainty
Monte-Carlo dropouts

ABSTRACT

Leveraging inexpensive and human intervention-based annotating methodologies, such as crowdsourcing and web crawling, often leads to datasets with noisy labels. Noisy labels can have a detrimental impact on the performance and generalization of deep neural networks. Robust models that are able to handle and mitigate the effect of these noisy labels are thus essential. In this work, we explore the open challenges of neural network memorization and uncertainty in creating robust learning algorithms with noisy labels. To overcome them, we propose a novel framework called "Bayesian DivideMix++" with two critical components: (i) DivideMix++, to enhance the robustness against memorization and (ii) Monte-Carlo MixMatch, which focuses on improving the effectiveness towards label uncertainty. DivideMix++ improves the pipeline by integrating the warm-up and augmentation pipeline with self-supervised pre-training and dedicated different data augmentations for loss analysis and backpropagation. Monte-Carlo MixMatch leverages uncertainty measurements to mitigate the influence of uncertain samples by reducing their weight in the data augmentation MixMatch step. We validate our proposed pipeline using four datasets encompassing various synthetic and real-world noise settings. We demonstrate the effectiveness and merits of our proposed pipeline using extensive experiments. Bayesian DivideMix++ outperforms the state-of-the-art models by considerable differences in all experiments. Our findings underscore the potential of leveraging these modifications to enhance the performance and generalization of deep neural networks in practical scenarios.

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable advancements in tackling complex and challenging problems such as image classification, outperforming human-level performances (Schmarje, Santarossa, Schröder, & Koch, 2021). One key contributing factor to the high performance of DNNs is the availability of large-scale training datasets, like ImageNet-21K (Ridnik, Ben-Baruch, Noy, & Zelnik-Manor, 2021), OpenImages (Kuznetsova et al., 2020), and JFT-3B (Zhai, Kolesnikov, Houlsby, & Beyer, 2022). However, curating such extensive datasets with high precision poses significant challenges. Data collection is labour-intensive and expensive in both sample accumulation and sample labelling (Liao, Kar, & Fidler, 2021). Emerging research has explored several cost-effective alternatives, like web crawling, human crowdsourcing, and creating annotations using semi-supervised models, thereby enabling the creation of large-scale training datasets. However, these methods and processes carry an inherent introduction of noise in the assigned labels (Northcutt, Jiang, & Chuang, 2021; Oyen, Kucer, Hengartner, & Singh, 2022; Wei et al., 2021). As shown in the literature, real-world datasets exhibit considerable label noise estimated between 8.0% to 38.5 Training with noisy labels impacts the generalization of the DNNs due to their over-parameterization (Allen-Zhu, Li, & Liang, 2019) and strong memorization capability (Zhang, Bengio, Hardt, Recht, & Vinyals, 2021).

Learning with Noisy Labels (LNL) has been a long-studied problem (Angluin & Laird, 1988) due to its importance in creating models that are robust towards noisy labels. Recent advances in LNL algorithms use a variety of strategies (Song, Kim, Park, Shin, & Lee, 2022), such as modifying the loss functions (Wu et al., 2021; Zhang, Niu, & Sugiyama, 2021), using more robust loss functions (Ma et al., 2020; Wang et al., 2019), adding regularization (Chen, Hu, Shen, Ai, & Suykens, 2022; Menon, Rawat, Reddi, & Kumar, 2020), sample re-weighting (Ren, Zeng, Yang, & Urtasun, 2018), and label aggregation (Wu et al., 2023). A more recent and promising class of LNL algorithms is **sample selection**, which relies on selecting clean samples for training the models. These algorithms work on the intuition that less noisy data leads to more robust DNNs (Arpit et al., 2017). Using small-loss for selecting samples is a well-established technique (Arazo, Ortego, Albert, O'Connor, & McGuinness, 2019; Li, Socher, & Hoi, 2020). Earlier

* Corresponding author.
E-mail address: bhalaji.nagarajan@ub.edu (B. Nagarajan).

https://doi.org/10.1016/j.neunet.2024.106122
Received 2 August 2023; Received in revised form 4 December 2023; Accepted 9 January 2024
Available online 10 January 2024
0893-6080/© 2024 Elsevier Ltd. All rights reserved.

sample selection-based methods suffered a significant challenge of error accumulation. Using two *co-operative* 'peer' networks reduced the error flow introduced by the noisy labels (Jiang, Zhou, Leung, Li, & Fei-Fei, 2018; Malach & Shalev-Shwartz, 2017). Co-teaching (Han et al., 2018) and Co-teaching+ (Yu et al., 2019) use clean samples of one network to train the other. On the other hand, JoCoR (Wei, Feng, Chen, & An, 2020) train two networks using a joint loss for each training example.

Advances in Semi-supervised learning have helped to reduce the need for labelled data by leveraging unlabelled data. MixMatch (Berthelot et al., 2019) integrated several dominant semi-supervised approaches into a single framework. The success of MixMatch, combined with the effectiveness of training 'peer' networks, led the way for the emergence of a prominent LNL benchmark algorithm - **DivideMix** (Li et al., 2020). DivideMix leveraged the per-sample loss distribution and modelled it using a Gaussian Mixture Model (GMM) to divide the samples into clean samples and noisy samples dynamically. Subsequently, DivideMix used the clean samples as a labelled set and noisy samples as an unlabelled set and employed an improved version of MixMatch for semi-supervised training. Ever since, DivideMix has served as the foundation for numerous LNL algorithms, such as Contrast2Divide (Zheltonozhskii, Baskin, Mendelson, Bronstein, & Litany, 2022), AugDesc (Nishi, Ding, Rich, & Hollerer, 2021), Probabilistic Noise Perception (Sun et al., 2022), ProMix (Wang, Xiao, Dong, Feng, & Zhao, 2022) and SplitNet (Kim, Ryoo, Cho, & Kim, 2022).

A typical behaviour of any DNN is to learn the patterns first and then gradually *memorize* all the samples (Arpit et al., 2017). In the case of datasets with noisy labels, the DNNs would be able to learn the clean labels first, followed by the noisy ones. Sample selection algorithms choose the small-loss samples as clean samples for subsequent training steps (Wei et al., 2020). The rationale behind this strategy is that small-loss samples are assumed to have a lower likelihood of being affected by label noise, thus making them more reliable for training the model. However, this approach to sample selection is arguable (Xia et al., 2021b). The reason is that small-loss samples might not always represent the true underlying data distribution or capture the challenging patterns present in the dataset. Also, this might not lead towards a generalized performance (Ji, Oh, Hyun, Kwon, & Park, 2021). Moreover, in most cases, the low-loss criterion uses the losses by the current prediction, making the estimation of the noisy class posterior unstable (Yao et al., 2020). Additionally, not all noisy samples have high losses. When a model encounters samples that are not so easy to learn, such as those in non-dominant classes, it would have a high loss (Xia et al., 2021b). Misleading predictions due to uncertainty can also impact the identification of clean samples (Huang, Bai, Zhao, Bai, & Wang, 2022), which, in turn, can affect the quality of training data for the subsequent steps.

Our proposal. With this understanding of LNL algorithms, we investigate the open challenges surrounding the memorization of noisy samples and the uncertainty of clean-noisy samples split that can affect the learning performance in the presence of noisy labels. We propose **Bayesian DivideMix++** using two novel components, DivideMix++ and Monte-Carlo MixMatch, to tackle these issues in the LNL algorithms. Specifically, we study the popular DivideMix algorithm for these challenges and improve the training pipeline, which, as shown by our results, leads to substantial improvements in the learning process. On this front, our main contributions to this work are outlined as follows:

- **DivideMix++:** DNN memorization has been a well-established challenge to robust LNL algorithms. We study the influence of pre-training and augmentation strategies in the challenge of memorization and establish a promising model called Bayesian DivideMix++, modifying both these phases of the training pipeline.
- **Monte-Carlo MixMatch:** We analyse the epistemic uncertainty present in the training data and improve the MixMatch component, which is one of the core components of the training process. We study the effectiveness of this improvement with respect to label uncertainty and show how using uncertainty is beneficial in sample selection.
- We evaluate the efficacy of our proposed pipeline using different benchmarks and real-world LNL datasets. We perform an extensive analysis of the results and show the improvements with respect to several state-of-the-art methods. We also present an ablation study to verify the importance of each component used in our proposal.

The rest of the paper is structured as follows: We present a comprehensive review of relevant literature in Section 2. We give the background information and rationale in Section 3. We explain the details of our proposed method in Section 4 and present the results used to validate our approach in Section 5. We provide detailed analysis in Section 6 and then present the concluding remarks in Section 7.

2. Related work

In this section, we provide an overview of the latest literature that is highly relevant to our work. We first present the works focussed on LNL algorithms and later highlight works based on Uncertainty quantification.

2.1. Learning with noisy labels

LNL has garnered significant research interest in recent years, and several works have been proposed using deep learning algorithms to tackle the problem of label noise (Song et al., 2022). LNL algorithms fall into different families of algorithms based on their mode of operation.

Loss correction methods. Loss correction methods modify the loss of samples using a noise transition matrix (Zhang, Niu, & Sugiyama, 2021). The transition matrices are constructed using the probability that a clean label flips into a noisy label (Patrini, Rozza, Menon, Nock, & Qu, 2016). Class2Simi (Wu et al., 2021) instead used a noise label similarity transformation to simplify the LNL problem. Dual-T (Yao et al., 2020) factorized the transition matrix into a product of two easy-to-estimate transition matrices. However, in these methods, estimating the label transition matrix is quite challenging as it is hard to calculate this probability accurately.

Regularization-based methods. Regularization-based methods find wide usage, as they limit the memorization of labels and make algorithms more robust. Using Data Augmentation (DA) strategies such as Mixup (Zhang, Cisse, Dauphin, & Lopez-Paz, 2017), adding dropouts (Chen et al., 2022), and using gradient clipping (Menon et al., 2020) have proved to be effective in LNL solutions. Advanced regularization techniques such as Progressive Early Stopping (Bai et al., 2021), Neighbour Consistency Regularization (Iscen, Valmadre, Arnab, & Schmid, 2022), Multi-Objective Interpolation Training (Ortego, Arazo, Albert, O'Connor, & McGuinness, 2021), k-nearest neighbour-based filtering (Bahri, Jiang, & Gupta, 2020), Robust early-learning (Xia et al., 2021a), Sparse over-parameterization (Liu, Zhu, Qu, & You, 2022) and Scalable Penalized Regression (Wang, Sun, & Fu, 2022) have been successful in creating robust LNL algorithms. Importantly, these methods often entail complex architectures and sensitive hyperparameters, making it difficult during training.

Noise-robust loss functions and architectures. Another category of LNL algorithms uses noise-robust loss functions. Mean Absolute Error (Ghosh, Kumar, & Sastry, 2017) was more robust than the standard cross-entropy loss. Symmetric cross-entropy learning (Wang et al., 2019) used a reverse cross-entropy to boost the cross-entropy loss in handling the overfitting class labels. Active Passive Loss (Ma et al., 2020) combined two robust loss functions mutually helping each other. Reweighting samples (Ren et al., 2018), making robust architectures such as

noise adaptation layer (Goldberger & Ben-Reuven, 2017), and using distillation (Li, Yang, et al., 2017) are some of the other methods that have increased the robustness of LNL algorithms. Recently, contrastive learning algorithms such as Sel-CL+ (Li, Xia, Ge, & Liu, 2022), supervised contrastive learning for label correction (Huang, Lin, & Xu, 2022), initializing with representations learned by contrastive learning (Ghosh & Lan, 2021) have proven successful in LNL training.

Sample selection methods. Of late, sample selection techniques have become very popular amongst the LNL community. These methods rely on selecting a possible clean subset of data and providing different training strategies for the clean and noisy subsets. The challenge is finding the criterion used in selecting clean or noisy samples. Several strategies are used in sample selection, such as clustering training data (Huang, Qu, Jia, & Zhao, 2019) and progressively removing the noisy samples from subsequent training steps. A common strategy in sample selection relies on per-sample loss, designating samples with minimal loss as clean samples (Jiang et al., 2018). Following this, a two-component beta-mixture model (Arazo et al., 2019) was employed to fit the loss distribution and split the training samples into clean and noisy samples. Alternatively, DivideMix (Li et al., 2020) used a GMM to model the loss distribution. GMMs have since been widely used in several methods. Semi-supervised approach (Ding, Wang, Fan, & Gong, 2018) based on a two-stage framework was used to identify samples that could possibly mislead the learning process. UNICON (Karim, Rizve, Rahnavard, Mian, & Shah, 2022), Gradient Switching Strategy (Yu et al., 2023), and SSS-Net (Cai, Zhang, Pedrycz, & Miao, 2023) are recent methods that focus on selecting clean samples. In general, sample selection methods work well. However, they suffer from accumulated errors caused by incorrect selection. With this notion, several studies have employed multiple DNNs (Han et al., 2018; Malach & Shalev-Shwartz, 2017; Tan, Xia, Wu, & Li, 2021), thereby reducing the confirmation bias.

Multi-network learning. Decoupling (Malach & Shalev-Shwartz, 2017) updated the model parameters using only instances on which the predictions of two networks are different. MentorNet (Jiang et al., 2018) employed curriculum-based learning to guide the student network. Co-teaching (Han et al., 2018) trained two DNNs simultaneously, allowing them to teach each other based on small-loss samples and Co-teaching++ (Yu et al., 2019) improved co-teaching by adapting the disagreement strategy. JoCoR (Wei et al., 2020) used co-regularization to bring the prediction of networks closer to each other, while JoSRC (Yao et al., 2021) used Jensen–Shannon divergence to estimate the likelihood of a sample 'being' clean. Recent advancements in various fields have spurred the emergence of innovative approaches that combine sample selection techniques with other novel concepts, resulting in improved performance compared to the existing methods. SELF (Nguyen et al., 2019) used gradual filtering of noisy labels during the training using running averages of predictions as ensembles. Similarly, SELC (Lu & Selc, 2022) worked on gradually correcting the noisy labels. Co-learning (Tan et al., 2021) instead used a shared encoder with two heads, one self-supervised and another supervised and maximized the agreement between them.

The DivideMix family. One of the most popular LNL multi-network benchmark algorithms in the family of sample selection is DivideMix (Li et al., 2020). DivideMix used two networks to train each other by employing GMMs to split the training data into clean and noisy samples. The GMMs are modelled based on the per-sample loss distribution. DivideMix used an improved MixMatch (Berthelot et al., 2019) to train the networks. Different stages in the DivideMix algorithm were further improved so as to mitigate the potential drawbacks in the training pipeline. The importance of the warm-up stage in LNL algorithms was studied in Contrast2Divide (Zheltonozhskii et al., 2022), which proposed using self-supervised pre-training to create better feature extractors. AugDesc (Nishi et al., 2021) explored DA techniques in LNL algorithms and proposed using two different augmentations — one for analysing the loss and the other for backpropagation. CCLM (Tatjer, Nagarajan, Marques, & Radeva, 2023) studied the per-sample loss modelling of DivideMix and proposed a class-conditional approach to split the clean and noisy samples. A dynamic class-conditional weighting focused on less-learned classes was used to improve the balance of losses in learning samples (Nagarajan, Marques, Mejia, & Radeva, 2022). Several methods adapt the training pipeline of DivideMix. Probabilistic Noise Perception (Sun et al., 2022) used different optimization objectives for the specific samples based on two networks — one to predict the category label and the other to predict the noise type. ProMix (Wang, Xiao, et al., 2022) combined the small-loss sample selection with a high confidence-based selection. SplitNet (Kim et al., 2022) used an additional complementary learning module to predict the clean-noisy label.

Sample selection algorithms focus on avoiding confirmation bias and try to use the entire noisy data to create robust LNL models. The main challenge faced by these algorithms revolves around DNN memorization. Given the importance of DivideMix in solving the LNL problem, we investigate the training pipeline of DivideMix with the notion of tackling this issue. Importantly, we delve into the role of warm-up and DAs, which are of prime importance during the training step. Our results highlight how the proposed pipeline addresses the issue of DNN memorization.

2.2. Uncertainty quantification

DNNs typically contain millions of trainable parameters, far more than the number of training samples. Despite their vast capacity, these models often encounter challenges in generalization (Zhang, Bengio, et al., 2021). Different uncertainty estimation techniques such as Deep Ensembles (Lakshminarayanan, Pritzel, & Blundell, 2017) and MC-Dropouts (Gal & Ghahramani, 2016) are employed to find the prediction confidence of the models. Bayesian probability theory provides tools to reason about model uncertainty (Gal & Ghahramani, 2016). Bayesian approaches offer a principled framework to capture uncertainty by introducing a distribution over model parameters and then sampling multiple sets of parameters to form a predictive distribution (Maddox, Izmailov, Garipov, Vetrov, & Wilson, 2019). Bayesian modelling allows capturing and modelling two distinct types of uncertainty: *aleatoric uncertainty*, which captures the noise in observations and *epistemic uncertainty*, which accounts for uncertainty in model parameters (Der Kiureghian & Ditlevsen, 2009). Epistemic uncertainty reflects the lack of knowledge of the model parameters and often arises due to limited data. Exact Bayesian inference is intractable in DNNs and, therefore, usually approximated; Bayesian approximation methods fall majorly into (1) Sampling-based, (2) Variational Inference, or (3) Hybrid approaches (Lange, Benjamin, Haefner, & Pitkow, 2022; Wang & Yeung, 2020). Abdar et al. (2021) presents an in-depth exploration of different uncertainty quantification techniques. By leveraging the confidence of model outputs, it is possible to handle the uncertain inputs and the special cases explicitly. Uncertainty measurements have been incorporated into various LNL algorithms previously. An iterative label noise filtering process using ensembles and MC-Dropouts (Köhler, Autenrieth, & Beluch, 2019) was introduced based on the assumption that the predictive uncertainties of clean and noisy data follow two different distributions. Uncertain Aware Co-Training (Ji et al., 2021) studied the relationship between uncertainty and clean labels using MC-Dropouts and used it as a criterion to choose the samples along with the small-loss trick. Entropy-based Debiasing framework (Oh, Lee, Byun, & Shin, 2022) used predictive uncertainty to identify minority group samples while preventing the models from learning the noisy labels. The reliability of uncertainty estimation methods in the presence of label noise was critically analysed (Pan, Yuan, Zhou, & Yao, 2022). Specifically, Goel and Chen (2021) performed an in-depth analysis of the effectiveness of using MC-Dropouts in the presence of label noise. Combats Noisy Labels by Concerning Uncertainty (CNLCU) algorithm (Xia

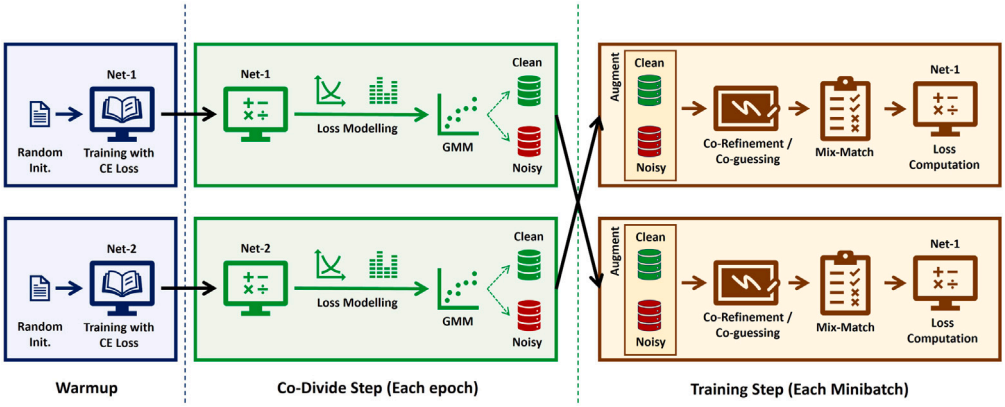


Fig. 1. DivideMix uses co-teaching of two models (Net-1 and Net-2). During each epoch, the loss distribution of each model is used to divide the dataset into clean and noisy data (Co-Divide), which is used to train the other model. During each mini-batch of the training step, the models are trained in a semi-supervised manner using MixMatch.

et al., 2021b) used the uncertainty of losses by adopting interval estimation instead of point estimation. CNLCU works on large loss samples, distinguishing between mislabelled or underrepresented samples. Uncertainty-aware Label Correction framework (Huang, Bai, et al., 2022) studied the impact of label noise on imbalanced datasets and used a combination of epistemic uncertainty and aleatoric uncertainty to boost the performance of the LNL algorithm.

Following the importance and the role of uncertainty in improving and interpreting DNNs, particularly in LNL problems, we analyse the epistemic uncertainty of the training data by converting the deterministic model into a Bayesian one. We identify the role of MixMatch as an important one in the training pipeline, where improving this step would provide a better robust learning pipeline. In this regard, we employ the measured uncertainty in order to create an improved version of MixMatch for data augmentation, which we call Monte-Carlo MixMatch. This step differs from the other techniques, which only use uncertainty computations to modify the loss function.

3. Background and rationale

First, we provide a concise overview of the baseline algorithm for LNL, DivideMix, which provides the background information for understanding the LNL algorithm pipeline. Later, we outline the key challenges we tackle using our proposed method.

3.1. DivideMix

Small-loss-based sample selection methods rely on the hypothesis that clean samples are faster to learn than noisy samples, leading to a lower loss for clean samples (Arpit et al., 2017). Formally, let $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$ denote a noisy training set of N samples, where x_i is the ith image and $\tilde{y}_i \in \{0,1\}^C$ is the one-hot label over C classes. The goal is to develop a robust learning algorithm capable of effectively handling and mitigating the impact of label noise, ultimately ensuring reliable and accurate predictions on unseen data. DivideMix algorithm (Li et al., 2020) trains two homogeneous models simultaneously, thereby avoiding the confirmation bias (Tarvainen & Valpola, 2017). Different random components, such as parameter initialization and batch sequences, make the models diverge from each other. We show an overview of the DivideMix algorithm in Fig. 1. Before employing the learning steps of the algorithm, a **warm-up** (sapphire-coloured (■) in Fig. 1) is performed for both models. During the warm-up phase, the models are trained for a few epochs using the standard cross-entropy loss. The warm-up allows the model to achieve initial convergence and enable better loss separations and feature representations.

Given a model with parameters θ, the cross-entropy loss $l(\theta)$ is defined as follows:

$$l(\theta) = \{l_i\}_{i=1}^{N} = \left\{ -\sum_{c=1}^{C} \tilde{y}_i{}^c log\left(p_{model}^c(x_i;\theta)\right) \right\}_{i=1}^{N} \quad (1)$$

where p_{model}^c is the model's output softmax probability for class c.

Once the model is capable of producing an informed loss, DivideMix exploits the above-mentioned hypothesis. To this end, a two-component GMM, g, is fit to the loss, $l(\theta)$, using the Expectation–Maximization algorithm. The resulting GMM is then used to produce a probability density function $p(\cdot)$, such that $p(x_i \mid l_i; g)$ yields the posterior probability of x_i being clean, given its loss l_i. It has been shown by Li et al. (2020) that using two different GMMs, each associated with a model, mitigates the confirmation bias. The GMM of one model is used to split the training data, and subsequently, the other model is trained on this data. This division strategy, named **Co-Divide** (jade-coloured (■) in Fig. 1), is determined by a threshold τ over the probability density function $p(\cdot)$, to split the training set into a labelled set ($\mathcal{X}$) and an unlabelled set ($\mathcal{U}$).

Next, DivideMix incorporates semi-supervised learning to improve the model's performance. During each training step (highlighted in saddle brown (■) in Fig. 1), an improved **MixMatch** (Berthelot et al., 2019) is applied on both the labelled set and the unlabelled set. MixMatch uses DAs on both labelled and unlabelled sets. A set of augmentations, M, is created for every sample in the batch. In the case of the labelled set, the model predictions are averaged across augmentations and refined using a temperature-sharpening step. For the unlabelled set, the model predictions are averaged across the augmentations to *guess* a pseudo-label for each sample.

Using a combined mini-batch of $\widehat{\mathcal{X}}$ and $\widehat{\mathcal{U}}$, the MixMatch algorithm uses MixUp (Zhang et al., 2017) to interpolate two random samples to produce a modified labelled set, $\mathcal{X}'$ and an unlabelled set, $\mathcal{U}'$. The sum of the cross-entropy loss for the modified labelled set ($\mathcal{X}'$), the mean-squared error for the unlabelled set ($\mathcal{U}'$), and a regularization term (as used by Arazo et al. (2019), Tanaka, Ikami, Yamasaki, and Aizawa (2018)) constitutes the total training loss. DivideMix improves MixMatch by: (i) Label co-refinement for the labelled set ($\widehat{\mathcal{X}}$): Combining ground-truth labels with the model's prediction and clean probability of the other model; (ii) Co-guessing for the unlabelled data ($\widehat{\mathcal{U}}$): Ensemble both models to predict unlabelled data.

3.2. The challenge of uncertainty

The interpretation of predictive probabilities as model confidence on classification tasks is often inaccurate (Gal & Ghahramani, 2016). Model confidence is crucial in capturing uncertain inputs and effectively handling those cases. Bayesian probability theory offers a solid

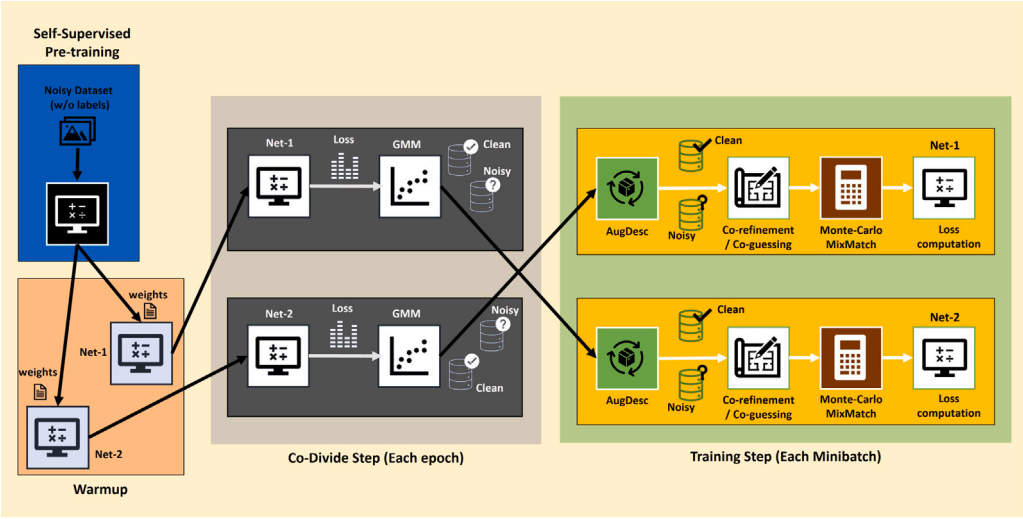


Fig. 2. Bayesian DivideMix++ uses co-teaching of two models (Net-1 and Net-2), initialized with **Self-supervised pre-trained weights** (■). During each epoch, the loss distribution of each model is used to divide the dataset into clean and noisy data (Co-Divide), which in turn is used to train the other. During each mini-batch of the training step, **AugDesc** (■) is used to augment the data splits followed by **Monte-Carlo MixMatch** (■) to train in a semi-supervised way.

mathematical framework to effectively reason about model uncertainty (Jospin, Laga, Boussaid, Buntine, & Bennamoun, 2022) and is robust to over-fitting. In the context of LNL, uncertainty estimation methods ensure a delayed memorization effect of noisy labels and contribute to improved generalization on the test set (Goel & Chen, 2021). By embracing the principles of Bayesian inference, we claim to achieve a more robust estimation of uncertainty, leading to enhanced model performance for LNL.

3.3. The challenge of DNN memorization

In a dataset with noisy labels, the DNNs learn to fit on the easy clean samples during the early learning stages. Later, the samples with noisy labels are memorized (Arpit et al., 2017). This phenomenon is well-studied (Zhang, Bengio, et al., 2021) and exploited to make the models more robust to label noise. Early learning and memorization have been fundamental to any high-dimensional learning task (Liu, Niles-Weed, Razavian, & Fernandez-Granda, 2020). With this regard, we dive into the role of *warm-up* and *data augmentations* in making the LNL algorithm more robust. The *warm-up* phase of DivideMix ensures two critical purposes: enhanced loss separability for the follow-up stages and better feature extraction. It is a well-analysed and documented procedure to initialize the training procedures using clean labelled datasets such as ImageNet (Li et al., 2020; Li, Wong, Zhao, & Kankanhalli, 2019; Patrini, Rozza, Krishna Menon, Nock, & Qu, 2017; Tanaka et al., 2018). However, the benefits of supervised pre-training are generally inconsistent on LNL algorithms, where the influence of factors such as noise level and domain gap plays a critical role in achieving high-quality features (Zheltonozhskii et al., 2022). The primary goal of any *data augmentation* technique is to enhance the generalization capability of the dataset. However, it is crucial to ensure that the use of DA does not negatively impact loss modelling and the convergence of the algorithms. LNL algorithms such as DivideMix typically employ simple augmentations such as random flips and crops (called *weak augmentations*). However, using the basic set of augmentations during the learning step is suboptimal (Nishi et al., 2021) as it limits the generalization abilities of the models. Incorporating additional *strong* augmentations such as AutoAugment (Cubuk, Zoph, Mane, Vasudevan, & Le, 2019) and RandAugment (Cubuk, Zoph, Shlens, & Le, 2020) is highly advantageous, especially in high-noise settings. The additional variations and transformations introduced by these augmentations contribute to a more diverse training set, effectively reducing overfitting and increasing the model's ability to generalize well.

4. Bayesian DivideMix++

In this section, we present our proposed **Bayesian DivideMix++** with a comprehensive explanation of each of the two novel components, uncertainty-aware DivideMix++ and Monte-Carlo MixMatch, and their role in enhancing the performance of LNL algorithms by tackling the above-listed challenges. We show the overview of our proposed method in Fig. 2.

4.1. Bayesian formulation of DivideMix

Training DNNs with dropouts can be interpreted as an approximate Bayesian inference of the weight's posterior (Gal & Ghahramani, 2016). Dropouts have been widely used in several applications to measure uncertainty and have been very effective. Integrating models with Monte-Carlo (MC) Dropout helps in reducing the decline in classification performance when data is more challenging or ambiguous (Goel & Chen, 2021).

In this work, we take advantage of the uncertainty information captured using MC-Dropouts to improve the MixMatch step of DivideMix. Let d_l denote a dropout at the lth layer, where $d_l \sim Bernoulli(p_b)$, with p_b being the dropout probability. Let us perform at inference time K forward passes of the LNL algorithm; hence, we obtain a distribution of K logits and predictions for each input data. Here, we use Shannon entropy (Lin, 1991) as a measure of uncertainty. For any sample x_i, the Shannon entropy $\mathcal{H}$ is calculated as:

$$\mathcal{H}(x_i) = -\sum_{c=1}^{C} p_c(x_i) log(p_c(x_i))$$

where $p_c(x_i)$ is the output softmax probability for class c.

Epistemic uncertainty affects the identification of clean samples from noisy samples (Huang, Bai, et al., 2022). The robustness property and ease of implementation of MC-Dropouts indicate that it is an effective and practical solution against noisy labels.

Monte-Carlo MixMatch. Here, we propose a new variant of MixMatch, called Monte-Carlo MixMatch (MC-MixMatch) (highlighted in saddle brown (■) in Fig. 2), based on the uncertainty of samples. MixMatch combines two samples randomly by interpolating them using a weighted combination. MixMatch uses a random parameter, λ, which is derived from a Beta distribution to combine the two samples. However, in our proposed component, MC-MixMatch, the combination is determined by the parameter λ_{unc}, which is derived from a Beta distribution

and weighted using the uncertainty of the samples $\mathcal{H}$. Given a pair of samples (x_1, x_2) and their corresponding labels $(\tilde{y_1}, \tilde{y_2})$, the mixed sample (x', y') is computed by:

$$\lambda \sim Beta(\alpha, \alpha) \tag{2}$$

$$\lambda' = max(\lambda, 1-\lambda) \tag{3}$$

$$\lambda_{unc} = 0.5 \times (\mathcal{H}(x_1) * \lambda' + 1)$$

$$x' = \lambda_{unc}\, x_1 + (1-\lambda_{unc})\, x_2$$

$$y' = \lambda_{unc}\, y_1 + (1-\lambda_{unc})\, y_2$$

where $\mathcal{H}(x_1)$ is the uncertainty of sample x_1. The λ_{unc} is computed, only using the uncertainty of the first sample $\mathcal{H}(x_1)$, with the notion that the second sample also undergoes a similar learning process during some part of the training step. The importance of the sample is determined using the uncertainty of the samples. Highly uncertain samples are weighted more in the interpolation, thereby giving more importance to those samples. Meanwhile, highly certain samples are weighted comparatively less, helping the model to generalize better. Note that by Eq. (3), the weight of sample x_1 is always higher than 0.5.

4.2. Diversity-increasing DivideMix++

In our proposed DivideMix++, we tackle the challenge of DNN memorization by leveraging the advantages of self-supervised pre-training and dedicated augmentation strategies within the training pipeline. First, we use self-supervised pre-training weights (shown in navy blue (■) in Fig. 2) to initialize the model rather than starting with random initializations or with supervised pre-training. By this step, we discard the presence of labels during the warm-up phase, which in turn leads to removing the influence of label noise. The pre-trained features are thus agnostic to noisy labels, which results in more effective and reliable feature representations (Zheltonozhskii et al., 2022).

The benefits of self-supervised pre-training are twofold. First, the model becomes more robust to the presence of noisy labels during subsequent supervised training, improving the model's ability to generalize and make accurate predictions. Second, the pre-training enhances the model's discriminative power by extracting more informative and representative features. Self-supervised pre-training thus provides a robust starting phase for subsequent training steps, laying a solid foundation for the model's feature representation.

The next improvement, we make is to modify each training batch by incorporating two separate sets of augmentations, one for loss analysis and another for gradient descent, instead of a single augmentation step (highlighted in birch green (■) in Fig. 2). This strategy allows us to leverage the benefits of diverse transformations and expand the augmentation space. By incorporating two different sets of augmentations, we introduce additional variations and increase the diversity within the training data. By expanding the augmentation set, we empower the model to learn from a richer set of training samples. However, applying augmentations during warm-up negatively impacts the loss convergence (Nishi et al., 2021). Hence, we perform this addition only to the training step and not during the warm-up phase. Self-supervised pre-training aids in better loss separation of clean and noisy samples and the modified augmentation strategy positively impacts the loss modelling during the training step.

4.3. Overall training

We train the proposed Bayesian DivideMix++ similar to the DivideMix algorithm, retaining the loss function:

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_u \mathcal{L}_{\mathcal{U}'} + \lambda_r \mathcal{L}_{reg} \tag{4}$$

where $\mathcal{L}_{\mathcal{X}}$ is the cross-entropy loss over the labelled set, $\mathcal{L}_{\mathcal{U}'}$ is the mean-squared error over the unlabelled set, $\mathcal{L}_{reg}$ is the regularization term, λ_u and λ_r are used to control the strength of the loss terms. Incorporating self-supervised pre-training during warm-up and AugDesc during training constitutes the DivideMix++, whereas MC-MixMatch is added to DivideMix++ to enhance the model generalization, thereby comprising the proposed Bayesian DivideMix++.

5. Experiments

In this section, we describe the experiment settings used to validate our approach. We evaluate the performance on two synthetic noise datasets - CIFAR-10 (Krizhevsky, Hinton, et al., 2009) and CIFAR-100 (Krizhevsky et al., 2009) at different noise rates, as well as on two real noise datasets — WebVision (Li, Wang, Li, Agustsson, & Van Gool, 2017) and Clothing1M (Xiao, Xia, Yang, Huang, & Wang, 2015). Comparisons with state-of-the-art models demonstrate the effectiveness of our approach.

5.1. Datasets

Both the **CIFAR datasets** consist of 50,000 training samples and 10,000 testing samples where each image is of size 32 × 32. We follow common practices in synthetic noise benchmarks and vary the amount of injected noise. We study two types of synthetic label noise - *Symmetric* and *Asymmetric*. Symmetric noise is introduced by replacing the labels of a percentage of samples with random labels (all possible classes) drawn from a uniform distribution. Asymmetric noise is closer to real-world label noise, where the labels are only replaced by similar classes. The noise is injected, following Patrini et al. (2017). The labels of CIFAR-10 are flipped as Truck→Automobile, Bird→Airplace, Deer→Horse, Dog↔Cat. For CIFAR-100, the classes are grouped into 20 super-classes of five (e.g. Aquatic Mammals contain Beaver, Dolphin, Otter, Seal and Whale), and the noise flips each class into the next circularly. Following the previous works (Li et al., 2020; Nishi et al., 2021; Zheltonozhskii et al., 2022), we test our method on 20%, 50%, 80% and 90% symmetric noise. We also test our approach on 40% asymmetric noise on the CIFAR-100 dataset.

WebVision 1.0 contains 2.4 million images from the 1000 ILSVRC12 ImageNet classes (Russakovsky et al., 2015). The dataset is collected from Flickr and Google images and is estimated to have a label noise of ≈20 Following previous work (Li et al., 2020), we use the first 50 classes of the Google image subset for comparisons. **Clothing1M** is a real-world apparel dataset consisting of 1 million images of 256 × 256 split into 14 categories. The cloth classes are based on online shopping website categories. The dataset is estimated to have a label noise of 38.5%. To ensure comparability, we maintain the dataset pre-processing the same as those in Li et al. (2020, 2019). We resize the images to 256 × 256, crop the middle 224 × 224, and perform normalization. We use the standard train/val/test split for all experiments. Both WebVision and Clothing1M include clean validation and evaluation sets.

5.2. Implementation details

For all experiments, we follow the setup similar to those in Li et al. (2020), Nishi et al. (2021) and Zheltonozhskii et al. (2022). For CIFAR datasets, we use 18-layer PreAct ResNet (He, Zhang, Ren, & Sun, 2016) as the backbone architecture. For self-supervised pre-training, we use SimCLR (Chen, Kornblith, Norouzi, & Hinton, 2020) trained weights. Following Zheltonozhskii et al. (2022), we keep a batch size of 64 and increase the number of epochs to 360 for all our experiments. We set the initial learning rate of 0.02 and reduced it by a factor of 10 after 150 epochs. Strong pre-trained weights reduce the warm-up duration, and we use a warm-up period of 5 epochs for both CIFAR datasets following Zheltonozhskii et al. (2022) (DivideMix uses 10 and 30 epochs for CIFAR-10 and CIFAR-100 respectively). The number of augmentations ($M = 2$) is fixed, as in other methods (Nagarajan et al., 2022; Nishi et al., 2021; Zheltonozhskii et al., 2022) for fair comparisons. The GMM

Table 1
Summary of **Experiment Settings.**

Parameter/Dataset	CIFAR	WebVision	Clothing1M
Architecture	18-layer PreAct ResNet	ResNet-50	ResNet-50
Batch Size	64	32	32
Epochs	360	100	80
Initial LR	0.02	0.01	0.002
LR Reduction Epochs	150	50	40
LR Reduction Factor	10	10	10
(Bayesian DivideMix++ Specific Parameters)			
Warm-up (Epochs)	5	1	1
GMM Threshold (τ)	0.5	0.5	0.5
Beta Parameter (α)	4	0.5	0.5
Sharpening Temperature (T)	0.5	0.5	0.5
Number of Augmentations (M)	2	2	2

Table 2
Unsupervised loss weights (λ_u **in Eq.** (4)) for CIFAR datasets.

Dataset	20%	50%	80%	90%
CIFAR-10	0	25	25	50
CIFAR-100	25	150	500	500

threshold (τ) is maintained as 0.5 for all noise ratios, except for the 90% noise ratio, where it is set as 0.6, following the original DivideMix implementations. The beta distribution parameter of MixMatch (α in Eq. (2)) is set to 4 and the sharpening temperature (T) is maintained at 0.5. We present a summary of experiment settings in Table 1.

The only parameter that has a considerable effect on the algorithm is the unsupervised loss weight (λ_u in Eq. (4)) and is fixed as in Table 2. For higher noise settings of CIFAR-100 (80% and 90%), we follow the values of Zheltonozhskii et al. (2022), whereas the rest are following Li et al. (2020). We use these values following the experiments of Zheltonozhskii et al. (2022), where higher λ_u benefited the training of models. For the Asymmetric 40% noise of CIFAR-100, we set λ_u to 150. We report the sensitivity of Bayesian DivideMix++ for hyperparameters in Section 6.4.3. We find the results to be as per the parameters reported by Zheltonozhskii et al. (2022), and therefore, we maintain the same for fair comparisons.

For WebVision and Clothing1M, we use ResNet-50 as the backbone architecture. For both WebVision and Clothing1M, we use the same hyperparameters: $\alpha = 0.5$, $\lambda_u = 0$, $M = 2$, $\tau = 0.5$, $T = 0.5$. We use a warm-up period of 1 epoch. For WebVision, we train the models for 100 epochs, with an initial learning rate of 0.01, reduced by a factor of 10 after 50 epochs. For Clothing1M, we train the models for 80 epochs, with an initial learning rate of 0.002, reduced by a factor of 10 after 40 epochs. For all experiments (CIFAR and real-noise), we use SGD with a momentum of 0.9 and weight decay of 5e-4. For CIFAR experiments, we train on an NVIDIA 2080 Ti GPU, whereas for the real noise dataset, we use an NVIDIA 3090 GPU.

Augmentation Policies: For all experiments, we follow Nishi et al. (2021) and use the AugDesc-WS augmentation policy. This augmentation policy corresponds to applying weak augmentations (Resize, Random crop, Flip) for the loss analysis and strong augmentations (including ImageNet policies and CIFAR policies of AutoAugment (Cubuk et al., 2019) along with the weak augmentations) for the optimization steps. For CIFAR and WebVision experiments, we use weak augmentations during the warm-up phase, whereas for Clothing1M experiments, we use strong augmentations during the warm-up phase. We do this change to adapt to the complex noise structure of Clothing1M following Nishi et al. (2021).

Measuring Uncertainty: Following the works of Boluki, Ardywibowo, Dadaneh, Zhou, and Qian (2020) and Rizve, Duarte, Rawat, and Shah (2021), we use $K = 10$ forward passes to calculate the uncertainty. For the CIFAR experiments, we use one dropout layer after the 4th ResNet block and one before the Fully Connected layer with a dropout ratio (p_b) of 0.1. We explain with experiments the decision of the placement of the dropout layer and dropout ratio in Section 6.4. We use max-scaling to normalize the uncertainty values obtained using Shannon entropy. As with DivideMix, the λ_{unc} is maintained $\in [0.5, 1]$. For high noise settings (80% and 90% symmetric noise of both CIFAR datasets), we use a settling period of 150 epochs before using uncertainty measurements in the training pipeline. For WebVision and Clothing1M experiments, we use a dropout ratio (p_b) of 0.2. We keep the settling period as 50 epochs for WebVision and 40 for Clothing1M experiments.

5.3. Results

We first provide an overview of the different state-of-the-art methods we compare against our proposed method. Later, we highlight the performance of our approach across various datasets. We discuss the results in detail and provide insights on the improvements.

5.3.1. Comparison methods

We compare the results with state-of-the-art LNL models: DivideMix (Li et al., 2020) proposes a semi-supervised learning framework based on per-sample loss distribution modelling; AugDesc (Nishi et al., 2021) presents DA strategies for LNL algorithms; Contrast2Divide (Zheltonozhskii et al., 2022) employs self-supervised pre-training to tackle the problem of warm-up; Sel-CL+ (Li et al., 2022) is a confidence-based selection method for supervised contrastive LNL; UNICON (Karim et al., 2022) uses Jensen–Shannon divergence-based sample selection replacing probabilistic modelling; ULC (Huang, Bai, et al., 2022) exploits uncertainty measurements in label correction for imbalanced LNL problems; LongReMix (Cordeiro et al., 2023) utilizes oversampling of smaller selected high-confidence clean sets to build a robust LNL algorithm; Lipschitz Regularization (Miao et al., 2023) uses Bayes' rule to detect clean samples and combines FixMatch and MixUp to improve the semi-supervised learning step. Note that all mentioned methods above use DivideMix as the baseline method. Also, note that all comparing methods use the same backbone for all the reported datasets as used by the proposed Bayesian DivideMix++.

5.3.2. Synthetic noise datasets

We show the results for CIFAR-10 experiments in Table 3 and for CIFAR-100 in Table 4 with different levels of **symmetric noise**. Following the comparison methods, we report the **best** test accuracy (%) over all epochs and the average of the last 10 epochs (**last**). We report the mean and standard deviation of the best and the last test accuracy computed over five runs. Note that we use the settling period described in Section 5.2 to obtain the results for high noise settings (80% and 90%). Bayesian DivideMix++ outperforms the state-of-the-art methods by a considerable margin across all noise ratios, which can be attributed to bringing together the self-supervised pre-training and augmentation strategies, thereby reducing the degree of memorization. Also, by including uncertainty measurements using MC-MixMatch in the learning step, the proposed Bayesian DivideMix++

Table 3
CIFAR-10 — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods on different symmetric noise settings.

Method / Noise ratio		20%	50%	80%	90%
DivideMix (ICLR, 2020)	Best	96.1	94.6	93.2	76.0
Li et al. (2020)	Last	95.7	94.4	92.9	75.4
DM + AugDesc (CVPR, 2021)	Best	96.3	95.4	93.8	91.9
Nishi et al. (2021)	Last	96.2	95.1	93.6	91.8
DM + C2D (WACV, 2022)	Best	**96.43±0.07**	95.32±0.12	94.40±0.04	93.57±0.09
Zheltonozhskii et al. (2022)	Last	96.23±0.09	95.15±0.16	94.30±0.12	93.42±0.09
Sel-CL+ (CVPR, 2022)	Best	95.5	93.9	89.2	81.9
Li et al. (2022)	Last	–	–	–	–
UNICON (CVPR, 2022)	Best	96.0	95.6	93.9	90.8
Karim et al. (2022)	Last	–	–	–	–
ULC (AAAI, 2022)	Best	96.1	95.2	94.0	86.4
Huang, Bai, et al. (2022)	Last	95.9	94.7	93.2	85.8
LongReMix (PR, 2023)	Best	96.3±0.1	95.1±0.1	93.8±0.2	79.9±2.7
Cordeiro, Sachdeva, Belagiannis, Reid, and Carneiro (2023)	Last	96.0±0.1	94.8±0.1	93.8±0.2	79.1±3.1
Lipschitz Reg. (PR, 2023)	Best	96.2	95.2	93.4	85.0
Miao, Wu, Xu, Zuo, and Meng (2023)	Last	95.7	94.8	93.1	84.3
Bayesian DivideMix++	Best	96.39±0.06	**95.68±0.09**	**95.25±0.08**	**94.46±0.15**
	Last	96.13±0.07	95.40±0.11	94.97±0.02	94.20±0.12

Table 4
CIFAR-100 — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods on different symmetric noise settings.

Method / Noise ratio		20%	50%	80%	90%
DivideMix (ICLR, 2020)	Best	77.3	74.6	60.2	31.5
Li et al. (2020)	Last	76.9	74.2	59.6	31.0
DM + AugDesc (CVPR, 2021)	Best	79.5	77.2	66.4	41.2
Nishi et al. (2021)	Last	79.2	77.0	66.1	40.9
DM + C2D (WACV, 2022)	Best	78.69±0.17	76.43±0.25	67.78±0.30	58.7±0.31
Zheltonozhskii et al. (2022)	Last	78.32±0.35	76.07±0.41	67.43±0.30	58.45±0.30
Sel-CL+ (CVPR, 2022)	Best	76.5	72.4	59.6	48.8
Li et al. (2022)	Last	–	–	–	–
UNICON (CVPR, 2022)	Best	78.9	77.6	63.9	44.8
Karim et al. (2022)	Last	–	–	–	–
ULC (AAAI, 2022)	Best	77.3	74.9	61.2	34.5
Huang, Bai, et al. (2022)	Last	77.1	74.3	60.8	34.1
LongReMix (PR, 2023)	Best	77.9±0.2	75.5±0.2	62.3±0.5	34.7±0.3
Cordeiro et al. (2023)	Last	77.5±0.2	74.9±0.2	61.7±0.5	30.7±5.9
Lipschitz Reg. (PR, 2023)	Best	78.8	75.3	62.6	37.5
Miao et al. (2023)	Last	77.4	74.3	61.0	34.7
Bayesian DivideMix++	Best	**80.02±0.03**	**78.31±0.14**	**70.01±0.23**	**61.15±0.34**
	Last	79.56±0.13	77.71±0.13	69.55±0.22	60.70±0.42

can perform better, especially in high-noise settings where the improvements are substantial. In the 90% noise setting, we improve by 0.27% in CIFAR-10 and 0.85% in CIFAR-100 experiments, respectively (details in Section 6.4). In particular, Bayesian DivideMix++ outperforms the state-of-the-art methods by large margins in CIFAR-100 experiments, especially in the higher noise settings, where the improvements are more significant (2.23% gain in 80% and 2.45% gain in 90% symmetric noise settings over the best comparison methods). This improvement can be attributed not only to better warm-up and augmentation but also to the higher levels of uncertainty in samples, which is directly related to the noisy labels. Bayesian DivideMix++ has on-par performances with DM + C2D in CIFAR-10 20% noise. The reason could be that the models show higher performances already, making it difficult to surpass using Bayesian DivideMix++. We present the results of the **asymmetric noise** setting on CIFAR-100 in Table 5. We report only on 40% because over 50% makes the classes indistinguishable (Li et al., 2020). Similar to the symmetric cases, the proposed method outperforms the comparison methods by a substantial margin, proving the effectiveness of reducing the effect of memorization and reducing uncertainty during the training process.

Table 5
CIFAR-100 — Asymmetric 40% Noise Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods.

Method		Test accuracy
DM + C2D (WACV, 2022)	Best	75.48±0.16
Zheltonozhskii et al. (2022)	Last	75.06±0.16
Sel-CL+ (CVPR, 2022)	Best	74.20
Li et al. (2022)	Last	–
UNICON (CVPR, 2022)	Best	74.80
Karim et al. (2022)	Last	–
LongReMix (PR, 2023)	Best	59.80±0.1
Cordeiro et al. (2023)	Last	54.90±0.4
Bayesian DivideMix++	Best	**76.52±0.12**
	Last	76.06±0.13

5.3.3. Real noise datasets

We show the results of Real Noise datasets in Table 6 for WebVision and in Table 7 for Clothing1M. Following the literature, we report the top-1 and top-5 accuracy of both WebVision and ILSVRC12 validation sets for WebVision experiments and test accuracy for the Clothing1M experiments. As in CIFAR experiments, we report the mean and standard deviation computed over five runs.

Performance on WebVision: As can be seen in Table 6, we observe a Top-1 accuracy gain of 0.70% compared to the best comparison method (DM + C2D) in the WebVision validation set. However, there is no significant improvement in Top-5 accuracy, which indicates that the model is able to focus on making precise and accurate predictions. However, a lower Top-5 accuracy shows that the model lacks a higher range of possible predictions. Similarly, in the case of the ImageNet validation set, our proposed method is on par or better than the

Table 6
WebVision — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods. Results are reported for both WebVision and ILSVRC12 (ImageNet) validation sets. († - *results reported in DM + C2D (Zheltonozhskii et al., 2022). DivideMix results are reported only on InceptionResNet-V2*).

Validation Set /	WebVision		ILSVRC12	
Method	Top-1	Top-5	Top-1	Top-5
DivideMix (ICLR, 2020) † Li et al. (2020)	76.32±0.36	90.65±0.16	74.42±0.29	91.21±0.12
DM + C2D (WACV, 2022) Zheltonozhskii et al. (2022)	79.42±0.34	92.32±0.33	**78.57±0.37**	93.04±0.10
UNICON (CVPR, 2022) Karim et al. (2022)	77.60	**93.44**	75.29	**93.72**
Bayesian DivideMix++	**80.12±0.28**	92.40±0.30	78.51±0.28	92.67±0.42

Table 7
Clothing1M — Performance comparison (test accuracy (%, mean ± std over five runs)) with SoTA methods.

Method	Test accuracy
DivideMix (ICLR, 2020) Li et al. (2020)	74.38[a]
DM + AugDesc (CVPR, 2021) Nishi et al. (2021)	74.57[a]
DM + C2D (WACV, 2022) Zheltonozhskii et al. (2022)	73.96[a]
ULC (AAAI, 2022) Huang, Bai, et al. (2022)	**74.9±0.2**
LongReMix (PR, 2023) Cordeiro et al. (2023)	74.38
Lipschitz Reg. (PR, 2023) Miao et al. (2023)	74.87
DivideMix++ (Deterministic)	74.09
DivideMix++ (with Dropouts)	74.13
Bayesian DivideMix++	74.81±0.09

[a] Denotes results acquired by us using published source code.

Table 8
Memorization statistics (average and standard deviation) in CIFAR-100 90% symmetric noise.

Method	Avg.	Std.
DivideMix++ (Deterministic)	0.0149	0.0019
Bayesian DivideMix++	**0.0142**	**0.0017**

performance of other models on Top-1 accuracy. However, it does not obtain higher Top-5 accuracy.

Performance on Clothing1M: We reproduce the experiments of Li et al. (2020), Nishi et al. (2021), Zheltonozhskii et al. (2022) and observe an interesting phenomenon. DivideMix++ uses the dataset itself for self-supervised pre-training. However, the results of deterministic DivideMix++ and DivideMix++ with MC-Dropouts indicate that the warm-up gain obtained in other experiments is not present in this case. These results also follow the individual experiments of using only self-supervised pre-training and only augmentation strategies, where the experiment with respect to augmentation was initialized with ImageNet pre-trained weights. This performance could be attributed to the complicated noise structure in Clothing1M as observed by Zheltonozhskii et al. (2022). We, therefore, initialize the subsequent addition of the Bayesian DivideMix++ experiment using ImageNet weights. The proposed method has a test accuracy gain of 0.43% and 0.24% compared with DivideMix (Li et al., 2020) and AugDesc (Nishi et al., 2021) respectively, which highlights the impact of the MC-MixMatch addition to the pipeline. Note that the reproduced state-of-the-art results are a little lower than those reported in the papers, and it could be the difference caused by random components in the training pipeline.

6. Analysis

In this section, we analyse the various experiments to evaluate the effectiveness of our proposed technique in addressing the above-described challenges. We use the model with the best performance of the five runs to do these analyses.

6.1. Robustness towards memorization

We examine the effectiveness of our proposed approach in addressing the challenge of memorization. DNNs, in general, begin to learn the easy samples first (low-loss samples). Similarly, LNL algorithms learn the clean samples first. Different algorithms use different techniques to make the models better learn the clean samples and also not memorize the noisy sample labels. This is an important factor to consider, as it determines the generalization and robustness of the models. We assess the learning behaviour of the models by individually analysing the performance in learning clean and noisy samples. We compare the performance of Bayesian DivideMix++ with that of DivideMix, self-supervised pre-training (DM + C2D) and DivideMix++ (Deterministic). We illustrate this comparison using CIFAR-100 with 90% symmetric noise setting and present the results in Fig. 3. The top row corresponds to the learning of clean samples while the bottom row represents the learning of noisy samples. DivideMix has an initial poor separation of loss terms, requiring more training steps to effectively separate the clean and noisy samples compared to the other methods. The fraction of correctly identified noisy samples, which is considerably low (the first column of Fig. 3), reinforces this behaviour. The introduction of self-supervised pre-training (DM + C2D) leads to better initial loss separation (second column of Fig. 3). With better augmentation strategies provided in DivideMix++ and Bayesian DivideMix++, the learning of clean samples becomes more accurate, as indicated by a higher fraction of correctly identified samples. Concerning learning noisy samples (second row of Fig. 3), DivideMix exhibits a higher level of memorization, where it tends to memorize the labels of the noisy samples. The introduction of self-supervised pre-training (DM + C2D) reduces the degree of memorization. Furthermore, by changing the augmentation scheme (DivideMix++) and by adding MC-MixMatch (Bayesian DivideMix++), we can learn a higher fraction of samples that were initially noisy. This can be seen from Table 8, where with the proposed scheme, the fraction of memorized samples has reduced. Similar to the behaviour of learning the clean samples, the proposed method works better in correctly identifying the noisy samples. The method also shows a reduced impact of memorization, highlighting the efficacy of our proposed scheme.

The results presented above detail the need for models robust against memorization. With the obtained results, we show the benefit of using DivideMix++ during the training of the LNL model and highlight the improvements towards making the models robust towards label noise.

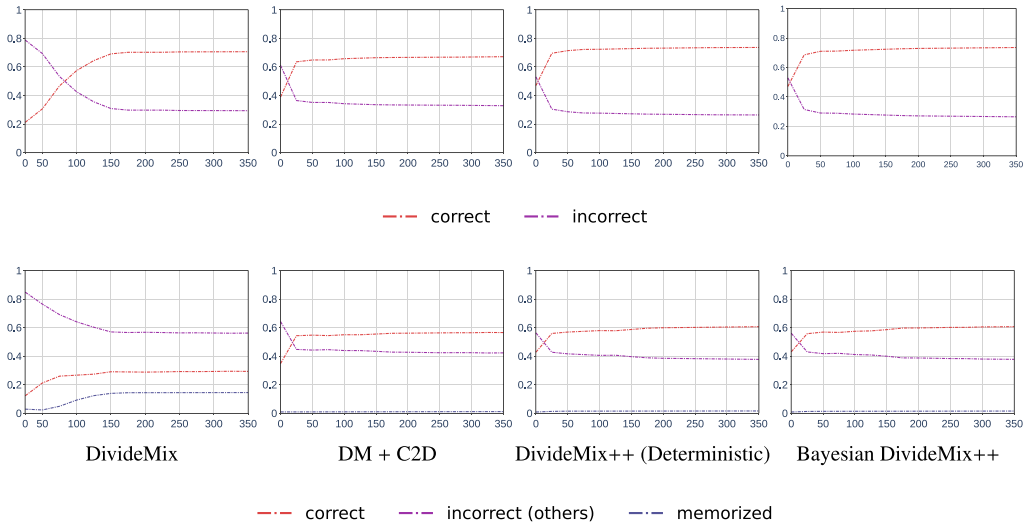


Fig. 3. Robustness towards Memorization. The plots are generated for CIFAR-100 Sym. 90% noise setting. The top row shows the fraction of clean samples that are predicted correctly and incorrectly. The second row shows the fraction of noisy samples that are predicted correctly, memorized and predicted incorrectly. The x-axis corresponds to the epochs and the y-axis corresponds to the fraction of samples.

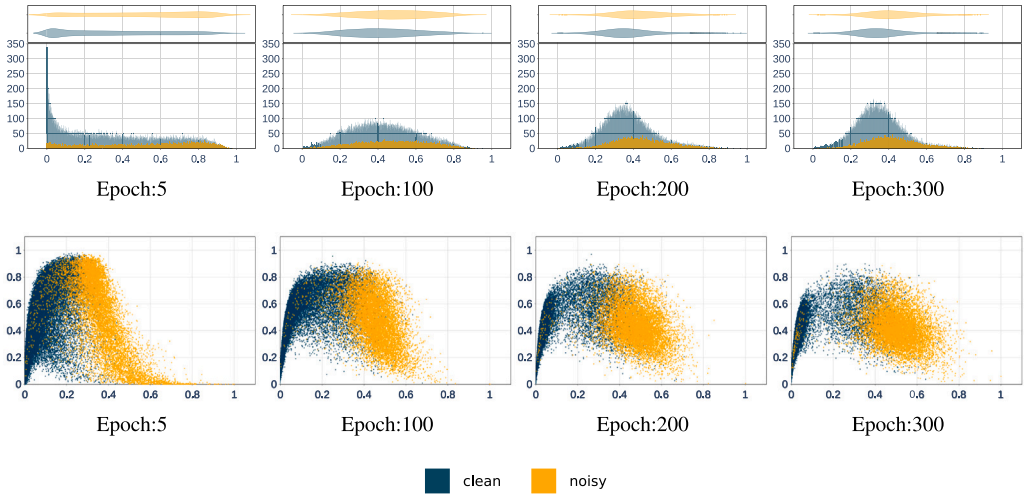


Fig. 4. Uncertainty of clean and noisy samples for CIFAR-100 20% sym. noise. The first row corresponds to the histogram of uncertainty values of clean and noisy samples. The second row corresponds to the uncertainty against the loss values.

6.2. Impact of uncertainty

We next analyse the impact of uncertainty on the training pipeline. We show the histogram of uncertainty values and uncertainty-loss plots over different epochs for CIFAR-100 20% symmetric noise in Fig. 4 and 90% symmetric noise in Fig. 5. In the early training stages (Epoch 5) of 20% symmetric noise setting (Fig. 4), a high number of clean samples have low uncertainty values. However, there is a considerable number of noisy samples that are highly uncertain. This can be visualized with the shape of the violin in the violin plots for Epoch 5. As the training progresses, both clean and noisy samples become less uncertain. Using the uncertainty-loss plots (bottom row of Fig. 4), we can see that in Epoch 5, uncertainty is high for a large number of clean and noisy samples. However, as the training progresses, the uncertainty of both clean and noisy samples is considerably reduced. In the 90% symmetric noise setting (Fig. 5), during the start of the training step (right after warm-up), we observe that most of the clean and noisy samples have high uncertainty. This can be confirmed by the shape of the violin plots. With regards to the relationship between sample loss and sample uncertainty (bottom row of Fig. 5), we can observe that, at epoch 5, uncertainty is generally high for all samples (even for low-loss samples). This can be attributed to the high noise ratio, where more wrongly labelled samples could potentially confuse the model. As the training progresses, the uncertainty of both clean and noisy samples decreases. After epoch 100, the uncertainty of low-loss samples, which are mostly clean, is low, and these are well separated from the noisy samples, highlighting the learning of the models. Although the uncertainty of noisy samples is high compared to that of the clean samples, the overall uncertainty of the models decreases. This can be seen with the shift and concentration of the violin of noisy samples.

Table 9
Uncertainty statistics (maximum and standard deviation) in CIFAR-100 20% and 90% symmetric noise. We measure uncertainty in DivideMix++ by adding MC-Dropouts.

Noise Ratio /	20%		90%	
Method	Max.	Std.	Max.	Std.
DivideMix++	0.9866	0.1448	0.9723	0.1737
Bayesian DivideMix++	**0.9176**	**0.1312**	**0.8930**	**0.1552**

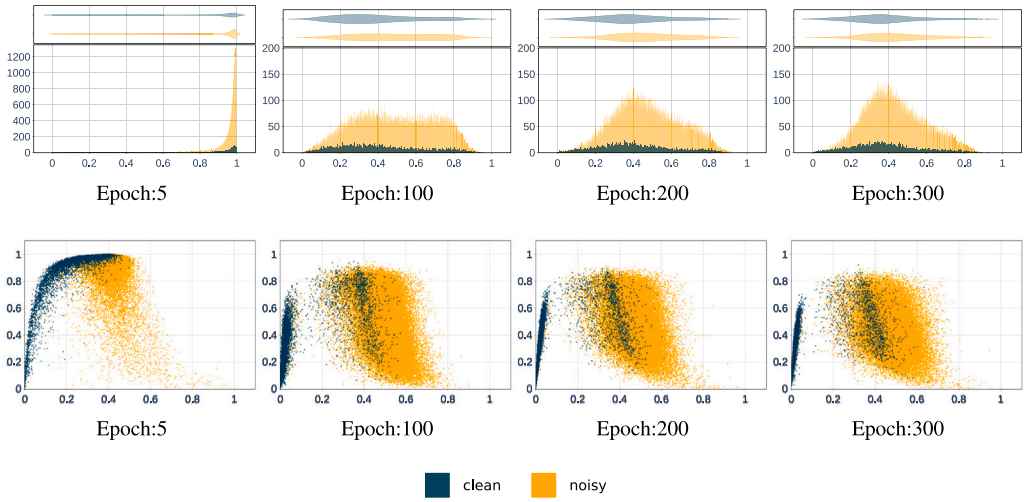


Fig. 5. Uncertainty of clean and noisy samples for CIFAR-100 90% sym. noise. The first row corresponds to the histogram of uncertainty values of clean and noisy samples. Note the change in the *y*-axis for plots other than Epoch:5. The second row corresponds to the uncertainty against the loss values.

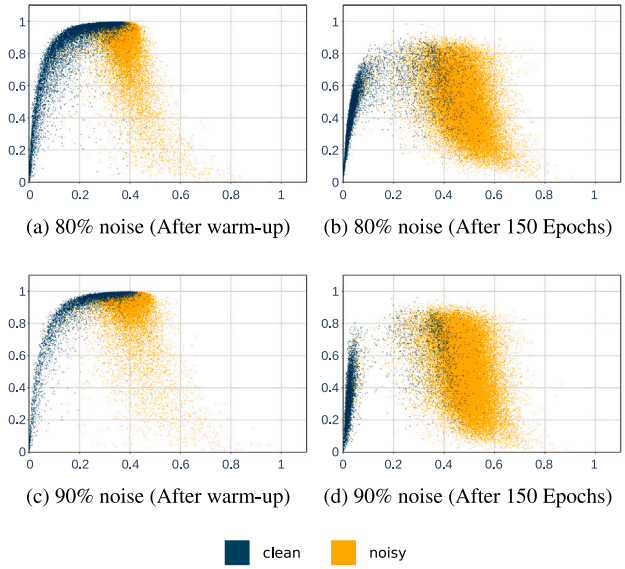


Fig. 6. Usage of MC-MixMatch. Uncertainty vs. loss plots after warm-up and after 150 epochs in high noise settings.

Table 10
Usage of MC-MixMatch. Difference in test accuracy (%) on CIFAR-100 (80% and 90% symmetric noise) using MC-MixMatch right after warm-up vs. after 150 epochs.

Dataset		CIFAR-10		CIFAR-100	
Method / Noise Ratio		80%	90%	80%	90%
After warm-up	Best	95.16	94.49	69.79	60.95
	Last	94.90	94.15	69.17	60.58
After 150 Epochs	Best	**95.35**	**94.64**	**70.24**	**61.56**
	Last	94.94	94.32	69.75	61.19

We report statistical measurements of the computed uncertainty values in Table 9. We highlight the maximum uncertainty values and standard deviation of uncertainty values of the training data. For both noise settings (20% and 90%), we see a decrease in the maximum value for Bayesian DivideMix++, highlighting a reduction in the uncertainty of the training samples. The lower standard deviation of the experiments highlights a reduction in the overall spread of uncertainty values.

Uncertainty in high noise settings. We observe that in high-noise settings, the uncertainty of both clean and noisy samples is very high. This is due to the high level of noise making it difficult for the models to learn from the clean samples. With this understanding, we study the impact of using uncertainty during the entire training steps against the impact when using uncertainty after allowing the models to learn further epochs without using MC-MixMatch (a settling period). We report the findings of this study on both CIFAR datasets in Table 10. In all cases (80% and 90% noise settings), the performance of models was higher when using MC-MixMatch after allowing the models to stabilize. We show the uncertainty-loss plots for this experiment in Fig. 6 and observe that using the uncertainty values after models learn better from clean samples is more beneficial than using it during the entirety of the training (better separation of clean and noisy samples in both noise settings as shown in Fig. 6(b) and Fig. 6(d)).

Making models aware of the label uncertainty makes them better equipped to handle the label noise. This can be seen through the different visualizations across different noise settings. MC-MixMatch helped the models to learn from samples that are more certain, thereby increasing the overall model confidence.

6.3. Loss separation

We also study the effectiveness of losses in separating the clean and the noisy samples. We compare the loss separation of DivideMix with our proposed method in Fig. 7 using CIFAR-100 90% symmetric noise. During the initial stages of training, both methods exhibit a limited ability to separate clean and noisy samples (Epoch 30 and Epoch 5 - first column of Fig. 7 are the first epochs after the warm-up phase in DivideMix and our proposed method respectively). Specifically, Epoch 5 of Bayesian DivideMix++ has a very high number of noisy samples which have large losses. As the training progresses, our proposed method exhibits a better separation of samples compared to that of DivideMix (bottom row of Fig. 7). The violin plots above the loss histograms depict the distribution of the loss values. A higher density of samples outside the violin indicates the inability of DivideMix to completely learn from the clean samples. In contrast, our proposed method shows a clearer distinction of samples, as evidenced by the majority of samples being well-separated. The presence of a smaller mode of clean samples overlapping with higher loss values might be due to the higher noise ratio. This highlights the improved capability of our proposed method to effectively separate clean and noisy samples, leading to enhanced performance in learning with noisy labels.

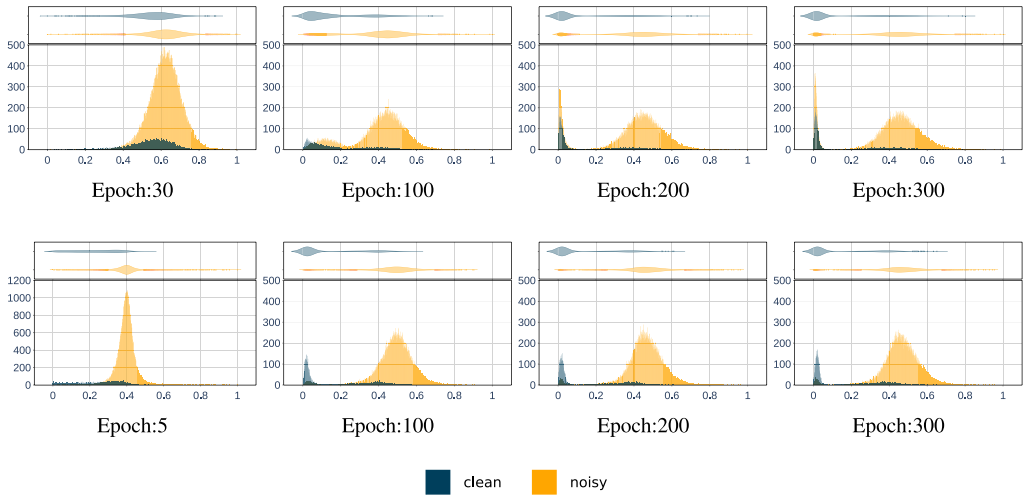


Fig. 7. Better loss separation. Progression of loss with respect to clean and noisy samples. As the training progresses, the loss is better separated. The first row corresponds to DivideMix, while the second row corresponds to Bayesian DivideMix++. DivideMix has a warm-up of 30 epochs (Hence, the first plot computed at 30 and not at 5). Also, note the change in the y-axis for Epoch:5 of Bayesian DivideMix++. All plots are generated for CIFAR-100 90% sym. noise.

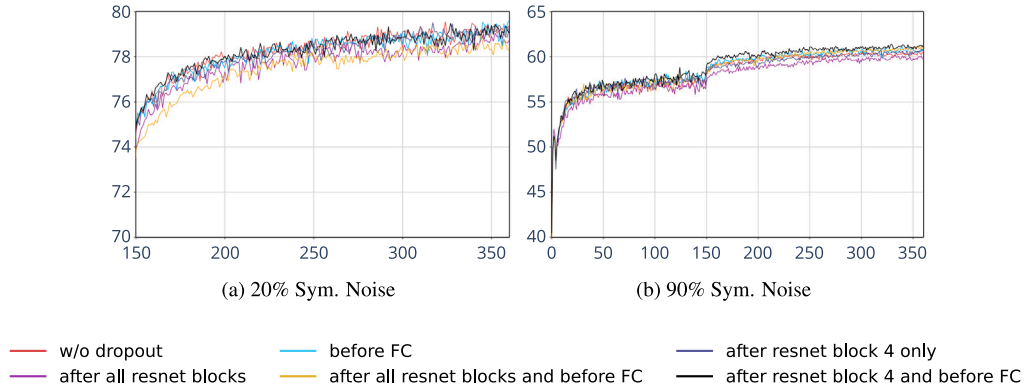


Fig. 8. Dropout position. Different configurations to select a better dropout position.

6.4. Ablation study

In this section, we provide an overview of the design decisions in our proposed method. Subsequently, we present a comprehensive analysis of the experiments conducted to evaluate the significance of each component in our proposed method. For all experiments, we use CIFAR-100 with 20% and 90% symmetric noise settings and report the test accuracy.

6.4.1. Exploring the position of MC-dropouts

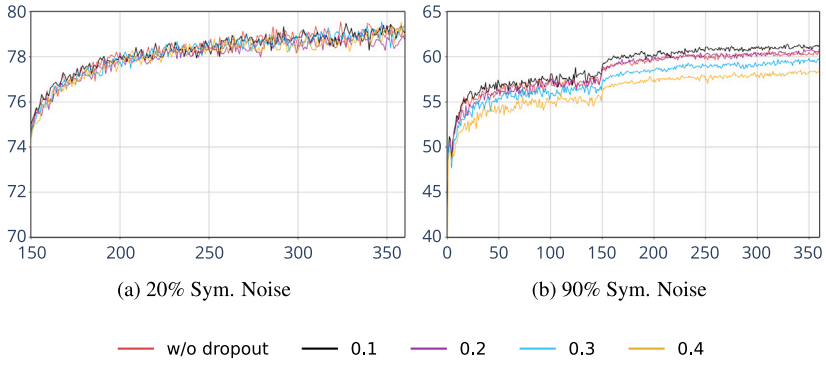
We investigate the influence of using dropout layers at different stages within the model architecture and report the performances in Table 11. For this experiment, we fixed the dropout ratio as 0.1. We placed the dropout layers at different places of the ResNet block (a ResNet-18 model has 4 ResNet blocks). Using the dropout layer after the 4th block fared better than using it after all blocks. This can be due to losing more generic features during the training. We also check the behaviour by applying dropout at the later stages of the model. The optimal selection is the combination of using it after the 4th block and before the Fully Connected (FC) layer, which could be due to not losing much generic information and better regularization during learning specific features. The difference in performance is substantial in the higher noise cases, whereas the difference is comparative in the lower noise cases. We show the test accuracy of different settings against the training epochs in Fig. 8. The difference between different dropout positions is very evident in the 90% symmetric noise settings (Fig. 8(b)) as compared to the 20% symmetric noise settings (Fig. 8(a)) highlighting the important role of dropouts in higher noise settings.

Table 11
Position of MC-Dropout. Difference in test accuracy (%) on CIFAR-100 (20% and 90% symmetric noise) with different combination of dropout layers. The dropout ratio was set to 0.1 in all cases.

Position of MC-Dropout		20%	90%
After all blocks	Best	79.05	60.15
	Last	78.77	59.86
Only after block 4	Best	**79.51**	60.85
	Last	79.13	60.61
Only before FC	Best	79.48	61.09
	Last	78.89	60.62
After all blocks & before FC	Best	79.00	61.18
	Last	78.44	60.99
After block 4 & before FC	Best	79.48	**61.39**
	Last	79.16	61.13

6.4.2. Exploring the dropout ratio

We study the effect of using different dropout ratios (p_b) in Table 12. We fixed the position of the dropout layers for this experiment. For this experiment, we use one dropout layer after the 4th ResNet block and one before the Fully Connected layer of the ResNet-18 model (Fig. 9). We vary the dropout ratio between 0.1 and 0.4 on both low-noise and high-noise settings. As with the experiments dealing with the position



(a) 20% Sym. Noise (b) 90% Sym. Noise

Fig. 9. Dropout ratio. Different configurations to select a better dropout ratio.

Table 12
Dropout Ratio. Difference in test accuracy (%) on CIFAR-100 (20% and 90% symmetric noise) when using different dropout ratios. Dropouts are placed after block 4 and before FC of the ResNet-18 model.

Dropout ratio		20%	90%
0.1	Best	79.48	**61.39**
	Last	79.16	61.13
0.2	Best	79.18	60.90
	Last	78.71	60.65
0.3	Best	**79.54**	59.82
	Last	79.02	59.54
0.4	Best	79.53	58.48
	Last	79.08	58.31

Table 13
Study of optimal λ_u on CIFAR-100 symmetric 20% noise.

λ_u	0	25	50	100	250
Best	**80.06**	80.01	79.81	79.02	75.99
Last	79.30	79.44	79.30	78.49	75.46

Table 14
Study of optimal λ_u on CIFAR-100 symmetric 90% noise.

λ_u	0	10	50	100	250	500	750	1000
Best	55.54	55.64	57.01	58.78	60.11	**61.56**	59.83	59.42
Last	54.71	55.18	56.47	58.36	59.70	61.19	59.23	59.04

Table 15
Ablations. Difference in test accuracy (%) on CIFAR-100 (20% and 90% symmetric noise) with different components.

Method		20%	90%
DivideMix	Best	77.30	31.50
	Last	76.90	31.00
DM + C2D	Best	78.69±0.17	58.70±0.31
	Last	78.32±0.35	58.45±0.30
DivideMix++ (Deterministic)	Best	79.55	60.71
	Last	79.19	60.36
DivideMix++ (with) MC-Dropouts	Best	79.48	61.39
	Last	79.16	61.13
Bayesian DivideMix++ (After warm-up)	Best	**80.01**	60.95
	Last	79.44	60.58
Bayesian DivideMix++ (After 150 Epochs)	Best	79.51	**61.56**
	Last	79.13	61.19

of dropout layers, the effect of using different dropout ratios is more pronounced in high-noise settings. We select the dropout ratio based on the high-noise settings, where we achieve higher performances with a smaller dropout of 0.1. Higher dropout ratios have a negative impact on the model performance, which can be due to more neurons randomly getting dropped during training, losing more meaningful information. Similar to the position of dropouts, it can be seen that the difference between different dropout ratios is more prominent in the higher noise settings (Fig. 9(b) as compared with Fig. 9(a)).

6.4.3. Hyperparameter sensitivity

Based on the previous works that were inspired by DivideMix algorithm (Li et al., 2020; Nishi et al., 2021; Zheltonozhskii et al., 2022), we identify the unsupervised loss weight (λ_u in Eq. (4)) as the most important hyperparameter. We follow the studies of Nagarajan et al. (2022), Zheltonozhskii et al. (2022) and investigate the performance of Bayesian DivideMix++ by varying λ_u on CIFAR-100 symmetric 20% and 90% noise settings. The sensitivity study is reported in Tables 13 and 14. For low noise settings, we find that lower λ_u values are better, whereas for higher noise settings, we require a higher λ_u value. Although a λ_u of 0 performs slightly better than the value of 25, the last accuracy is better in the later experiment, showing better overall training. We, therefore, set the λ_u to be 25, which also follows the results obtained by Zheltonozhskii et al. (2022). All the results thus provided in this work are obtained using the hyperparameter settings of Zheltonozhskii et al. (2022).

6.4.4. Evaluation of different components

We perform detailed ablations to show the importance of each component and report the results in Table 15. We highlight the insights of each component as follows:

- DivideMix++ (Deterministic) exhibits substantial performance gains over the DivideMix algorithm (*Row 1*) and the one by using Self-supervised pre-training (*Row 2*). While the impact of self-supervised pre-training is noticeable in the 90% noise setting, it falls short of achieving complete robustness against memorization. However, combining effective augmentation strategies, DivideMix++ (*Row 3*) achieves greater resilience to memorization effects, resulting in higher performance gains.
- By converting DivideMix++ to a Bayesian model using MC-Dropouts, we achieve comparable results to its deterministic counterparts (*Row 4*). The conversion of DivideMix++ to a Bayesian model allows us to harness the benefits of uncertainty estimation without compromising the overall performance achieved by the deterministic counterparts.
- MC-MixMatch enhances the performance of DivideMix++ by leveraging uncertainty measures to assign importance to individual samples (*Row 5 and 6*). By considering the uncertainty estimates, MC-MixMatch effectively prioritizes samples during training, leading to improved performance. In higher noise settings, where the initial uncertainty is not figurative of the actual data, a waiting period is needed to provide good uncertainty measurements.
- The combination of DivideMix++ and MC-MixMatch (Bayesian DivideMix++) has yielded significant benefits for the DivideMix algorithm, highlighting the crucial role of addressing both DNN memorization effects and uncertainty during the learning process.

Table 16
Computation time (seconds) per-epoch for each operation of Bayesian DivideMix++ and DivideMix.

Method	Model initialization	Warmup	AugDesc	MixMatch
DivideMix	11.51 s	74.05 s	1.15 s	8.25 s
Bayesian DivideMix++	11.45 s	74.87 s	0.66 s	26.27 s

6.4.5. Training time analysis

We analyse the training time of our proposed method, Bayesian DivideMix++ and compare it against the baseline DivideMix algorithm (Li et al., 2020). Bayesian DivideMix++ uses the same training pipeline as that of DivideMix (Li et al., 2020) except for the proposed contributions. We show the breakdown of computation time for each operation in Table 16. We integrate self-supervised pre-trained weights and change the augmentation pipeline, which does not contribute to a difference in the training time. However, Monte-Carlo MixMatch is the only component that affects the training time. The computation time of Monte-Carlo MixMatch depends on the number of iterations used to compute uncertainty. In this case, we calculate the uncertainty over 10 iterations, which leads to the increase in training time (8.25 s to 26.27 s). This increase is a tradeoff between having more meaningful data and higher computation costs associated with computing uncertainty. The focus of our paper is to show the importance of handling label uncertainty in LNL algorithms. More efficient alternate methods for measuring uncertainty can alleviate the computation time. However, this is beyond the focus of our paper. Additionally, it has to be noted that there is no difference in the inference time of Bayesian DivideMix++ as the proposed components have no noticeable impact on it.

6.5. Limitations

The combination of DivideMix++ and MC-MixMatch in our proposed Bayesian DivideMix++ has produced promising results. However, we identified some limitations in our proposal that can serve as a potential research direction.

- **Computational Complexity:** DivideMix++ follows a complex modelling technique by incorporating components such as the co-teaching of two DNNs, generating pseudo-labels, selecting clean samples and MixMatch of labelled and unlabelled data. This addition of components makes the training more expensive, particularly when dealing with large-scale datasets. Uncertainty estimation adds to the computational complexity of the algorithm, as it needs multiple inference steps to compute uncertainty.
- **Hyperparameter Selection:** Although this paper and several others before us have attempted to experiment with different hyperparameters of the DivideMix pipeline, training still heavily relies on multiple hyperparameters, making the selection of appropriate values a challenging task.
- **Dropout Rate and Position:** MC-Dropout is a popular technique for estimating uncertainty. However, the choice of dropout rate and the placement of dropout positions is critical when using MC-Dropouts. It requires careful tuning, especially considering the noise settings.
- **Noise Level Estimation:** The training pipeline highly relies on estimating the noise level in the dataset. Estimating noise levels accurately in real-world scenarios is very challenging. Inaccurate estimation can lead to improper sample selection, thereby affecting the algorithm performance.

6.6. Broader impact

LNL has been studied extensively over the years due to the challenges associated with annotating training data accurately. The difficulties in obtaining perfect training data have highlighted the need for robust LNL algorithms. With label noise, the DNNs could end up learning incorrect patterns and subsequently make poor predictions on unseen data. An efficient LNL algorithm is crucial to effectively handle and mitigate the effect of noisy labels during the training phase. LNL algorithms increase the robustness and generalization ability of the models in practical settings, resulting in more accurate predictions. Overall, LNL enables leveraging huge, readily available, but imperfectly labelled data in training models that can excel in real-world applications, thereby improving the performance and reliability of models.

7. Conclusions and future works

Learning with noisy labels has been a longstanding and actively researched problem. The need for robust algorithms is of prime importance as noisy labels pose significant challenges in training accurate and reliable models. Two critical challenges in developing robust LNL algorithms are the DNN memorization of noisy labels and label uncertainty. Both are of prime importance to enhance the generalization of models in the presence of label noise. With this regard, we proposed Bayesian DivideMix++ addressing the two issues: DivideMix++ provided a better warm-up phase and improved the DA pipeline. MC-MixMatch, which incorporated MC-Dropouts, was used to reduce the impact of uncertain samples in the training step. The combination of both components gave substantial performance improvements and provided benefits in addressing the DNN memorization and uncertainty effects. We validated our proposed pipeline on popular benchmarks of different noise settings and achieved state-of-the-art performance.

Future works. We measured uncertainty during the MixMatch step of Bayesian DivideMix++ and used it as a weight to combine samples. Currently, the co-divide step is based on the inference of previous epochs, and uncertainty is present in the model predictions. This uncertainty could be measured and used in the subsequent steps. Our future work could thus focus on investigating the impact of uncertainty during the loss-separation step. Another future work is exploring the benefits of using class-conditional learning as an alternative to the current global noise learning strategy.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00), IDEATE (AEI-MICINN, PID2022-141566NB-I00), A-BMC (AEI-MICINN, CNS2022-135480), CERCA Programme/Generalitat de Catalunya, and Agencia Nacional de Investigación y Desarrollo de Chile (ANID) (Grant No. FONDECYT INICIACIÓN 11230262). Ricardo Marques acknowledges the support of the Serra Húnter Programme. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. The authors thankfully acknowledge the computer resources at FinisTerrae III and the technical support provided by the Galician Supercomputing Center (CESGA) (RES-IM-2023-2-0025).

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, *76*, 243–297.

Allen-Zhu, Z., Li, Y., & Liang, Y. (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in Neural Information Processing Systems*, *32*.

Angluin, D., & Laird, P. (1988). Learning from noisy examples. *Machine Learning*, *2*, 343–370.

Arazo, E., Ortego, D., Albert, P., O'Connor, N., & McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International conference on machine learning* (pp. 312–321). PMLR.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., et al. (2017). A closer look at memorization in deep networks. In *International conference on machine learning* (pp. 233–242). PMLR.

Bahri, D., Jiang, H., & Gupta, M. (2020). Deep k-nn for noisy labels. In *International conference on machine learning* (pp. 540–550). PMLR.

Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., et al. (2021). Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, *34*, 24392–24403.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, *32*.

Boluki, S., Ardywibowo, R., Dadaneh, S. Z., Zhou, M., & Qian, X. (2020). Learnable Bernoulli dropout for Bayesian deep learning. In *International conference on artificial intelligence and statistics* (pp. 3905–3916). PMLR.

Cai, K., Zhang, H., Pedrycz, W., & Miao, D. (2023). SSS-Net: A shadowed-sets-based semi-supervised sample selection network for classification on noise labeled images. *Knowledge-Based Systems*, Article 110732.

Chen, Y., Hu, S. X., Shen, X., Ai, C., & Suykens, J. A. (2022). Compressing features for learning with noisy labels. *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607). PMLR.

Cordeiro, F. R., Sachdeva, R., Belagiannis, V., Reid, I., & Carneiro, G. (2023). Longremix: Robust learning with high confidence samples in a noisy label environment. *Pattern Recognition*, *133*, Article 109013.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 113–123).

Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 702–703).

Der Kiureghian, A., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112.

Ding, Y., Wang, L., Fan, D., & Gong, B. (2018). A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE winter conference on applications of computer vision* (pp. 1215–1224). IEEE.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning* (pp. 1050–1059). PMLR.

Ghosh, A., Kumar, H., & Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1*.

Ghosh, A., & Lan, A. (2021). Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2703–2708).

Goel, P., & Chen, L. (2021). On the robustness of monte carlo dropout trained with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2219–2228).

Goldberger, J., & Ben-Reuven, E. (2017). Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., et al. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, *31*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, proceedings, Part IV 14* (pp. 630–645). Springer.

Huang, Y., Bai, B., Zhao, S., Bai, K., & Wang, F. (2022). Uncertainty-aware learning against label noise on imbalanced datasets. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 6* (pp. 6960–6969).

Huang, B., Lin, Y., & Xu, C. (2022). Contrastive label correction for noisy label learning. *Information Sciences*, *611*, 173–184.

Huang, J., Qu, L., Jia, R., & Zhao, B. (2019). O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3326–3334).

Iscen, A., Valmadre, J., Arnab, A., & Schmid, C. (2022). Learning with neighbor consistency for noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4672–4681).

Ji, D., Oh, D., Hyun, Y., Kwon, O. M., & Park, M. J. (2021). How to handle noisy labels for robust learning from uncertainty. *Neural Networks*, *143*, 209–217.

Jiang, L., Zhou, Z., Leung, T., Li, L. J., & Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning* (pp. 2304–2313). PMLR.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., & Bennamoun, M. (2022). Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48.

Karim, N., Rizve, M. N., Rahnavard, N., Mian, A., & Shah, M. (2022). Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9676–9686).

Kim, D., Ryoo, K., Cho, H., & Kim, S. (2022). SplitNet: Learnable clean-noisy label splitting for learning with noisy labels. arXiv preprint arXiv:2211.11753.

Köhler, J. M., Autenrieth, M., & Beluch, W. H. (2019). Uncertainty based detection and relabeling of noisy image labels. In *CVPR workshops* (pp. 33–37).

Krizhevsky, A., Hinton, G., et al. (2009). *Learning multiple layers of features from tiny images*. ON, Canada: Toronto.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., et al. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, *128*(7), 1956–1981.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, *30*.

Lange, R. D., Benjamin, A. S., Haefner, R. M., & Pitkow, X. (2022). Interpolating between sampling and variational inference with infinite stochastic mixtures. In *Uncertainty in artificial intelligence* (pp. 1063–1073). PMLR.

Li, J., Socher, R., & Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394.

Li, W., Wang, L., Li, W., Agustsson, E., & Van Gool, L. (2017). Webvision database: Visual learning and understanding from web data. arXiv preprint arXiv:1708.02862.

Li, J., Wong, Y., Zhao, Q., & Kankanhalli, M. S. (2019). Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5051–5059).

Li, S., Xia, X., Ge, S., & Liu, T. (2022). Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 316–325).

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., & Li, L. J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1910–1918).

Liao, Y. H., Kar, A., & Fidler, S. (2021). Towards good practices for efficiently annotating large-scale image classification datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4350–4359).

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, *37*(1), 145–151.

Liu, S., Niles-Weed, J., Razavian, N., & Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, *33*, 20331–20342.

Liu, S., Zhu, Z., Qu, Q., & You, C. (2022). Robust training under label noise by over-parameterization. In *International conference on machine learning* (pp. 14153–14172). PMLR.

Lu, Y., & Selc, W. H. (2022). Self-ensemble label correction improves learning with noisy labels, 3. arXiv preprint arXiv:2205.01156.

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., & Bailey, J. (2020). Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning* (pp. 6543–6553). PMLR.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., & Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, *32*.

Malach, E., & Shalev-Shwartz, S. (2017). Decoupling "when to update" from "how to update". *Advances in Neural Information Processing Systems*, *30*.

Menon, A. K., Rawat, A. S., Reddi, S. J., & Kumar, S. (2020). Can gradient clipping mitigate label noise? In *International conference on learning representations*.

Miao, Q., Wu, X., Xu, C., Zuo, W., & Meng, Z. (2023). On better detecting and leveraging noisy samples for learning with severe label noise. *Pattern Recognition*, *136*, Article 109210.

Nagarajan, B., Marques, R., Mejia, M., & Radeva, P. (2022). Class-conditional importance weighting for deep learning with noisy labels.. In *VISIGRAPP* (pp. 679–686).

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., & Brox, T. (2019). Self: Learning to filter noisy labels with self-ensembling. arXiv preprint arXiv:1910.01842.

Nishi, K., Ding, Y., Rich, A., & Hollerer, T. (2021). Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8022–8031).

Northcutt, C., Jiang, L., & Chuang, I. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, *70*, 1373–1411.

Oh, D., Lee, D., Byun, J., & Shin, B. (2022). Improving group robustness under noisy labels using predictive uncertainty. arXiv preprint arXiv:2212.07026.

Ortego, D., Arazo, E., Albert, P., O'Connor, N. E., & McGuinness, K. (2021). Multi-objective interpolation training for robustness to label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6606–6615).

Oyen, D., Kucer, M., Hengartner, N., & Singh, H. S. (2022). Robustness to label noise depends on the shape of the noise distribution in feature space. arXiv preprint arXiv:2206.01106.

Pan, C., Yuan, B., Zhou, W., & Yao, X. (2022). Towards robust uncertainty estimation in the presence of noisy labels. In *Artificial neural networks and machine learning–ICANN 2022: 31st international conference on artificial neural networks, Bristol, UK, September 6–9, 2022, proceedings, Part I* (pp. 673–684). Springer.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., & Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1944–1952).

Patrini, G., Rozza, A., Menon, A., Nock, R., & Qu, L. (2016). Making neural networks robust to label noise: a loss correction approach. *Stat*, *1050*, 13.

Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2018). Learning to reweight examples for robust deep learning. In *International conference on machine learning* (pp. 4334–4343). PMLR.

Ridnik, T., Ben-Baruch, E., Noy, A., & Zelnik-Manor, L. (2021). Imagenet-21k pretraining for the masses. arXiv preprint arXiv:2104.10972.

Rizve, M. N., Duarte, K., Rawat, Y. S., & Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. arXiv preprint arXiv:2101.06329.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

Schmarje, L., Santarossa, M., Schröder, S. M., & Koch, R. (2021). A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access*, *9*, 82146–82168.

Song, H., Kim, M., Park, D., Shin, Y., & Lee, J. G. (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Sun, Z., Shen, F., Huang, D., Wang, Q., Shu, X., Yao, Y., et al. (2022). Pnp: Robust learning from noisy labels by probabilistic noise prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5311–5320).

Tan, C., Xia, J., Wu, L., & Li, S. Z. (2021). Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM international conference on multimedia* (pp. 1405–1413).

Tanaka, D., Ikami, D., Yamasaki, T., & Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5552–5560).

Tarvainen, A., & Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, *30*.

Tatjer, A., Nagarajan, B., Marques, R., & Radeva, P. (2023). CCLM: Class-conditional label noise modelling. In *Iberian conference on pattern recognition and image analysis* (pp. 3–14). Springer.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., & Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 322–330).

Wang, Y., Sun, X., & Fu, Y. (2022). Scalable penalized regression for noise detection in learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 346–355).

Wang, H., Xiao, R., Dong, Y., Feng, L., & Zhao, J. (2022). ProMix: Combating label noise via maximizing clean sample utility. arXiv preprint arXiv:2207.10276.

Wang, H., & Yeung, D. Y. (2020). A survey on Bayesian deep learning. *ACM Computing Surveys (CSUR)*, *53*(5), 1–37.

Wei, H., Feng, L., Chen, X., & An, B. (2020). Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13726–13735).

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., & Liu, Y. (2021). Learning with noisy labels revisited: A study using real-world human annotations. arXiv preprint arXiv:2110.12088.

Wu, M., Li, Q., Yang, F., Zhang, J., Sheng, V. S., & Hou, J. (2023). Learning from biased crowdsourced labeling with deep clustering. *Expert Systems with Applications*, *211*, Article 118608.

Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., et al. (2021). Class2simi: A noise reduction perspective on learning with noisy labels. In *International conference on machine learning* (pp. 11285–11295). PMLR.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., et al. (2021a). Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*.

Xia, X., Liu, T., Han, B., Gong, M., Yu, J., Niu, G., et al. (2021b). Sample selection with uncertainty of losses for learning with noisy labels. arXiv preprint arXiv:2106.00445.

Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2691–2699).

Yao, Y., Liu, T., Han, B., Gong, M., Deng, J., Niu, G., et al. (2020). Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in Neural Information Processing Systems*, *33*, 7260–7271.

Yao, Y., Sun, Z., Zhang, C., Shen, F., Wu, Q., Zhang, J., et al. (2021). Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5192–5201).

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., & Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International conference on machine learning* (pp. 7164–7173). PMLR.

Yu, X., Jiang, Y., Shi, T., Feng, Z., Wang, Y., Song, M., et al. (2023). How to prevent the continuous damage of noises to model training? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12054–12063).

Zhai, X., Kolesnikov, A., Houlsby, N., & Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12104–12113).

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), 107–115.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

Zhang, Y., Niu, G., & Sugiyama, M. (2021). Learning noise transition matrix from only noisy labels via total variation regularization. In *International conference on machine learning* (pp. 12501–12512). PMLR.

Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M., & Litany, O. (2022). Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1657–1667).

CHAPTER 3

Self-Supervised Learning

Introduction

Acquiring large volumes of labelled training data is a significant bottleneck in many DL applications. This data is expensive to collect, time-consuming to label, and often limited in availability. Self-supervised Learning is a powerful alternative learning paradigm to learn meaningful representations directly from massive raw data without labelled supervision. Contrastive learning, a prominent SSL approach works on pulling together different views of the sample closer to each other in the latent representation space. Neighbour contrastive learning, a sub-category of contrastive SSL focuses on contrasting different views of a sample with its nearest neighbours and bringing the representations close in the representation space.

Despite the advancements in contrastive SSL, several research questions remain open for exploration. The following questions drive our research on this topic (Numbers in brackets represent the corresponding paper):

- *How can we design effective contrastive pretext tasks? (#1, #2)*
- *How to effectively use neighbours in the Neighbour contrastive learning methods? (#2, #3)*
- *Are all selected neighbours "good" and informative? (#3)*

Our research investigates these critical aspects to enhance the capabilities of contrastive learning for SSL tasks.

Works Published Within This Chapter

1. *Nil Ballús, **Bhalaji Nagarajan**, and Petia Radeva. "Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition." IbPRIA, 2022.*
2. *Imanol G. Estepa, Ignacio Sarasúa, **Bhalaji Nagarajan**, and Petia Radeva. "All4One: Symbiotic Neighbour Contrastive Learning via Self-Attention and Redundancy Reduction." ICCV, 2023. **(GGS Rating: A++).***
3. *Imanol G. Estepa, Jesús M. Rodríguez-de-Vera, **Bhalaji Nagarajan**, and Petia Radeva. "Good Fences Makes Good Neighbours." ICCV-W, 2023.*

Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition

Nil Ballús[1(✉)], Bhalaji Nagarajan[1], and Petia Radeva[1,2]

[1] Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona, Spain
nballuri7@alumnes.ub.edu, {bhalaji.nagarajan,petia.ivanova}@ub.edu
[2] Computer Vision Center, Cerdanyola, Barcelona, Spain

Abstract. Self-supervised Learning has been showing upbeat performance in several computer vision tasks. The popular contrastive methods make use of a Siamese architecture with different loss functions. In this work, we go deeper into two very recent state of the art frameworks, namely, SimSiam and Barlow Twins. Inspired by them, we propose a new self-supervised learning method we call Opt-SSL that combines both image and feature contrasting. We validate the proposed method on the food recognition task, showing that our proposed framework enables the self-learning networks to learn better visual representations.

Keywords: Self-supervised · Contrastive learning · Food recognition

1 Introduction

The advent of Deep Neural Networks (DNNs) has increased the performance of several challenging computer vision problems [24]. The success of this multi-fold advancements can be attributed to large amounts of annotated data [20]. However, building large labelled datasets has several challenges [21]. The labelling process is tedious, costly and time consuming. Moreover, it is not possible to create datasets for every possible scenario, and the difficulty in generalization is one of the main bottlenecks in these supervised models [23]. One of the recent possible alternatives to this approach is the Self-Supervised Learning (SSL). SSL focuses on models that adapt to learn different underlying representations of the data without use of supervised datasets [19]. SSL frameworks use a pretext learning task which requires only unlabeled data. This task enables the DNNs to construct data representations that are semantically meaningful and are later used to solve any downstream task such as image recognition [17].

Food Recognition is a fast growing, but challenging area of study [11]. However, food recognition is a very complex computer vision task, due to the nature of images [22]. Food images are very hard to study as they are often mixed or composed of many items in a single plate. The high intra-class and low inter-class variance makes it difficult to work for any food recognition model. The images

P. Radeva—IAPR Fellow.

© Springer Nature Switzerland AG 2022
A. J. Pinho et al. (Eds.): IbPRIA 2022, LNCS 13256, pp. 655–666, 2022.
https://doi.org/10.1007/978-3-031-04881-4_52

are very diverse in visual appearances and the complexity of background scenes add additional obstacles to train the models [27]. It is therefore very important for the DNNs to learn the underlying information of food images and not just fit with the given label information. Hence, the success of SSL framework has given a new way on the food recognition research direction.

The key challenge in any SSL framework is to learn effective image representation [31]. Siamese architectures [1] offer a very efficient architectures for learning data representations. In the Contrastive Learning algorithms, the models learn to reduce the distance between representations of the same image by using different augmentations of the same image. These pairs of images are known as 'positive pairs'. The earlier studies incorporated loss terms to increase the distance between negative pairs. However, it hugely increased the complexity of the learning algorithms. Recent studies have shown to achieve state of the art performance using only the positive pairs [13]. SimSiam [8] and Barlow Twins [33] are contrastive algorithms that were able to learn better visual representations and achieved higher performance on the downstream tasks. SimSiam works on learning representations specific to the pair of positive pairs whereas Barlow Twins uses features across different sets of images to learn the visual representation. Inspired in the SimSiam and Barlow Twins, we propose a new method called Opt-SSL, which modifies the contrastive learning loss function incorporating both image and feature comparison. Our main contributions are:

- First, we propose a new SSL which loss function incorporates cross-correlation matrices in the dimension of features along with the dimension of batches.
- Second, we do an extensive analysis of the new loss function applied to food recognition and show that our proposed method increases the performance of the baseline being at the same time computationally effective.

The rest of the paper is organized as follows. We discuss the related work in Sect. 2. We provide the rationale behind the proposed technique in Sect. 3 and explain the proposed method in Sect. 4. The results used to validate the proposed method are explained in Sect. 5 followed by conclusions at the end.

2 Related Work

SSL frameworks can be categorized into three classes of algorithms based on the objective of the SSL methods - Generative [18], Contrastive [33] and Generative-contrastive methods [23]. Generative SSL methods define pretext tasks that aim to reconstruct parts of the original input while learning effective latent representation for the data [18]. Contrastive SSL methods use contrastive pretext tasks that compare the similarity of inputs' representations and aim to get similar latent representations for similar inputs and dissimilar latent representations for non-similar inputs [33]. Generative-Contrastive methods define pretext tasks to generate fake samples from the input data and the objective is to distinguish them from real samples. These last methods arose from the idea of generative SSL, but instead of relying on sample reconstruction, they learn to reconstruct

the original data distribution by minimizing the distributional divergence [12]. A detailed summary of the algorithms are presented in [23].

The fundamental principle behind the SSL methods relies on the cross-view prediction framework [3], where different representations of the same image are used to learn the representations. Contrastive learning algorithms focuses on learning to compare between different representations through a Noise Contrastive Estimation (NCE) [14] objective function, where the distance between different augmentations of the same image are brought close compared to the negative samples. One of the earlier approaches in Contrastive methods was to use clustering algorithms [5]. Deep Cluster [5] used clustering to yield pseudo-labels and used a discriminator to predict the labels of the images. Local aggregation [34] used an objective function that directly optimized a local soft-clustering metric instead of the cross entropy loss used in Deep Cluster. VQ-VAE [28] substituted each 1-dimensional vector in the matrix to the nearest one in an embedding dictionary. SwAV [6] used online clustering under balanced partition constraint to compute the assignment from one view and predicted it from another view. The main disadvantage of the cluster based approach is the two-stage training process [23], which gave rise to instance discrimination-based methods.

Contrastive Multiview Coding [29] used multiple different views of an image as positive samples and used another image as negative to make positive pairs near in the embedding space. InfoMin [30] studied the importance in selecting less mutual information between views of the same image for better-augmented views in contrastive learning. SimCLR [7] used a nonlinear transformation between different data augmentations and combined with a contrastive loss to improve the representations. However, SimCLR requires a large batch size to work well. MoCo [15] used different learning updates for the main network and the encoder network. BYOL [13] combined asymmetric learning with two networks - online and target, where the online network was trained using an augmentation of the image to predict the representation of the target with a different augmentation of the image. BYOL used only positive pairs and was able to learn faster and more efficient data representations.

SimSiam [8], similar to BYOL, modified both the network architecture and updated the parameters to introduce asymmetry. SimSiam used a predictor network to update the model parameters using one view of the network and used stop-gradient to prevent the model from collapsing. Barlow Twins [33] used a cross-correlation matrix between the two networks which were passed with different representations of the same image. The objective here is to make the distance as closer to the identity matrix as possible.

3 Rationale

In this section, we introduce the rationale behind our proposed method contrasting both SimSiam and Barlow Twins and introducing the Opt-SSL framework.

SimSiam [8] used a Siamese network where the two sub-networks of the Siamese network consist of an encoder and a predictor. The encoder consists

of a backbone followed by a projection MLP head. The outputs of the encoder and the predictor are used to compute the loss function, which is a negative cosine similarity loss. The training is done using batches of images and a loss function is computed over each batch in order to optimize the goal. One of the main contributions of the SimSiam is the 'stop-gradient', which was proposed to avoid model collapsing.

On the other hand, Barlow Twins [33] used a loss function based on Horace Barlow's redundancy-reduction principle [2] applied to a Siamese network. This model also consists of an encoder, followed by a projector. The outputs of the projector are used in the loss function. The objective function is based on a cross-correlation matrix between the outputs of both projectors and is comprised of two terms - an *invariance term* and a *redundancy reduction term*. The *invariance term* aims to make the representations invariant to the distortions applied while the *redundancy reduction term* aims to decorrelate as much as possible the features of the embeddings, so as to reduce the redundancy between features.

The main conceptual difference between SimSiam and Barlow Twins is that former learns by directly contrasting the representations of different images, whereas the later does the contrasting along the features, trying to minimize their redundancy.

4 A New Self-Supervised Learning Method (Opt-SSL)

In this paper, we use an asymmetric Siamese network, inspired in the excellent results achieved by the SimSiam [8] and BYOL [13]. The proposed framework can be seen in Fig. 1. Here, we use a process $\mathcal{T}$, wherein for each image x_i in a batch $X = \{x_i\}_{i=1}^N$, we generate two augmented views $x_i^A \sim \mathcal{T}(x_i)$ and $x_i^B \sim \mathcal{T}(x_i)$. This process leads to the creation of two batches of augmented images, $X^A = \{x_i^A\}_{i=1}^N$ and $X^B = \{x_i^B\}_{i=1}^N$. The two distorted batches are processed by an encoder f, which shares weights between the two batches, obtaining $Z^A = f(X^A)$ and $Z^B = f(X^B)$. Next, we use a predictor h to transform the output of one branch, obtaining $P^A = h(Z^A)$. Finally, the output of the first branch P^A is contrasted with the output of the second branch Z^B, by means of the proposed loss function explained bellow.

Another important component included in our training pipeline is the stop-gradient operation [8]. We implement it by treating Z^A and Z^B as constants in the computation of the loss function. This means that the backpropagation method is only carried out through the variables P^A and P^B. The encoder f will not receive gradients from Z^A and Z^B, but it will receive them from P^A and P^B to update the weights.

During the training phase, we also interchange the inputs so as to obtain $P^B = h(f(X^B))$ and $Z^A = f(X^A)$. These terms are added to the loss function, making it symmetric. Following the general conventions, we assume that $Z^A = (z_{ij}^{(A)})$, $Z^B = (z_{ij}^{(B)})$, $P^A = (p_{ij}^{(A)})$, $P^B = (p_{ij}^{(B)})$ where $\forall i \in \{1, ..., N\}$ and $\forall j \in \{1, ..., D\}$ $f(x_i^A) = (z_{i1}^{(A)}, z_{i2}^{(A)}, ..., z_{iD}^{(A)})$, $f(x_i^B) = (z_{i1}^{(B)}, z_{i2}^{(B)}, ..., z_{iD}^{(B)})$, $h(f(x_i^A)) = (p_{i1}^{(A)}, p_{i2}^{(A)}, ..., p_{iD}^{(A)})$ and $h(f(x_i^{(B)})) = (p_{i1}^{(B)}, p_{i2}^{(B)}, ..., p_{iD}^{(B)})$ are the

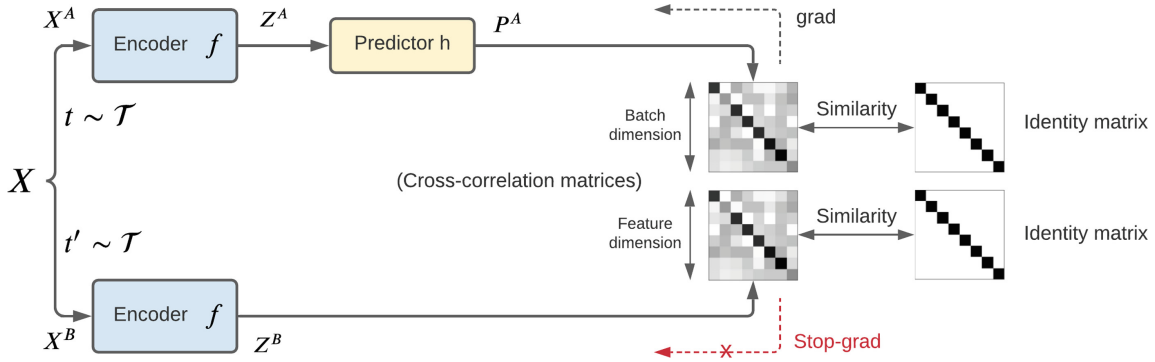


Fig. 1. Proposed Opt-SSL method.

representations of the two distorted views of the image x_i, N is the batch size, and D is the dimension of the representations. The symmetric loss function $\mathcal{L}_{Opt\text{-}SSL}$, is defined as follows:

$$\mathcal{L}_{Opt\text{-}SSL} = \mathcal{L}_{i\text{-}diag} + \lambda_1 \cdot \mathcal{L}_{i\text{-}off\text{-}diag} + \mathcal{L}_{f\text{-}diag} + \lambda_2 \cdot \mathcal{L}_{f\text{-}off\text{-}diag} \ , \tag{1}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}_{>0}$ are constants that balance the terms importance, the values of λ_1 and λ_2 were empirically determined and set to be equal (e.g. 0.5).

$$\mathcal{L}_{i\text{-}diag} = \sqrt{\frac{1}{2N}\Big(\sum_{i=1}^{N}(1-c_{ii}^{(1)})^2 + \sum_{i=1}^{N}(1-c_{ii}^{(2)})^2\Big)} \ , \tag{2}$$

$$\mathcal{L}_{i\text{-}off\text{-}diag} = \sqrt{\frac{1}{2N(N-1)}\Big(\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(c_{ij}^{(1)})^2 + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(c_{ij}^{(2)})^2\Big)} \ , \tag{3}$$

$$\mathcal{L}_{f\text{-}diag} = \sqrt{\frac{1}{2D}\Big(\sum_{i=1}^{D}(1-c_{ii}^{(3)})^2 + \sum_{i=1}^{D}(1-c_{ii}^{(4)})^2\Big)} \ ,$$

$$\mathcal{L}_{f\text{-}off\text{-}diag} = \sqrt{\frac{1}{2D(D-1)}\Big(\sum_{i=1}^{D}\sum_{j=1,j\neq i}^{D}(c_{ij}^{(3)})^2 + \sum_{i=1}^{D}\sum_{j=1,j\neq i}^{D}(c_{ij}^{(4)})^2\Big)} \ .$$

where we define each term of the loss function as follows:

$$c_{ij}^{(1)} = \frac{\sum_{d=1}^{D} z_{id}^{A} p_{jd}^{B}}{\sqrt{\sum_{d=1}^{D}(z_{id}^{A})^2}\sqrt{\sum_{d=1}^{D}(p_{jd}^{B})^2}} \ \forall i,j \in \{1,...,N\},$$

$$c_{ij}^{(2)} = \frac{\sum_{d=1}^{D} z_{id}^{B} p_{jd}^{A}}{\sqrt{\sum_{d=1}^{D}(z_{id}^{B})^2}\sqrt{\sum_{d=1}^{D}(p_{jd}^{A})^2}} \ \forall i,j \in \{1,...,N\}.$$

$(c_{ij}^{(1)})$ and $(c_{ij}^{(2)})$ are the cross-correlation matrices between Z^A and P^B, Z^B and P^A respectively along the dimension of the features.

$$c_{ij}^{(3)} = \frac{\sum_{n=1}^{N} z_{ni}^A p_{nj}^B}{\sqrt{\sum_{n=1}^{N}(z_{ni}^A)^2}\sqrt{\sum_{n=1}^{N}(p_{nj}^B)^2}} \; \forall i, j \in \{1, ..., D\},$$

$(c_{ij}^{(3)})$ is the cross-correlation matrices between Z^A and P^B along the dimension of the batch. We get $(c_{ij}^{(4)})$ by replacing Z^A by Z^B and P^B by P^A.

Note that the first term, $\mathcal{L}_{i\text{-}diag}$, of the loss function (Eq. (1)) is similar to the loss function used in SimSiam. However, the second term, $\mathcal{L}_{i\text{-}off\text{-}diag}$, is added to compare the negative pairs of images. The third term, $\mathcal{L}_{f\text{-}diag}$, is comparable to the *invariance term* of Barlow Twins while the last term, $\mathcal{L}_{f\text{-}off\text{-}diag}$, is analogous to the *redundancy reduction term* of Barlow Twins. However, $\mathcal{L}_{Opt\text{-}SSL}$ is not the sum of SimSiam's and Barlow Twins' loss functions. We treat the image pairs differently compared to both the methods (Eq. 1).

$\mathcal{L}_{i\text{-}diag}$ includes the diagonal elements of matrices $(c_{ij}^{(1)})$ and $(c_{ij}^{(2)})$. These diagonal elements correspond to the cosine similarity between the predictor and the encoder representations of both distorted views of a same image. In this term, the positive pairs of views are contrasted. As each element is the cosine similarity between a positive pair of images, we tend to go towards unity (perfect correlation). $\mathcal{L}_{i\text{-}off\text{-}diag}$, whereas, includes the off-diagonal elements of matrices $(c_{ij}^{(1)})$ and $(c_{ij}^{(2)})$. These elements correspond to the cosine similarity between the predictor and the encoder representations of both distorted views of different images. Here, the negative pairs are contrasted and we tend this term to go towards 0 (no correlation).

$\mathcal{L}_{f\text{-}diag}$ includes the diagonal elements of matrices $(c_{ij}^{(3)})$ and $(c_{ij}^{(4)})$. The i^{th} diagonal element of these matrices is the cosine similarity between the vector formed by the i^{th} feature of the predictor of all the views in a batch and the vector formed by the i^{th} feature of the encoder. This term makes the representations invariant to the distortions applied, making them to go towards unity. $\mathcal{L}_{f\text{-}off\text{-}diag}$ includes the off-diagonal elements of matrices $(c_{ij}^{(3)})$ and $(c_{ij}^{(4)})$. The element $c_{ij}^{(k)}, i \neq j$, of these matrices is the cosine similarity between the vector formed by the i^{th} feature of the predictor of all the views in a batch and the vector formed by the j^{th} feature of the encoder. This term aims to decorrelate as much as possible the features of the embeddings, so to reduce the redundancy between the features and is made to go towards 0.

Regarding the complexity of our proposal, Barlow Twins has a complexity of $O(m^2)$, where m is the number of features used in the latent space. SimSiam has a complexity of $O(n)$, where n is the batch size. In contrast, Opt-SSL has a complexity of $O(m^2) + O(n^2)$, since we are comparing both the positive and negative samples.

Alternatively to the Opt-SSL, we propose a reduced version, which we call Reduced Opt-SSL consisting of its first two terms (Eq. 1). Note that the complexity of the Reduced Opt-SSL is $O(n^2)$.

5 Validation

In this section, we describe the dataset used, we present the implementation setting and the evaluation metrics, and finally we discuss the obtained results.

5.1 Dataset

We use a popular food dataset, called Food-101 [4], to validate our proposed framework. Food-101 consists of 101 classes, with 1000 images (750 training, rest test) per class. We use the standard train-test split [26] for our experiments. The test set is used as validation set.

5.2 Implementation Details

Inspired in [33], the image augmentation process $\mathcal{T}$ used in Opt-SSL to generate the two distorted image views consists of the following transformations, applied sequentially: random resize cropping to 224×224, random horizontal flipping, color jittering, grayscale conversion, Gaussian blurring, and solarization. Random resize cropping is applied by default, whereas the rest are applied based on a probability score similar to the ones used in BYOL [13] and Barlow Twins [33].

The encoder f of the SSL network consists of a backbone, which is also a ResNet-50 [16], but without the final classification layer, followed by a projection MLP head. The MLP head has 3 fully-connected (fc) layers, each one with batch normalization applied. In the first two layers (hidden layers), a ReLU layer is applied as activation layer, but the last layer does not have any activation function. All the three layers have 2048 output units. The predictor h has 2 fc layers. Only the first layer has batch normalization applied and uses a ReLU as activation function. The input and output dimensions of h are both 2048 and the output of the hidden layer is 512. We use default PyTorch initialization for the projection MLP head and the predictor. The weights of each fc layer are initialized by a uniform distribution $\mathcal{U}(\sqrt{-k}, \sqrt{k})$, where $k = 1/\#(\text{input units})$.

For the SSL pretext tasks, we use a ResNet-50 pre-trained on ImageNet [10] with an input image of size $224 \times 224 \times 3$. We use a batchsize of 32 and SGD optimizer with a weight decay of 0.0001, momentum of 0.9, and a learning rate of $lr \times$ batch size/256, starting with $lr = 0.05$. We make use of the cosine decay schedule. For reproducing the results of SimSiam and Barlow Twins, we use the default settings defined in the original papers, except the batch size, where we use 32 batch size. For all the studies, we train the SSL models for 150 epochs.

For the downstream tasks, we use the classifier architecture formed using the encoder of the SSL models which acts as backbone (ResNet-50 without the final classification layer), followed by a dropout layer (with a dropout rate of 0.4), and a classification layer with softmax activation. We use RandAugment [9] and standard data augmentations to reduce overfitting. We also used Stochastic Gradient Descent (SGD) with Warm Restarts [25] with a learning rate of 0.001, weight decay of 0.0005, momentum as 0.9 and T_{mul} of 2. We applied a batch size of 64. We used PyTorch framework in Nvidia RTX 20i or all our experiments.

Table 1. Validation accuracy of different SSL frameworks.

SSL model	λ	k-NN acc.		Linear classifier
		Top-1 acc. (%)	Top-5 acc. (%)	Val. Acc. (%)
Barlow twins [33]	0.0051	49.89	79.07	87.39
SimSiam [8]	–	54.33	81.62	88.60
Reduced Opt-SSL	0.5	**63.52**	**87.48**	88.53
Opt-SSL	0.5	62.06	86.43	**88.63**

5.3 Results and Analysis

We used two evaluation strategies to validate our proposed method. For the first evaluation, we computed the overall accuracy using a k-Nearest Neighbors (k-NN) classifier similar to the one by Wu et al. [32]. We used $k = 200$ and $\tau = 10$ for all the experiments. Table 1 compares the k-NN performance and the accuracy of the linear classifier using the three SSL methods. We can see that SimSiam has better learned representations compared to the Barlow Twins. Apart from the architecture and configuration setting variations between them, the most important difference is that SimSiam learns by directly contrasting the representations of different images, while the Barlow Twins method does the contrasting along the features, trying to minimize their redundancy. However, we can see that a hybrid combination of both approaches, as with the proposed Opt-SSL, is much better to learn the visual representations, which shows a 7.73% increase in the top-1 accuracy compared to SimSiam in the k-NN classifiers and a little higher on the linear classifier performance. The second optimal solution is the Reduced Opt-SSL that contains only the first two terms in Eq. (1). In fact, one can see that the Reduced Opt-SSL gave the best results when using the k-NN classifier but the best final linear classifier was for the Opt-SSL.

5.4 Empirical Study

Next we show an empirical study where we considered mainly the reduced version of Opt-SSL (Reduced Opt-SSL) in order to optimize the computation. The results can be shown to be easily extrapolated to the Opt-SSL loss function, as it is an extension of the previous one.

Normalizing the Opt-SSL's Loss Function. Each term of the loss function behaves differently and it is important to normalize the terms accordingly. We take the square root of the average of the atomic values of each term (Eq. (1)). We empirically show the importance of this normalization by comparing the performance of the first two terms of Eq. (1) with the non-normalized loss terms (Eq. (4)).

Table 2. Effect of normalizing the loss terms.

Experiment	λ	Top-1 acc. (%)	Top-5 acc. (%)
Non-normalized Reduced Opt-SSL	0.0	37.61	67.59
Reduced Opt-SSL	0.0	58.48	84.53
Non-normalized Reduced Opt-SSL	1.0	53.02	80.52
Reduced Opt-SSL	1.0	58.94	84.78

$$\begin{aligned}\mathcal{L}^{non-norm} = &\Big(\sum_{i=1}^{N}(1-c_{ii}^{(1)})^2 + \sum_{i=1}^{N}(1-c_{ii}^{(2)})^2\Big) \\ &+\lambda\cdot\Big(\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(c_{ij}^{(1)})^2 + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}(c_{ij}^{(2)})^2\Big).\end{aligned} \tag{4}$$

From Table 2, we see that the performance improves with normalization of the loss terms. We also study the behaviour with different values of λ and find the improvement consistent.

Negative Sample Contrasting. As discussed in Sect. 4, the first term of Opt-SSL's loss function is similar to that of SimSiam. As an intermediate definition of the loss functions of SimSiam and the Reduced Opt-SSL, we extended the original loss function of SimSiam to include a second term to compare the negative samples represented by c_{ij}. To this aim, we defined an extended loss function of SimSiam as follows:

$$\mathcal{L}_{SimSiam}^{ext} = \mathcal{L}_{SimSiam} + \lambda\cdot\frac{1}{2N(N-1)}\Big(\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}|c_{ij}^{(1)}| + \sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}|c_{ij}^{(2)}|\Big),$$

where $\mathcal{L}_{SimSiam}$ is the original SimSiam loss function, $\lambda \in \mathbb{R}_{>0}$ is a constant to balance the importance between the terms, and $(c_{ij}^{(1)})$ and $(c_{ij}^{(2)})$ are the cross-correlation matrices defined in Sect. 4.

In Table 3, we show the results with different values of λ of both the SimSiam with negative sample contrasting and the Reduced Opt-SSL loss function. In all the experiments, we observe that the loss terms of Opt-SSL behave better than the original SimSiam model. It can further be seen through the experiment with $\lambda = 0$, where the loss of Reduced Opt-SSL behaves in the same way as SimSiam, but shows a higher performance. We observe that the loss function of Reduced Opt-SSL (Eq. 2), which differs from SimSiam in the way the cosine similarity of samples is treated, gives a better performance.

Selection of λ. In Table 4, we show the experiments with respect to the selection of λ using Reduced Opt-SSL. We see that 0.5 shows the highest performance.

Table 3. Study of the behaviour of Loss Terms.

Experiment	λ	Top-1 acc. (%)	Top-5 acc. (%)
Original SimSiam	–	54.33	81.62
SimSiam with negative	0.5	57.47	83.71
Sample contrasting	1.0	57.88	83.92
Reduced Opt-SSL	0.0	58.48	84.53
	0.5	59.00	84.91
	1.0	58.94	84.78

Table 4. Selection of λ.

λ	Top-1 acc. (%)	Top-5 acc. (%)
0.0	58.48	84.53
0.5	59.00	84.91
0.75	58.82	84.81
1.0	58.94	84.78

Table 5. Study of different data augmentation techniques in SSL.

Experiment	λ	Top-1 acc. (%)	Top-5 acc. (%)
SimSiam (SimSiam DA)	–	54.33	81.62
SimSiam (Proposed DA)	–	61.76	86.48
Reduced Opt-SSL (SimSiam DA)	0.5	59.00	84.91
Reduced Opt-SSL (Proposed DA)	0.5	63.52	87.48

Data Augmentation Techniques. In this experiment, we check the importance of data augmentation techniques used in Opt-SSL. We compare the performance of data augmentation techniques used in SimSiam with the data augmentation techniques defined for Sect. 5.2. Just with changing the strategies, the original SimSiam model performed better. We also compare the performance of the image correlation terms (Eqs. (2) and (3)) with the proposed loss term and the best performance is achieved with the proposed method. From Table 5, we see that the combination of our proposed method and data augmentation techniques is optimal.

6 Conclusions

In this paper, we study the behaviour of the state of the art methods, specially, SimSiam and Barlow Twins and proposed a new modified SSL method, called Opt-SSL. In this work, we combine the feature representation learned by Barlow Twins with the image representation learned by SimSiam. We show

the importance of each component of the proposed loss function using different experiments. We validated our method using a challenging food recognition task and shew to improve the results compared to the original state of the art SSL methods. One of the main advantages of our proposal is that it considers positive and negative pairs in an efficient way assuring that the resulting self-supervised approach is more beneficial. Next steps involve including annotations from supervised learning datasets to apply the self-learning heuristics to treat differently images from the same class and different classes.

Acknowledgements. This work was partially funded from the European Union's Horizon 2020 Research and Innovation programme under Open Call budget of the grant agreement No. 857159 (SHAPES), TIN2018-095232-B-C21, SGR-2017 1742 and CERCA Programme/Generalitat de Catalunya. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

1. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910 (2019)
2. Barlow, H.B., et al.: Possible principles underlying the transformation of sensory messages. Sensory Commun. **1**(01), 1–18 (1961)
3. Becker, S., Hinton, G.E.: Self-organizing neural network that discovers surfaces in random-dot stereograms. Nature **355**(6356), 161–163 (1992)
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 446–461. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_29
5. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV, pp. 132–149 (2018)
6. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607. PMLR (2020)
8. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR, pp. 15750–15758 (2021)
9. Cubuk, E., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: practical automated data augmentation with a reduced search space. In: CVPR Workshops, pp. 702–703 (2020)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE CVPR, pp. 248–255 (2009)
11. El Khoury, C.F., Karavetian, M., Halfens, R.J., Crutzen, R., Khoja, L., Schols, J.M.: The effects of dietary mobile apps on nutritional outcomes in adults with chronic diseases: a systematic review and meta-analysis. J. Acad. Nutr. Diet. **119**(4), 626–651 (2019)
12. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS, vol. 27 (2014)
13. Grill, J.B., et al.: Bootstrap your own latent: a new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)

14. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: 13th ICAIS, pp. 297–304. JMLR Workshop (2010)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR, pp. 9729–9738 (2020)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
17. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: a survey. IEEE Trans. PAMI **43**, 4037–4058 (2020)
18. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1× 1 convolutions. In: 32nd NIPS, pp. 10236–10245 (2018)
19. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1920–1929 (2019)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
21. Liao, Y.H., Kar, A., Fidler, S.: Towards good practices for efficiently annotating large-scale image classification datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4350–4359 (2021)
22. Liu, C., Liang, Y., Xue, Y., Qian, X., Fu, J.: Food and ingredient joint learning for fine-grained recognition. IEEE Trans. Circ. Syst. Video Technol. **31**, 2480–2493 (2020)
23. Liu, X., et al.: Self-supervised learning: generative or contrastive. IEEE Trans. Knowl. Data Eng. (2021)
24. Liu, X., Deng, Z., Yang, Y.: Recent progress in semantic image segmentation. Artif. Intell. Rev. **52**(2), 1089–1106 (2018). https://doi.org/10.1007/s10462-018-9641-3
25. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
26. Martinel, N., Foresti, G.L., Micheloni, C.: Wide-slice residual networks for food recognition. In: 2018 WACV, pp. 567–576. IEEE (2018)
27. Meng, L., et al.: Learning using privileged information for food recognition. In: 27th ACM ICMM, pp. 557–565 (2019)
28. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: NIPS, pp. 14866–14876 (2019)
29. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12356, pp. 776–794. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58621-8_45
30. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? arXiv preprint arXiv:2005.10243 (2020)
31. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: unsupervised learning of invariances. Neural Comput. **14**(4), 715–770 (2002)
32. Wu, Z., Xiong, Y., Yu, S., Lin, D.: Unsupervised feature learning via non-parametric instance-level discrimination. arXiv preprint arXiv:1805.01978 (2018)
33. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230 (2021)
34. Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. In: IEEE CVPR, pp. 6002–6012 (2019)

This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

All4One: Symbiotic Neighbour Contrastive Learning via Self-Attention and Redundancy Reduction

Imanol G. Estepa
Universitat de Barcelona,
Barcelona, Spain
igonzaes42@alumnes.ub.edu

Ignacio Sarasúa
NVIDIA
isarasua@nvidia.com

Bhalaji Nagarajan
Universitat de Barcelona,
Barcelona, Spain
bhalaji.nagarajan@ub.edu

Petia Radeva
Universitat de Barcelona,
Barcelona, Spain
Computer Vision Center,
Cerdanyola (Barcelona), Spain
petia.ivanova@ub.edu

Abstract

Nearest neighbour-based methods have proved to be one of the most successful self-supervised learning (SSL) approaches due to their high generalization capabilities. However, their computational efficiency decreases when more than one neighbour is used. In this paper, we propose a novel contrastive SSL approach, which we call All4One, that reduces the distance between neighbour representations using "centroids" created through a self-attention mechanism. We use a Centroid Contrasting objective along with single Neighbour Contrasting and Feature Contrasting objectives. Centroids help in learning contextual information from multiple neighbours whereas the neighbour contrast enables learning representations directly from the neighbours and the feature contrast allows learning representations unique to the features. This combination enables All4One to outperform popular instance discrimination approaches by more than 1% on linear classification evaluation for popular benchmark datasets and obtains state-of-the-art (SoTA) results. Finally, we show that All4One is robust towards embedding dimensionalities and augmentations, surpassing NNCLR and Barlow Twins by more than 5% on low dimensionality and weak augmentation settings. Source code is available in **https://github.com/ImaGonEs/all4one**.

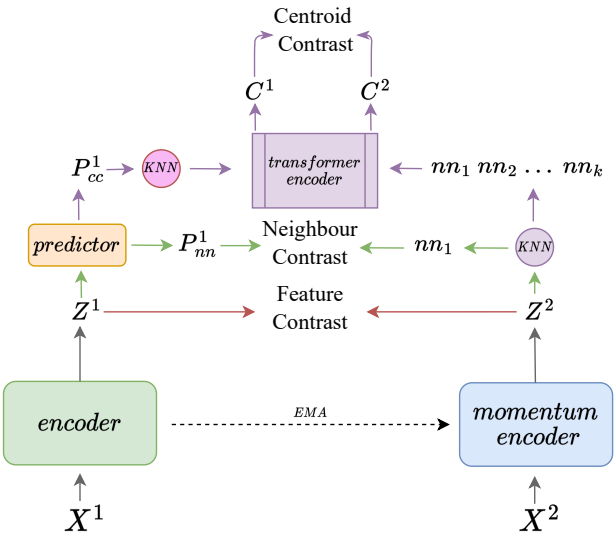


Figure 1: **Simplified architecture of All4One.** All4One uses three different objective functions that contrast different representations: Centroid objective contrasts the contextual information extracted from multiple neighbours while the Neighbour objective assures diversity [14]. Additionally, the Feature contrast objective measures the correlation of the generated features and increases their independence.

1. Introduction

Deep learning (DL) models strongly depend on the availability of large and high-quality training datasets whose construction is very expensive [25]. Self-supervised learning (SSL) claims to allow the training of DL models without the need for large annotated data, which serves as a milestone in speeding up the DL progression [25]. The

most popular SSL approaches rely on instance discrimination learning, a strategy that trains the model to be invariant to the distortions applied to a single image defined as positive samples [6, 18, 31]. As all the views belong to the same image, consequently, they belong to the same semantic class. Bringing them together in the same feature space encourages the model to create similar representations for similar images. The encouraging results of initial works such as SimCLR [6] and BYOL [15] boosted multiple improvements that address common problems of instance discrimination such as lack of diversity between samples and model collapse.

Neighbour contrastive learning hinges on the fact that data augmentations do not provide enough diversity in selecting the positive samples, as all of them are extracted from the same initial image [14]. To solve it, Nearest neighbour Contrastive Learning (NNCLR) [14] proposes the use of nearest neighbours (NN) to increase the diversity among the positive samples which in turn boosts the generalization of the model. Instead of bringing together two distortions created from the same image, they increase the proximity between a distorted sample and the NN of another distorted sample. However, relying entirely on the first neighbour holds back the real potential of the approach. MSF [21] proposes the use of k neighbours to increase the generalization capability of the model. However, MSF suffers from high computation as the objective function needs to be computed for each neighbour (k times). Apart from the low diversity of positive samples, instance discrimination approaches suffer from model collapse, a scenario where the model learns a constant trivial solution [15]. Barlow Twins [38] proposes a redundancy reduction-based approach that naturally avoids the collapse by measuring the correlation among the features on the generated image representations. However, this collapse avoidance suffers from the requirement of projecting embeddings in high dimensions.

In our work, we contrast information from multiple neighbours in a more efficient way by avoiding multiple computations of the objective function. This way, we are able to increase the generalization from neighbour contrastive approaches while avoiding their flaws. For that, we propose the use of a new embedding constructed by a self-attention mechanism, such as a transformer encoder, that combines the extracted neighbour representations in a single representation containing contextual information about all of them. Hence, we are able to contrast all the neighbours' information on a single objective computation. We make use of a Support Set that actively stores the representations computed during the training [14] so that we can extract the required neighbours. In addition, we integrate our approach with a redundancy reduction approach [38]. Making the computed cross-correlation matrix close to the identity reduces the features redundancy of the same image representation while also making them invariant to their distortions. This idea contrasts the representations in a completely different way than the rest of instance discrimination approaches [28, 30, 14]. For this reason, we increase the richness of the representations learnt by the model by combining the neighbour contrast approach with the redundancy reduction objective that directly contrasts the features generated by the encoder and aims to increase their independence. In addition, the need for high-dimensional embeddings of redundancy reduction feature contrast approaches [38] is alleviated thanks to our SSL objective combination.

As a summary, in this paper, we introduce a new symbiotic SSL approach, which we call All4One, that leverages the idea of neighbour contrastive learning while combining it with a feature contrast approach (see Figure 1). All4One integrates three different objectives that prove to benefit each other and provide better representation learning. Our contributions are as follows: (i) We define a novel objective function, centroid contrast, that is based on a projection of sample neighbours in a new latent space through self-attention mechanisms. (ii) Our proposal, All4One, is based on a combination of centroid contrast, neighbour contrast and feature contrast objectives, going beyond the single neighbour contrast while avoiding multiple computations of the objective function; (iii) We demonstrate how contrasting different representations (neighbours and distorted samples) using InfoNCE [28] based and feature contrast objectives benefit the overall performance and alleviate individual flaws such as the reliance of high-dimensional embeddings on feature contrast approaches; (iv) We show that All4One, by contrasting the contextual information of the neighbours and multiple representations, outperforms single nearest neighbour SSL on low-augmentation settings and low-data regimes, proving much less reliance on augmentations and increased generalization capability; and (v) We prove that All4One outperforms single-neighbour contrastive approaches (among others) by more than 1% in different public datasets and using different backbones.

2. Related Works

Self-supervised Learning. In SSL methods, a model is trained to learn intermediate representations of the data in a completely unsupervised way to be transferred later to multiple tasks [25]. Since the introduction of contrastive loss [10], several works such as SimCLR [6] and MoCo [18] proved their usefulness in multiple downstream tasks by proposing variations of InfoNCE [28], a contrastive objective function inspired by Noise Contrastive Estimation (NCE) [26]. In image representation learning, this objective function works with the assumption of positive pairs (z^a, z^+), formed by two representations that share the same semantic class, and negative pairs (z^a, z^-), which are formed by image representations that do not belong to the

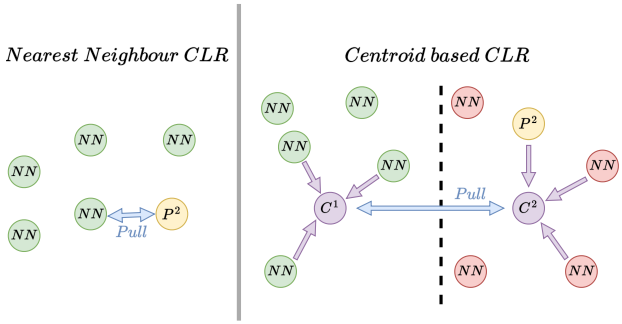


Figure 2: **Neighbour contrast comparison.** While the common neighbour contrastive approaches only contrast the first neighbour, we create representations that contain contextual information from the k NNs and contrast it in a single objective computation.

same semantic class. Thus, the positive pairs are pulled together in the same feature space while the negative pairs are repelled to avoid model collapse. Later, BYOL [15] proved that it is possible to achieve the same effect without using negative samples by avoiding the collapse with the introduction of architectural changes such as a predictor. Additionally, Barlow Twins [38] introduced a novel objective function based on the redundancy reduction principle instead of InfoNCE that naturally avoided the collapse.

Overall, discriminative frameworks have been obtaining exceptional results [5, 7, 8, 6] due to the improvements done by introducing new architectures [18, 5, 9], applying new objective functions individually (e. g. redundancy reduction) [27, 4, 2], alternative augmentation settings [32] or even proposing novel strategies such as NN based approaches [14, 21], providing an increased generalization by contrasting distortions with NNs.

During training, NN-based approaches store the representations in a queue and extract them by applying a k-NN algorithm that uses one of the representations in the positive pair as a query. This way, one of the pair representations is swapped by its neighbour for the loss computation. Nevertheless, using a single neighbour per sample holds back the potential of the approach meanwhile using k neighbours reduces the efficiency of the NN approaches as the objective function needs to be computed multiple times.

Self-attention. Since the introduction of the Transformer [33] architecture, self-attention-based models have proved to be one of the most successful approaches in Computer Vision (CV). In fact, multiple works analyse the behaviour of the self-attention mechanism and combine it with other well-known tools such as k-NN algorithms [35, 16, 22, 34]. In SSL, Self-attention has been widely used on generative frameworks [17, 3, 40], where they train the transformer backbone to reconstruct the given masked image. However, these reconstruction objectives used in works such as iBOT [40], BEiT [3] and MAE [17] are computationally expensive and rely on vision transformers exclusively. In our research, different to previous MIM works, we still maintain a contrastive objective. We take advantage of the capacity of the transformer encoder to mix the neighbour representations into a single one that contains information from the neighbours and contrast it using a variation of InfoNCE [28].

3. The All4One Symbiosis

In order to increase the performance and efficiency of previous neighbour contrastive methods, we propose a symbiotic SSL framework that combines three different approaches into one. We show our proposed All4One pipeline in Figure 3, where we show the three different objectives: the first objective is a neighbour contrast objective (green path); the second objective is a centroid contrast objective (purple path), which is carried out by the application of self-attention mechanisms [33] and the final objective is a redundancy reduction-based feature contrast objective (red path). During training, initial representations are transformed depending on the followed path to adapt them to the objective of the path.

The pipeline is composed of a pair of encoders or neural networks, f, and a pair of projectors, g. The projector consists of a basic MLP that transforms the output of the encoders [6]. In this case, we apply a momentum encoder, which is a smoothed version of the online encoder, similar to BYOL [15]. The pipeline is iterated by a batch of images, X. For each image, two batches of augmented/distorted images, X^1 and X^2 are generated using a data augmentation pipeline. Then, both distorted images are fed to the encoder and, next, to the projector. This sequence of encoder-projector is defined as momentum or online branch depending on the encoder used. We call the momentum branch the encoder-projector sequence that contains a momentum projector and vice-versa [15]. The output of momentum and online branches can be defined as $Z^1 = g^\xi(f^\xi(X^1))$ and $Z^2 = g^\theta(f^\theta(X^2))$ respectively, being g and f the projectors and encoders of each branch. Next, we brief each of the objectives used in the proposal.

3.1. Neighbour Contrast

NNCLR [14] is the most popular neighbour contrastive approach. Instead of contrasting two distortions of the same image, it uses the simple KNN operator to extract from a queue or Support Set [14] the NN of the first distortion and contrast it against the second distortion using a variant of InfoNCE [28]. For each i-th pair in the batch, the neighbour contrast loss is defined as:

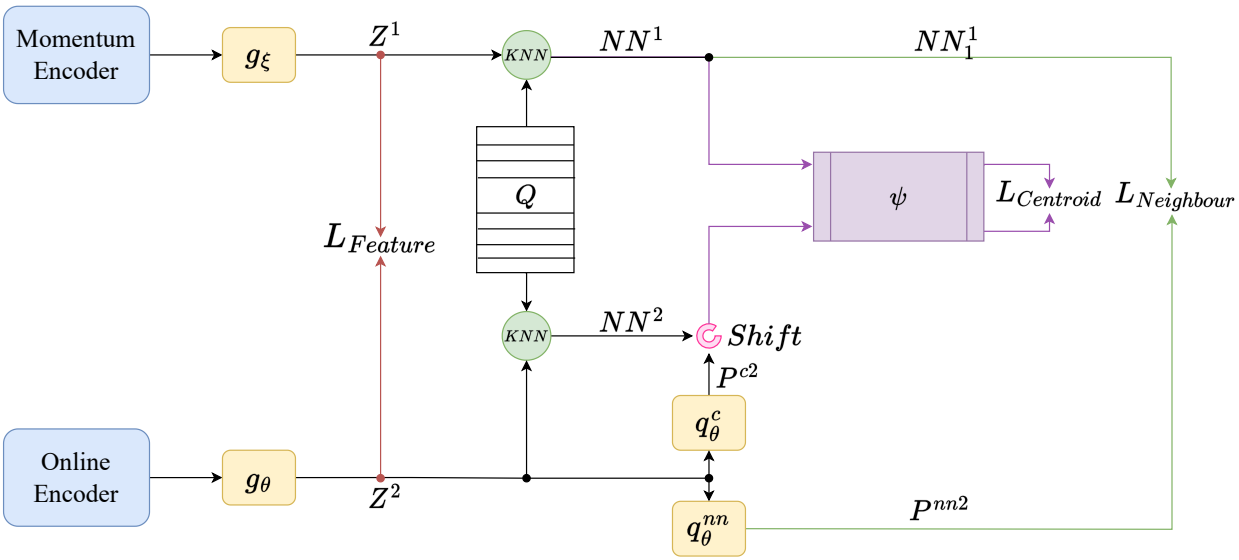


Figure 3: **Complete architecture of All4One framework.** Feature, Centroid and Neighbour contrast objective functions are indicated by red, purple, and green respectively.

$$L_i^{NNCLR} = -log(\frac{exp(nn_i^1 \cdot p_i^2/\tau)}{\sum_{k=1}^{N} exp(nn_i^1 \cdot p_k^2/\tau)}) \quad (1)$$

where nn_i is the i-th NN of the representation z, p_i is the second distortion, τ refers to the temperature constant and N is the number of samples in the batch. This way, it increases the generalization of the model through the use of more diverse samples. Note that, in our case, the second predictor q_θ^{nn} is used for the Neighbour Contrast objective.

3.2. Centroid Contrast

According to MSF [21], using a single NN could be holding back the potential of the approach. In fact, they showed that contrasting the second distorted image with multiple NN could provide a better SSL framework that also obtains higher accuracy regarding the selection of the neighbours. Nevertheless, contrasting multiple neighbours hurts the computational efficiency of the model, as they need to compute the objective function k times, where k is the number of extracted neighbours. For this reason, the improvement is severely constrained by computational resources. Following the idea of using multiple neighbours, we introduce an alternative proposal that does not require multiple-loss computations. We compile the relevant information from the extracted k neighbours to create a pair of representations, defined as "centroids" that contain contextual information about all the neighbours and pull them together in the feature space applying a variation of the InfoNCE [28] objective function. This way, the generalization of the model is improved without contrasting multiple neighbours one by one.

Once Z^1 and Z^2 are computed from the pair of distorted images, we calculate the cosine similarity between each Z (query representation) and the Support set Q, a queue that stores the computed representations [14]. Next, we extract a sequence with the K most similar representations ($nn_i^1 = KNN(z_i^1, Q)$) for each representation in both batches. Then, for each Z^1 and Z^2, we obtain their respective batch of sequences of NNs, NN^1 and NN^2. As we try to avoid contrasting the neighbours one by one, we introduce a new element to the pipeline: a transformer encoder, ψ. Given NN^1 and NN^2, we input ψ with each sequence nn_i^1 to compute the self-attention of the sequences.

Given a sequence of neighbour representations nn_i^1, we obtain a single representation c_1 that contains as much information as possible about the input sequence nn_i^1. When computing self-attention [33], we mix the representations of the input sequence in a weighted manner so that a new enriched vector of representations is returned. Each element of this enriched vector contains contextual information about all the neighbours in the sequence. During training, for each sequence in NN^1, the process is made up of the following steps: (i) for each sequence Seq_i in NN^1, we add sinusoidal positional encoding [33]; (ii) then, we feed the transformer encoder ψ with Seq_i; (iii) inside the transformer encoder, self-attention is computed and a new sequence is returned Seq_i^c; (iv) finally, we select the first representation $Seq_i^c{}_1$ in the returned sequence Seq_i^c as our centroid c_i as we aim to contrast a single representation

that contains context information from the rest of the neighbours. After selecting the first representation on all sequences, we obtain a batch of representations defined as C^1.

On the online branch, a slightly different process is followed. A second MLP similar to the projector, the predictor, is used to change the feature space of Z^2 batch and obtain $P^{c2} = q_\theta^c(Z^2)$ batch of transformed representations. More concretely, we pass Z^2 through the centroid predictor q_θ^c. Then, we replace the fifth neighbour in each sequence nn_i^2 of NN^2 by p_i^{c2}. Finally, we reorder each sequence so p_i^{c2} is the first element. We define this process as $Shift$ operation. This is done to introduce the distorted image in the sequence, thereby impacting the back-propagation. More information about the *Shift* operation is provided in Section 1 of the supplementary material. Once modified NN^2 is created, we pass it through the transformer encoder to obtain C^2 by following the previously explained process. Finally, we contrast C^1 and C^2 using a variation of the InfoNCE [28] loss function aiming to bring the neighbour centroids together (see Figure 2). For each centroid pair, the centroid loss can be defined as:

$$L_i^{centroid} = -log\left(\frac{exp(c_i^1 \cdot c_i^2/\tau)}{\sum_{n=1}^{n} exp(c_i^1 \cdot c_n^2/\tau)}\right) \tag{2}$$

3.3. Reducing Redundancy: Feature Contrast

The application of the redundancy reduction principle to increase the independence of the features is one of the most successful approaches in the SSL SoTA [38]. The Barlow Twins' main idea is that, instead of focusing on the images, to directly contrast the features by computing a cross-correlation matrix C_{ij}. Then, it aims to increase the invariance of the features by equating the diagonal elements c_{ii} of C_{ij} to one (invariance term) while also decreasing the redundancy between the features by reducing the correlation between different features (i^{th} and j^{th} features). Inspired by this approach, we increase the richness of our framework by introducing a feature contrast objective function that measures the correlation of the features and aims to increase their independence.

To do so, we construct the output of the momentum and online projectors as $(Z)_{ij}$ matrices where each element represents an exact feature j of a single augmented image representation, i. Then, we use L_2 normalization on both matrices in the batch dimension [38] and we compute the cosine similarity between the transposed matrix, Z^{1T} and Z^2 to obtain the cross-correlation matrix, CC^1. We compute this term symmetrically. This is done by swapping the branches and computing the similarities. Finally, two cross-correlation matrices CC^1 and CC^2 are obtained. Redundancy reduction feature contrast objective [2] is computed as follows:

$$\begin{aligned} L_{Red.} = &\frac{1}{2}\sqrt{\frac{1}{2D}\sum_{i=1}^{D}((1-cc_{ii}^1)^2 + (1-cc_{ii}^2)^2)} \\ &+\frac{1}{2}\sqrt{\frac{1}{2D(D-1)}\sum_{i=1}^{D}\sum_{j\neq i}^{D}((cc_{ij}^1)^2 + (cc_{ij}^2)^2)} \end{aligned} \tag{3}$$

As it can be noted, the first term increases the correlation between the elements that represent the same feature among the distorted images (diagonal elements), while the second term decreases the correlation between elements that represent different features (off-diagonal elements).

3.4. Final Objective: The All4One Objective

Once all objectives are computed, the final loss function is formed by summing the previously defined objectives. The All4One objective is defined as:

$$L_{All4One} = \sigma L_{NNCLR} + \kappa L_{Centroid} + \eta L_{Red} \tag{4}$$

where σ, κ and η are determined through loss progression as 0.5, 0.5 and 5, respectively. Depending on the dataset, giving more importance to each objective could increase the performance of the models. However, the common benchmarks used for validation are balanced, so we keep the three All4One objectives uniformly weighted. By combining different objectives, the All4One objective improves the learning of representations. We show the improvements over other methods below.

4. Experiments

In this section, we first describe the implementation details of All4One and its training. Then, we evaluate it using the common image classification linear evaluation pipeline on different datasets: CIFAR-10 [23], CIFAR-100 [23], ImageNet-1K [24] and ImageNet-100, a reduced ImageNet of 100 classes. We extensively study the different components of our proposal and discuss the various design decisions in detail. Finally, we also extend our proposal to a transformer-based backbone using ViT [13] and also show the efficacy of our proposal on other downstream tasks.

4.1. Implementation Details

Architecture. All4One follows a momentum instance discrimination pipeline (Figure 3). The momentum branch includes the usual momentum projector, g_ξ [15]. The online branch, on the contrary, includes a projector, g_θ and double predictor, q_θ^{nn} and q_θ^c. MLP projectors are formed by 3 fully connected layers of size [2048, 2048, 256], while the MLP predictor uses 2 fully connected layers with a dimensionality of [4096, 256]. Similar to NNCLR [14], all

fully-connected layers, except the last ones, are followed by batch-normalization [19].

MLP components of a SSL method filter the features generated by the encoder, keeping those that are useful for the downstream task [1]. In All4One, there are multiple objectives that are applied in completely different representations, so the predictors involved should filter the features independently. For this reason, we propose the use of two different predictors that work separately for each objective. Finally, our approach introduces a small Transformer encoder, ψ, that is applied for both branches and a shared $KNN(\cdot,\cdot)$ operator that extracts image representations from the Support Set, Q, given a query. The final objective of our proposal follows Eq. (4), which brings together both contrasting approaches.

Training. We train All4One on CIFAR-10, CIFAR-100, ImageNet100 (using a ResNet-18 backbone) and the complete ILSVRC2012 ImageNet (using a ResNet-50 backbone) without any class label or annotation. During the training, all backbones are initialized with default Solo-learn [11] initialization. All MLP components use the PyTorch default initialization. The transformer encoder uses three transformer encoder layers with 8 heads each [33]. Following SimCLR [6], lr is adapted using $lr * batchsize/256$ formula and readapted for each layer using LARS [37] (only for CNNs). For ResNet-18 and ResNet-50, we use the SGD optimizer and on the ViT-Small backbone, we use AdamW. We find it optimal to use 1.0, 0.00015 and 0.1 as base learning rates for the CNN backbones, ViT and the internal transformer encoder respectively, which are gradually reduced by using a warm-up cosine annealing scheduler. When using AdamW, we detach the lr of the transformer encoder by setting it to a constant of 0.1. For the complete ILSVRC2012 ImageNet experiments, we adapt the learning rates to 1.5 and 0.45 for All4One and NNCLR respectively. All experiments are run with the same batch size and for the same amount of epochs. Finally, the queue or Support set size is set to 98304, following the settings of NNCLR [14]. The rest of the hyperparameters of the model are directly extracted from NNCLR [14]. All the training processes are done on a single NVIDIA RTX 3090 GPU, except for ImageNet experiments where the evaluations are done on 4xNVIDIA V100 GPUs. Training curve comparison is shown in Section 4 of the supplementary material.

4.2. All4One Evaluation

CIFAR and ImageNet100 Linear Evaluations. We compare our approach against the current SoTA SSL frameworks. We first show the evaluations on the CIFAR datasets and the ImageNet-100 dataset. ImageNet-100 is a reduced ImageNet version with 100 classes and the images are 224x224 (as compared to CIFAR which has 32x32).

	CIFAR-10		CIFAR100		ImageNet100	
Method	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
BYOL	92.58	99.79	70.46	91.96	80.16	95.02
DC V2	88.85	99.58	63.61	88.09	75.36	93.22
DINO	89.52	99.71	66.76	90.34	74.84	92.92
MoCoV2+	92.94	99.79	69.89	91.65	78.20	95.50
MoCoV3	93.10	99.80	68.83	90.57	80.36	95.18
ReSSL	90.63	99.62	65.92	89.73	76.92	94.20
SimCLR	90.74	99.75	65.78	89.04	77.64	94.06
Simsiam	90.51	99.72	66.04	89.62	74.54	93.16
SwAV	89.17	99.68	64.88	88.78	74.04	92.70
VIbCReg	91.18	99.74	67.37	90.07	79.86	94.98
VICReg	92.07	99.74	68.54	90.83	79.22	95.06
W-MSE	88.67	99.68	61.33	87.26	67.60	90.94
BT	92.10	99.73	70.90	91.91	80.38	95.28
NNCLR	91.88	99.78	69.62	91.52	79.80	95.28
All4One	**93.24**	**99.88**	**72.17**	**93.35**	**81.93**	**96.23**

Table 1: **Linear evaluation results on CIFAR-10, CIFAR-100 and ImageNet100.** The results are extracted from Solo-learn Self-supervised learning library [11].

We report Top-1 and Top-5 linear accuracies for all the datasets. As can be seen from Table 1, our approach clearly outperforms the previous SoTA approaches, including the ones that inspired our own approach. We gain 1.36%, 2.55% and 2.13% over NNCLR on CIFAR-10, CIFAR-100 and ImageNet-100 respectively, and similarly improve by 1.14%, 1.27% and 1.55% over Barlow Twins. This emphasizes the fact that we are able to outperform both the feature contrast approach and the neighbour contrast approach by a considerable margin by combining them and adding the new contrastive centroid loss.

Linear Evaluation on ImageNet. In table 2a, we compare NNCLR [14] and All4One on the complete ILSVRC2012 ImageNet [12] dataset for linear evaluation task. Considering the computational resources for ImageNet, we perform the linear evaluation only for 100 epochs with an effective batch size of 1024. We are able to outperform NNCLR on the larger ImageNet. This highlights the improvements in the performance of our approach on larger datasets.

Linear evaluation on Transformer Backbone. We study the All4One behaviour on a different backbone by replacing the ResNet with a Transformer backbone. We use a ViT-Small on CIFAR-100 for this study. In Table 2b, we see how All4One outperforms NNCLR verifying the independence of our approach on backbones. We have an improvement of 1.15% compared to the ViT version of NNCLR.

Semi-supervised ImageNet100 and ImageNet Evaluations. We perform semi-supervised evaluation following the experiments in NNCLR [14]. We fine-tune the Ima-

Method	Top-1	Top-5
NNCLR	65.74	86.90
All4One	**66.60**	**87.51**

(a) **Linear evaluation on ILSVRC2012 ImageNet.** For NNCLR, all hyperparameters except for the batch size (we use 1024 for both approaches) are the ones recommended in the original paper [14].

Method	Top-1	Top-5
NNCLR	68.55	90.94
All4One	**69.7**	**91.65**

(b) **Linear evaluation on CIFAR-100 using ViT-Small backbone.** Same hyperparameter settings are used for both methods.

	ImageNet100		ImageNet	
Method	1%	10%	1%	10%
NNCLR	54.14	75.49	37.51	58.74
All4One (Ours)	**58.73**	**76.95**	**38.96**	**60.14**

(c) **Semi-supervised learning results (Top-1 linear accuracy) on ImageNet100 and ImageNet.**

	Food-101	Caltech-101	Dogs	Pets
NNCLR	69.52	90.15	67.47	**83.34**
All4One	**71.16**	**91.10**	**68.07**	81.57

(d) **Transfer learning evaluation (Top-1 linear accuracy).**

Table 2: **All4One linear evaluation experiments.**

geNet pre-trained model (ResNet-50 on 100 epochs) on 1% and 10% subsets of the datasets. The results are presented in Table 2c. As can be seen, All4One generalizes better than NNCLR, outperforming it for both ImageNet100 and ImageNet on 1% and 10% subsets of the data.

Transfer Learning. Finally, we evaluate All4One and NNCLR on transfer learning downstream tasks for Food101 [20], Caltech-101 [36], Dogs [39] and Pets [29] dataset. For all datasets, we freeze the ImageNet pre-trained model (ResNet-50 on 100 epochs) and train a single linear classifier on top of it for 90 epochs on train splits while performing a sweep over the lr to obtain the best-performing one. Then, we evaluate the performance on the validation split for the Food101 dataset and test split for the rest of the datasets. The results are shown in Table 2d. As can be seen, All4One outperforms NNCLR in 3 out of 4 datasets, further validating the increased generalization capabilities of All4One.

4.3. Ablation Study

First, we show the importance of each objective defined by our approach. Then, we analyse the dimensionality and augmentation robustness of our model. Finally, we present some design choices such as the number of layers used by the transformer encoder and the number of extracted neighbours. All the ablations are done following exactly the same settings defined in Section 4.1 if not stated otherwise.

Objective Importance. As explained, our approach introduces redundancy reduction and novel neighbour contrast objectives. For this reason, we find it interesting to analyse, one by one, the importance of each of them. In Table 3, we report the performance of each objective of the framework. In addition, we also study the importance of EMA ($v2$ vs $v3$). We see that the addition of EMA boosts the overall performance of the model.

As can be seen with the different versions of All4One, all three objectives are important regarding the overall performance. Intuitively, both NNCLR [14] and Centroid-based objectives focus on contrasting neighbour image representations, so it is possible that, during the training, both objectives may partially overlap. This is not the case for the redundancy reduction, as it focuses on the features instead, causing its removal to be more critical than the others. Moreover, we designed the redundancy objective to use the representations z_i^1 and z_i^2 to compute the loss rather than using the neighbours. This fact adds more richness to the final loss function, as a total of three different representations (original neighbour nn_i^1, centroid derived from the neighbours c_i^1 and image representation z_i^1) are used for the unified objective.

On the other hand, we check that the Centroid objective is the one that increases the most on the NN retrieval accuracy, reaching 86.16% when combined with the NNCLR [14] and EMA architecture (experiment $v3$). As expected, contrasting contextual information from multiple neighbours encourages the model to create better representations easier to distinguish just by using a simple KNN operator. However, the introduction of different objectives, such as the redundancy reduction objective, forces the model to generalize more and, consequently, perform better on the downstream tasks.

Dimensionality Robustness. Redundancy reduction approaches such as Barlow Twins [38] highly depend on the dimensionality of the embeddings. Other approaches such as Opt-SSL [2] required high dimensional embeddings to provide SOTA results. Our approach, however, manages to outperform Barlow Twins with much lower dimensional embeddings as it does not only depend on the redundancy reduction. Even if the dimensionality of the embedding is low, the symbiosis formed avoids the decrease in performance, as can be seen in Table 4a. Another factor of consideration to use low dimensions is that $KNN(\cdot, \cdot)$ suffers

Method	NNCLR obj.	Cen. obj.	Feat. obj.	EMA	Top-1	k-NN Top-1	NN Top-1
All4One$_{v0}$	✓				69.62	62.16	68.8 (77.8*)
All4One$_{v1}$		✓			67.4	59.61	82.8
All4One$_{v2}$	✓	✓			71.02	63.21	85.28
All4One$_{v3}$	✓	✓		✓	71.08	63.83	**86.16**
All4One$_{v4}$		✓	✓	✓	71.31	63.72	80.6
All4One$_{v5}$	✓		✓	✓	71.64	64.58	78.8
All4One$_{v6}$	✓	✓	✓	✓	**72.17**	**64.84**	**82.16**

Table 3: **All4One objective function ablation study (using CIFAR-100).** All4One$_{v0}$ is equal to vanilla NNCLR. NN retrieval accuracy marked by * represents the NN retrieval obtained by increasing the queue size from 65503 to 98304 [14]

	Top-1	k-NN Top-1
Barlow Twins (2048)	71.21	63.11
Barlow Twins (256)	62.14	54.64
All4One (256)	**72.17**	**64.84**

(a) **Dimensionality analysis using CIFAR-100 dataset.**

Method	Top-1	k-NN Top-1
Barlow Twins [38]	39.66	30.89
NNCLR [14]	35.39	27.64
All4One (Ours)	**44.7**	**33.99**

(b) **Augmentation analysis using CIFAR-100.**

Layer number	Top-1	k-NN Top-1
3	72.17	64.84
6	71.86	64.50
9	71.75	64.52

(c) **Number of transformer layers.**

Number of NN	Top-1	k-NN Top-1
5	72.17	64.84
10	72.00	64.54
15	71.92	64.63
20	71.79	64.6

(d) **Number of NNs extracted.**

Table 4: **All4One ablation experiments.** Evaluated on CIFAR-100 for linear and k-NN classification.

from the curse of dimensionality, which decreases its efficiency when high dimensional embeddings are used.

Augmentation Dependency. SSL frameworks depend heavily on the augmentations used by the Pretext task generator to create hard positive samples. However, neighbour contrastive approaches [14, 21] empirically proved that they are naturally more robust to augmentation removals. NNCLR [14] proposes the removal of all the augmentations except random crop augmentation to compare NNCLR frameworks robustness with SimCLR [6] and BYOL [15] frameworks. We follow the same to check the robustness of All4One compared to NNCLR and Barlow Twins. As it can be seen in Table 4b, our approach proves to be more robust than Barlow Twins [38] and NNCLR [14] to augmentation removal. Intuitively, adding a term that uses multiple neighbours increases, even more, the richness of the loss function and makes the augmentations less relevant.

Number of Transformer Layers. We increase the transformer encoder complexity by increasing the number of encoder layers (Table 4c). We infer that more complex encoders decrease the overall performance of the model. In SSL paradigms, the main goal is to train an encoder or the backbone. We hypothesize that simplifying the transformer encoder encourages the backbone to produce more useful features, rather than letting the transformer encoder create them with their feed-forward layers, thus decreasing performance.

Number of Neighbours. We study the effect of the neighbours' number on All4One (Table 4d). Similar to previous works [14, 21], All4One is robust regarding the number of neighbours extracted, obtaining the best results when 5 of them are used. As we increase the number of neighbours, the contextual information obtained from the Self-Attention operations performed by the transformer encoder may get more sparse, decreasing slightly the model performance while also increasing the required computation time.

4.4. Discussions

Dimensionality of the Embeddings. Feature contrast approaches such as Barlow Twins [38] state high dimensional embeddings as a requirement for their frameworks. However, we prove that combining the basic ideas behind feature contrast with another instance discrimination strategy drastically reduces this dependency. Usually, different SSL approaches tend to be applied individually and obtain improvements. We show that by combining and complementing these ideas could lead to higher improvements.

NN Retrieval Increase and Comparison. In neighbour contrast approaches, the number of times the KNN operator retrieves a neighbour from the same semantic class as the query (NN retrieval accuracy) has been defined as critical. However, we prove that increasing this accuracy does

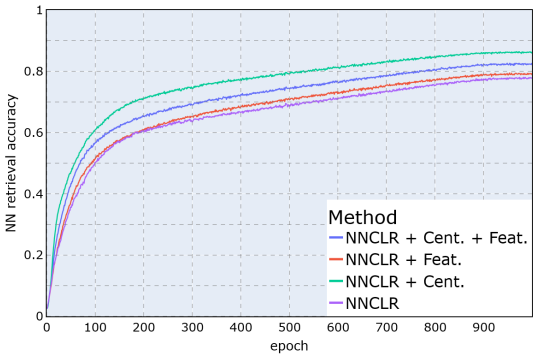


Figure 4: **Top-1 NN retrieval accuracy comparison.**

Figure 5: **NN extractions performed by All4One.**

not imperatively lead to a high overall accuracy increase. In fact, the best-performing version of All4One is not the one that obtains the highest NN retrieval accuracy. Hyperparameters such as the Support Set size or even the pretext task defined affect directly this accuracy. In Figure 4, we show how adding different pretext tasks affects the original NNCLR [14] method regarding the NN retrieval accuracy. As can be seen, adding the feature contrast strategy to vanilla NNCLR slightly boosts this performance. Also, combining NNCLR with a centroid pretext task provides a NN retrieval accuracy of 86%. However, when the three of them are combined, the NN retrieval accuracy does not surpass the 86% mark, even if it is the best overall performing version of All4One. This shows that the NN retrieval accuracy is not as critical as it is stated. In fact, as can be seen in Figure 5, neighbour contrast frameworks aim to bring together images that are similar so, even if the retrieved neighbour does not belong to the same semantic class, bringing it together with a very similar image on the feature space would not bring down the overall performance of the model. More NN extractions can be found in Section 5 of supplementary material.

4.5. Limitations

Even though All4One produced promising results, we identify some limitations.

Computation Efficiency. All4One, due to the introduction of three different objective functions, is more efficient than NN approaches that use multiple neighbours, but less efficient than NNCLR [14], which only contrasts a single neighbour. This is a limitation on low computation constraints. We provide a computation complexity analysis in the supplementary material.

Increased Number of Hyperparameters. The final All4One objective function introduces 3 additional hyperparameters to tune. Also, the transformer encoder uses a different learning rate, which also adds an extra hyperparameter to the overall framework.

Transformer Encoder Parameters. Several advances are available in terms of training transformers. The stability of the transformer encoder when using different settings compared to that of the other components is not known.

5. Conclusions

We propose a symbiotic approach that leverages NN contrastive learning by contrasting contextual information from multiple neighbours in an efficient way via self-attention. Also, we integrate a feature contrast objective function beneficial to the overall framework. All4One proves to generalize better and provide richer representations, outperforming previous SoTA contrastive approaches thanks to the integration of its different objectives. This highlights its exceptional performance in low data regimes, low dimensionality scenarios and weak augmentation settings. In the future, we plan to extend All4One to more complex backbones and investigate its application in diverse downstream tasks such as Object Detection and Instance Segmentation.

Acknowledgements

This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00), PID2022-141566NB-I00 (AEI-MICINN), and CERCA Programme / Generalitat de Catalunya. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain.

References

[1] Srikar Appalaraju, Yi Zhu, Yusheng Xie, and István Fehérvári. Towards Good Practices in Self-supervised Representation Learning, Dec. 2020. arXiv:2012.00868 [cs]. 6

[2] Nil Ballús, Bhalaji Nagarajan, and Petia Radeva. Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition. In Armando J. Pinho, Petia Georgieva, Luís F. Teixeira, and Joan Andreu Sánchez, editors, Pattern Recognition and Image Analysis, Lecture Notes in Computer Science,

pages 655–666, Cham, 2022. Springer International Publishing. 3, 5, 7

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254, 2021. 3

[4] H. Barlow. Redundancy reduction revisited. Network: Computation in Neural Systems, 12(3):241–253, Jan. 2001. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/net.12.3.241.253. 3

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In Advances in Neural Information Processing Systems, volume 33, pages 9912–9924. Curran Associates, Inc., 2020. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the 37th International Conference on Machine Learning, pages 1597–1607. PMLR, Nov. 2020. ISSN: 2640-3498. 2, 3, 6, 8

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. In Advances in Neural Information Processing Systems, volume 33, pages 22243–22255. Curran Associates, Inc., 2020. 3

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning, Mar. 2020. arXiv:2003.04297 [cs]. 3

[9] Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. pages 15750–15758, 2021. 3

[10] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 539–546 vol. 1, June 2005. ISSN: 1063-6919. 2

[11] Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A Library of Self-supervised Methods for Visual Representation Learning. Journal of Machine Learning Research, 23(56):1–6, 2022. 6

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, June 2009. ISSN: 1063-6919. 6

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 5

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a Little Help From My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations. pages 9588–9597, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In Advances in Neural Information Processing Systems, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 2, 3, 5, 8

[16] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond Self-Attention: External Attention Using Two Linear Layers for Visual Tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–13, 2022. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. 3

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16000–16009, 2022. 3

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. pages 9729–9738, 2020. 2, 3

[19] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Mar. 2015. arXiv:1502.03167 [cs]. 6

[20] Parneet Kaur, Karan Sikka, and Ajay Divakaran. Combining weakly and webly supervised learning for classifying food images. arXiv preprint arXiv:1712.08730, 2017. 7

[21] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean Shift for Self-Supervised Learning. pages 10326–10335, 2021. 2, 3, 4, 8

[22] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking Graph Transformers with Spectral Attention. In Advances in Neural Information Processing Systems, volume 34, pages 21618–21629. Curran Associates, Inc., 2021. 3

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012. 5

[25] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-Supervised Learning: Generative or Contrastive. IEEE Transactions on Knowledge and Data Engineering, 35(1):857–876, Jan. 2023. Conference Name: IEEE Transactions on Knowledge and Data Engineering. 1, 2

[26] Zhuang Ma and Michael Collins. Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency, Sept. 2018. arXiv:1809.01812 [cs, stat]. 2

[27] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting Self-Supervised Learning via Knowledge Transfer. pages 9359–9367, 2018. 3

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, Jan. 2019. arXiv:1807.03748 [cs, stat]. 2, 3, 4, 5

[29] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In 2012 IEEE conference on

computer vision and pattern recognition, pages 3498–3505. IEEE, 2012. 7

[30] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. Advances in neural information processing systems, 29, 2016. 2

[31] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding, Dec. 2020. arXiv:1906.05849 [cs]. 2

[32] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? arXiv preprint arXiv:2201.05119, 2022. 3

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, Dec. 2017. arXiv:1706.03762 [cs]. 3, 4, 6

[34] Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and Rong Jin. KVT: k-NN Attention for Boosting Vision Transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, Computer Vision – ECCV 2022, Lecture Notes in Computer Science, pages 285–302, Cham, 2022. Springer Nature Switzerland. 3

[35] Lemeng Wu, Xingchao Liu, and Qiang Liu. Centroid Transformers: Learning to Abstract with Attention, Mar. 2021. arXiv:2102.08606 [cs, stat]. 3

[36] Xinxing Xu, Joey Tianyi Zhou, IvorW Tsang, Zheng Qin, Rick Siow Mong Goh, and Yong Liu. Simple and efficient learning using privileged information. arXiv e-prints, pages arXiv–1604, 2016. 7

[37] Yang You, Igor Gitman, and Boris Ginsburg. Large Batch Training of Convolutional Networks, Sept. 2017. arXiv:1708.03888 [cs]. 6

[38] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In Proceedings of the 38th International Conference on Machine Learning, pages 12310–12320. PMLR, July 2021. ISSN: 2640-3498. 2, 3, 5, 7, 8

[39] Peisen Zhao, Lingxi Xie, Ya Zhang, and Qi Tian. Universal-to-specific framework for complex action recognition. IEEE Transactions on Multimedia, 23:3441–3453, 2020. 7

[40] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. arXiv preprint arXiv:2111.07832, 2021. 3

This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Good Fences Make Good Neighbours

Imanol González Estepa
Universitat de Barcelona,
Barcelona, Spain
igonzaes42@alumnes.ub.edu

Jesús M. Rodríguez-de-Vera
Universitat de Barcelona,
Barcelona, Spain
jmolinro33@alumnes.ub.edu

Bhalaji Nagarajan
Universitat de Barcelona,
Barcelona, Spain
bhalaji.nagarajan@ub.edu

Petia Radeva
Universitat de Barcelona,
Barcelona, Spain
Computer Vision Center,
Cerdanyola (Barcelona), Spain
petia.ivanova@ub.edu

Abstract

Neighbour contrastive learning enhances the common contrastive learning methods by introducing neighbour representations to the training of pretext tasks. These algorithms are highly dependent on the retrieved neighbours and therefore require careful neighbour extraction in order to avoid learning irrelevant representations. Potential "Bad" Neighbours in contrastive tasks introduce representations that are less informative and, consequently, hold back the capacity of the model making it less useful as a good prior. In this work, we present a simple yet effective neighbour contrastive SSL framework, called "Mending Neighbours" which identifies potential bad neighbours and replaces them with a novel augmented representation called "Bridge Points". The Bridge Points are generated in the latent space by interpolating the neighbour and query representations in a completely unsupervised way. We highlight that by careful selection and replacement of neighbours, the model learns better representations. Our proposed method outperforms the most popular neighbour contrastive approach, NNCLR, on three different benchmark datasets in the linear evaluation downstream task. Finally, we perform an in-depth three-fold analysis (quantitative, qualitative and ablation) to further support the importance of proper neighbour selection in contrastive learning algorithms.

1. Introduction

Deep Learning (DL) algorithms have made remarkable strides across a wide range of applications [13]. The success of DL can be attributed to larger architectures, powerful computation capabilities and more importantly, the availability of large training data [2]. Collecting large volumes of labelled data is often expensive, time-consuming, and very scarce in many domains [40]. Self-supervised Learning (SSL) is an alternative learning paradigm that enables models to learn meaningful representations by exploiting massive raw data without annotated supervision [16]. SSL models are label agnostic and learn representations that are generic across several tasks [1]. They capture the underlying relationships, structure or semantics of the data using a pretext task [37]. Downstream tasks based on the pretext trained models are therefore able to perform better on fine-tuning using task-specific labels [25, 32, 41]. Well-designed pretext tasks which learn proper representations rather than

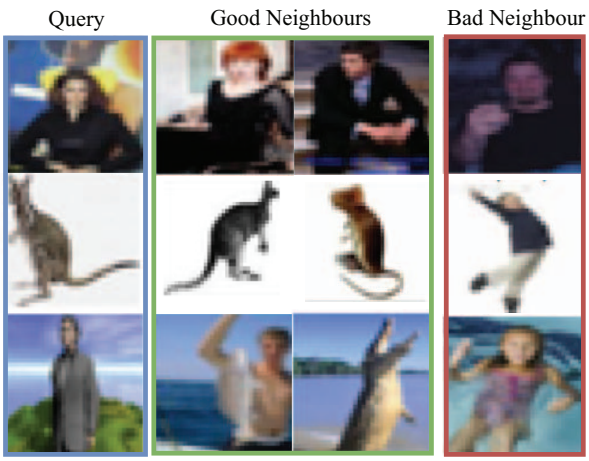


Figure 1. **Sample images** showing "good" and "bad" neighbours.

free-style learning would be better priors in various downstream tasks.

Pretext tasks can be classified in general into generative, contrastive or generative contrastive [29]. Generative models use an encoder-decoder architecture to reconstruct the sample [22, 23, 36]. Contrastive Learning (CL) algorithms, on the other hand, work on pulling together different augmentations (views) of the sample closer (positives) to each other while repelling those from other instances (negatives) [21]. CL algorithms use several similarity measurements such as NCE Loss [18], InfoNCE loss [33], and Redundancy-reduction loss [44] to contrast different views. SimCLR [8], a breakthrough SSL method used two views of the same image to learn the visual representations. MoCo [19] extended SimCLR by using a dynamic queue to store representations of views. Self-distillation methods such as BYOL [17], SimSiam [9], and DINO [6, 34] rely on different encoders to map the different views to each other. Other methods such as SWaV [5] and Barlow Twins [44] use correlation to infer relationships between views.

One of the fundamental design criteria in the CL algorithms is the generation of positive views from a given sample [1]. Data augmentation serves as a common approach to generate different diverse views from the sample image. SSL methods learn by contrasting the different views to learn representations that are invariant to these transformations [4]. However, there is a potential pitfall in solely using data augmentation to create different views. The augmentations alone would not be able to cover all variations of a given class [14].

Neighbour Contrastive Learning (NCL) algorithms are based on the notion that data augmentations (views) may not provide sufficient diverse information in selecting positive samples [14]. NCL algorithms contrast different views of the image with their nearest neighbours and learn to bring them in close proximity. This allows for better-learned representations as the contrasted pairs are often from different source samples. Algorithms such as NNCLR [14], MeanShift [24], All4One [15] are able to learn from new data points that would be different from those generated using views. SNCLR [10] used cross-attention to compute the importance of neighbours and used them as soft neighbours. A common entity in these methods is that they use a support set (queue) to store the representations of samples and use algorithms such as k-nearest neighbours [14] or mean shift [24] to retrieve one or few nearest neighbours, which in turn act as positive samples during CL.

One of the critical aspects for the proper functioning of these algorithms lies in the careful selection of neighbours [14]. Fig. 1 shows some examples of "good" and "bad" neighbours. "Good" neighbours are essential to learn proper representations of data distribution as they share similar features. Good representations possess local smoothness, sparse activation for specific inputs, temporal and spatial coherence, hierarchically organized explanatory factors, and simple dependencies [3]. "Good" neighbours do not need to be from the same semantic class, rather should produce representative features. "Bad" neighbours, on the other hand, may introduce noise or confusion in the representations that might lead to less effective representations. It is therefore crucial to identify good neighbours that can positively help SSL models to learn proper representations of the data. With this aim, we explore the question of *What constitutes a good neighbour?* We propose a neighbour correction framework that identifies potential "bad", *not so* helpful neighbours and uses the identified neighbour representations to generate new synthetic representations that are effective and also different from the representations created using different views of the samples.

The main contributions of our work are characterized as follows: (1) We present a neighbour correction framework through which we identify potential "bad" neighbours that can harm the pretext training process. (2) We introduce a mechanism to generate representation in the latent space, called "Bridge Points" from those identified neighbours such that they move closer to the instances in the latent space. (3) With a detailed analysis of the performance of our method on different benchmarks, we show the importance of neighbour selection in CL frameworks.

2. Related Works

In this section, we present an overview of the latest self-supervised visual representation learning literature that is relevant to our work.

Self-supervised Learning. SSL involves training a model without using any kind of supervised signal in an attempt to force it to learn intermediate representations that could be later transferred to multiple downstream tasks [1]. Existing SSL methods can be grouped into generative and discriminative algorithms [29]. While the former requires the use of visual transformers and reconstruction tasks, the latter has been able to maintain good results with a low budget thanks to their CL pretext tasks [46, 34]. CL works on grouping similar samples closer and moving diverse samples farther from each other [21]. In the context of learning image representations, the objective function relies on positive pairs, where both representations belong to the same semantic class and negative pairs, consisting of representations from different semantic classes. The goal is to bring the positive pairs closer together in the feature space, while simultaneously pushing away the negative pairs to avoid the collapse of the model. In recent years, this principle has been leveraged into several alternatives that work on clusters [5], using only positive samples [17] using neighbours as positives [14, 24, 10, 15]. Neighbour-based algorithms

are characterized by their enhanced generalization capacity inherited from the use of diverse neighbour representations obtained by algorithms such as k-NN.

Neighbour Contrast Approaches. Nearest neighbour (NN) is a simple and effective machine learning algorithm applied in several computer vision tasks [6, 35, 39]. NN-based SSL methods leverage the relationships between samples in the pretext training to enhance the quality of the learned representations. By exploiting the proximity or similarity between samples, these methods encourage the model to capture meaningful patterns, structures, or semantics from the data. NNCLR [14] was the first SSL method that explicitly adopted the NN approach. NNCLR implemented a memory queue, called a support set, to store the representations of samples and contrasted representations between views of samples and their first nearest neighbour mined from this support set. Mean Shift [24] used a mean-shift algorithm to group several neighbours together without contrasting them directly. SNCLR [10] used a cross-attention module to measure the correlation between samples and used this score to identify positive samples in CL. Recently, All4One [15] combined the neighbour contrast with feature contrast and transformer-based centroid contrast to learn representations from different latent spaces. The core idea in all the above-listed approaches is exploiting neighbours to learn relationships between samples. However, they do not control or measure the neighbours extracted, which could lead to a performance decrease. Our work differentiates from them by emphasizing the importance of a good neighbour selection and proposes useful replacements for the ones that should be discarded.

Feature Space Augmentations. Image data augmentations play a critical role in supervised learning [45, 42, 20, 38] and in most of CL-SSL approaches [8, 44, 17, 10, 15]. Creating two different samples from the exact same initial sample allowed unsupervised CL, as no labels are required for the correct selection of the contrasted samples [8]. Several pipelines have been proposed in order to enhance the augmented sample and their usefulness [17, 4]. All these augmentations are directly applied to the images so when augmentations are required for latent representations, it is not really effective. On the contrary, latent space augmentations can be perfectly applied with negligible computational efficiency loss. Adding random Gaussian noise, and extrapolating or interpolating feature space representations are the most common approaches to create new augmented representations [12, 7]. In recent years, these kinds of augmentations have been used to address diverse problems such as long-tailed instance segmentation [43], pose prediction [28] and multimodality [30]. However, the lack of visual control has made latent augmentations less popular than their counterpart. In our work, we propose a novel application of these latent augmentations in an NCL task in order to create interpolated representations. These interpolations, when contrasted, improve the capabilities of the trained model by enabling the model to capture more discriminative and meaningful patterns. This way, they provide meaningful replacements for neighbours where the extracted ones do not provide useful information for the CL task.

3. Rationale

NNCLR [14], which marked the inception of NCL algorithms, proved that changing from augmented representations increased the diversity of contrasted samples and, consequently, improved the performance of models on several downstream tasks. However, NNCLR also showed that a semi-supervised selection of neighbours achieved better results compared to an unsupervised selection. This highlights the fact that not all neighbours are completely useful. NCL algorithms often use a k-NN to extract the nearest neighbours of samples by computing the distances in latent space. These neighbours are later used in learning meaningful representations of the data. High-quality neighbour representations, therefore, directly impact the performance of the trained models [14]. Improving the quality of neighbour extractions in an unsupervised manner poses several challenges. Identifying what constitutes a *"good neighbour representation"* is not straightforward. All NCL models compute their neighbours in feature space, making their analysis more difficult. Additionally, there are no direct measures of quality between neighbour representations. Moreover, it is very challenging to differentiate an augmentation of a sample from its neighbour representation. Considering these complexities, we rise several important questions in this work: "*Can an image augmentation be a neighbour?*", "*How do we measure the quality of a neighbour in CL?*", "*How do we identify potential bad neighbours in feature space?*" and, most importantly, "*What does constitute a good neighbour?*". In this work, we hypothesize that good neighbours are those that are different from data augmentations, but in close proximity to the samples, whereas bad neighbours are those that are farthest from the augmentations. Based on this hypothesis, we provide "Mending Neighbours", a method based on neighbour selection and replacement neighbour generation.

4. Mending Neighbours

The foundations of our proposed pipeline are established with inspirations from the NNCLR [14] framework. The proposed pipeline is shown in Fig. 2. It contains two *braches*, each composed by an encoder followed by an MLP projector, together defined as f. One of the branches has an additional MLP predictor. Each f transforms the input

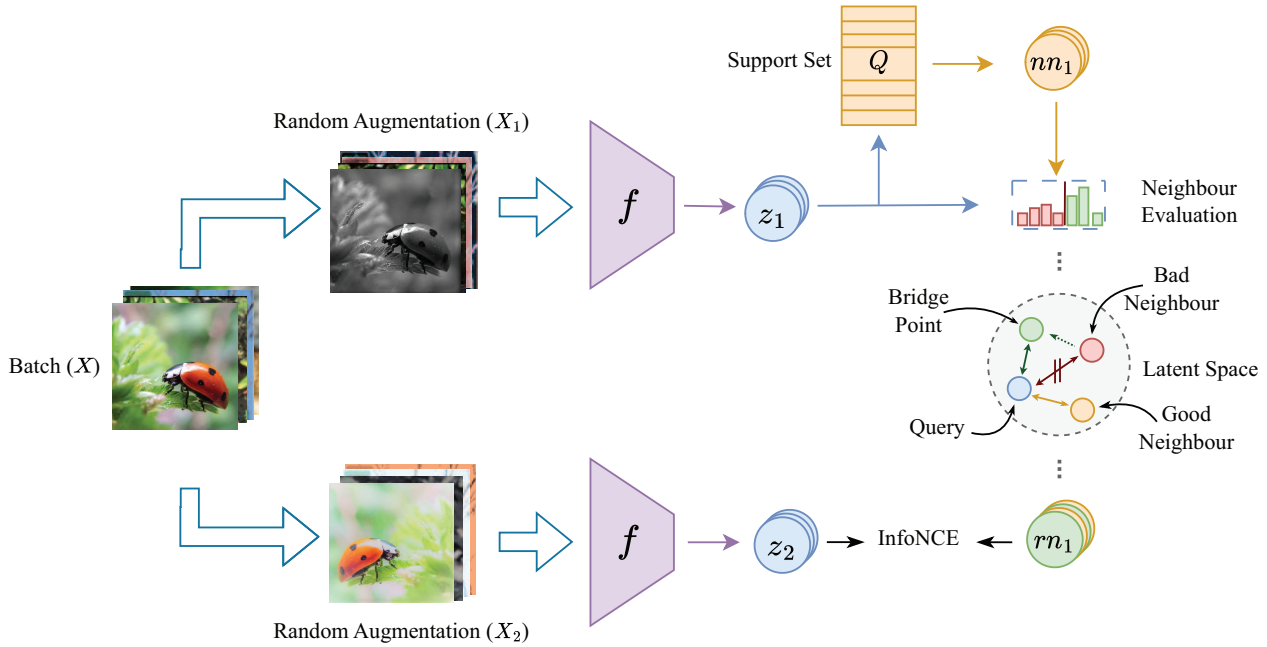


Figure 2. **Overview of our proposed method.**

image into an SSL representation. For a given mini-batch $\mathcal{X}$, we augment the samples twice, one for each branch to obtain the augmented batches $\mathcal{X}^1$ and $\mathcal{X}^2$. These batches are passed through their respective branches to obtain their respective representations $\mathcal{Z}^1$ and $\mathcal{Z}^2$. On every iteration, z_i is extracted from its respective representation batch and is used as a query for the k-NN algorithm that extracts its neighbour representation nn_i from a fixed-sized Support Set, $\mathcal{Q}$.

We extract the neighbours following NNCLR [14], which is defined as follows:

$$\mathcal{NN}(z_i, \mathcal{Q}) = argmin\ (Sim(z_i, \mathcal{Q})) \tag{1}$$

where $Sim(z_i, \mathcal{Q})$ is defined as $||z_i - \mathcal{Q}||_2$. Next, we use a simple neighbour evaluation approach to identify the "good" and the potential "bad" neighbours. We evaluate the goodness, gd_i of each neighbour by storing the similarity between the query sample and its nearest neighbour representation in the feature space. This is defined as:

$$\mathcal{G}(z_i, \mathcal{Q}) = min\ (Sim(z_i, \mathcal{Q})) \tag{2}$$

We use the mean gd_i of the whole batch as a threshold to split the neighbours into "good" and potential "bad" ones.

For the identified as "bad" neighbours, we present an unsupervised feature space interpolation between the "bad" neighbour nn_i and the query sample z_i. This interpolation allows us to create representations in the feature space that has the characteristics of both the query and the neighbours. We augment the potential "bad" neighbour directly in the feature space by creating an interpolation or Bridge Point (BP) bp_i between neighbour representation and its query. Though the identified neighbour can deteriorate the learning of the model, they still would contain representative information as they are the most similar in the Support Set. Formally, the interpolation is defined as:

$$bp_i = (z_i - nn_i) * \lambda + nn_i \tag{3}$$

λ is used to control the strength of the interpolation.

The final neighbour replacement function $\mathcal{R}$ is defined as follows:

$$\mathcal{R}(z_i, nn_i, bp_i) = \left\{ \begin{array}{ll} nn_i, & \text{if } gd_i > \frac{1}{n}\sum_{k=1}^{n} gd_k \\ bp_i, & otherwise \end{array} \right\} \tag{4}$$

This approach aims to detect the "bad" neighbours while also replacing them with representations created in the feature space between the query and the "bad" neighbours. The final loss is determined as:

$$\mathcal{L}_i = -log\left(\frac{exp(rn_i^1 \cdot z_i^2/\tau)}{\sum_{k=1}^{N} exp(rn_i^1 \cdot z_k^2/\tau)}\right) \tag{5}$$

where rn_i^1 represents the output of the defined replacement function $\mathcal{R}$ and τ is the temperature constant. The loss is computed symmetrically.

5. Validation

In this section, we first show the experimental settings of our proposed framework and then present our results high-

lighting the need to use "good" neighbours in NCL. We use three popular image classification benchmarks: CIFAR-10 [26], CIFAR-100 [26], and ImageNet-100, a reduced ImageNet of 100 classes [27] to validate our method. We compare our proposed method to the NCL SoA, specially to the benchmark NCL algorithm NNCLR [14].

5.1. Implementation Details

For all datasets, we use a ResNet-18 encoder in a self-supervised manner. We use solo-learn [11], a Pytorch-based SSL framework for all our implementations. Regarding the architecture, we follow the implementations of NNCLR [14] and use a common shared-weights dual encoder-projector architecture with a predictor at the end of the second branch. We create the projectors using 3 fully-connected layers of size [2048, 2048, 256] and the predictor using 2 fully-connected layers of size [4096, 256]. All fully-connected layers are followed by batch normalization. For all experiments, we initialize the backbones with solo-learn initialization parameters [11]. We follow the hyper-parameter settings as defined by solo-learn for all datasets except for the queue size of CIFAR experiments, where we increase it to 98304 following NNCLR [14]. We empirically set the interpolation hyperparameter λ to 0.2 for CIFAR10 and ImageNet100 datasets and 0.5 for CIFAR100. We train all the models using a single NVIDIA RTX 3090 GPU.

5.2. Results

We analyse the benefits of our proposed approach using linear evaluation on the three benchmark datasets, following common SSL evaluation schemes. We further present quantitative results based on neighbour retrieval and similarity metrics. We also show visual qualitative results highlighting the advantages of having "good" neighbours in NCL.

5.2.1 Linear Evaluation

For linear evaluation, we freeze the SSL-trained models and use them as backbones or feature extractors for a common linear classification task. Following the solo-learn pipeline [11], we perform the linear evaluation across all training epochs and report the best Top-1 accuracy. We present the linear evaluation results in Table 1. Our "Mending neighbour" approach outperforms NNCLR [14] on the three benchmarks showing the advantages of having a smarter selection and replacement of neighbours. Selection of good neighbours leads to better-learned models that act as a better prior in the linear classification task. One interesting point to note is that the datasets with a high number of classes show a bigger improvement in terms of performance. This could be due to the fact that the higher the number of classes, the easier for the k-NN to fail in retrieving a "good" neighbour as more confusing classes could be present in the support set. However, this is not the case for CIFAR-10, which has a total of 10 well-differentiated classes.

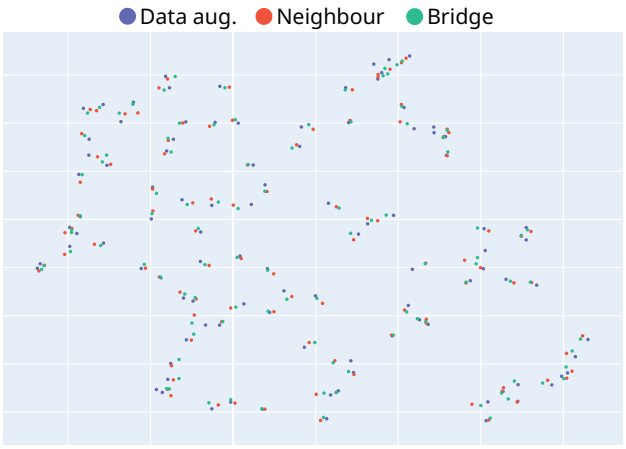


Figure 3. **UMAP visualization** of the best epoch (100 samples).

5.2.2 Quantitative Results

In addition to the linear evaluation, we also analyse the accuracy of the neighbours extracted for both NNCLR and the proposed approach using the k-NN accuracy. We show the k-NN accuracy for both CIFAR datasets in Table 2. This measures the number of times the extracted neighbour belongs to the same class as that of the query. As can be seen in Table 2, our approach increases the retrieval accuracy of the neighbours in both cases, implying that the generated bridge point representations of the encoder contain higher representative information than those obtained using the NNCLR neighbours.

We also measure the similarity or goodness of the extracted neighbours for both methods on CIFAR-100. The goodness is computed using the equation 2. Our approach is able to preserve the good neighbours while also providing good alternatives to the replaced ones. Consequently, our approach shows higher *goodness* than NNCLR on the non-replaced neighbours, while having a lower score on the replaced ones.

5.2.3 Qualitative Results

Bridge Point Analysis. We visualize 100 random samples of the best training epoch using UMAP [31] along with their respective neighbours and BP. In Fig. 3, one can see that in several Aug-BP-NN trios, the created BPs are located in the middle of the augmentation and nearest neighbours. This proves the effectiveness of our proposed approach to obtain representations that mostly are representative of both the query augmentation and the extracted neighbour. For the neighbours that are being replaced, new representations

Method	CIFAR-10	CIFAR-100	ImageNet-100
NNCLR [14]	92.13	69.19	79.80*
Ours	**92.25**	**70.77**	**80.10**

Table 1. **Linear evaluation results** showing Top-1 Test Accuracy. *- Results extracted from solo-learn [11].

	CIFAR10	CIFAR100
NNCLR	93.11	78.11
Ours	94.76	87.2

Table 2. k**-NN accuracy** showing neighbour retrieval accuracy.

	Replaced NN	Non-replaced NN
NNCLR	-	93.14
Ours	84.66	96.37

Table 3. ***Goodness*** between queries and extracted neighbours.

are created close to where good neighbours are supposed to be located.

In order to visualize our BPs, we implement a U-Net++ [47] based encoder-decoder architecture for an image reconstruction task. We initialize the encoder part of the U-Net with the weights of our pre-trained encoder and freeze it. Then, we train the decoder for a single epoch. We simulate a previously stored epoch of our pre-trained model by passing the same query and neighbour images through the encoder to obtain their representations and compute the BPs using Equation 4. Once the BPs are computed, we can simply pass them through the decoder to obtain their image visualization. We show the reconstructed queries, neighbours and BPs in Fig. 4. Most of the BPs resemble the original NN, but are enhanced with the characteristics of the query, making them contain information from both NN and queries. The created BPs combine properties such as colours from the query and the neighbour (first row), make mixed samples (second row), or remove portions of the query. Ultimately, we find the resemblance of the examples to the ones that could be obtained by common image augmentation techniques such as MixUp [45]. However, while those techniques augment the images by applying modifications to the pixels, our approach acts directly in the learnt feature space, which is more efficient and completely unsupervised.

Neighbour Selection and Replacement. The hypothesis of the existence of bad neighbours consequently implies the existence of neighbours that are good for the CL task and should not be replaced by BPs. We use the histogram of goodness values as in Fig. 5 of all extracted neighbours to analyze the goodness of neighbours. In Fig. 5, we show the histogram for the best epoch of pre-training, which represents the whole training set. As can be seen, while most

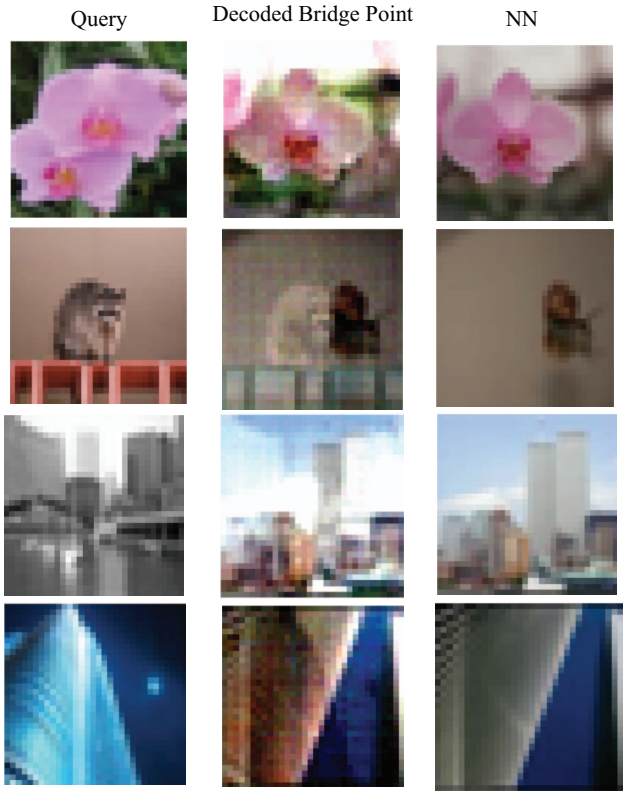


Figure 4. **Decoded BP visualizations** of Query, BP and NN using encoder-decoder image reconstruction.

of the extracted neighbours achieve a very high goodness value, there is still a considerable number of neighbours that possibly do not manage to be good enough for the task. However, deciding an exact threshold that divides good and bad neighbours, is a hard task when we take into account that these values vary during the whole training. At first, when the encoder is not well-trained, the generated representations do not provide the same richness as the ones generated on the final part of the training, making the goodness value fluctuate. For this reason, the selected threshold should be dynamic. In fact, this fluctuation also applies to the different batches that are computed during the training. As can be seen in Figure 6, the mean goodness value (marked in red) deviates depending on the batch. Given these observations, we find that the batch mean threshold is an effective alternative that is dynamic, and adaptative with respect to the training batches.

Finally, we show the effectiveness of the batch-mean

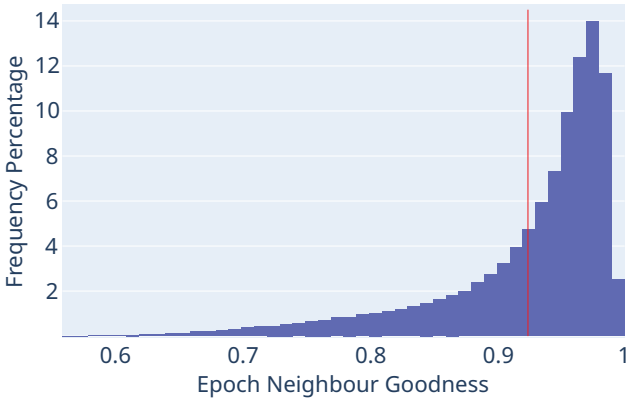


Figure 5. **Histogram of Goodness** for the complete best epoch. The threshold is shown using a red vertical line.

threshold using Figure 7. Overall, it can be observed that the replacements align with the notion of a bad neighbour. Most of the bad neighbours belong to a different class while still sharing some features with the original query. However, due to the vast variation of the augmentations used in the pretraining phase, our approach manage to also replace neighbours that could be considered good ones. By looking at their goodness score, we observe that these samples could possibly be very near to the threshold used for that batch. Positively, bridge points tend to share information from both augmentation and the neighbours, therefore the impact of not using the original good neighbour is decreased.

5.3. Ablation Study

We empirically analyze the four components of our approach by a careful ablation study: the representation used as a replacement, the origin of the bridge point used, the replacement strategy type and, finally, the used threshold. For each ablation experiment, we exclusively modify the component to analyse from our best experimental setup. All ablations are done on the CIFAR-100 dataset for a linear classification downstream task.

Replacement Representation Type. In this experiment, we analyse the effect of using different alternatives to the bridge point as a replacement representation. As a first alternative, we replace the bad neighbours with the original query augmentation. As shown in Table 4, this approach manages to outperform the baseline, proving once again the effect of bad neighbours on the model. However, this replacement does not provide any kind of diversity to the contrastive task, and therefore the effect would be the same as switching between NNCLR [14] and SimCLR [8] loss functions depending on the quality of the neighbour. As a second option, we add random Gaussian noise to the query augmentations before contrasting them. This increases the diversity, but it does not produce good evaluation results. The main idea of our proposed bridge points resides in the hypothesis of generating points that could have the potential to be good neighbours i. e. points with proper diversity that would be useful and not confusing to the model. This might not be obtained by the use of simple image augmentations or uncontrolled latent augmentations such as using Gaussian noise.

Bridge Point Type. In this analysis, we explore the effect of using augmentation as the second term in Eq. 4. This way, the bridge point would be based on the query augmentation instead of the neighbour (extrapolation). The bridge point based on the query augmentation does not provide good diversity and, in fact, performs worse than just using the augmentation. This is because the first term is meant to be added to the neighbour for a correct interpolation. Additionally, if we completely interpolate the augmentations instead of just changing the second term, we can observe an improvement. However, it is still less diverse than our proposed bridge point.

Replacement Strategy Type. To prove the effectiveness of our batch mean replacement, we ablate the replacement strategy by experimenting with two different alternatives. First, we show the effects of replacing all neighbours with bridge points. This improves the baseline, however, is held back by the fact that some neighbours do not require a replacement. Good neighbours, contrary to the bridge points, do not contain any explicit information from the query. For this reason, they provide useful information that is better and more diverse than the bridge point generated. In fact, just randomly replacing half of the neighbours with bridge points is enough to outperform the all-replace alternative. However, a random replacement is less stable compared to the proposed approach.

Threshold Type. We compare our batch mean threshold with a static threshold and a threshold based on the epoch mean. The batch mean threshold provides more dynamism than the epoch mean threshold or the static threshold. For the cases we explored, the additional dynamism of our selected threshold strategy keeps a better balance of the borderline samples than the epoch mean strategy. Depending on the batch, some higher goodness samples are replaced and some lower goodness samples are maintained, which proves to be beneficial for the model. On the contrary, the epoch mean threshold is more restrictive, which leads the model to lower performance. In the case of the static threshold, we do not find it suitable for this task, as it introduces an extra hyperparameter that is very hard to tune in a way that makes the strategy useful for the whole pretraining phase. Overall, our strategy empirically outperforms the other two

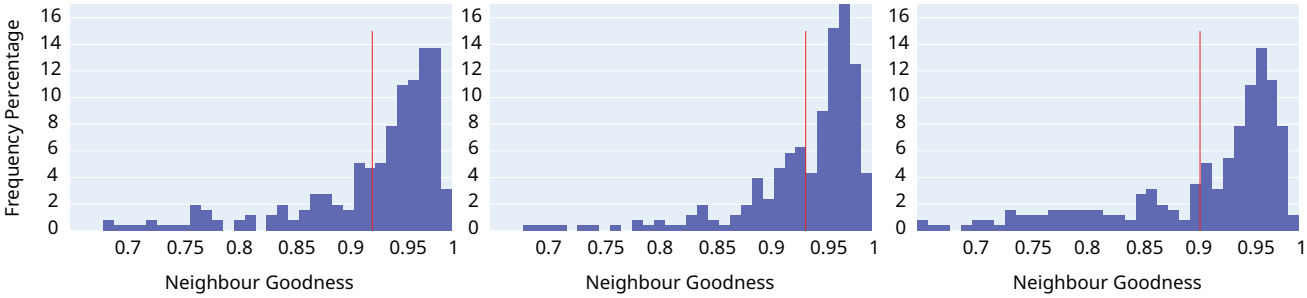


Figure 6. **Histogram of Goodness** for three different batches of the best epoch. The threshold is shown using a red vertical line.

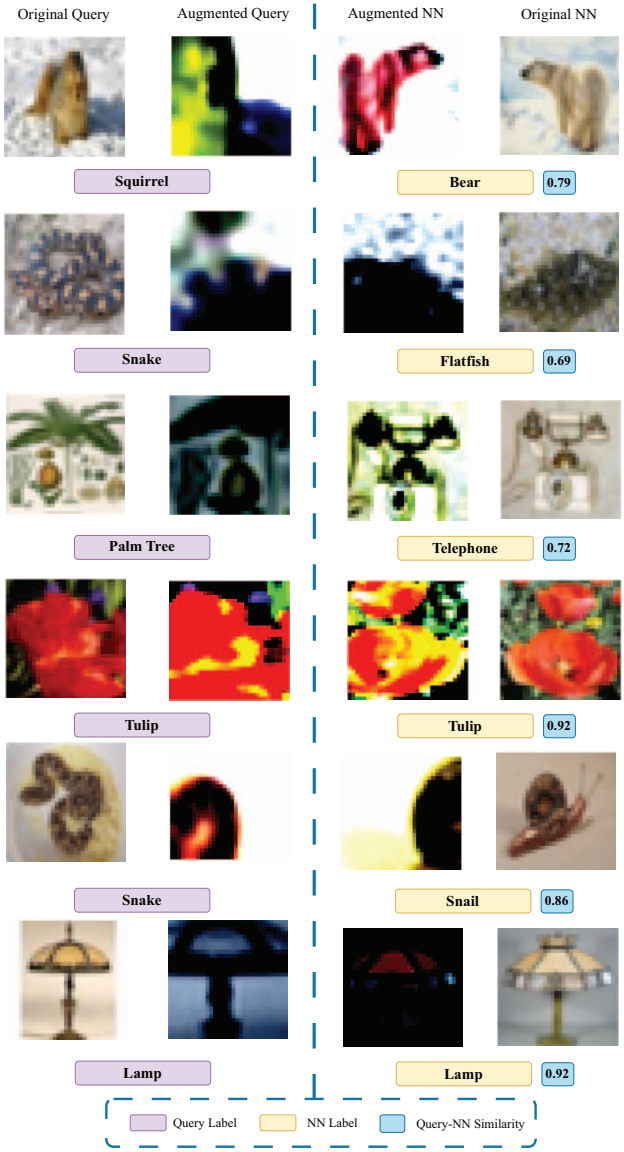


Figure 7. **Replaced NN Visualization.**

strategies both in performance and simplicity, as it does not require any further tuning.

Method	Top-1
NNCLR	69.19
Replacement Representation Type	
Data Augmentation	69.38
Noisy Data Augmentation	69.10
Bridge Point Type	
Data Augmentation Extrap.	68.85
Data Augmentation Interp.	70.03
Replacement Type	
Replace All Neighbours	69.51
Replace Random Neighbours	70.10
Threshold Type	
Epoch Mean Threshold	69.99
Static Threshold 0.8	69.62
Ours	**70.77**

Table 4. Ablation study.

5.4. Limitations

While the current study provides valuable insights for NCL, there are still some limitations in the current proposed scheme. We carefully elucidate the limitations that can serve as potential research directions.

Better Threshold Strategy. In the proposed Mending Neighbours approach, we use batch mean threshold due to its effectiveness and simplicity. However, it is difficult to address borderline neighbours where some good neighbours are also replaced. A better threshold strategy could avoid replacing these neighbours.

Measure of Goodness. Based on our initial hypothesis, we use cosine similarity as a Goodness metric. However, it is simple in terms of providing distance information. A more advanced metric could provide a better measure of goodness and help in better selecting bad neighbours.

Entanglement of Features. We hypothesize that the common entanglement of the features generated by the encoder holds back the whole neighbour selection pipeline. The introduction of a disentanglement process similar to

the ones applied in generative algorithms could increase the independence of the features, making them more differentiable and easy to create improved bridge points.

6. Conclusions

In our work, we analyze the current NCL SoTA approaches and identify critical aspects that can affect the performance of NCL algorithms. We propose a novel neighbour correction framework, called "Mending Neighbours" that correctly identifies potential "bad neighbours" and replaces them with a bridge point, a novel representation created directly in the latent space using neighbours and queries. The generated bridge points are more useful than a "bad neighbour" in NCL algorithms and this provides important informative prior information for downstream tasks. We validated our method using different SSL benchmarks and metrics and highlighted our improvements over NNCLR, a popular benchmark NCL algorithm. With in-depth quantitative, qualitative and ablation analysis we showed a measure of neighbour quality and obtained a scheme to identify what constitutes a good neighbour. In future, we plan to generate good neighbours through advanced generative processes that could provide representations of higher quality.

Acknowledgements

This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00), PID2022-141566NB-I00 (AEI-MICINN), and CERCA Programme / Generalitat de Catalunya. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. The authors thankfully acknowledge the computer resources at FinisTerrae III and the technical support provided by the Galician Supercomputing Center (CESGA) (RES-IM-2023-2-0025).

References

[1] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023. 1, 2

[2] Ms Aayushi Bansal, Dr Rewa Sharma, and Dr Mamta Kathuria. A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (CSUR)*, 54(10s):1–29, 2022. 1

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[4] Florian Bordes, Randall Balestriero, and Pascal Vincent. Towards democratizing joint-embedding self-supervised learning. *arXiv preprint arXiv:2303.01986*, 2023. 2, 3

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 2

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2, 3

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3, 7

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 2

[10] GE Chongjian, Jiangliu Wang, Zhan Tong, Shoufa Chen, Yibing Song, and Ping Luo. Soft neighbors are positive supporters in contrastive visual representation learning. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 3

[11] Victor G Turrisi Da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. Solo-learn: A library of self-supervised methods for visual representation learning. *The Journal of Machine Learning Research*, 23(1):2521–2526, 2022. 5, 6

[12] Terrance DeVries and Graham W. Taylor. Dataset Augmentation in Feature Space. Feb. 2017. 3

[13] Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021. 1

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. 2, 3, 4, 5, 6, 7

[15] Imanol G Estepa, Ignacio Sarasúa, Bhalaji Nagarajan, and Petia Radeva. All4one: Symbiotic neighbour contrastive learning via self-attention and redundancy reduction. *arXiv preprint arXiv:2303.09417*, 2023. 2, 3

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 1

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach

to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2, 3

[18] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 2

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[20] Shaoli Huang, Xinchao Wang, and Dacheng Tao. Snapmix: Semantically proportional mixing for augmenting fine-grained data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1628–1636, 2021. 3

[21] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. 2

[22] Amina Kammoun, Rim Slama, Hedi Tabia, Tarek Ouni, and Mohmed Abid. Generative adversarial networks for face generation: A survey. *ACM Computing Surveys*, 55(5):1–37, 2022. 2

[23] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020. 2

[24] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10326–10335, 2021. 2, 3

[25] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, 2022. 1

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 5

[28] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9090–9098, 2018. 3

[29] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021. 2

[30] Zichang Liu, Zhiqiang Tang, Xingjian Shi, Aston Zhang, Mu Li, Anshumali Shrivastava, and Andrew Gordon Wilson. Learning multimodal data augmentation in feature space. *arXiv preprint arXiv:2212.14453*, 2022. 3

[31] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. 5

[32] Juliette Millet, Charlotte Caucheteux, Yves Boubenec, Alexandre Gramfort, Ewan Dunbar, Christophe Pallier, Jean-Remi King, et al. Toward a realistic model of speech processing in the brain with self-supervised learning. *Advances in Neural Information Processing Systems*, 35:33428–33443, 2022. 1

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[35] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018. 3

[36] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 2

[37] Ravid Shwartz-Ziv and Yann LeCun. To compress or not to compress–self-supervised learning and information theory: A review. *arXiv preprint arXiv:2304.09355*, 2023. 1

[38] Shashanka Venkataramanan, Ewa Kijak, Laurent Amsaleg, and Yannis Avrithis. Alignmixup: Improving representations by interpolating aligned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19174–19183, 2022. 3

[39] Bram Wallace and Bharath Hariharan. Extending and analyzing self-supervised learning across domains. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 717–734. Springer, 2020. 3

[40] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023. 1

[41] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 1

[42] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 3

[43] Yuhang Zang, Chen Huang, and Chen Change Loy. Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3457–3466, 2021. 3

[44] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 2, 3

[45] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3, 6

[46] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2

[47] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer, 2018. 6

CHAPTER 4

Quantizing Uncertainty

Introduction

Understanding how DL models arrive at their predictions is crucial for ensuring transparency and facilitating their application across diverse tasks. Uncertainty Quantification offers valuable insights into model behaviour, enabling us to assess not only what the model predicts, but also its confidence in those predictions. We can identify areas where the model is confident, and conversely, where predictions might be less reliable. This knowledge empowers us to make more informed decisions. Various strategies exist for quantifying uncertainty. Bayesian approaches, including Monte-Carlo Dropouts, are widely used.

This research delves into the role of uncertainty quantification in developing more robust and transparent DL models. We address the following key questions (Numbers in brackets represent the corresponding paper):

- *How can the inherent uncertainty within the model influence the prediction process? (#1, #2, #3)*
- *How the quantified uncertainty be effectively leveraged to enhance models? (#2, #3)*
- *How insights gained from uncertainty quantification be used to improve the learning process? (#2, #3)*

By investigating these core questions, we aim to unlock the full potential of uncertainty quantification for creating better, more transparent, and more adaptable DL models.

Works Published Within This Chapter

1. ***Bhalaji Nagarajan***, *Marc Bolaños, Eduardo Aguilar, and Petia Radeva. "Deep Ensemble-Based Hard Sample Mining for Food Recognition." Journal of Visual Communication and Image Representation, 2023.* ***(JCR: Q2, IF: 2.6).***
2. *Eduardo Aguilar,* ***Bhalaji Nagarajan****, Beatriz Remeseiro, and Petia Radeva. "Bayesian Deep Learning for Semantic Segmentation of Food Images." Computers and Electrical Engineering, 2022.* ***(JCR: Q1, IF: 7.7).***
3. *Eduardo Aguilar,* ***Bhalaji Nagarajan****, and Petia Radeva, "Uncertainty-Aware Selecting for an Ensemble of Deep Food Recognition Models." Computers in Biology and Medicine, 2022.* ***(JCR: Q2, IF: 4.3).***

Computers and Electrical Engineering 103 (2022) 108380

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Bayesian deep learning for semantic segmentation of food images ☆

Eduardo Aguilar [a,*], Bhalaji Nagarajan [b], Beatriz Remeseiro [c], Petia Radeva [b,d]

[a] Department of Computing and Systems Engineering, Catholic University of the North, Avenida Angamos 0610, Antofagasta, 1270709, Antofagasta, Chile
[b] Departament de Matematiques i Informatica, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, Barcelona, 08007, Barcelona, Spain
[c] Department of Computer Science, Universidad de Oviedo, Campus de Gijon s/n, Gijon, 33203, Asturias, Spain
[d] Computer Vision Center, Cerdanyola (Barcelona), Spain

ARTICLE INFO

Keywords:
Deep learning
Uncertainty quantification
Bayesian inference
Image segmentation
Food analysis

ABSTRACT

Deep learning has provided promising results in various applications; however, algorithms tend to be overconfident in their predictions, even though they may be entirely wrong. Particularly for critical applications, the model should provide answers only when it is very sure of them. This article presents a Bayesian version of two different state-of-the-art semantic segmentation methods to perform multi-class segmentation of foods and estimate the uncertainty about the given predictions. The proposed methods were evaluated on three public pixel-annotated food datasets. As a result, we can conclude that Bayesian methods improve the performance achieved by the baseline architectures and, in addition, provide information to improve decision-making. Furthermore, based on the extracted uncertainty map, we proposed three measures to rank the images according to the degree of noisy annotations they contained. Note that the top 135 images ranked by one of these measures include more than half of the worst-labeled food images.

1. Introduction

Administrative policies around the globe, ranging from local counties to the World Health Organization, have been centered around better health care for everyone. Health is one of the core targets in the 2030 Agenda for Sustainable Development and is identified to impact each of the other goals in one way or another [1]. As outlined by the new Industry 5.0 paradigm, human-centric solutions are the future of things to come [2]. Industry 5.0, formally charted on January 2021, focuses on providing resilient, sustainable solutions that can be integrated into the existing social and environmental priorities [3]. Digitization of healthcare goes hand-in-hand with Industrial progress. Artificial Intelligence (AI) and Big Data have enabled the creation of better healthcare solutions that are cost-effective and efficient. Global healthcare has moved from being a centralized *organization* to a more personalized *service*. Recent decision-making in many of these customized approaches involves assistive technologies such as computer vision and machine learning solutions. Technologies are closer to the common person as never before, and have helped in improving the living conditions.

Expeditious growth in deep learning and computer vision algorithms, along with the availability of large-scale food datasets, serve as pillars in the development of food systems. Paying attention to the multi-class food segmentation problem, we can find some interesting works in the literature. Aslan et al. [4] proposed a method for dietary monitoring that solves the semantic food

☆ This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Sunder Ali.

* Corresponding author.
E-mail address: eaguilar02@ucn.cl (E. Aguilar).

https://doi.org/10.1016/j.compeleceng.2022.108380
Received 29 June 2022; Received in revised form 31 August 2022; Accepted 2 September 2022
Available online 18 September 2022
0045-7906/© 2022 Elsevier Ltd. All rights reserved.

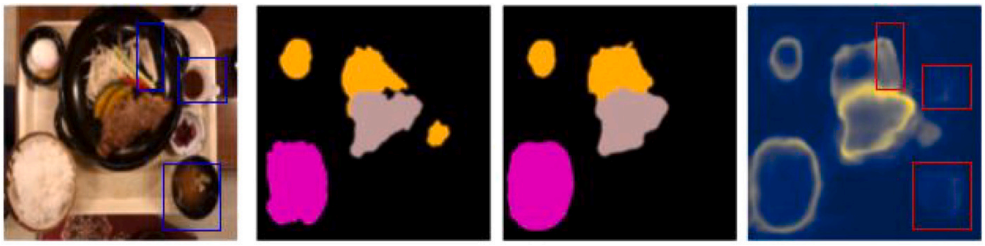


Fig. 1. Noisy label present in a sample image from the UECFOODPIX dataset. From left to right: original image, ground-truth (GT) labels, predicted segmentation mask, and the uncertainty map. The patches in the original image (blue boxes) and the uncertainty map (red boxes) represent the noisy label where uncertainty in the prediction is present. Best viewed in color.

segmentation task employing a DeepLabv2 pre-trained on the MSCOCO dataset and fine-tuned on the UNIMIB2016 dataset. They also explored the binary food segmentation problem by applying Dense Conditional Random Fields (Dense CRF), obtaining more refined boundaries and fewer false positives. Chiang et al. [5] developed a system to analyze the composition of food images, in terms of calories and nutrients, which uses a Mask Region-based Convolutional Network method (mask R-CNN) with ad-hoc post-processing. Additionally, they introduced their dataset for food recognition, known as Ville Cafe. Also in the context of monitoring diet and nutritional intake, Freitas et al. [6] built a dataset with Brazilian food and presented a comparative study with five segmentation models, including the aforementioned DeepLabv3+ and mask R-CNN. Another architecture used for semantic food segmentation is GourmetNet [7], which incorporates both channel and spatial attention through a multi-scale feature representation. The method achieved state-of-the-art performance on two datasets, UNIMIB2016 and UECFOODPIX. Wu et al. [8] provided FoodSeg103, a new dataset with 9490 images annotated with 154 ingredient classes and pixel-wise masks. They also proposed ReLeM, a multi-modality learning approach that was evaluated on their dataset and compared with three well-known semantic segmentation methods.

Semantic segmentation encounters various challenges with respect to data collection [9]. The quantity of data needed to train any deep learning algorithm is typically huge. This is even more important for segmentation algorithms, which need information about each and every pixel. Collecting ground truth pixel labels for such volumes of data is expensive. For instance, in the Cityscapes dataset [10], widely used in training self-driving cars, it took close to 90 min to annotate a single image. With less accurate labels, the performance of the models deteriorates. Recent advances in active learning frameworks [11] have helped to some extent, whilst self-supervised learning algorithms have resulted in more confident predictions by using better label-error maps [12]. However, both active learning and self-learning approaches require high-quality labels to learn the unlabeled data. The problem is that annotation efforts, both manual and automated, lead to noisy labels due to the complexity of the task [13]. The most important aspect of learning from noisy labels is to accurately characterize the uncertainty of the label noise [14]. Confident Learning (CL) algorithms work on the assumption that label noise is often class-conditional and it can be learned directly from the class labels [15]. MultiNET [16], an improved version of the original CL algorithm, uses aggregated outputs from multiple deep networks followed by a detection threshold to improve the noise detection rate. Using the CL algorithm in a teacher network allowed the teacher model to identify potential label noise and subsequently, by assigning soft-corrected masks, better students were created achieving more confident models [17].

Modern deep learning methods have reached real-world applications and, therefore, it is important for them to be *certain* about the predictions. In several critical applications, incorrect segmentation often has catastrophic results, such as cases involving self-driving cars [18]. Softmax probabilities were used as a measure of confidence. However, this is not always reliable [19] as seen in cases of adversarial examples. The Bayesian formulation is a popular way to estimate the model confidence in terms of uncertainty, which is treated as a measure of the trustworthiness of any deep learning algorithm. In fact, most of the semantic segmentation algorithms that implement uncertainty modeling are based on the Bayesian inference approach. Kendall et al. [20] presented Bayesian SegNet, the first probabilistic semantic segmentation approach using deep learning. The proposed method is an extension of SegNet, an encoder–decoder Neural Network (NN) architecture, to a Bayesian Convolutional Neural Network (CNN) that produces a probabilistic segmentation as output using MC-dropout [19]. Recently, Dechesne et al. [21] presented Bayesian U-Net, also based on MC-dropout [19] for uncertainty estimation but with the popular U-Net [22] for semantic segmentation. For their part, Mukhoti and Gal [23] proposed three metrics to evaluate Bayesian models designed for semantic segmentation, using also MC-dropout for experimentation purposes. All these works highlight the effectiveness of the Bayesian methods that provide accurate semantic segmentation and a reliable uncertainty map.

Food applications require a high level of certainty in their predictions since they are often used to make critical decisions such as what should constitute the next meal, what should be the quantity of certain food components, identify allergens of prepared foods, etc. However, to the best of our knowledge, there is no previous research work focused on uncertainty modeling applied to semantic food segmentation. Inspired by the good results obtained with the Bayesian methods described above, in this article we will extend the state-of-the-art semantic segmentation food models with a Bayesian approach. In addition, we delve into the analysis of quantified uncertainty to assess the benefit of the uncertainty map in identifying noisy labels present in the data. The experimentation carried out shows how error predictions can be identified with these algorithms and how we take advantage of uncertainty map information to identify images with noisy labels (see Fig. 1). There are two key contributions of this work. (1) We present the Bayesian version of two benchmark food segmentation methods: DeepLabv3+ and GourmetNet using MC-Dropout. Our

proposal provides both segmentation results and uncertainty measures related to each prediction. (2) We analyzed the uncertainty maps to identify possible wrong or mislabeled data. The proposed methods were validated on several public food datasets, where superior performance was observed in all of them compared to the baseline architectures.

The rest of the paper is organized as follows. We explain the proposed method in Section 2. We detail the experimental setup in Section 3. We discuss the results in Section 4, followed by the conclusions in Section 5.

2. Bayesian semantic segmentation

In Bayesian learning, the posterior predictive distribution that we want to compute is given by:

$$p(y|x,\mathcal{D}) = \int p(y|x,w)p(w|\mathcal{D})dw,$$

where $\mathcal{D}$ denotes the training set, x the input image, y the output class label, w the weights of the neural network, $p(y|x,w)$ the likelihood (e.g., the softmax output of the CNN), and $p(w|\mathcal{D})$ the posterior distribution.

In particular, for deep learning methods, a full Bayesian inference is intractable. However, instead of calculating the exact posterior distribution, it can be approximated. Different methods have been proposed to approximate a Bayesian inference, including MC-dropout [19] that is the most popular strategy used in semantic segmentation [20,21,23]. MC-dropout can be interpreted as a variational inference technique where a Bernoulli distribution is placed on the weights (w) of the neural network. In variational inference, the real posterior distribution is approximated by minimizing its Kullback–Leibler (KL) divergence with respect to a variational distribution $q(w)$.

Interestingly, Gal and Ghahramani [19] showed that using a dropout layer with a dropout rate of p on a hidden layer has the same effect as placing a Bernoulli distribution with parameter p on the weights of that layer. In addition, they also found that minimizing the negative logarithm of the likelihood with the standard optimization algorithm provides an effect equivalent to minimizing the KL divergence. Thus, with MC-dropout (see Algorithm 1), Bayesian inference can be performed using the traditional training procedure by simply putting a dropout layer after each training layer (e.g., convolutional layer). Then, one can sample the results provided by the posterior distribution during the prediction phase by performing T forward passes for the same input data while keeping the dropout layer active. Finally, with the mean of the sampled results, the final prediction is calculated and the uncertainty is estimated with entropy and mutual information.

Algorithm 1: Bayesian approximation using MC-dropout for image segmentation

input: Target image I, Trained model with dropout layers M, T forward passes;
output: Semantic segmentation SS, Entropy map E, Mutual information map MI;
$acc_prob \leftarrow 0$;
$acc_prob_log_prob \leftarrow 0$;
$t \leftarrow 0$;
while *t is less than T* **do**
  // Model prediction for input image using active dropout layers;
  $logits \leftarrow M.pred(I)$;
  $prob \leftarrow softmax(logits)$;
  $prob_log_prob \leftarrow prob \times log(prob)$;
  $acc_prob \leftarrow acc_prob + prob$;
  $acc_prob_log_prob \leftarrow acc_prob_log_prob + prob_log_prob$;
  $t \leftarrow t + 1$;
end
$mean_prob \leftarrow \frac{1}{T} \times acc_prob$;
// $mean_prob$ and $acc_prob_log_pro \in \mathbb{R}^{width \times height \times nclasses}$;
$E \leftarrow -sum(mean_prob \times log(mean_prob), axis = -1)$;
$MI \leftarrow E + \frac{1}{T} \times sum(acc_prob_log_prob, axis = -1)$;
$SS \leftarrow argmax(mean_prob, axis = -1)$;
return SS, E, MI;

In order to perform Bayesian inference with MC-dropout, some adjustments to the architecture of the models are required; in particular, the incorporation of dropout layers after trainable layers. For the latter, previous Bayesian semantic segmentation deep learning approaches differ in criteria regarding the numbers of dropout layers and where these layers are placed. For encoder–decoder based segmentation methods, the dropout layer has been placed after all [20,21] or some [20] convolutions and deconvolution blocks, or only in the middle flow of the network [23]. In [20], the authors noted in practice that using a dropout layer after each trainable layer was representing a too strong regularization, causing the network to learn very slowly. Alternatively, they explored a number of variants and found a good trade-off between accuracy and uncertainty quantification by placing dropout layers after four central encoders and decoders. More closely related to our work is the Bayesian semantic segmentation proposed in [23]. Here, the authors also used DeepLabv3+ as the base architecture, but with a different backbone, in this case the Xception network. They proposed to add the dropout layer only in the backbone, after every four Xception modules in the middle of the network, arguing that higher level features in the deeper layer are better modeled using probabilistic weights.

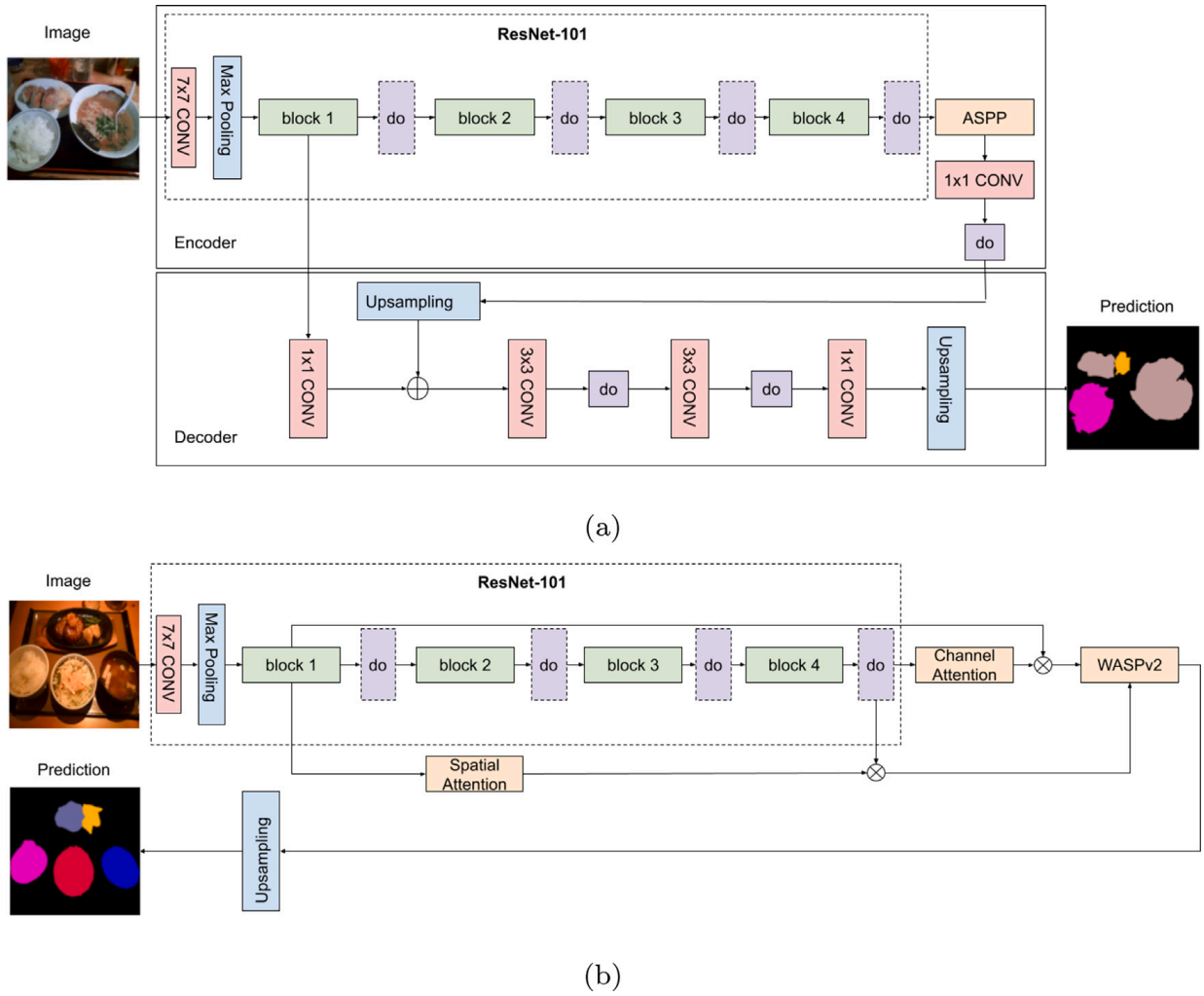


Fig. 2. BayesianDeepLabv3+ (a) and BayesianGourmetNet (b) with ResNet101 as backbone. Dropout (do) layers added to both networks are marked by dashed lines.

As we mentioned before, the best results in semantic segmentation of food images have been obtained with the DeepLabv3+ and GourmetNet architectures, both of them using ResNet101 [24] as the backbone. Inspired by the works discussed above, we propose to approximate a Bayesian inference in both networks by means of MC-dropout incorporating a total of four extra dropout layers placed after each residual block belonging to the backbone network (see Fig. 2). In this manner, it will be possible to obtain both the prediction and its corresponding uncertainty. A brief description of DeepLabv3+ and GourmetNet can be seen in the following subsections.

2.1. *DeepLabv3+*

DeepLabv3+ [25] is based on an encoder–decoder structure to perform the semantic image segmentation task (see Fig. 2(a)). On the encoder side, this architecture uses the original DeepLabv3 [26], which is composed of a backbone (e.g., Xception or ResNet101) for feature extraction and an Atrous Spatial Pyramid Pooling (ASPP) for capturing multi-scale information. Specifically, the encoding results correspond to the last feature map before the DeepLabv3 logits layer. On the decoder side, there is a simple, but effective approach to refine the segmentation mask by retrieving object segmentation details. First, the low-level features extracted from the backbone are convolved using a 1×1 convolution to reduce the number of channels, and then concatenated with the bilinear upsampled encoder output by a factor of 4. After that, some 3×3 convolutional layers are applied to the concatenation in order to refine the extracted features. Next, 1×1 convolutions are applied on top of the network where the number of convolutions equals the number of classes. Finally, a bilinear upsampling by a factor of 4 is used to provide the segmentation result.

2.2. *GourmetNet*

GourmetNet [7] is a segmentation architecture designed for food segmentation that incorporates a Waterfall Atrous Spatial Pooling (WASPv2) module to capture multi-scale features, coupled with dual attention modules to capture the context (see Fig. 2(b)).

Fig. 3. Image samples with their respective annotations for the UECFOODPIX (left group), UNIMIB2016 (middle group), and Food201 (right group) datasets. Notice that in UECFOODPIX there are two annotations that correspond to the GrabCut version (first) and the manual annotation version (second) of this dataset.

GourmetNet uses the ResNet101 as a backbone architecture and modifies the last block to learn multi-scale features. The authors employed atrous convolutions instead of regular convolutions. On the other hand, GourmetNet uses spatial attention to capture the low-level features and channel attention to capture high-level features. Both the low-level and high-level features are refined in the attention modules and fed into the WASPv2 module. The WASPv2 module uses a waterfall-like configuration of atrous convolutional layers to increase the field of view. Utilizing the multi-scale representations from both the attention modules and increasing the field of view led to better features. The final layer of the module is similar to an in-built decoder, which produces the segmentation results.

3. Experiments

This section describes the datasets used for experimental purposes. After that, the experimental setup is explained in detail. Finally, the measures used to evaluate the performance are shown.

3.1. Datasets

Three public food datasets were selected to perform semantic segmentation of food images by means of the proposed approach and baseline architectures. These are: (1) UNIMIB2016 [27], an Italian food dataset composed of 1027 food images and 73 food categories collected in a self-service canteen, in a semi-controlled environment and using a smartphone camera; (2) UECFOODPIX [28,29], a popular large-scale public dataset comprised primarily of Japanese foods consisting of 10,000 images and 102 food categories with annotations corresponding to plate masks automatically extracted from previously labeled bounding boxes (UECFOODPIX) or with pixel-wise annotation (UECFOODPIXComplete); and (3) Food201 [30], the first food dataset for semantic segmentation, but not widely used because annotations were not available until recently. The dataset is composed of a subset of images from Food101 [31], specifically 12,093 images which include pixel-wise annotations for 201 different foods that may be present in the images. Unlike UNIMIB2016, UECFOODPIX and Food201 contain images with noisy labels, different resolutions and large visual variations for the same foods. An example of the images contained in each dataset can be seen in Fig. 3.

3.2. Experimental setup

Two state-of-the-art semantic food segmentation methods were selected to perform the experiments, these are: DeepLabv3+ [25] and GourmetNet [7]. In both methods the ResNet101 [24] was used as the backbone. In addition, we adapt these methods to approximate Bayesian inference with the aim to extract and analyze the uncertainty in each given prediction in the segmentation of food images. For the latter, the backbone is adapted by adding a dropout layer with a probability of $p = 0.1$ after each residual block in order to estimate the uncertainty by means of the MC-dropout approach [19]. We named the Bayesian-based methods as BayesianDeepLabv3+ and BayesianGourmetNet.

For training we use the backbone pre-trained on ImageNet [32] and fine-tune the entire network for 100 epochs with a Cross-Entropy Loss and Stochastic Gradient Descent optimizer. For the optimization, we set a momentum of 0.9 and a weight decay of 0.0005. Regarding the learning rate (LR), it depends on the dataset and was set at 10 times higher for the weights of the layers located at the top of the backbone. Specifically, for UECFOODPIX and Food201 the base LR was 0.001 and for UNIMIB2016 it was 0.01. For DeepLabv3+-based models, the LR is decayed in a polynomial manner after each iteration with an exponential factor of 0.9. For GourmetNet-based models, the LR is decayed in a multi-step manner at epochs 40 and 70 with a factor of 0.3.

For the Bayesian semantic segmentation models, MC-dropout was used with $T = 100$ during the prediction phase to compute the uncertainty, where T corresponds to the multiple forward passes through the model performed with the dropout turned on.

With respect to the data, for comparison purposes, a 320×320 input size was used for UECFOODPIX and Food201, and an input size of 480×360 for UNIMIB2016. The original images were resized according to the nearest filter and no other transformation was applied to them.

All experiments were performed on a server with a graphic card of 11 GB of VRAM, using Pytorch as the deep learning framework.

3.3. Validation

All the experiments were evaluated quantitatively and qualitatively. In the quantitative case, we based the evaluation in three standard semantic segmentation metrics: Accuracy (Acc), mean Accuracy ($mAcc$) and mean Intersection over Union ($mIoU$). In addition, we use two metrics proposed for uncertainty quantification [23]: predictive entropy ($\hat{\mathbb{H}}$), which captures both aleatoric and epistemic uncertainty; and mutual information ($\hat{\mathbb{I}}$), which only captures the epistemic uncertainty. The metrics mentioned above are formally detailed below:

$$\hat{\mathbb{H}}(y|x, \mathcal{D}_{train}) = -\sum_{c} \left(\frac{1}{T}\sum_{t} p(y=c|x, \hat{w_t}) \log\left(\frac{1}{T}\sum_{t} p(y=c|x, \hat{w_t})\right)\right)$$

$$\hat{\mathbb{I}}(y, w|x, \mathcal{D}_{train}) = \hat{\mathbb{H}}(y|x, \mathcal{D}_{train}) + \frac{1}{T}\sum_{c,t} p(y=c|x, \hat{w_t}) \log(p(y=c|x, \hat{w_t}))$$

where $p(y=c|x, \hat{w_t})$ is the softmax probability of the input x being in class c, and $\hat{w_t}$ are the model weights for the forward pass t.

On the other hand, for the qualitative analysis, we will interpret the uncertainty estimated with the proposed approach with the intention of making a visual interpretation of the meaning of the epistemic and aleatoric uncertainty present in the food images. Additionally, we will assess whether uncertainty information is useful in automatically determining mislabeled or incorrect data.

4. Results

This section reports the results obtained in the experiments using the traditional semantic segmentation measures (Acc, $mAcc$, $mIoU$) and also the analysis of the uncertainty associated with the predictions.

4.1. Results based on standard measures for semantic food segmentation

To evaluate the results of the state-of-the-art models and the proposed methods in a comparative way, all of them were trained using the same hyperparameters, as explained in Section 3.2. Table 1 shows the results for the four datasets considered: UNIMIB2016, UECFOODPIX, UECFOODPIXComplete, and Food201. For all datasets, an improvement in the state-of-the-art models performance is evident when the dropout layers are incorporated into the backbone network (bb w/d) based on the proposed design. In particular, GourmetNet shows a bigger increase than DeepLabv3+ and surprisingly a great improvement is shown in UNIMIB2016, with about 4% more in terms of $mIoU$. The latter suggests that GourmetNet is more prone to strongly overfitting the network than DeepLabv3+ when training on a dataset with a small amount of data. Furthermore, it can be seen that the proposed Bayesian version of the state-of-the-art models not only allows us to obtain more information about the certainty of the segmented pixels, but also provides the best results regardless of the target dataset. BayesianGourmetNet shows better behavior with clean label datasets (UNIMIB2016 and UECFOODPIXComplete). On the contrary, BayesianDeepLabv3+ shows comparable or even better behavior than BayesianGourmetNet with more complex datasets (UECFOODPIX and Food201), which contain images of several food classes obtained in uncontrolled environments that are semi-automatically labeled and, therefore, may contain some degree of noise in the labels. On the other hand, when comparing the results in terms of Acc and $mAcc$, it can be seen that the correct segmentation is mostly balanced between classes in most of the datasets, with the exception of Food201, where there is a clear bias towards the classes with the highest occurrence in the images (e.g., background). Likewise, it is observed that GourmetNet-based models tend to provide a better balance in performance between classes.

Table 2 shows the class-wise results provided by GourmetNet-based models on the UECFOODPIXComplete and Food201 datasets, respectively. Specifically, it shows the three classes that obtain the best (Top 3) and worst (Bottom 3) results, the mean, standard deviation (Std Dev), and median. All these metrics were calculated in terms of IoU. In the UECFOODPIXComplete case, it can be seen that there is coincidence in the classes with better (classes id 13 and 14) and worse (class id 45) results, and in all of them our approach provides better performance. Classes id 13 (*croissant*) and 14 (*roll bread*) can be considered easy food classes because the color, shape and texture are almost constant in the samples that belong to these two classes. For this reason, the model is able to learn them very well. On the contrary, class 45 (*fried fish*) can be considered as a fine-grained food class because the visual appearance is very similar to other fried foods present in the dataset, thus increasing the complexity of its segmentation. Overall, there is a noticeable improvement (more than 3% in terms of mean IoU and about 4.5% in terms of median IoU) when BayesianGourmetNet is compared to GourmetNet.

In the Food201 case, it is also observed that the proposed method provides an improvement in the prediction of the best segmented classes. Again, the best results are obtained in easy food classes, for example in *edamame* (class id 76) and *macarons* (class id 124). It is interesting to note that, in this dataset, there are several classes that have not been learned, that is, with IoU equal to 0. Specifically, 71 classes were not learned by GourmetNet and 77 by BayesianGourmetNet models. Most of these classes correspond to sauces, drinks (e.g., class id 208: *chocolate*) and ingredients (e.g., class id 68: *croutons*, class id 42: *onions*, class id 143: *orange slice*) that are underrepresented, so the model does not have enough information to learn them. Although BayesianGourmetNet has more unlearned food classes than GourmetNet, this method improves the overall results by more than 1.5% in terms of mean IoU and about 2.5% in terms of median IoU. From these results it can be inferred that the model discards the learning of classes with almost 0 IoU in favor of the rest of the classes.

Table 1
Results for semantic food segmentation on four datasets. Best performances are shown in bold.

Dataset	Model	Acc	mAcc	mIoU
UNIMIB2016	BayesianDeepLabv3+	**0.9843**	**0.8271**	**0.7717**
	DeepLabv3+ (bb w/d)	0.9836	0.8243	0.7657
	DeepLabv3+	0.9822	0.8033	0.7469
	BayesianGourmetNet	**0.9861**	**0.8646**	**0.8076**
	GourmetNet (bb w/d)	0.9856	0.8606	0.8014
	GourmetNet	0.9829	0.8323	0.7685
UECFOODPIX	BayesianDeepLabv3+	**0.8408**	0.7012	**0.5907**
	DeepLabv3+ (bb w/d)	0.8375	**0.7020**	0.5810
	DeepLabv3+	0.8383	0.6961	0.5783
	BayesianGourmetNet	**0.8456**	**0.7196**	**0.5926**
	GourmetNet (bb w/d)	0.8419	0.7175	0.5867
	GourmetNet	0.8419	0.6947	0.5789
UECFOODPIXComplete	BayesianDeepLabv3+	**0.8729**	**0.7615**	**0.6421**
	DeepLabv3+ (bb w/d)	0.8664	0.7574	0.6311
	DeepLabv3+	0.8706	0.7571	0.6307
	BayesianGourmetNet	**0.8805**	**0.7817**	**0.6616**
	GourmetNet (bb w/d)	0.8767	0.7756	0.6506
	GourmetNet	0.8707	0.7587	0.6288
Food201	BayesianDeepLabv3+	**0.7696**	0.3296	**0.2520**
	DeepLabv3+ (bb w/d)	0.7659	**0.3341**	0.2464
	DeepLabv3+	0.7674	0.3290	0.2473
	BayesianGourmetNet	**0.7681**	**0.3366**	**0.2510**
	GourmetNet (bb w/d)	0.7629	0.3327	0.2449
	GourmetNet	0.7552	0.3155	0.2345

bb w/d = backbone with dropout layer after each residual block.

Table 2
Class-wise results in terms of IoU obtained by BayesianGourmetNet (BGNet) and GourmetNet (GNet) on the UECFOODPIXComplete (UFPC) and Food201 datasets.

Dataset	Model	Top 3 (class id)	Bottom 3 (class id)	Mean	Std Dev	Median
UFPC	BGNet	0.9566 (013) 0.9381 (014) 0.9330 (028)	0.1825 (045) 0.2298 (089) 0.2627 (090)	0.6616	0.1884	0.6794
	GNet	0.9565 (013) 0.9220 (000) 0.9182 (014)	0.1796 (045) 0.2657 (101) 0.2912 (056)	0.6288	0.1762	0.6348
Food201	BGNet	0.8829 (076) 0.8604 (124) 0.7997 (000)	0.0000 (208) 0.0000 (142) 0.0000 (143)	0.2510	0.2546	0.1920
	GNet	0.8806 (076) 0.8318 (124) 0.8268 (111)	0.0000 (208) 0.0000 (068) 0.0000 (142)	0.2345	0.2420	0.1677

4.2. Qualitative results based on the uncertainty map analysis

Creating pixel annotations for image segmentation is a time-consuming task. Common approaches start annotations with a semi-automatic technique (e.g., GrabCut) and then manually refine the labels to reduce the numbers of mislabeled or unlabeled data [29,30]. Arguably, reviewing all images manually to make corrections is also time consuming. Keeping in mind the fact that uncertainty modeling allows us to improve our understanding of what the model learns from the data, we propose to take advantage of uncertainty map analysis to discover the data that should be reviewed in order to prioritize or select for manual refinement. We propose to perform qualitative analysis of the uncertainty maps based on: (1) the segmented mask presented in the full image, (2) only considering the background mask, and (3) only using the foreground mask (any food).

4.2.1. Full image segmentation mask

The qualitative analysis was done using the full image segmentation mask in order to identify images annotated with noisy labels. In this case, the uncertainty present in the images was analyzed regardless of the nature of the noisy labels; that is, whether it occurs because the food categories are mislabeled or because the background images are labeled as some food category.

Let us consider $maxH = -ln(\frac{1}{C})$ the maximum $\hat{\mathbb{H}}$, where C corresponds to the number of classes. Then, the first evaluation measure (EM_1) proposed to prioritize the images in descending order is:

$$EM_1(x) = \sum_p ind(\hat{\mathbb{H}}(x_p) > maxH * \lambda)$$

where x_p corresponds to the pth pixel of the input image x, $ind(*)$ is an indicator function that returns 1 when the condition is true and 0 otherwise, and λ is a threshold to determine when a pixel has an uncertain prediction. In our case, we use $\lambda = 0.1$.

The proposed metric EM_1 was used in the UECFOODPIX dataset to order the training data and then select some of the first images to visually analyze the obtained results. In most cases, we were able to identify images with a large number of incorrectly labeled pixels, mainly images fully labeled as a background even though food was present there. An example of the images obtained with our metric can be seen in the first row of Fig. 4. As can be observed, the model is forced to learn the labels as a background. However, a high $\hat{\mathbb{H}}$ is present in the areas where food appears, while a low $\hat{\mathbb{I}}$ appears throughout the image. Therefore, as expected, a high aleatoric uncertainty in $\hat{\mathbb{A}}$ is obtained when noisy labels are present.

4.2.2. Foreground mask

In this case, instead of using the full image segmentation mask, the uncertainty was calculated in those pixels that belong to the foreground mask. The aim is to discover those images that have food pixels mislabeled with some other food category or background. The high uncertainty present in this region is expected to allow us to identify those pixels with noisy labels.

The second evaluation measure (EM_2) proposed to prioritize the images in descending order is:

$$EM_2(x) = EM_1(\hat{x}),$$

where $\hat{x}$ corresponds to a subset of the image x placed in the foreground of the GT mask provided by UECFOODPIX.

When this measure was applied to order the images, we found that most of the first images contain a very small food region labeled with a large uncertainty. An example can be seen in the second row of Fig. 4. In this case, even if the image is mislabeled, if we focus on the foreground region, the labels are correct. The high $\hat{\mathbb{H}}$ in this region is not caused by noisy labels, but we assume that it is caused by epistemic uncertainty; that is, the absence of more correctly labeled data with the same visual content.

4.2.3. Background mask

The qualitative analysis of the uncertainty map was also performed using only the background mask. In this case, it is expected that those images whose pixels are confused with some food category will be identified by analyzing the high uncertainty present in the pixels that belong to the background mask.

The third evaluation measure (EM_3) proposed to prioritize the images in descending order is:

$$EM_3(x) = EM_1(\check{x}),$$

where $\check{x}$ corresponds to a subset of the image x placed in the background of the GT mask provided by UECFOODPIX.

Similarly to the foreground mask measure, after ordering the images with this measure, we observed that the first selected images contain a small background region with high uncertainty. An example can be seen in the third row of Fig. 4, where it is observed that there is a high uncertainty in the region poorly labeled as background. However, this same image does not present high uncertainties in the regions where mislabeling between different foods occurs. The reason could be that the difference between the background and some food category is large. However, since the data is tagged at the dish level and not at the ingredient level, the model expects that the ingredient may be part of different foods depending on the context and, therefore, it is not able to compute correctly in some cases the certainty between different food categories.

4.3. Quantitative results based on the uncertainty map analysis

In addition to the qualitative analysis, we performed a quantitative analysis regarding the capabilities of the proposed measures, specifically EM_1, to order the images according to the degree of noise of the labels. For this purpose, we first computed the numbers of labels incorrectly annotated in the UECFOODPIX based on the annotations of UECFOODPIXComplete in all the training images. Next, we sorted the images in descending order considering the worst labeled first. Then, the intersection of the worst labeled images with respect to the images selected with the proposed measures was computed for the first 45, 90, 135, 180, and 225 images. Also, instead of using EM_1, the intersection was evaluated using Mean $\hat{\mathbb{H}}$ (mH) and Random Selection (RS). In the case of RS, we report the mean intersection achieved by 1000 random subsets. The results are reported in Table 3, where we show that our measure provides the highest degree of intersection. Specifically, more than 50% coincidence is evident when we analyze the first 135 images. These results support the effectiveness of our measure in finding images with noisy labels.

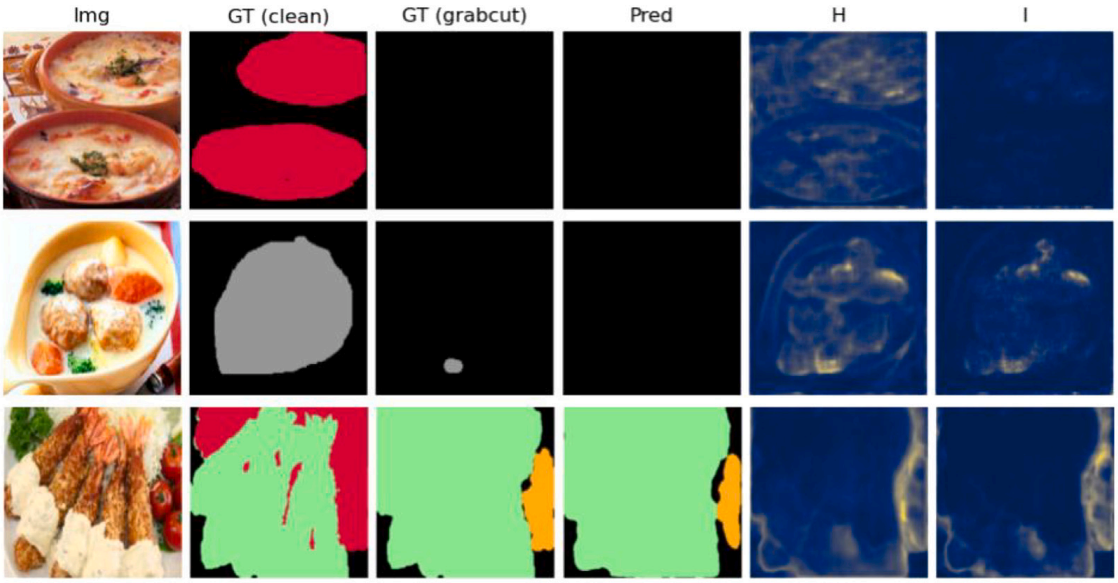


Fig. 4. Qualitative results obtained in the training set of UECFOODPIX. The first, second, and third rows represent images selected using the criteria EM_1, EM_2, and EM_3, respectively. Note that Img, GT (clean), GT (GrabCut), Pred, H, and I correspond to the target image, the clean labels provided by UECFOODPIXComplete, the GrabCut labels provided by UECFOODPIX, the predicted mask, the uncertainty measured with the entropy, and the uncertainty measured with the mutual information.

Table 3
Degree of intersection between a subset of the worst labeled images belonging to UECFOODPIX with respect to a subset of images selected by different criteria: Mean 𝕀 (mH), Mean EM_1 (mEM_1), and Random Selection (RS).

#Images	mH	mEM_1	RS
45	0.2444	0.3556	0.0051
90	0.3778	0.4556	0.0100
135	0.4593	0.5111	0.0150
180	0.4278	0.4444	0.0200
225	0.3733	0.3911	0.0249

4.4. Analysis of the successful and unsuccessful segmentation

Some successful and unsuccessful segmentation results on the test set of the three food datasets can be seen in Figs. 5 and 6, respectively. In general, across all datasets, the best results were achieved when the images contained few food instances and these instances correspond to food categories with few ingredients. Furthermore, we can see from Fig. 5 that the predictions contain a high degree of uncertainty in the contours of the food region (the boundary between the food and the bottom) where a small error in the segmentation is detected. In particular, we can notice that the method loses the details in the contour part providing a smoother prediction. In the Food201 case (third row of Fig. 5), a high uncertainty is obtained in the region of the food produced mainly by the shadow that is placed on it. Regarding the unsuccessful segmentation results (Fig. 6), we observe in UNIMIB2016 and UECFOODPixComplete that errors occur because the model incorrectly segments food with other food categories and not with the background. In theses cases, high uncertainty is captured in the wrong prediction regions. A more complex case is evidenced in Food201, where the ingredients are mixed on the plate and do not separate well. In addition, we can observe noisy labels in the GT. Unlike the results of other datasets, the output is completely incorrect and highly uncertain for the segmentation of this image. Although UECFOODPIX has nearly the same strong imbalance of food instances contained in each image as Food201 (see Fig. 7), the latter is a more challenging dataset due to the large number of food categories and the complexity of the annotations themselves, where we noticed significant differences in performance from the rest of the tested datasets. On the contrary, UNIMIB2016 can be considered a less complex dataset due to its acquisition in a semi-controlled environment, which allows to easily differentiate the background with respect to different foods. However, there is still work to be done to avoid segmentation errors between different food categories.

4.5. Comparison with the results of the state-of-the-art methods

Finally, Table 4 includes a comparison between our proposal and the state-of-the-art models in three food datasets. We note that, despite using a standard training procedure without data augmentation, the proposed Bayesian methods outperform the results previously obtained on UNIMIB2016 and UECFOODPIXComplete, and provide comparable performance in Food201. As for the

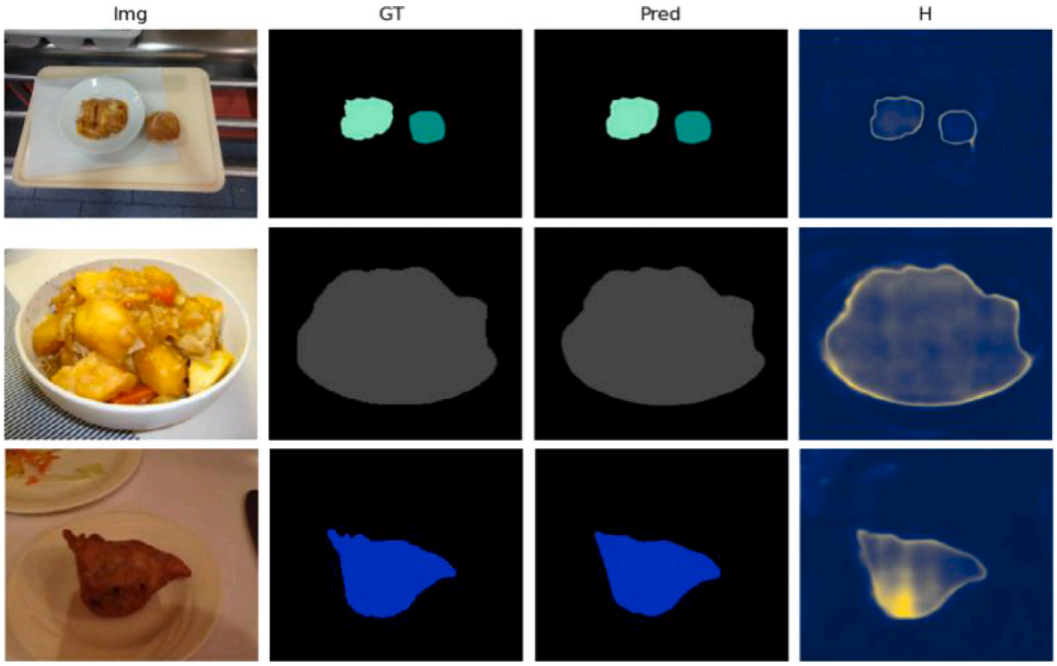


Fig. 5. Images in the test set for the UNIMIB2016 (top), UECFOODPIXComplete (middle), and Food201 (bottom) datasets with the highest performance.

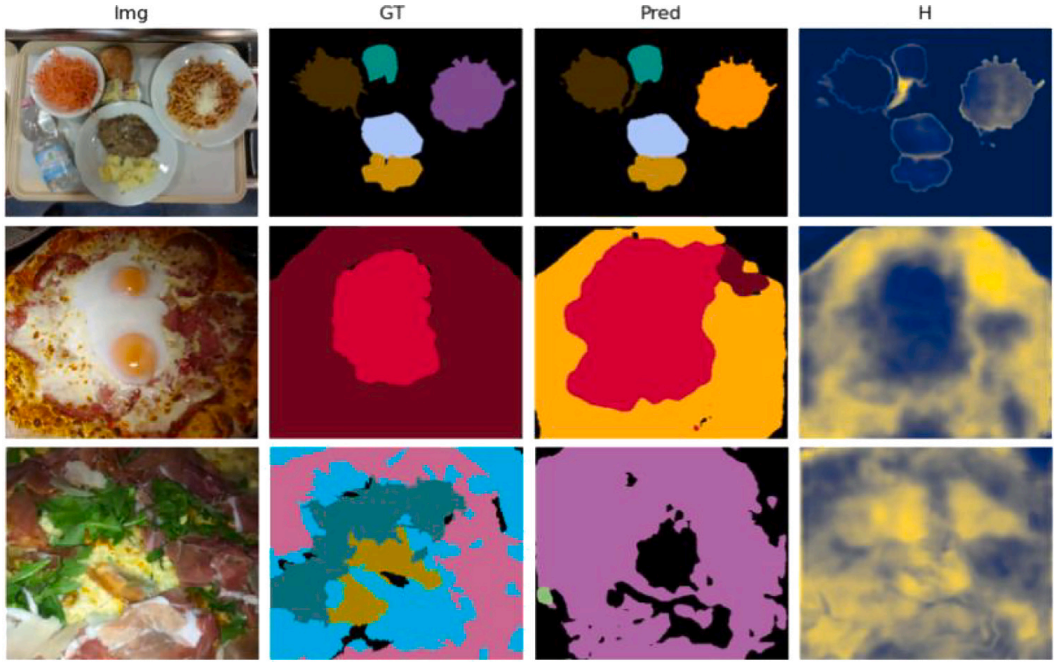


Fig. 6. Images in the test set for the UNIMIB2016 (top), UECFOODPIXComplete (middle), and Food201 (bottom) datasets with the lowest performance.

UECFOODPIXComplete dataset, we trained exactly the same model proposed by Sharma et al. [7] (GourmetNet), but achieved about 2.5% less $mIoU$ than results reported elsewhere. We believe that the performance difference is due to some hyperparameters used for training that are not properly reported in the article. Although the results of the base architecture were 2.5% lower than the reported ones, the proposed Bayesian version based on it outperformed the state-of-the-art by 1.0% in terms of $mIoU$. With respect to the Food201 dataset, Meyers et al. [30] proposed a more complex pipeline that requires training a multi-label classifier, extracting the context, and refining the CNN output before computing CRF. Also, the input data has 512 pixels for the maximum side, which is higher than the one used for our experiments. Despite this, our proposed end-to-end method provides results comparable to those reported in this work.

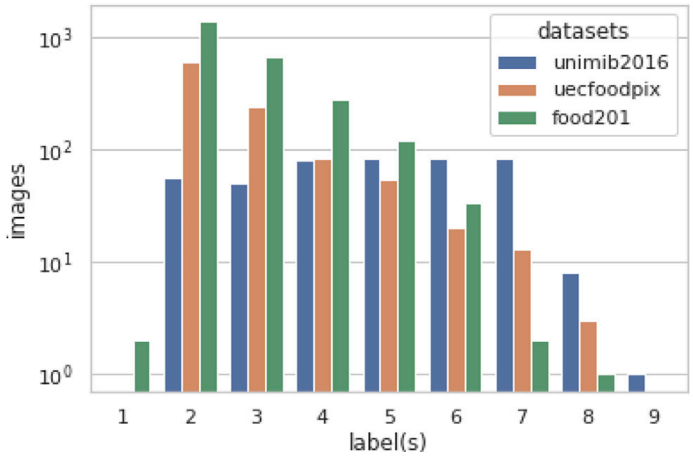


Fig. 7. Total images according to the number of pixels with different labels contained.

Table 4
Performance of semantic segmentation methods on three food datasets. Best performances are shown in bold.

Dataset	Method	Acc	mAcc	mIoU
UNIMIB2016	DeepLabv2 [4]	–	–	0.43
	SegNet [33]	–	–	0.44
	Sharma et al. [7]	–	–	0.72
	BayesianDeepLabv3+	0.98	0.83	0.77
	BayesianGourmetNet	**0.99**	**0.87**	**0.81**
UECFOODPIXComplete	Okamoto et al. [29]	–	0.67	0.56
	Sharma et al. [7]	–	–	0.65
	BayesianDeepLabv3+	0.87	0.76	0.64
	BayesianGourmetNet	**0.88**	**0.78**	**0.66**
Food201	Myers et al. [30]	0.76	0.33	**0.25**
	BayesianDeepLabv3+	**0.77**	0.33	**0.25**
	BayesianGourmetNet	**0.77**	**0.34**	**0.25**

5. Conclusions

We propose an adapted version of two state-of-the-art semantic segmentation methods used to segment food images in order to perform Bayesian inferences. Specifically, the backbone considered in both methods was adapted by incorporating a dropout layer after each residual block in order to approximate Bayesian inference using the MC-dropout technique. The resulting methods were named BayesianDeepLabv3+ and BayesianGourmetNet. Both of them outperformed the baseline results in terms of IoU on three public food datasets. Furthermore, these methods capture the uncertainty involved in the generated segmentation. The latter was useful to deepen the analysis of the uncertainty map by providing new measures to discover the incorrectly labeled images. The results of this analysis demonstrate the benefits of our approach both qualitatively and quantitatively, with special emphasis on food categories mislabeled as background. Finally, it is worth mentioning that this research work contributes to the generation of healthcare technologies, particularly in the improvement of food monitoring from images, which through human–machine collaboration facilitates the maintenance of a healthy diet and/or the prevention of health problems in those people who must limit monitor their diet due to allergies, intolerance or chronic diseases.

In future research, we plan to develop an algorithm that takes advantage of the proposed measures to select data labeled with noise and manage them during the training procedure minimizing the error incurred from this type of data.

CRediT authorship contribution statement

Eduardo Aguilar: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Visualization, Supervision. **Bhalaji Nagarajan:** Software, Investigation, Writing – original draft. **Beatriz Remeseiro:** Writing – review & editing, Visualization. **Petia Radeva:** Writing – review & editing, Supervision.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.compeleceng.2022.108380.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially funded by 20211005001-VRIDT-UCN, TIN2018-095232-B-C21, PID2019-109238GB-C21, Measurer EIT Digital, Logmeal4Shape, and CERCA Programme/Generalitat de Catalunya, Spain. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the valuable feedback of Prof. Ludmila Kuncheva on the research work. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

[1] Acharya S, Lin V, Dhingra N. The role of health in achieving the sustainable development goals. Bull World Health Organ 2018;96(9):591.
[2] Breque M, De Nul L, Petridis A. Industry 5.0: towards a sustainable, human-centric and resilient European industry. Luxembourg, LU: European Commission, Directorate-General for Research and Innovation; 2021.
[3] Maddikunta PKR, Pham Q-V, Prabadevi B, Deepa N, Dev K, Gadekallu TR, Ruby R, Liyanage M. Industry 5.0: A survey on enabling technologies and potential applications. J Ind Inf Integr 2022;26:100257.
[4] Aslan S, Ciocca G, Schettini R. Semantic food segmentation for automatic dietary monitoring. In: IEEE 8th international conference on consumer electronics-berlin. 2018, p. 1–6.
[5] Chiang M-L, Wu C-A, Feng J-K, Fang C-Y, Chen S-W. Food calorie and nutrition analysis system based on mask R-CNN. In: IEEE 5th international conference on computer and communications. 2019, p. 1721–8.
[6] Freitas CN, Cordeiro FR, Macario V. MyFood: A food segmentation and classification system to aid nutritional monitoring. In: 33rd SIBGRAPI conference on graphics, patterns and images. 2020, p. 234–9.
[7] Sharma U, Artacho B, Savakis A. Gourmetnet: Food segmentation using multi-scale waterfall features with spatial and channel attention. Sensors 2021;21(22):7504.
[8] Wu X, Fu X, Liu Y, Lim E-P, Hoi SC, Sun Q. A large-scale benchmark for food image segmentation. In: 29th ACM international conference on multimedia. 2021, p. 506–15.
[9] Guo Y, Liu Y, Georgiou T, Lew MS. A review of semantic segmentation using deep neural networks. Int J Multimedia Inf Retr 2018;7(2):87–93.
[10] Cordts M, Omran M, Ramos S, Scharwächter T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B. The cityscapes dataset. In: CVPR workshop on the future of datasets in vision, Vol. 2. 2015, p. 1–4.
[11] Siddiqui Y, Valentin J, Nießner M. Viewal: Active learning with viewpoint entropy for semantic segmentation. In: IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 9433–43.
[12] Chen H, Jin Y, Jin G, Zhu C, Chen E. Semisupervised semantic segmentation by improving prediction confidence. IEEE Trans Neural Netw Learn Syst 2021.
[13] Karimi D, Dou H, Warfield SK, Gholipour A. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Med Image Anal 2020;65:101759.
[14] Northcutt C, Jiang L, Chuang I. Confident learning: Estimating uncertainty in dataset labels. J Artificial Intelligence Res 2021;70:1373–411.
[15] Abad ZSH, Lee J. Detecting uncertainty of mortality prediction using confident learning. In: 43rd annual international conference of the IEEE engineering in medicine & biology society. 2021, p. 1719–22.
[16] Popowicz A, Radlak K, Lasota S, Szczepankiewicz K, Szczepankiewicz M. Combating label noise in image data using MultiNET flexible confident learning. Appl Sci 2022;12(14):6842.
[17] Zhang M, Gao J, Lyu Z, Zhao W, Wang Q, Ding W, Wang S, Li Z, Cui S. Characterizing label errors: confident learning for noisy-labeled image segmentation. In: International conference on medical image computing and computer-assisted intervention. 2020, p. 721–30.
[18] Rao Q, Frtunikj J. Deep learning for self-driving cars: Chances and challenges. In: 1st international workshop on software engineering for AI in autonomous systems. 2018, p. 35–8.
[19] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: International conference on machine learning. 2016, p. 1050–9.
[20] Kendall A, Badrinarayanan V, Cipolla R, Leap M. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In: British machine vision conference. 2017, p. 1–12.
[21] Dechesne C, Lassalle P, Lefèvre S. Bayesian U-Net: Estimating uncertainty in semantic segmentation of earth observation images. Remote Sens 2021;13(19):3836.
[22] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer assisted intervention. 2015, p. 234–41.
[23] Mukhoti J, Gal Y. Evaluating bayesian deep learning methods for semantic segmentation. 2018, arXiv preprint arXiv:1811.12709.
[24] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[25] Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European conference on computer vision. 2018, p. 801–18.
[26] Florian L-C, Adam SH. Rethinking atrous convolution for semantic image segmentation. In: IEEE/CVF conference on computer vision and pattern recognition. 2017, p. 1–14.
[27] Ciocca G, Napoletano P, Schettini R. Food recognition: a new dataset, experiments, and results. IEEE J Biomed Health Inf 2016;21(3):588–98.
[28] Ege T, Shimoda W, Yanai K. A new large-scale food image segmentation dataset and its application to food calorie estimation based on grains of rice. In: 5th international workshop on multimedia assisted dietary management. 2019, p. 82–7.
[29] Okamoto K, Yanai K. UEC-FoodPIX complete: A large-scale food image segmentation dataset. In: International conference on pattern recognition. 2021, p. 647–59.
[30] Meyers A, Johnston N, Rathod V, Korattikara A, Gorban A, Silberman N, Guadarrama S, Papandreou G, Huang J, Murphy KP. Im2Calories: towards an automated mobile vision food diary. In: IEEE international conference on computer vision. 2015, p. 1233–41.
[31] Bossard L, Guillaumin M, Gool LV. Food-101 — Mining discriminative components with random forests. In: European conference on computer vision. 2014, p. 446–61.
[32] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012;25.

[33] Aslan S, Ciocca G, Schettini R. Semantic segmentation of food images for automatic dietary monitoring. In: 26th signal processing and communications applications conference. 2018, p. 1–4.

Eduardo Aguilar is a Doctor in Mathematics and Computer Science from the University of Barcelona. He is currently an Assistant Professor in the Department of Computer and Systems Engineering at Universidad Católica del Norte. His main interest is in the research and application of robust Deep Learning algorithms for the semantic analysis of images.

Bhalaji Nagarajan is a Ph.D. student at the University of Barcelona. His Ph.D. thesis centers around sample analysis of deep learning (DL) training data and is focused on DL for food recognition. His research interests include uncertainty quantification, self-supervised learning, and learning from noisy labels.

Beatriz Remeseiro received her Ph.D. degree (2014) in Computer Science from the University of A Coruña. She is currently a tenured Associate Professor at the University of Oviedo and a board member of the Spanish Association for Artificial Intelligence (AEPIA). Her main research interests include computer vision and deep learning, mainly applied to real-world problems.

Petia Radeva is a Full Professor at the Universitat de Barcelona (UB), PI of the Consolidated Research Group "Computer Vision and Machine Learning at UB", IAPR Fellow, and Senior researcher in the Computer Vision Center. Associate editor of Pattern Recognition and Journal of Visual Communication and Image Representation, PI of European and national projects on Computer Vision food intake monitoring.

Computers in Biology and Medicine 146 (2022) 105645

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Uncertainty-aware selecting for an ensemble of deep food recognition models

Eduardo Aguilar [a,*], Bhalaji Nagarajan [b], Petia Radeva [b]

[a] Department of Computing and Systems Engineering, Catholic University of the North, Avenida Angamos 0610, Antofagasta, 1270709, Antofagasta, Chile
[b] Department of Mathematics and Computer Science, University of Barcelona, Gran Via de les Corts Catalanes 585, Barcelona, 08007, Barcelona, Spain

ARTICLE INFO

Keywords:
Food recognition deep learning ensemble learning model selection convolutional neural networks uncertainty estimation

ABSTRACT

Deep learning is a machine learning technique that has revolutionized the research community due to its impressive results on various real-life problems. Recently, ensembles of Convolutional Neural Networks (CNN) have proven to achieve high robustness and accuracy in numerous computer vision challenges. As expected, the more models we add to the ensemble, the better performance we can obtain, but, in contrast, more computer resources are needed. Hence, the importance of deciding how many models to use and which models to select from a pool of trained models is huge. From the latter, a common strategy in deep learning is to select the models randomly or according to the results on the validation set. However, in this way models are chosen based on individual performance ignoring how they are expected to work together. Alternatively, to ensure a better complement between models, an exhaustive search can be used by evaluating the performance of several ensemble models based on different numbers and combinations of trained models. Nevertheless, this may result in being high computationally expensive. Considering that epistemic uncertainty analysis has recently been successfully employed to understand model learning, we aim to analyze whether an uncertainty-aware epistemic method can help us decide which groups of CNN models may work best. The method was validated on several food datasets and with different CNN architectures. In most cases, our proposal outperforms the results by a statistically significant range with respect to the baseline techniques and is much less computationally expensive compared to the brute-force search.

1. Introduction

Healthy-eating has become one of the buzzwords in the modern society. Nutritional patterns [1] are a key factor in maintaining the well-being of individuals and of late, more and more health platforms [2] have become very popular and have been instrumental in maintaining the dietary intake of people. With the fast-paced development in the fields of computer vision and deep learning, image-based food journals have become the latest trend [3]. Automated food recognition is an integral component to these food journals [4]. However, food recognition is an onerous computer vision task, owing to the nature and complexity of food images [5]. The main factors that contribute to the complexity of food image analysis are the shape, volume, texture, color and composition of food classes that can be represented in numerous ways [6]. In addition to this, the background, the combination and the layout of food dishes make automated food recognition highly complicated. Also, food images suffer from high intra-class variance and low inter-class variance. Fig. 1 shows some sample images that highlight the complexity of food images often used in food recognition.

Convolutional Neural Networks (CNNs) allow to solve many complex computer vision tasks on par with human performance and surpass machine learning based approaches [7]. However, one of the important reasons for a well-trained deep learning model is the large volume of training data. The datasets should be able to provide sufficient training samples representing the wide variability of the task. Several public food recognition datasets [8–11] are available that can be used to train such models. However, these datasets are often from a constrained environment or represent only a particular cuisine. Food images are very common in social platforms such as Facebook and Instagram due to the increased number of posts at various eating places and also on various food recipe websites. However, a considerable effort is needed in making these data trainable and therefore there are a very few datasets having large number of food images [12,13] available to the research community. It has to be noted that available datasets cover still considerably

* Corresponding author.
E-mail address: eaguilar02@ucn.cl (E. Aguilar).

https://doi.org/10.1016/j.compbiomed.2022.105645
Received 27 January 2022; Received in revised form 4 May 2022; Accepted 14 May 2022
Available online 21 May 2022
0010-4825/© 2022 Elsevier Ltd. All rights reserved.

Fig. 1. Sample images showing the complexity of food.

a small percentage of the common food items across the world. A real-world food recognition model would be expected to work on data other than those that are represented by the training data, which sheds importance on the model that is used in the task.

Several popular CNN architectures, such as, AlexNet [14], GoogLeNet [15,16], Inception [17] and Residual Neural Network (ResNet) [18] have been directly adapted for food recognition. However, these direct adaptations provide limited results. With increased computational capabilities and more powerful architectures, ensemble of various models is a common approach [19–21]. Ensembling reduces the overfitting of models and also aids in improving the performance of food recognition [21]. When different models are used on the same task, it can be observed that the patterns of each architecture differ and therefore the results are not always overlapping [22]. This implies that ensembles provide complimentary information, a fact that when combined boosts the overall performance of the models.

In this work, we propose to use epistemic uncertainty [23] in selecting the ensembles in order to achieve higher performance with a previously limited number of base classifiers, subject to computational limitations in real-world solutions. The epistemic uncertainty analysis provides a good clue as to the confidence of the model with respect to its prediction, a factor that has not been fully exploited for the selection of base classifiers. Our contributions are the followings:

Table 1
A summary of deep learning-based work related to visual food analysis.

Reference	Year	Classification Method	Learning Approach	Tasks Addressed	Food Datasets
[16]	2016	CNN	Single-Task	Food Recognition	Food-101, UECFood100, UECFood256
[10]	2016	CNN	Multi-Task	Food and Ingredient Recognition	VIREO172
[20]	2017	Deep network combination (AlexNet, GoogLeNet, ResNet)	Ensemble	Food Recognition	Food-101, Indian Food Database
[21]	2017	Decision Template Scheme (ResNet, Inception)	Ensemble	Food Recognition	Food-11, Food-101
[28]	2018	Comparative of several classifier using feature extracted from ResNet and GoogLeNet	Single-Task	Food Recognition	Food-5K, Food-11, Food-101
[29]	2018	Wide-Slice Residual Network (WISeR)	Single-Task	Food Recognition	Food-101, UECFood100, UECFood256
[30]	2018	CleanNet	Single-Task	Food Recognition	Food-101
[36]	2018	CNN with Spatial Pyramid Pooling	Single-Task	Food Recognition	Food-101, UECFood256
[14]	2019	NutriNet (AlexNet-based CNN)	Single-Task	Food and Beverage Recognition	UNIMIB2016, Private dataset
[34]	2019	Ingredient-Guided Cascaded Multi-Attention Network (IG-CMAN)	Multi-Task	Food and Ingredient Recognition	Food-101, VIREO172, ISIA-Food200
[63]	2019	Regularized Uncertainty-based Multi-Task Learning (RUMTL). It was validated using ResNet50	Multi-Task	Food, Ingredient and Cuisine Style Recognition	MAFood121, VIREO172
[31]	2020	Class Incremental Extreme Learning Machine using features selected from those extracted with InceptionResNetV2	Single-Task	Food Recognition	Food-101, UECFood100, UECFood256, PFID, PAKISTAN FOOD
[62]	2020	Uncertainty-based Hybrid Classification. It was validated using ResNet50	Single-Task	Food Recognition	Food-101, MAFood121
[38]	2020	Joint-learning Distilled Network (JDNet)	Teacher-Student	Food Recognition	Food-101, UECFood256
[19]	2020	Voting-based fine-tuned CNNs (VGG, GoogLeNet, ResNet, Inception)	Ensemble	Food Recognition	Food-101, UECFood100, UECFood256
[35]	2021	Graph Convolutional Network (GCN)	Single-Task	Food Recognition	Food-101, UECFood256
[39]	2021	TL-Ensemble (Soft-Voting in fine-tuned InceptionV3 and DenseNet)	Ensemble	Food Recognition	MAFood121, Food24
[41]	2021	Snapshot Ensembling with MobileNetV3	Ensemble	Food Recognition	Food-101, UECFood100, UECFood256, Malaysian food

- We propose the first Ensemble Selection method to perform the visual food recognition task by mean of CNNs.
- We present a novel measure to perform the Ensemble Selection based on the analysis of the epistemic uncertainty.
- The advantage of the proposed approach was demonstrated in several datasets compared to the baseline approach showing improving of the accuracy meanwhile keeping reasonably fast final food classification.

The remainder of this manuscript is organized as follows: Previous related works are reviewed in the next section. The proposed technique is discussed in Section 3. In Section 4, we introduce the experimental setup and discuss the results obtained in each of the experiments. Finally, conclusions and future research are presented in the last section.

2. Related work

The literature related to the proposed method is briefly discussed covering three fields of research, namely: 1) Food Recognition, 2) Uncertainty Modeling and 3) Ensemble Selection. In addition, the closest visual food analysis works are summarized in Table 1 that indicates the classification method, the learning approach, the task to be solved and the dataset where the method was validated.

2.1. Food recognition

Automated food recognition algorithms have been crucial and central to several food computing tasks, such as in dietary assessment, food perception and food recommendation [4]. Classical machine learning approaches involve segmentation, feature extraction, feature selection or reduction, texture information and classification stages [8,24–26]. This approach was superseded by deep networks after the successful demonstration of their effectiveness in various application areas. Most existing literature on food recognition uses a standard CNN architecture on the food datasets by fine-tuning the model on food data [8,16,27]. Another approach is using the latent space from different layers of CNNs to train different classification algorithms [28]. All these techniques have been very effective on simpler food images, that is, in images acquired in a controlled environment and particularly for those images belonging to a class of foods that show little intra-class variability and little similarity between classes (e.g. edamame).

The recent research in food recognition focused on more advanced deep network architectures. Wide Residual Networks [29] were used to extract the common vertical food characteristics (e.g. vertical arrangement of ingredients in hamburgers) and they were used to improve the classification network. The architecture label cleaning network (CleanNet) [30] was used to handle noisy food data, present when data is automatically collected from the web without manual verification, which is a recurring problem in a complex task such as food recognition. NutriNet [14] was proposed to handle larger image resolutions (512×512) by modifying the standard AlexNet architecture to extract more knowledge from higher-resolution data. Class Incremental Extreme Learning Machines [31] was used in order to handle the open-ended and dynamic nature of the food datasets, which is reflected in an increasing number of food images and food classes that can be generated by creating new food recipes or by different ways of serving the same kind of food. Additional information such as GPS and restaurant information was combined to improve the classification performance [32,33]. Ingredients were also used as supportive information [34] to aid the classifiers in order to better learn food data. The semantic embeddings are found to be correlated with the image representations and this relation has been exploited using Graph Convolution Networks [35].

Another vertical in food recognition research is based on creating models that have less footprint as most of the real-world applications are in mobile platform. An optimized CNN model with spatial pyramid pooling layers [36] was used to reduce the architecture size without compensating on the performance. Network-In-Network [37] architectures are used in real time mobile food recognition apps due to their ease in adapting to the mobile platform and the trade-off they provide. Teacher-student learning is an approach carried out to decrease the full-sized networks. A Joint-Learning Distilled Network (JDNet) [38] used a joint learning framework where both the teacher and the student networks were trained simultaneously.

As it can be seen from the research directions on food recognition, it is evident that attaining higher accurate results is very mandatory. It is due to the direct impact on real-world applications. Ensemble of Deep Networks is a common technique used to improve the performance of algorithms and also to avoid over fitting of networks. FoodNet [20] used the latent space from AlexNet, GoogLeNet and ResNet architectures. Each of the networks was trained separately on food images and then the extracted features were fused together. Tasci in Ref. [19] further enhanced this idea by adding voting combinations for the output of the network architectures which boosted the performance further. Decision Templates [21] were used in fusing the features of ResNet and Inception architectures. TL_Ensemble [39] considered a soft-voting approach (average voting) to integrate the prediction of two different CNN architectures: Inception and Densely Connected Convolutional Networks (DenseNet) [40]. Recently, a method combining the snapshot ensembles approach with MobileNetV3 proved effective for analyzing food images on devices with limited resources [41].

2.2. Uncertainty modeling

There is no doubt that there is *a before* and *after* in image understanding solutions with the arrival of CNNs. The Big Data era together with an increase in computational capacities (e.g. GPU, processor and memory) are enabling factors that have allowed to create deep CNN solutions for various real-world problems. The industry has also been captivated by these solutions, increasingly incorporating them into its business. However, it is not *all rosy in the garden*. Although the Deep Learning algorithms have provided successful results in the experimental condition, when the solution is applied in real-world conditions, in some cases it does not behave as expected. Uncertainty modeling is a research field that tries to understand why models might or might not work. There are two categories of uncertainty [23]: a) aleatory, representing an unexpected behavior of the model produced by noise that could be present in the observation (e.g. noise in the measuring instrument) and b) epistemic, representing an unexpected behavior of the model produced by the lack of sufficient data to achieve a correct fit of the distribution and, therefore, a better generalization. The estimation of the uncertainty can be carried out using a Bayesian neural network (BNN) approach, but in the case of deep learning algorithms, which are characterized by having several parameters and using a large amount of data for their training, it becomes intractable [42–44]. Instead, a stochastic variational inference has been used frequently to capture uncertainty [42,43,45–47]. Of all of them, Monte Carlo (MC)-dropout is the simplest one, but not less efficient method to estimate the uncertainty that is equivalent to using Bernoulli to approximate a variational inference in the BNN [48].

Lately, novel uncertainty-based methods have been proposed to successfully solve a wide range of computer vision tasks. In image classification, uncertainty modeling has been effective in improving classification on unbalanced datasets [49], robustness against adversarial attacks [50] and robustness against noise in face data [51]. For image segmentation, it has been useful for domain adaption [52], weighting the predictions from a multi-view inputs from a 3D data [53], and to create a metric and annotations to solve the night-time semantic segmentation through a curriculum model adaptation from the correspondence of day-night images content [54]. For object detection, it has been applied to model bounding boxes [55,56] and improve the 3D pedestrian location [57]; as to many more (e.g. RGB-D saliency detection [58], facial landmark locations [59], multi-modal learning [60],

image processing [61], etc). Despite the growing interest in uncertainty-based models, we have no evidence that they have been used for food analysis other than ourselves.

In our previous works relating to uncertainty-aware techniques, we have introduced various strategies to improve the performance of food recognition models. Uncertainty-based hierarchical classifiers were used to consider the local classifier performance along with the flat classifier and it showed improved results on several food datasets [62]. Multi-task framework based on homoscedastic uncertainty was used to predict different food-related tasks [63]. Sample-level uncertainty was also used to improve the data augmentation that in turn offered improved performance [64]. In this work, we further investigate the role of uncertainty in selecting the ensemble of several models in order to improve the performance of the food recognition task.

2.3. Ensemble selection

The complex visual appearance of food images results in a real challenge for the computer vision community to analyze them. Recent performance achieved through deep learning algorithms is encouraging, but far from accurate for real-life food recognition problems where thousands of dishes need to be classified. Ensemble learning becomes mandatory to ensure more robust and accurate classification than individual models [65] to address this task. Most of the traditional Ensemble Learning methods can be mainly grouped into three different strategies: bagging [66], boosting [67] and stacking [68]. Bagging methods consist of combining the prediction of a committee of trained models with different subsets of the training data produced by sampling with replacement. Boosting methods involve incorporating one model at a time into the ensemble in order to reduce misclassification of previous models. And Stacking methods work by weighting the predictions given by the committee members through a machine learning algorithm (e.g. logistic regression) trained for that purpose. Particularly for the object recognition task, the common strategy adopted to form an ensemble is through a committee of neural networks, which consists simply of a collection of the same CNN models trained on the same data by varying the weights initialization [69–71]. With this approach, final predictions are calculated by averaging the output of all models.

In some real-world solutions, both accuracy and computational resources are important factors to consider when designing the classification method. Ensemble Learning allows us to improve the accuracy, but also increases the prediction time or the computational resource required (e.g. models running in parallel to perform recognition). Therefore, it is desirable to find a trade-off between the number of committee members with respect to the solution cost. Ensemble Selection (ES) [72,73], also known as ensemble pruning or selective ensemble, is a research field that deals with alleviating the problem of indistinctly considering all base classifiers (members) to build an ensemble.

There are two approaches to find a suitable subset of classifiers [74]: a) Static Ensemble Selection (SES), where all unseen data are classified with the ensemble selected in the training phase; and b) Dynamic Ensemble Selection (DES), which attempts to find the most competent classifiers for each unseen data. Regarding SES, one of the earliest methods in the literature corresponds to a forward step-wise selection approach to perform the ensemble selection, which consists of a greedy procedure iteratively adding models. Each next model tries to optimize the performance metric (e.g. accuracy, root-mean-squared-error, among other) on the validation set, in order to fill an initially empty ensemble [73]. From there, different selection criteria have been proposed to greedy select the ensemble [75–78]. A diversity measure dubbed Uncertainty Weighted Accuracy (UWA) in Refs. [75,76] takes into account the contribution of four events that represent the pairwise combinations given from the classification results of the candidate model and the current ensemble. In addition, they weight the events according to the uncertainty on the ensemble's decision, calculated as the proportion of models within the ensemble that correctly or incorrectly classifies a particular example. Rather than using a single criterion such as accuracy or diversity, recent approaches [77,78] propose a measure that incorporates both properties simultaneously. In Ref. [77], three new measures were proposed: Simultaneous Diversity and Accuracy (SDAcc), which adapt the measure proposed in Ref. [76] with the aims of correctly classifying all the samples, focusing on the hardest examples; Diversity-Focused-Two (DFTwo) that modifies the SDAcc measure by eliminating those events that take the same decision into account, in order to pay more attention to the ensemble diversity; and Accuracy-Reinforcement (AccRein) that modifies the SDAcc measure by eliminating the event where the candidate model and the current ensemble incorrectly classify the example in order to reinforce the contribution of adding the correct classifier. On the other hand, the work in Ref. [78] takes into account both the margin of examples [79] and the diversity ensemble to assess the importance of individual classifiers. Unlike greedy procedures, a dynamic programming approach that encourages extra diversity was proposed in Ref. [80]. With respect to DES, novel approaches have been proposed that highlight the importance of dynamic selection because a particular subset of the ensemble may not always be optimal for all unknown samples [81–83]. The latest approaches have provided improvements in the strategies to compose the region of competence [74,84,85], reinforcement learning modeling for DES [82] and integration of SES and DES [83], to name just a few.

Although ensemble selection is a very active research topic in the machine learning community [80,83,86], little work has been done on deep CNNs. One can find the SES for visual recognition tasks based on Bayesian Evidence [87] and two DES: 1) a method for medical image classification based on the Dirichlet distribution and the Mahalanobis distance [88], and 2) a method for facial expression recognition based on the geodetic distance calculated from a graph-based pruning approach [86]. On the other hand, one can notice that the pool of candidate models to build an ensemble is usually small in size when deep learning algorithms are used, mainly due to the time required to train them. In this case, when the set is small, greedy selection based on the validation set and a random choice has been shown to provide competitive results [89].

Our research, placed in the SES field, proposes a new metric that takes advantage of the epistemic uncertainty information to select a suitable ensemble subset. Despite the DES approach being interesting, the process involved in selecting the most competitive models for each new sample to be classified implies more demand for computational resources for the solution than SES. The latter is quite relevant in real-time applications where fast and accurate solutions are needed.

3. Forward step-wise Uncertainty-Aware Model Selection

In this section, we explain in depth the steps involved in the proposed Forward Step-wise Uncertainty-Aware Model Selection (FS_UAMS) method of selecting which models to ensemble from a set of deep CNN models.

Let us consider the object recognition problem (e.g. food recognition) with $x^{(i)} \in \mathbb{R}^{wxh}$ inputs features and $y^{(i)} \in \mathbb{R}$ outputs, where i corresponds to the i-th images and wxh are the image dimensions (width and height). As with any supervised machine learning algorithm, in deep CNN the goals are to find the model parameters W of the hypothesis function $h(x)$ that minimize the error between the predicted labels ($y^{(i)}$) and the ground-true labels ($\widehat{y}^{(i)}$). For this purpose, the weights of the deep CNN model (W) are fixed during model training by minimizing the loss function $l(h(x^{(i)}), \widehat{y^{(i)}})$ through a chosen optimization algorithm like: SGD, ADAM, among others.

$$\underset{W}{minimize} \sum_{i=1}^{M} l(h(x^{(i)}), \widehat{y^{(i)}})$$

The results of the optimization after training will often correspond to

a local minimum of the error, mainly because the error surface in the weight space of a deep CNN model is like a landscape with many hills and valleys. Here, the importance of using an ensemble of models is rescued, which smooths the bias produced in the classification by considering the output of several models that converge to different local minima, resulting in better performance (in terms of accuracy) and an improvement in the robustness against noisy inputs.

Particular applications, for example in critical real-time solutions, require accurate and fast models. As well known, by adding more models in an ensemble learning scheme, you can get better accuracy, but worse prediction time. Therefore, it is desirable to find a correct trade-off between both performance metrics. A common strategy for selecting CNN models within a pool of trained models in order to form the ensemble is commonly done by random selection or based on the model's performance in the validation set. Instead, we propose a method that selects and forwards step-wise the models according to the Epistemic Uncertainty analysis.

Let us consider a set $T = \{m_1, m_2, \ldots, m_N\}$ with N trained CNN models and a set $S = \{\varnothing\}$, initially empty, which corresponds to a subset T where N' selected models ($N' < N$) are stored. In the proposed FS_UAMS approach, the Epistemic uncertainty (EU) is analyzed for the purpose of identifying the most confident of the models in its prediction given the input images and, from this information, identifying which models could work best together. Let us define $EU_n \in \mathbb{R}^M$ as EU across all (M) training images for the n-th model as follows:

$$EU_n = \{EU_n^1, \ldots, EU_n^M\},$$

where:

$$EU_n^i = \sum_{y_i} - P_n(y_i|x_i) log(P_n(y_i|x_i))$$

and $P_n(y_i|x_i)$ can be estimated through the MC-dropout method [43], which corresponds to the average of the L softmax outputs obtained from L forward passes, applied to the image x_i during the prediction phase using the model m_n, with dropout turned on. After computing EU_n for all models, we select the first model to form the ensemble according to the minimum mean EU ($\overline{EU_n}$) checking the EU of all models within T. Then, an iterative procedure is carried out to select the following models to form the ensemble based on a proposed criterion that consists of analyzing how different the distribution of the uncertainty is among all images between the selected models ($m_j \in S$) with respect to the candidate model ($m_i \in T$). The candidate model with the smallest difference (minimum distance) is chosen and the procedure starts over until a total of K required models are reached (see Algorithm 3).

Algorithm 1. Forward Step-wise Uncertainty-Aware Model Selection

Algorithm 1: Forward Step-wise Uncertainty-Aware Model Selection

input: Selected models $S \leftarrow \emptyset$, Trained models T, Target models K;
output: Selected models S;
if S *is* $\emptyset$ **then**
  Select the m_i model from T according to the minimum average of epistemic uncertainty across all trained images;
  Update $S \leftarrow S \cup \{m_i\}$;
  Drop m_i from T;
end
while $|S|$ *is less than* K **do**
  $D \leftarrow \emptyset$ // Set that stores the proposed uncertainty-aware distance to select the best model;
  for $m_i \in T$ **do**
    $d_i = 0$;
    for $m_j \in S$ **do**
      $d_i = d_i + ||EU_i - EU_j||_2$
    end
    Update $D \leftarrow D \cup \{d_i\}$;
  end
  Select the m_i model from T according to the minimum value stored in D;
  Update $S \leftarrow S \cup \{m_i\}$;
  Drop m_i from T;
end
return S;

The EU, also known as model uncertainty, is the uncertainty produced by the lack of enough data for model training. For a particular image, the EU can be interpreted as a value that measures the confidence of the model's prediction. The smaller the EU, the greater the confidence for the given prediction and vice versa. In our case we propose to take advantage of this information selecting in the first place the model with greater confidence (smaller uncertainty) and then incorporate more models in the ensemble to slightly disturb the confidence provided for the first models.

Regarding the time complexity of the Algorithm 3, in the worst case (selecting $|T|-1$ models) it has an order of $O(\frac{|T|^3}{6})$ and even less when a small number of candidates are selected (e.g. order $O(\frac{|T|^3}{12})$ selecting $T/2$ models). In practical terms, using a reasonable pool of CNN models, with a size of $|T|=10$, where we intend to select $K=5$ models, the order of complexity of the proposed algorithm tends to be quadratic. Note that the selection process adds an additional cost to the training time to provide the final ensemble model. However, once the models are selected, the proposed method incurs no additional costs at execution time. The latter only depends on the number of selected models and the architecture of each one.

4. Experiments

In this section, we present the datasets used, the evaluation measures, the experimental setup and the results obtained with the proposed FS_UAMS approach in comparison with the baseline selection strategies.

4.1. Datasets for food recognition

In order to evaluate the performance of the proposed method for the food recognition purpose, we selected three challenging food datasets that share the property of having a wide range of dishes, large number of images, and unbalanced data between classes. We describe each one below.

MAFood121 [63] is an international multi-task food image dataset comprising of 21, 175 images belonging to 121 food dishes distributed evenly across 11 most popular types of cuisine. The dataset provides annotations to the images for 3 food-related recognition tasks: a) dish (single-label), b) cuisine (single-label), and c) categories/food groups (multi-label). For training purposes, images are stratified across the dish class, resulting in 72.5% images for training, 12.5% for validation, and the remaining 15% for testing. The images were collected from 4 different sources: 3 public food datasets [8,9,90] and Google Search Engine. As our experiments are focused on food recognition, we only consider the dish annotation for all images.

UECFood256 [9] is a Japanese food images dataset comprising of 31, 395 images belonging to 256 food dishes. In addition to the Japaneses dishes, it also includes traditional dishes from several other countries. The dataset provides two annotations for each image: a) dish and b) dish location (bounding box coordinates). Furthermore, the authors provide an official 5-fold cross-validation split that includes 28, 280 images with a minimum, maximum and average of images per dish class of 87, 637 and 110.47. In our experiments, a Holdout approach was applied, where the first official fold was used for testing purposes and the remainder for training. Regarding the training data, a stratified random sample was applied to split the data into 90% for training and 10% for validation. The bounding box information was not used in our experiments.

VIREO172 [10] is a Chinese multi-task dataset comprising of 110, 241 images belonging to 172 dishes collected by Baidu and Google image search. VIREO Food-172 has annotations for two tasks: a) dish and b) ingredients (only visible). The dataset is distributed as 60% for training, 10% for validation and the remaining 30% for testing. The ingredients annotations were not used in our experiments.

Table 2
Results in terms of accuracy using an ensemble consisting of five ResNet50 models.

Model	MAFood121	UECFood256	VIREO172
FSMS_VAL	85.02%	68.87%	86.35%
FS_UAMS	**85.84%**	**68.94%**	**86.48%**
P(MS-Rand>FS_UAMS)	2.38%	46.83%	7.14%

Table 3
Results in terms of accuracy using an ensemble consisting of five InceptionV3 models.

Model	MAFood121	UECFood256	VIREO172
FSMS_VAL	88.70%	**73.10%**	89.17%
FS_UAMS	**88.95%**	73.01%	**89.26%**
P(MS-Rand>FS_UAMS)	20.24%	25.00%	7.94%

4.2. Validation metrics

Overall accuracy (*Acc*) is the metric selected to evaluate the performance of individual models in the multi-class food recognition problem. Formally, it can be defined as follows:

$$Acc = \frac{1}{M}\sum_{i=1}^{M} ind(h(x^{(i)}), \widehat{y^{(i)}}),$$

where *ind*(*, *) denotes an indicator function that returns 1 when the predicted label ($h(x^{(i)})$) for the image $x^{(i)}$ is equal to its ground-truth label ($\widehat{y^{(i)}}$) and 0, otherwise.

In our case, $h(x^{(i)})$ is computed for the ensemble of deep models by mean of the average of the softmax outputs along all the CNNs that form the ensemble. The equation is defined as follows:

$$h(x^{(i)}) = argmax(\{\overline{softmax(f^{W}(x^{(i)})_1)}, \ldots, \overline{softmax(f^{W^n}(x^{(i)})_C)}\}),$$

with

$$\overline{softmax(f^{W}(x^{(i)})_c)} = \frac{1}{N}\sum_{n=1}^{N} softmax(f^{W^n}(x^{(i)})_c)$$

and

$$softmax(f^{W^n}(x^{(i)})_c) = \frac{\exp(f^{W^n}(x^{(i)})_c)}{\sum_{c'}\exp(f^{W^n}(x^{(i)})_{c'})}$$

where $f^{W^n}(x^{(i)})_c$ corresponds to the final activation of the n-th CNN for the sample $x^{(i)}$ on the c-th label and $c' = 1, \ldots, C$.

4.3. Experimental setup

The proposed FS_UAMS method is evaluated considering two deep CNN architectures: ResNet50 [70] and InceptionV3 [71]. The models were selected taking into account the computational cost ($\leq$200 MB per single sample and model parameters in memory), the training and execution time ($\leq$6 GFLOPS) and the good performance previously shown in food recognition [19,29,39]. Both CNN models were modified by removing the output layer, and instead, we added a dropout layer with a probability of 0.5 and an output layer with softmax activation. The neurons in the output layer equal the number of dishes in the target dataset, specifically 121 for MAFood121, 256 for UECFood256 and 172 for VIREO172.

A total of 10 models was trained for each CNN architecture with the same training hyper-parameters, but randomly changing the weight initialization and shuffling the input images. All of them were re-trained from a pre-trained model on ImageNet [69], during 50 epochs with a

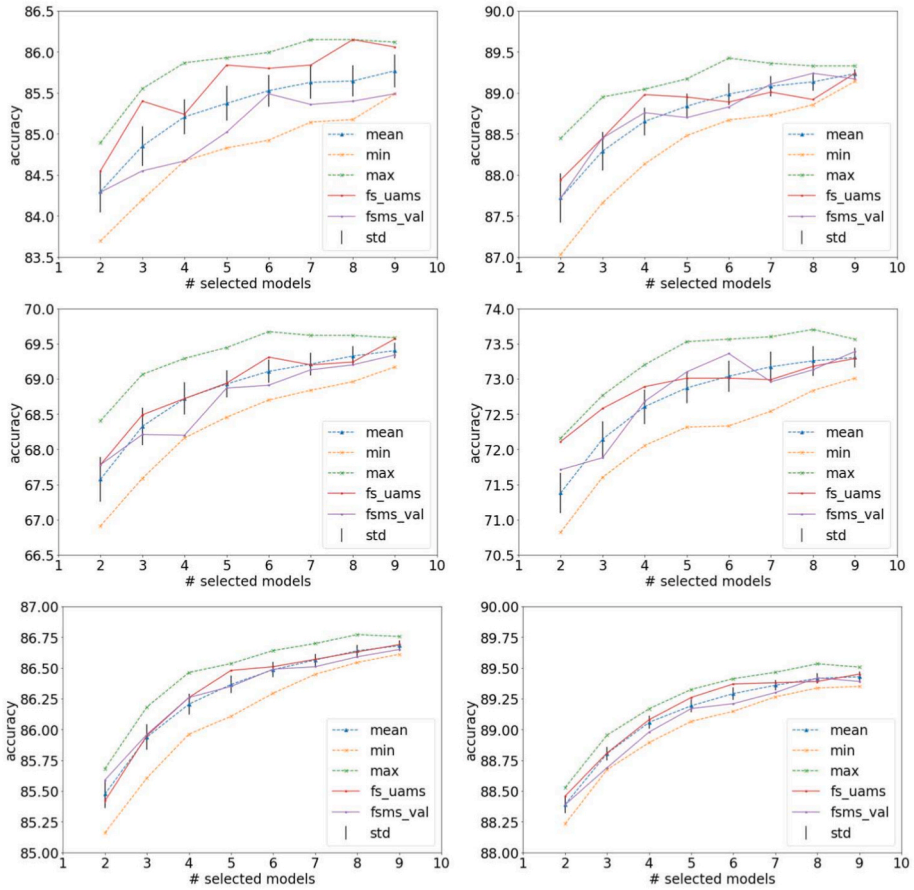


Fig. 2. Performance in terms of accuracy obtained for food recognition using ensemble approaches with different numbers of models. The results on MAFood121, UECFood256 and VIREO172 are shown from the first to the third row, respectively; and the columns group the results with ResNet50 (left) and InceptionV3 (right).

patience of 10, a batch size of 20, an initial learning rate of $2e-4$ and a decay of 0.2 every 8 epochs. As for the data preparation, the images were normalized within the range $[-1.1]$ and the ImageNet mean was subtracted in all of them. Regarding the data augmentation used during the training, the images were first resized to 256x256x3 in the case of ResNet50 and 342x342x3 for InceptionV3. Subsequently, traditional strategies like random crops and horizontal flips were applied. The dimensions of the random crops were 224x224x3 and 299x299x3 for ResNet50 and InceptionV3, respectively. The training was done using categorical cross entropy loss and Adam's optimizer in Keras framework with Tensorflow as backend. The best models of each training were chosen according to the highest accuracy obtained in the validation set.

In addition to the FS_UAMS, two other strategies for model selection were applied for comparative purposes. Selection was done according to the results on the validation sets and randomly selection.

4.4. Results

In order to assess the advantages of the proposed approach with respect to the traditional model selection methods, specifically model selection based on the results of the validation set (FSMS_VAL) or random selection (MS-Rand), ten models are trained for each type of CNN chosen (ResNet50 and InceptionV3) and across three public food datasets. Then, the same trained models have been considered to build the ensembles using the proposed FS_UAMS method or the other methods mentioned above. Note that all reported results refer to the performance of each method evaluated on the test sets. For the UECFood256 case, the validation set is considered as test set. In this way, we compare the performance in the classification of unseen data, which implies comparing the generalization capabilities of each method.

Table 2 and Table 3 summarize the overall accuracy achieved by the ensembles built from each model selection method, with a total of 5 models for each type of CNN architecture, respectively. From the tables, we can see that the proposed method outperforms the results in most cases and in all of them it is a much better option than random selection. Particularly, a significant improvement is provided for FS_UAMS in MAFood121 and VIREO172, where in some cases the probability of obtaining better models is less than 8%. Although it is true that in the case of UECFood256 there is no significant difference between FSMS_VAL with respect to FS_UAMS, our proposal is still better than a random choice.

On the other hand, the performance of each model selection methods (FS_UAMS and FSMS_VAL) and various statistical measures, for all datasets, are shown in Fig. 2. As we can see, ensembles of different size (number of base classifiers) were built and evaluated. Specifically, two to nine different models based on the same CNN architecture were considered for each ensemble. Regarding the statistical measures, the mean, standard deviation (std), minimum (min) and maximum (max) accuracy are presented. The results are distributed for each dataset (rows) and CNN architecture (columns). A noticeable improvement in terms of accuracy can be observed with the proposed FS_UAMS on MAFood121 (first row) and UECFood256 (second row) when we form ensembles with less or equal to five base classifiers, and even more important in some cases our approach achieves almost the maximum performance, which corresponds to the optimal value of the search space when we evaluated all the possible combinations of base classifiers. Regarding the result on VIREO172 (third row), a clear improvement can also be seen with InceptionV3 (right column) mainly with reduced ensemble size. However, with ResNet50 (left column), while it is true that our proposal works better than FSMS_VAL, in most cases the

Table 4
Data showing the percentage of images that change their predicted class positively (N_to_P) and negatively (P_to_N) after adding a new ResNet50 model to the ensemble by mean of the FSMS_VAL approach. *T* denotes the total of images that change their predicted class with respect to the previous ensemble.

	MAFood121			**UECFood256**			**VIREO172**		
# Models	T	N_to_P	P_to_N	T	N_to_P	P_to_N	T	N_to_P	P_to_N
2	157	0.6497	0.3503	419	0.6563	0.3437	1363	0.6383	0.3617
3	94	0.6383	0.3617	255	0.5647	0.4353	822	0.5754	0.4246
4	79	0.4810	0.5190	157	0.5287	0.4713	604	0.5828	0.4172
5	55	0.6000	0.4000	139	0.5899	0.4101	478	0.5314	0.4686

Table 5
Data showing the percentage of images that change their predicted class positively (N_to_P) and negatively (P_to_N) after adding a new ResNet50 model to the ensemble by mean of the FS_UAMS approach. *T* denotes the total of images that change their predicted class with respect to the previous ensemble.

	MAFood121			**UECFood256**			**VIREO172**		
# Models	T	N_to_P	P_to_N	T	N_to_P	P_to_N	T	N_to_P	P_to_N
2	155	0.7032	0.2968	376	0.6862	0.3138	1388	0.6571	0.3429
3	79	0.6709	0.3291	243	0.5844	0.4156	907	0.5976	0.4024
4	67	0.4627	0.5373	145	0.5448	0.4552	626	0.5799	0.4201
5	65	0.6462	0.3538	115	0.5565	0.4435	490	0.5755	0.4245

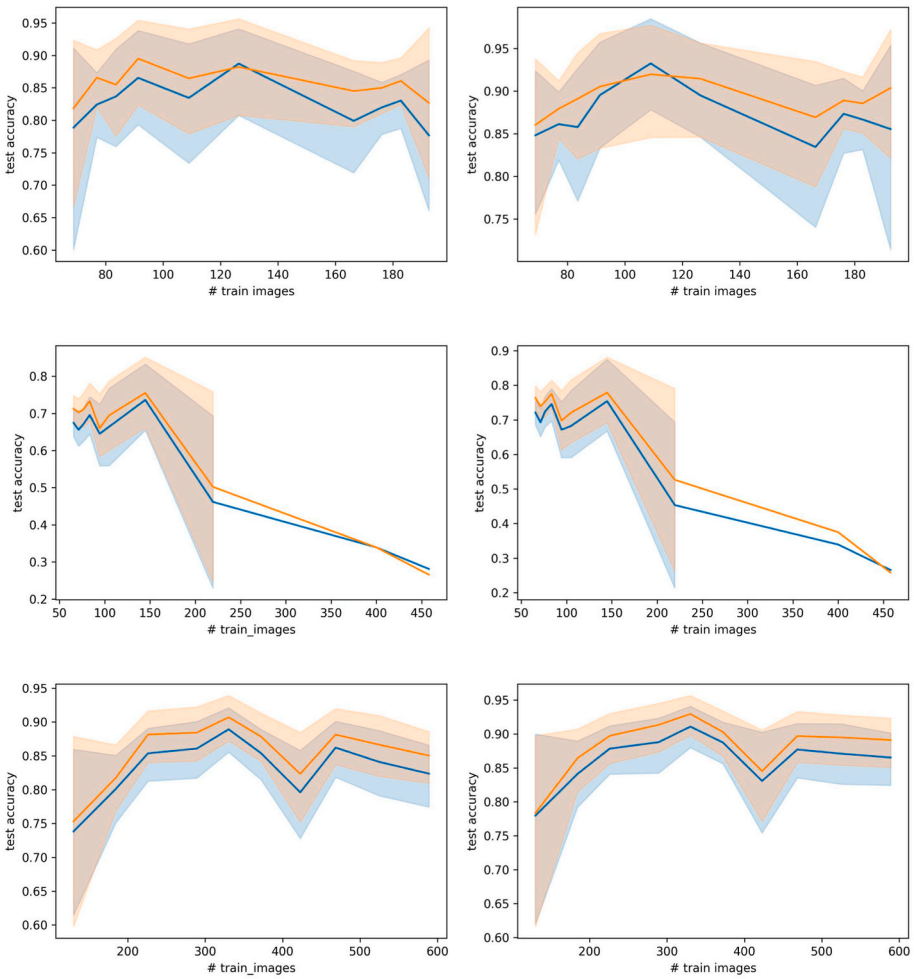


Fig. 3. Plot illustrating the number of training images versus test accuracy achieved. The blue and orange lines correspond to the BC and the ensemble built from FS_UAMS, respectively.

improvement is not too significant. A surprising aspect to highlight with respect to the FSMS_VAL strategy is that in several ensembles built, regardless of the dataset or the CNN architecture, poor performance has been evidenced, at some point close to the minimum value corresponding to the worst combination of the base classifiers. We think the reason for this is that the high dependency on the accuracy in the validation set affects the generalization capability especially when using a dataset with a small or medium number of images per class (<200 on

Table 6
Results for the classes on top-5 accuracy according to the Base Classifier (BC) performance for MAFood121, UECFood256, and VIREO172. The ID column refers to the class identification, BC and Ensemble correspond to the results obtained for BC and the ensemble built from our FS_UAMS approach for the respective classes.

MAFood121			**UECFood256**			**VIREO172**		
ID	BC	Ensemble	ID	BC	Ensemble	ID	BC	Ensemble
59	1.0000	0.9412	174	1.0000	0.9500	93	0.9902	0.9951
2	1.0000	1.0000	158	1.0000	1.0000	165	0.9876	0.9917
54	1.0000	0.9091	243	1.0000	0.9565	113	0.9859	0.9859
4	1.0000	1.0000	27	0.9583	0.9583	98	0.9823	0.9823
116	1.0000	1.0000	77	0.9546	1.0000	171	0.9811	0.9849

Table 7
Results on the classes with the worst-5 accuracy according to the BC performance for MAFood121, UECFood256, and VIREO172. The ID column refers to the class identification, BC and Ensemble correspond to the results obtained for BC and the ensemble built from our FS_UAMS approach for the respective classes.

MAFood121			**UECFood256**			**VIREO172**		
ID	BC	Ensemble	ID	BC	Ensemble	ID	BC	Ensemble
72	0.2778	0.3889	89	0.1429	0.0952	54	0.4685	0.4505
118	0.4546	0.4546	140	0.1500	0.3000	136	0.5172	0.4483
42	0.5455	0.6364	30	0.1905	0.2857	63	0.5298	0.5860
50	0.5484	0.6129	142	0.2105	0.3158	30	0.5730	0.6255
34	0.5714	0.6667	190	0.2222	0.2222	91	0.5859	0.6061

Table 8
MCNemar's test evaluated in ensembles built from FSMS_VAL and FS_UAMS, with respect to the BC and to the ensemble that achieved the minimum performance (Min). For this evaluation, two members are considered to form the ensembles. *** = p-value $\leq$ 0.01, ** = p-value $\leq$ 0.05, * = p-value $\leq$ 0.1 and NS = Not Significant.

	MAFood121		**UECFood256**		**VIREO172**	
Methods	χ^2	p-value	χ^2	p-value	χ^2	p-value
ResNet50						
BC vs FSMS_VAL	5,36	**	19.76	***	61.41	***
BC vs FS_UAMS	14,24	***	11.52	***	45.18	***
Min vs FSMS_VAL	2.27	NS	6.07	**	10.27	***
Min vs FS_UAMS	3.33	*	4.25	**	3.80	*
InceptionV3						
BC vs FSMS_VAL	2.96	*	36.71	***	58.50	***
BC vs FS_UAMS	8.53	***	54.40	***	115.65	***
Min vs FSMS_VAL	3.67	*	4.51	**	1.60	NS
Min vs FS_UAMS	4.10	**	10.03	***	4.63	**

Table 9
Performance of Deep Learning methods for visual food recognition on MAFood121, UECFood256 and VIREO172 datasets.

Method	Accuracy	Ensemble learning	Multiple annotations
MAFood121			
Aguilar et al. [62]	81.62%	no	yes
RUMTL [63]	83.82%	no	yes
TL-Ensemble [39]	84.95%	yes	no
ResNet50	83.16%	no	no
InceptionV3	86.94%	no	no
FS_UAMS	88.95%	yes	no
UECFood256			
DeepFood [16]	54.70%	no	no
Tahir et al. [41]	68.50%	yes	no
WISeR [29]	72.71%	yes	no
ResNet50	66.27%	no	no
InceptionV3	69.72%	no	no
FS_UAMS	73.01%	yes	no
VIREO172			
Arch-D [10]	82.06%	no	yes
RUMTL [63]	85.19%	no	yes
IG-CMAN [34]	90.63%	no	yes
ResNet50	84.45%	no	no
InceptionV3	87.33%	no	no
FS_UAMS	89.26%	yes	no

average). The latter is the case of MAFood121 and UECFood256 datasets. Unlike FSMS_VAL, our proposal based on epistemic uncertainty analysis looks more stable in its performance. In summary, regardless of the number of models used to form the ensemble, in most cases our approach provides better performance than the mean, min, and FSMS_VAL.

In addition to the traditional metric (accuracy) to compare the overall model performance, we analyze if the ensemble of two members built from FSMS_VAL and FS_UAMS strategies provides an improvement statistically significant with respect to the base classifier, which is selected from the best result on test, and the ensemble that provides the minimum accuracy. We validate the statistical significance of the performance improvement using the McNemar's Test [91], which is recommended in cases where it is costly or impractical to use cross-validation [92] (e.g. deep learning models trained on large datasets). For the Test, a 2x2 contingency table was prepared for each pair of methods. On the diagonal, we put the number of miss-classification and correct classification obtained by both methods in the same samples. In the rest, we put the number of miss-classification obtained for one method that are correct for the other in the same sample, and vice versa. The results of the MCNemar's Test can be seen in Table 8, where we observe that there is a statistically significant difference in the performance shown by both strategies compared to the base classifier. It is interesting to note when making the comparison with respect to the ensemble that showed minimum accuracy. In two cases the difference in performance with respect to the ensemble built following the FSMS_VAL strategy does not guarantee to obtain a statistical significance. Summarizing, FS_UAMS provides a statistically significant performance difference in all cases (p-value $\leq$ 0.1), in most of them highly significant (p-value $\leq$ 0.01) and generally a significance level higher than FSMS_VAL.

Table 4 and Table 5 present the total number of images that change the classification results (positively or negatively) after adding a new base classifier to the ensemble by mean of FSMS_VAL or FS_UAMS respectively. When both tables are compared with each other, we can see that in almost all cases the proposed approach tends to maintain the previously well-predicted classes better than FSMS_VAL. That is, with our approach, fewer images are obtained that negatively change their prediction, corresponding to images that are well predicted with the previous ensemble and bad for the current one, which shows that our metric based on epistemic uncertainty analysis is able of finding models

that best complement each other.

The results achieved by the proposed method in each of the classes were also analyzed. Fig. 3 shows the performance achieved for classes with a similar number of training images. To group the classes, a clustering algorithm, *kmeans++* with $k = 10$, was used on the number of training images contained in each class. In this figure, the straight line represents the mean accuracy across all classes within each group, and the background color represents the standard deviation. Regarding the colors, the blue color is used for the BC and the orange color for the proposed Ensemble method. In general terms, the magnitude of improvement when we compare BC with respect to the Ensemble models is almost uniform for classes with different amount of images. Contrary to expectations, the performance improvement of the different classes is not directly proportional to the number of images available in the training sets. Surprisingly, in some cases, classes with more training data perform even worse than classes with fewer samples (see Fig. 3, second row). When analyzing the results obtained in each dataset independently, it is observed that the proposed approach behaves differently for unbalanced data between classes. In MAFood121, a more balanced accuracy can be observed. Regarding UECFood256, a greater increase is observed for classes with few samples and a decrease in performance for classes with large samples. With respect to VIREO172, a greater improvement is observed for the classes with more data. The differences in behavior may be due to the distribution of the data in the training set. Although all datasets used are unbalanced, in some cases there is a strong imbalance in a small number of classes (e.g. UECFood256) or the degree of imbalance between classes is very large (e.g. VIREO172).

Table 6 and Table 7 show the *top-5* and *worst-5* classes predicted for BC and also the performance achieved on these classes with the proposed approach to reflect the improvement obtained in different food classes. In most cases, the performance of the worst classes improves significantly. On the other hand, some higher classes have reduced their performance. From this, it can be inferred that ensemble method tends to balance classification results rather than providing very high performance for a few classes.

Finally, in Table 9, we show the performance achieved on the three public food datasets for the state-of-the-art methods, the base classifier (ResNet50 and InceptionV3), and an ensemble of five InceptionV3 members constructed using the proposed FS_UAMS strategy. In the UECFood256 case, only works that use all uncropped images with the ground-truth bounding box are considered. As expected, our ensemble model provides a much better performance than the base classifier, the improvement is about 2% for MAFood121, 3% for UECFood256 and 2% for VIREO172. Furthermore, we outperform state-of-the-art results in MAFood121 with a wide range and provide competitive results in UECFood256 and VIREO172 using the simplest models to form the ensemble and without requiring additional annotations.

5. Conclusions

In this article, we presented a novel SES method to perform multi-class food recognition problem through an ensemble of Convolutional Neural Networks that employs the Epistemic Uncertainty. Unlike previous ES approaches that use the accuracy or diversity measure to select the base classifiers, in our case we proposed a new measure to select the models based on an Epistemic Uncertainty analysis. The validation of the proposed FS_UAMS ES method was performed on three public food image datasets and was tested using different CNN baseline networks. As a result, we can observe that our approach allowed us to improve the performance in most cases when compared to the baseline ES methods (randomly selection and validation set accuracy). Furthermore, it alleviates the computational cost involved in an exhaustive search of all possible combinations and reduces the bias on the validation set. The proposed method has been validated considering the standard way of generating ensemble in deep learning, that is, combining models belonging to the same CNN architecture. As future work, the compatibility of different architectures to extract the EU will be evaluated to combine with the proposed approach, models based on several architectures. In addition, this research will be extended to develop an uncertainty-aware ensemble selection for multi-label object recognition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially funded by TIN2018-095232-B-C21, SGR-2017 1742, Measurer EIT Digital, Logmeal4Shape and CERCA Programme/Generalitat de Catalunya. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

[1] E. Moguel, J. Berrocal, J. García-Alonso, Systematic literature review of food-intake monitoring in an aging population, Sensors 19 (15) (2019) 3265.
[2] V. Bruno, C.J. Silva Resende, A survey on automated food monitoring and dietary management systems, J. of health & medical informatics 8 (3) (2017).
[3] F. Cordeiro, E. Bales, E. Cherry, J. Fogarty, Rethinking the mobile food journal: exploring opportunities for lightweight photo-based capture, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 3207–3216.
[4] W. Min, S. Jiang, L. Liu, Y. Rui, R. Jain, A survey on food computing, ACM Comput. Surv. 52 (5) (2019) 1–36.
[5] Y. Wang, J.-j. Chen, C.-W. Ngo, T.-S. Chua, W. Zuo, Z. Ming, Mixed dish recognition through multi-label learning, in: Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities, 2019, pp. 1–8.
[6] L. Zhou, C. Zhang, F. Liu, Z. Qiu, Y. He, Application of deep learning in food: a review, Compr. Rev. Food Sci. Food Saf. 18 (6) (2019) 1793–1811.
[7] J. Ahmad, H. Farman, Z. Jan, Deep learning methods and applications, in: Deep Learning: Convergence to Big Data Analytics, Springer, 2019, pp. 31–42.
[8] L. Bossard, M. Guillaumin, L. Van Gool, ECCV, in: Food-101–mining Discriminative Components with Random Forests, Springer, 2014, pp. 446–461.
[9] Y. Kawano, K. Yanai, Automatic expansion of a food image dataset leveraging existing categories with domain adaptation, in: ECCV, Springer, 2014, pp. 3–17.
[10] J. Chen, C.-W. Ngo, Deep-based ingredient recognition for cooking recipe retrieval, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 32–41.
[11] G. Ciocca, P. Napoletano, R. Schettini, Food recognition: a new dataset, experiments, and results, IEEE.J. Biomed. Opt.Health Inf 21 (3) (2016) 588–598.
[12] P. Kaur, K. Sikka, W. Wang, S. Belongie, A. Divakaran, Foodx-251: a Dataset for Fine-Grained Food Classification, 2019 arXiv preprint arXiv:1907.06167.
[13] X. Chen, H. Zhou, Y. Zhu, L. Diao, Chinesefoodnet: A Large-Scale Image Dataset for Chinese Food Recognition, 2017 arXiv preprint arXiv:1705.02743.
[14] S. Mezgec, B.K. Seljak, Using deep learning for food and beverage image recognition, in: 2019 IEEE Int. Conf. On Big Data (Big Data), 2019, pp. 5149–5151. IEEE.
[15] H. Wu, M. Merler, R. Uceda-Sosa, J.R. Smith, Learning to make better mistakes: semantics-aware visual food recognition, in: Proceedings of the 24th ACM International Conference on Multimedia, 2016, pp. 172–176.
[16] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, Y. Ma, Deepfood: deep learning-based food image recognition for computer-aided dietary assessment, in: International Conference on Smart Homes and Health Telematics, Springer, 2016, pp. 37–48.
[17] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, S. Cagnoni, Food image recognition using very deep convolutional networks, in: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 2016, pp. 41–49.
[18] Z.-Y. Ming, J. Chen, Y. Cao, C. Forde, C.-W. Ngo, T.S. Chua, Food photo recognition for dietary tracking: system and experiment, in: International Conference on Multimedia Modeling, Springer, 2018, pp. 129–141.
[19] E. Tasci, Voting Combinations-Based Ensemble of Fine-Tuned Convolutional Neural Networks for Food Image Recognition, Multimedia Tools and Applications, 2020, pp. 1–22.
[20] P. Pandey, A. Deepthi, B. Mandal, N.B. Puhan, Foodnet: recognizing foods using ensemble of deep networks, IEEE Signal Process. Lett. 24 (12) (2017) 1758–1762.
[21] E. Aguilar, M. Bolaños, P. Radeva, Food recognition using fusion of classifiers based on cnns, in: ICIAP, Springer, 2017, pp. 213–224.
[22] J. Kittler, M. Hatef, R.P. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[23] A. Kendall, Y. Gal, What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? NIPS, 2017, pp. 5574–5584.

[24] Y. Kawano, K. Yanai, Foodcam: a real-time mobile food recognition system employing Fisher vector, in: International Conference on Multimedia Modeling, Springer, 2014, pp. 369–373.
[25] G.M. Farinella, D. Allegra, M. Moltisanti, F. Stanco, S. Battiato, Retrieval and classification of food images, Comput. Biol. Med. 77 (2016) 23–39.
[26] N. Martinel, C. Piciarelli, C. Micheloni, An ensemble feature method for food classification, Mach. Graph. Vis. 26 (2017).
[27] K. Yanai, Y. Kawano, Food image recognition using deep convolutional network with pre-training and fine-tuning, in: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2015, pp. 1–6. IEEE.
[28] P. McAllister, H. Zheng, R. Bond, A. Moorhead, Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets, Comput. Biol. Med. 95 (2018) 217–233.
[29] N. Martinel, G.L. Foresti, C. Micheloni, Wide-slice residual networks for food recognition, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 567–576. IEEE.
[30] K.-H. Lee, X. He, L. Zhang, L. Yang, Cleannet: transfer learning for scalable image classifier training with label noise, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5447–5456.
[31] G.A. Tahir, C.K. Loo, An open-ended continual learning for food recognition using class incremental extreme learning machines, IEEE Access 8 (2020) 82328–82346.
[32] L. Herranz, S. Jiang, R. Xu, Modeling restaurant context for food recognition, IEEE Trans. Multimed. 19 (2) (2016) 430–440.
[33] H. Wang, W. Min, X. Li, S. Jiang, Where and what to eat: Simultaneous restaurant and dish recognition from food image, in: Pacific Rim Conference on Multimedia, Springer, 2016, pp. 520–528.
[34] W. Min, L. Liu, Z. Luo, S. Jiang, Ingredient-guided cascaded multi-attention network for food recognition, in: Proceedings of the 27th ACM International Conference on Multimedia, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1331–1339, https://doi.org/10.1145/3343031.3350948. URL, https://doi.org/10.1145/3343031.3350948.
[35] H. Zhao, K.-H. Yap, A.C. Kot, Fusion learning using semantics and graph convolutional network for visual food recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1711–1720.
[36] E.J. Heravi, H.H. Aghdam, D. Puig, An optimized convolutional neural network with bottleneck and spatial pyramid pooling layers for classification of foods, Pattern Recogn. Lett. 105 (2018) 50–58.
[37] R. Tanno, K. Okamoto, K. Yanai, Deepfoodcam: a dcnn-based real-time mobile food recognition system, in: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 2016, p. 89.
[38] H. Zhao, K.-H. Yap, A.C. Kot, L. Duan, Jdnet: a joint-learning distilled network for mobile visual food recognition, IEEE J. of Selected Topics in Signal Processing 14 (4) (2020) 665–675.
[39] A. Fakhrou, J. Kunhoth, S. Al Maadeed, Smartphone-based food recognition system using multiple deep cnn models, Multimed. Tool. Appl. 80 (21) (2021) 33011–33032.
[40] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conf. On CVPR, 2017, pp. 4700–4708.
[41] G.A. Tahir, C.K. Loo, Explainable deep learning ensemble for food image analysis on edge devices, Comput. Biol. Med. 139 (2021), 104972.
[42] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight Uncertainty in Neural Network, ICML, 2015, pp. 1613–1622.
[43] Y. Gal, Z. Ghahramani, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning, ICML, 2016, pp. 1050–1059.
[44] M. Sensoy, L. Kaplan, M. Kandemir, Evidential Deep Learning to Quantify Classification Uncertainty, NIPS, 2018, pp. 3179–3189.
[45] D. Molchanov, A. Ashukha, D. Vetrov, Variational dropout sparsifies deep neural networks, in: ICML-volume 70, JMLR. org, 2017, pp. 2498–2507.
[46] C. Louizos, M. Welling, Multiplicative normalizing flows for variational bayesian neural networks, in: ICML-volume 70, JMLR. org, 2017, pp. 2218–2227.
[47] P. Van Molle, T. Verbelen, B. Vankeirsbilck, J. De Vylder, B. Diricx, T. Kimpe, P. Simoens, B. Dhoedt, Leveraging the bhattacharyya coefficient for uncertainty quantification in deep neural networks, Neural Comput. Appl. (2021) 1–17.
[48] Y. Gal, R. Islam, Z. Ghahramani, Deep bayesian active learning with image data, in: Proc. Of the 34th ICML-Volume 70, JMLR. org, 2017, pp. 1183–1192.
[49] S. Khan, M. Hayat, S.W. Zamir, J. Shen, L. Shao, Striking the right balance with uncertainty, in: Proc. Of the IEEE Conf. on CVPR, 2019, pp. 103–112.
[50] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, in: ICML, vol. 2, 2016, p. 7.
[51] J. Chang, Z. Lan, C. Cheng, Y. Wei, Data uncertainty learning in face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5710–5719.
[52] M. Cai, F. Lu, Y. Sato, Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14392–14401.
[53] Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, H. Roth, 3d semi-supervised learning with uncertainty-aware multi-view co-training, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3646–3655.
[54] C. Sakaridis, D. Dai, L.V. Gool, Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7374–7383.
[55] J. Choi, D. Chun, H. Kim, H.-J. Lee, Gaussian yolov3: an accurate and fast object detector using localization uncertainty for autonomous driving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 502–511.
[56] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2888–2897.
[57] L. Bertoni, S. Kreiss, A. Alahi, Monoloco: monocular 3d pedestrian localization and uncertainty estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6861–6871.
[58] J. Zhang, D.-P. Fan, Y. Dai, S. Anwar, F.S. Saleh, T. Zhang, N. Barnes, Uc-net, Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8582–8591.
[59] A. Kumar, T.K. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, C. Feng, Luvli face alignment: estimating landmarks' location, uncertainty, and visibility likelihood, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8236–8246.
[60] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, J. Schneider, Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2095–2104.
[61] Z. Zhang, A. Romero, M.J. Muckley, P. Vincent, L. Yang, M. Drozdzal, Reducing uncertainty in undersampled mri reconstruction with active acquisition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2049–2058.
[62] E. Aguilar, P. Radeva, Uncertainty-aware integration of local and flat classifiers for food recognition, Pattern Recogn. Lett. 136 (2020) 237–243.
[63] E. Aguilar, M. Bolaños, P. Radeva, Regularized uncertainty-based multi-task learning model for food analysis, J. Vis. Commun. Image Represent. 60 (2019) 360–370.
[64] E. Aguilar, B. Nagarajan, R. Khatun, M. Bolaños, P. Radeva, Uncertainty Modeling and Deep Learning Applied to Food Image Analysis, BIODEVICES, 2020, pp. 9–16.
[65] G. Huang, Y. Li, G. Pleiss, Z. Liu, J.E. Hopcroft, K.Q. Weinberger, Snapshot ensembles: train 1, get M for free, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.
[66] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[67] R.E. Schapire, The Boosting Approach to Machine Learning: an Overview, Nonlinear Estimation and Classification, 2003, pp. 149–171.
[68] D.H. Wolpert, Stacked generalization, Neural Network. 5 (2) (1992) 241–259.
[69] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, NIPS, 2012, pp. 1097–1105.
[70] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. Of the IEEE Conf. on CVPR, 2016, pp. 770–778.
[71] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conf. On CVPR, 2016, pp. 2818–2826.
[72] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1–2) (2002) 239–263.
[73] R. Caruana, A. Niculescu-Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, p. 18.
[74] S. García, Z.-L. Zhang, A. Altalhi, S. Alshomrani, F. Herrera, Dynamic ensemble selection for multi-class imbalanced datasets, Inf. Sci. 445 (2018) 22–37.
[75] I. Partalas, G. Tsoumakas, I.P. Vlahavas, Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection, ECAI, 2008, pp. 117–121.
[76] I. Partalas, G. Tsoumakas, I. Vlahavas, An ensemble uncertainty aware measure for directed hill climbing ensemble pruning, Mach. Learn. 81 (3) (2010) 257–282.
[77] Q. Dai, R. Ye, Z. Liu, Considering diversity and accuracy simultaneously for ensemble pruning, Appl. Soft Comput. 58 (2017) 75–91.
[78] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, M. Xu, Margin & diversity based ordering ensemble pruning, Neurocomputing 275 (2018) 237–246.
[79] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, et al., Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.
[80] O.A. Alzubi, J.A. Alzubi, M. Alweshah, I. Qiqieh, S. Al-Shami, M. Ramachandran, An optimal pruning algorithm of classifier ensembles: dynamic programming approach, Neural Comput. Appl. 32 (20) (2020) 16091–16107.
[81] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, Dynamic classifier selection: recent advances and perspectives, Inf. Fusion 41 (2018) 195–216.
[82] Z. Liu, K. Ramamohanarao, Instance-based ensemble selection using deep reinforcement learning, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–7. IEEE.
[83] T.T. Nguyen, A.V. Luong, M.T. Dang, A.W.-C. Liew, J. McCall, Ensemble selection based on classifier prediction confidence, Pattern Recogn. 100 (2020), 107104.
[84] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, Prototype selection for dynamic classifier and ensemble selection, Neural Comput. Appl. 29 (2) (2018) 447–457.
[85] D.V. Oliveira, G.D. Cavalcanti, R. Sabourin, Online pruning of base classifiers for dynamic ensemble selection, Pattern Recogn. 72 (2017) 44–58.
[86] D. Li, G. Wen, X. Li, X. Cai, Graph-based dynamic ensemble pruning for facial expression recognition, Appl. Intell. 49 (9) (2019) 3188–3206.
[87] Y.-D. Kim, T. Jang, B. Han, S. Choi, Learning to select pre-trained deep representations with bayesian evidence framework, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5318–5326.

[88] A.G. Pacheco, T. Trappenberg, R.A. Krohling, Learning dynamic weights for an ensemble of deep models applied to medical imaging classification, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.
[89] F. Perez, S. Avila, E. Valle, Solo or ensemble? choosing a cnn architecture for melanoma classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, 0–0.
[90] C. Güngör, F. Baltacı, A. Erdem, E. Erdem, Turkish cuisine: a benchmark dataset with Turkish meals for food recognition, in: 2017 25th Signal Processing and Communications Applications Conference (SIU), 2017, pp. 1–4, https://doi.org/10.1109/SIU.2017.7960494.
[91] Q. McNemar, Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika 12 (2) (1947) 153–157.
[92] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923.

J. Vis. Commun. Image R. 95 (2023) 103905

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Deep ensemble-based hard sample mining for food recognition ☆

Bhalaji Nagarajan [a,*], Marc Bolaños [b], Eduardo Aguilar [a,c,d], Petia Radeva [a,d]

[a] Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007 Barcelona, Spain
[b] AIGecko Technologies SL, Barcelona, Spain
[c] Dept. de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Avenida Angamos 0610, Antofagasta, 1270709, Chile
[d] Computer Vision Center, Cerdanyola (Barcelona), Spain

ARTICLE INFO

MSC:
68T05
68T10
68T30
68T37

Keywords:
Knowledge representation
Hard-sample mining
Food recognition
Deep ensembles
Data augmentation

ABSTRACT

Deep neural networks represent a compelling technique to tackle complex real-world problems, but are over-parameterized and often suffer from over- or under-confident estimates. Deep ensembles have shown better parameter estimations and often provide reliable uncertainty estimates that contribute to the robustness of the results. In this work, we propose a new metric to identify samples that are hard to classify. Our metric is defined as *coincidence score* for deep ensembles which measures the agreement of its individual models. The main hypothesis we rely on is that deep learning algorithms learn the low-loss samples better compared to large-loss samples. In order to compensate for this, we use controlled over-sampling on the identified "hard" samples using proper data augmentation schemes to enable the models to learn those samples better. We validate the proposed metric using two public food datasets on different backbone architectures and show the improvements compared to the conventional deep neural network training using different performance metrics.

1. Introduction

Food computing [1] has become an active area of research due to its widespread applications in managing health, including dietary management [2,3] and nutritional analysis [4]. Automatic food recognition is fundamental to most food computing tasks [5,6], which aims to categorize an input image by taking into account the main content that appears in it. Food recognition can be of different granularity [7] - ranging from coarse categories, such as fruits, vegetables, and desserts to fine-grained identification of specific food items like apples, bananas, and chocolate cake. The goal is to accurately determine the type of food present in an image. This capability facilitates a wide range of applications related to the overall food understanding. Food recognition poses significant computer vision challenges, primarily due to the inherent complexity of food images [8]. Food images show high intra-class and low inter-class variability. The visual appearance of food can significantly vary with different cooking methods and cuisines. Moreover, food can exhibit substantial differences within the same food class. Additionally, food images lack distinctive spatial layouts and rigid structures [7]. The presence of randomly distributed ingredients across the food platter compounds the challenge [9]. The fine-grained nature of food classes adds to the complexity of food images [10]. Collectively these factors contribute to the complexity of food recognition tasks, rendering them highly challenging.

The rapid advancement of Deep Learning (DL) techniques has accelerated the development of more sophisticated and effective food recognition models. The success of DL-based methods can be attributed to the ability of neural networks to learn any prediction function considering a sufficient number of neurons, layers and data [11]. Early improvements in performance were basically due to adding more and more layers to the model. A deeper model can represent a more complex function to map the input to the desired output. However, it is also more likely to overfit during its training. With more data, overfitting can be avoided, resulting in better performance on unseen data [12].

DL models are a data-hungry methodology, due to their high dependence on a large amount of training data to provide better model performance [13]. They require large-scale training datasets such as ImageNet [14] which have hundreds of samples representing each class. However, carefully curating such datasets is difficult due to the labelling cost and complexity of collecting samples [15]. Most widely used public food datasets are often downloaded from web sources and annotated with crowdsourcing tools. Intrinsically, these datasets

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail addresses: bhalaji.nagarajan@ub.edu (B. Nagarajan), marc.bolanos@aigecko.com (M. Bolaños), eaguilar02@ucn.cl (E. Aguilar), petia.ivanova@ub.edu (P. Radeva).

https://doi.org/10.1016/j.jvcir.2023.103905
Received 6 April 2023; Received in revised form 26 June 2023; Accepted 29 July 2023
Available online 31 July 2023
1047-3203/© 2023 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Fig. 1. Sample images from class *Roll Cold Noodle* in the Food-1K dataset. The top row shows samples with high coincidence scores ( ) and the bottom row with low coincidence scores ( ). **Samples with low scores are difficult for the models to learn.**.

possess several impediments such as noise (labelling errors and quality of images) and imbalance (distribution of samples in each class), along with the *de facto* computer vision challenges such as occlusions, blur and lighting.

It is natural that most of the available images in a dataset are of fairly good quality and well-defined. These samples constitute the portion of the training set that we call as **easy samples**. DL models tend to learn the underlying patterns of these data faster. However, there would also be samples that are drastically different in terms of their representative features and also are poorly represented in terms of numbers. These samples cannot be removed from the training set, because they are still representative of the problem at hand and often are a factor in the generalization of these models. We consider them as **hard samples** - samples that the models cannot learn sufficiently well due to their complexity or poor representation in the training set. We show sample images of the 'Roll Cold Noodle' class in Fig. 1. The samples at the top row show easy samples, whereas the bottom row shows hard samples. Although the hard samples remain representative of the class, they do not exemplify the typical characteristics associated with this class. These samples cannot be removed as they provide the necessary diversity in the dataset. It is highly likely that this information of easy and hard samples is not taken into consideration during training, the models would result in getting biased towards those samples that are widely present. In this work, we show the importance of these hard samples that are *not easy to learn*. We argue that paying more attention to them during the training process could be more beneficial for model training. In particular, the variability in food images is often not uniformly captured in public databases built automatically from food images available on the web, leading to more *hard* samples in them.

Specifically for some computer vision problems, proper data collection is a quite challenging task due to intrinsic properties present in the data itself [16,17], which increases the possibility of having to deal with hard samples. In the domain of food recognition, the aforementioned challenges significantly increase the likelihood of encountering hard samples within food datasets. This may be one of the reasons why, although DL methods have been shown to be very effective for some object classification tasks, obtaining highly accurate results for food data implies solving a higher level of difficulty.

The objective functions of DL models can be represented as a high-dimensional landscape containing many hills and valleys [18] and the goal of the learning process is to reach the lowest point in the search space. Apart from training individual models, ensemble learning has been shown to be effective in higher performances [19]. Training different models using exactly the same data but with random weight initialization often converges to a different solution. By combining several individual models, the generalization performance is often higher compared to individual models [20] and has been shown to be more robust to uncertainty and out-of-distribution issues [21]. With this regard, in this work, we propose a novel methodology that takes advantage of a deep ensemble scheme to discover hard samples in food recognition. We compute a new measure, *coincidence score* that estimates the correlation of different models of a deep ensemble. With this measure, we guide the training of each individual model focusing on the hard samples, thereby increasing its learning capacity. We create controlled over-sampling of those identified hard samples using data augmentation techniques. Data augmentation allows increasing the size and variability of a dataset without the need to acquire more real data [22]. However, uncontrolled data augmentation leads to overfitting those specific samples. Therefore keeping it controlled is necessary. On this regard, the main contributions of this paper are summarized as follows:

- We characterize the learning behaviour of deep ensemble models with respect to each sample using a new proposed *coincidence score*. It measures the importance of the samples with respect to the learning process.
- We propose a controlled over-sampling method that helps in the learning process of the food recognition model.
- We validate the proposed learning process using two public food datasets on different backbones and show how the prediction on the selected samples improves without losing the generalization ability of the models.
- We show how treating hard samples increases the confidence of the models in food recognition. We argue that it is important for models to be accurate and be certain about the decisions they make especially when they are applied in real-world scenarios [23].

The remaining sections of the paper are structured as follows: We provide a comprehensive review of related works in Section 2. We outline the underlying rationale behind our proposed technique and explain the details of our proposed method in Section 3. We present the results used to validate our method in Section 4 followed by concluding remarks at the end.

2. Related work

In this section, we review the latest articles that are most related to the proposed technique.

2.1. Sample understanding

One of the recent areas of the data-centric approach is to explore the nature of samples used in training the models. Importance sampling is a popular technique, where samples that could fasten the learning process as well as improve the generalization of the algorithms are identified [24]. Gradient norm-based importance sampling was one of the earliest works in applying importance sampling [25]. The general idea is to use the loss values to observe the sampling distribution for each mini-batches [26]. Self-paced learning algorithms [27] and curriculum learning [28] based methods have been successful in learning samples based on an importance criterion, where the easier samples are learned first and gradually the hard samples are learned [29]. Co-learning has also been employed to select samples using another network in order to maximize the convergence speed [30]. Different strategies to train the samples have been beneficial in improving the training speed and most importantly have been helpful in improving the test errors [31].

Example difficulty has been an interesting field of research, where the difficulty could be either due to the statistical or learning aspect of the example. Prediction depth [32] is used to measure the sample difficulty, where the samples are classified based on their likelihood of mislabelling, learning with the presence of labels as well as learning with and without labels. Compression-sensitivity [33] and c-score ranking [34] have established the benefits of learning the nature of samples in the training dataset and improving the performance of the algorithms by treating different samples differently. The variance of Gradients [35] is used as a metric to rank data based on difficulty and also acts as an out-of-distribution detection measure. By identifying the samples that do or do not contribute to the learning process, the algorithms can result in being faster and also more efficient [36].

2.2. Data augmentation as over-sampling strategy

Deep Neural Networks (DNNs) tend to memorize the samples in the training set due to a large number of learnable parameters resulting in overfitting. The generalization of DNNs is very important to make them usable in real-world applications. One of the common strategies to reduce overfitting is to use Data Augmentation (DA) schemes [37]. DA creates invariant samples from the original dataset samples and allows the creation of data points that are capable of minimizing the distance between the training and the test sets [22]. Typically, the size of the datasets is increased using different label-preserving transformations [38]. One of the areas of using DA to increase the size of datasets is to avoid the problem of class imbalance [39] using oversampling techniques. Oversampling re-balances the class distributions so that the models are able to overcome the data bias caused due to the majority classes. By oversampling, the samples of the minority classes are augmented to create additional samples [40]. DA techniques have been very powerful tools to overcome the problem of creating large DL training datasets. However, the best DA policies have to be manually designed. The procedure is both time-consuming and costly as it increases the number of experiments needed to reach an optimal DA technique that can be helpful in either avoiding overfitting or helping in oversampling. In recent years, learned policies have gained importance where the DA policies are automatically searched [41–43]. In all the above literature, the DA policies are applied to the entirety of the datasets or to a specific class of the datasets in order to increase the performance of the DNNs. It is interesting to observe that different DA techniques have been beneficial to specific classes [44] that are grouped using epistemic uncertainty.

DNNs have a tendency to make over-confident estimates [45] and when those estimates are used to make decisions, there is a bias in the decision-making process. Ensemble strategies have been beneficial in reducing the bias in decision-making and also in improving the performance of the models [19]. Deep ensembles have a better approximation of the hypothesis function and overcome the local minima that an individual DNN would be stuck in. The key benefit of using ensembles is that it reduces the variance in the prediction error [20]. In this work, we follow the same line of sample understanding whereby we identify those samples in order to oversample the data so that we can achieve better algorithm performance. With ensembles being better approximators of the hypothesis function, we use an ensemble of models in order to capture the hard samples. We use DA schemes to create additional samples from identified hard samples and train the individual classifiers again in order to improve the algorithm's performance.

2.3. Food recognition

As seen in any DL application, the main dependency is the availability of large datasets. However, considering the complexity of food images, it is very challenging to create such datasets. Recently, efforts have been made to create large-scale food datasets [7,46]. Food applications have reached the common people much faster and therefore it is necessary for the models to generalize well on unseen data. Food decisions are very critical as they directly affect the health of individuals and therefore making accurate decisions is a mandate [47]. Transfer learning has been successfully used in food recognition using different architectures [48,49]. As more and more complex architectures came into existence, those models were adapted for food recognition [50]. Context information is very important in food recognition and attention networks have been widely used to learn both local and global features [46]. Ingredient information coupled with information from the food images has been successful in achieving state-of-the-art performance in several food datasets [51,52]. Online continuous learning framework [53] adapted to food classification was successful in learning data continuously. Graph Neural Networks are used to learn inter-class relations between images and semantics [54]. However, food recognition in general is treated as a transfer learning problem with an emphasis on deeper architectures. There are very few food-related literature works being data-centric; that is, focusing on the samples for training [36,55]. As shown in other areas, this is an interesting and highly relevant research area in the DL community.

3. Proposed method

In this section, we explain our proposal for learning hard samples based on sample importance computed from an ensemble of homogeneous DNNs. First, we discuss the rationale behind the proposed approach and then we show the components of our pipeline.

3.1. Rationale

In any DNN learning process, the input pair $(x, \hat{y})$ from a distribution D is learned by repeatedly iterating it through a model, H. The learning model minimizes an objective function and the aim of this process is to obtain an optimal H among several possible hypotheses. Each input x gives a corresponding prediction y. The prediction y and its comparison to the ground truth label $\hat{y}$ are used to bring the model closer towards an optimal solution in the search space by back-propagating the error during each learning step. The model adjusts its learnable parameters and the process is repeated until the learning converges. The amount of data used for training the models is small compared to the size of the search space in a DNN and therefore different learning algorithms identify different hypotheses which give a near-optimal performance on the distribution D [20]. The aim of an ensemble is to reduce the risk of getting a wrong classifier, where none of the models is able to represent the true function and also to avoid the local minima in the search space [56]. Different models in an ensemble learn the same samples differently owing to different starting points in the search space. This increases the overall performance of the models in terms of performance and also increases the confidence of these models.

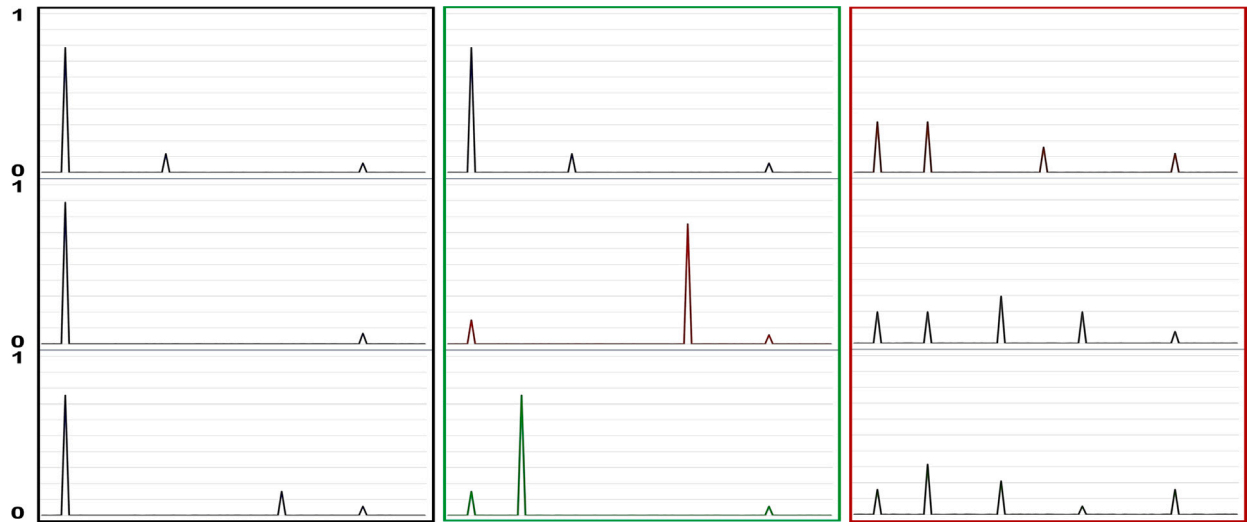


Fig. 2. Behaviour of three different models (rows) in a deep ensemble (the x-axis corresponds to the class index and the y-axis corresponds to the likelihood obtained by individual models in the ensemble). Left column: all three models predict the same class; middle column: all models predict different classes; and right column: models give low likelihood to all classes.

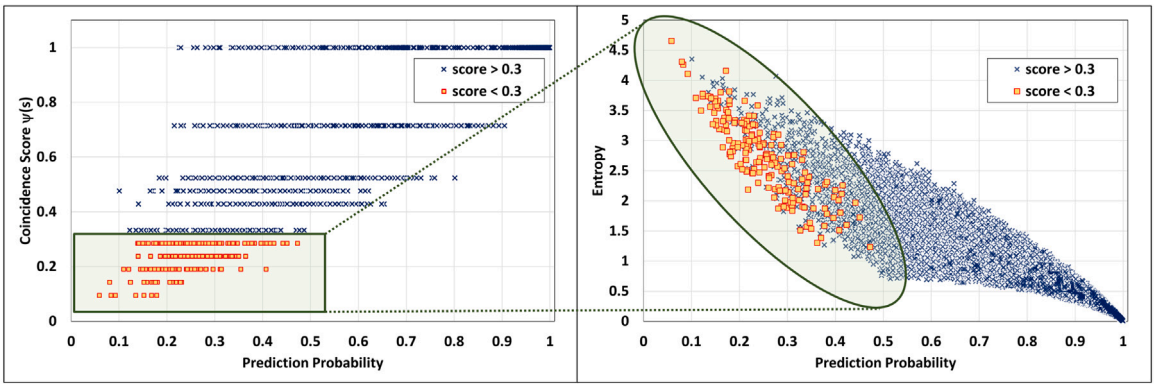


Fig. 3. Coincidence score vs. Prediction probability of samples (left plot). The relation between coincidence score and entropy is highlighted - a low coincidence score relates to high entropy.

The confidence of models can be evaluated using different metrics. High confidence relates to low uncertainty and in this work, we use Shannon entropy [57], a common uncertainty measure, to analyse the confidence of the predictions.

In a deep ensemble, we can observe different behaviour of models like: (1) Most/all models agree on the prediction of the sample, (2) One or more models do not agree on the prediction, (3) The samples are not learned; that is, models are giving not high likelihood for any of the classes, etc. We illustrate this behaviour using Fig. 2. The correlation between the models is a way to understand how well the samples are learned. Based on this model behaviour scheme, we establish a sample importance criterion and use this parameter to identify the hard samples. Hard samples are the ones that the ensembles have difficulty in learning. After identifying these samples, we use controlled DA on those samples in order to emphasize on them during the learning process. With this sample's importance, we argue that the overall performance of the models (both accuracy and confidence) improves.

3.2. Hard-sample mining

When training DNNs, it can be observed that not all samples behave in the same way. Some samples are learned in a few learning steps whereas some samples require larger training epochs [31]. Even when an optimal solution is found during training of the DNNs, still there would be samples that are not learned by the DNN. These samples can be repeated, emphasized by giving them more importance or weighted; or used for the DA in order to be learned. With this in mind, we propose to identify those samples that are difficult to learn during the traditional training process.

First, a deep ensemble, M, is created from models, $m_1, m_2, \ldots, m_n$ that are trained under similar conditions. The predicted probability vector p is obtained for each sample using each model and $y_i = argmax(p)$ is the prediction for that sample for the model, i. The class predicted by the models will form the basis for subsequent computations. Using this, we compute a metric called *coincidence score* for each sample. We define the coincidence score (ψ) for a given sample s as follows:

$$\psi(s) = \frac{2}{n * (n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ind(y_i == y_j) \tag{1}$$

where n is the number of models in the ensemble; y_i and y_j are the predicted class for the models i and j, respectively; and $ind(y_i == y_j)$ is an indicator function that returns 1 when y_i is equal to y_j and 0 otherwise. This score is used to measure the relationship of models in the ensemble. The coincidence score ranges between 0 and 1, where $\psi(s) = 1$ if all the models give the same prediction class. Note that the values of the coincidence score are in the range of $[0, 1]$ with $1/2 * n * (n-1)$ different possible values (see Fig. 3 (left)).

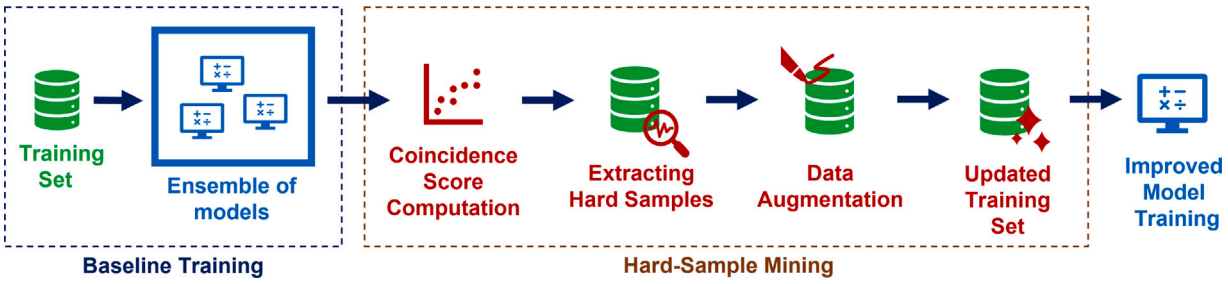


Fig. 4. Overall Training Pipeline. An ensemble is created during the baseline training; the dataset is updated using the hard samples mined using the coincidence score.

Using this coincidence score (Eq. (1)), we group samples that are in the lowest agreeable range as samples that are hard for the models to learn (coincidence groups). In Fig. 3, we show that the sample groups with the smallest coincidence score have high entropy and low probability score. The coincidence score can be visualized with respect to the uncertainty and also the accuracy of each sample. The samples are either not well learned or models do not agree on the prediction class of the samples. The hard samples often do not contribute to the performance of the ensembles and often have high entropy values and low accuracy values. However, these points are not completely useless in the learning process. It can be shown that with a different variation of these data points, it is possible to learn the relevant representations thereby increasing the performance of the models. It has to be noted that by oversampling these samples, it is quite possible to overfit the training set. Therefore, a controlled DA is needed to make benefit from this approach. We use random augmentations of the identified hard samples to increase the number of samples in the training set. This allows us to create new samples in the search space that differ from the original ones, thereby allowing the models to learn better from those samples. We show the overall training pipeline in Fig. 4. During the baseline training, an ensemble of homogeneous models is created. During the hard-sample mining phase, the coincidence score allows us to mine hard samples and create controlled data augmentation of those samples. Finally, an improved model is trained using the updated training data. With different experiments, we show how controlled DA on these samples would increase the model performances.

4. Experiments and results

In this section, we first provide a brief introduction to the datasets and then discuss the training and evaluation pipeline. We then show the results and provide a detailed analysis on them.

4.1. Datasets

In order to validate the proposed pipeline, we use two popular food datasets - UECFood-256 [58] and Food-1K [7]. UECFood-256 is a public Asian food object detection and multi-label dataset that was created using food images crawled from web sources. It has 28,898 images containing around 32 K food items categorized into 256 classes with each class containing at least 100 images. For our experiments, we crop individual dishes using the bounding boxes provided as ground truth in order to create a single-label dataset. For UECFood-256, we use 80% of samples for training, while the rest for testing. Food-1K was introduced in the ICCV-2021 LargeFoodAI[1] workshop comprises of 500 K images from 1000 classes. The dataset was created using the images crawled from the Meituan website and includes both eastern and western classes. All images in the dataset have a single food class label. Owing to the large-scale nature of Food-1K, we randomly select 100 classes to validate our proposed method, similar to the approaches of [53,59]. We use the training and validation split provided by the workshop for our experiments.

4.2. Training and evaluation procedure

We use different popular object classification architectures as base architectures to validate our pipeline. We keep the hyper-parameters constant before and after applying the DA strategies. However, we vary the hyper-parameters according to the datasets. A minimum of 5 models are needed for better uncertainty quantification [21]. We create an ensemble using 7 homogeneous models (the number of models is selected as a middle point based on the study of [21], where a maximum of 10 models have been studied). Once we train the models of the ensemble, we compute the coincidence score (Eq. (1)) and group the samples based on this coincidence score. We retrain the models after creating additional samples from these groups. We show with different experiments how controlled DA has helped in improving the individual models.

We use standard ResNet-50 [60] for Food-1K (100) experiments and EfficientNet-B0 [61] for UECFood-256 experiments. The base architectures are pre-trained with ImageNet weights. In order to show the independence of architectures, we also use EfficientNet-B0 to train on Food-1K (100) dataset. We use the conventional standard DA techniques to generalize the network apart from the ones we use to oversample the hard samples. We show the different hyperparameters used for our experiments – the ensemble learning for hard sample mining and the individual models (trained with or without augmented samples) – in Table 1. Additionally, for UECFood-256, we use class weighting to balance the loss function. Class weights are computed using the compute_class_weight function of scikit-learn library.[2] The pre-processing of images is done based on the respective preprocess functions of the backbones.[3] We train all the models until convergence and then compute the coincidence score. We augment the hard samples and re-train the individual models with the modified datasets. Note that all settings are maintained uniformly for the baseline and the retraining step. We develop all the experiments using the Keras framework with Tensorflow as the backend. All models are trained using a single NVIDIA RTX2080Ti GPU.

4.2.1. Data augmentation

Learned DA policies have been widely used to create additional data in order to improve the performance of DNNs. However, it requires expertise and several experimental evidence to arrive at the best DA policies. Learning these policies has been of interest recently and has been successful in achieving better performance compared to the traditional approaches [62]. Following this, we use RandAugment [63]

[1] https://foodai-workshop.meituan.com/foodai2021.html#index

[2] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

[3] https://keras.io/api/applications/

Fig. 5. Example data augmentations of images created using RandAug. The first row (■) corresponds to the original images from the Food-1K dataset. The other two rows (■) show different data augmentation images created.

Table 1
Hyperparameters used in model training.

Dataset	Food-1K (100)	UECFood-256
Loss	Categorical Cross Entropy	
Batch Size	32	
Total Epochs	100	
Optimizer	Adam	
Initial LR	0.001	
Patience	10	5
LR Scheduler	–	Cosine Decay with Restart
Image Size	224 × 224	384 × 384
Data Augmentation	rotation, brightness, flip (horizontal)	rotation, brightness, flip (horizontal), shift, shear

to create additional samples for our experiments. We use the hyper-parameters of RandAugment as used on ImageNet. We use $N = 2$ and $m = (6, 12)$ to create new samples. We use ImgAug[4] library to create additional samples from the ones that are identified to have low coincidence scores. The rest of the images are kept as it is. Some sample images along with their DA samples are shown in Fig. 5. The first row corresponds to the original images and the other two rows correspond to the DA images. Note that, RandAugment is only used to create new samples and is not used as a DA strategy during the training of proposed models.

4.2.2. Evaluation metrics

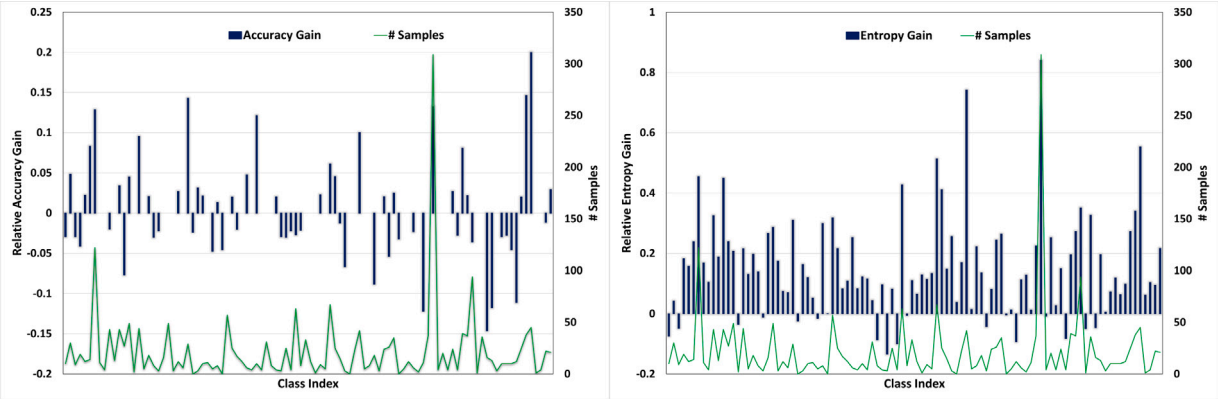
We evaluate the performance of the models using the accuracy of the validation set (or test set, in case the dataset contains only training and test splits). For all the experiments, we report the median accuracy of the individual models as a performance metric. We use median accuracy as it allows us to select the model corresponding to the accuracy value. It also avoids the overestimation or underestimation of the results, since models with high or low accuracy have less effect on the median than on the average. Apart from the quantitative results, we also use Shannon entropy to identify the behaviour of our method towards confidence in predictions. We use the accuracy versus entropy plots to show the learning progression of the models.

4.3. Performance comparison

We show the median validation accuracy provided by the base learners of different ensemble models in Table 2. We compare the proposed training scheme with the baseline and with the random selection method. We use general transfer learning as the baseline experiments, where ImageNet pre-trained weights are used to learn on the original food recognition datasets. For the other two experiments, we only modify the datasets, i.e. oversample the identified samples and retrain the individual models. For the random selection experiments, we randomly augment 5% of samples so as to keep the number of samples consistent with that of the proposed method. With the results in Table 2, it can be seen that our proposed method works across different datasets (Rows 2 and 3) and also across different architectures (Rows 1 and 2). In all three experiments, we show that the proposed method outperforms both the baseline and the random selection methods. As shown by the results, random augmentation improves over the baseline models. However, selective augmentation of hard samples outperforms random augmentation. The difference in the results obtained between our proposal and the random selection methods highlights the importance of our approach to perform hard sample mining and focus data augmentation on that subset of data.

We further analyse the results by showing the relative gain in the performance of the proposed technique against the baseline in Fig. 6. We show the relative gain for Food-1K (100) trained on ResNet-50 using accuracy and entropy metrics against the baseline performance. We measure class-wise accuracy and entropy for both the baseline and proposed method and report the differences as gain (in the case of entropy, the gain is the reduction in entropy) in performance. The number of samples augmented per class is also shown in the secondary axis. It can be seen that the hard samples are present in all the classes in varying degrees. With respect to the entropy, most of the classes fare better than the baseline, which can be attributed to the models

[4] https://github.com/aleju/imgaug



(a) Relative Accuracy Gain (b) Relative Entropy Gain

Fig. 6. Relative Gain in Performance for Food-1K(100) trained on ResNet-50.

Fig. 7. Qualitative results: The ground truth is shown at the top (■). Predictions in red (■) correspond to the baseline model, whereas those in green correspond to the proposed approach (■).

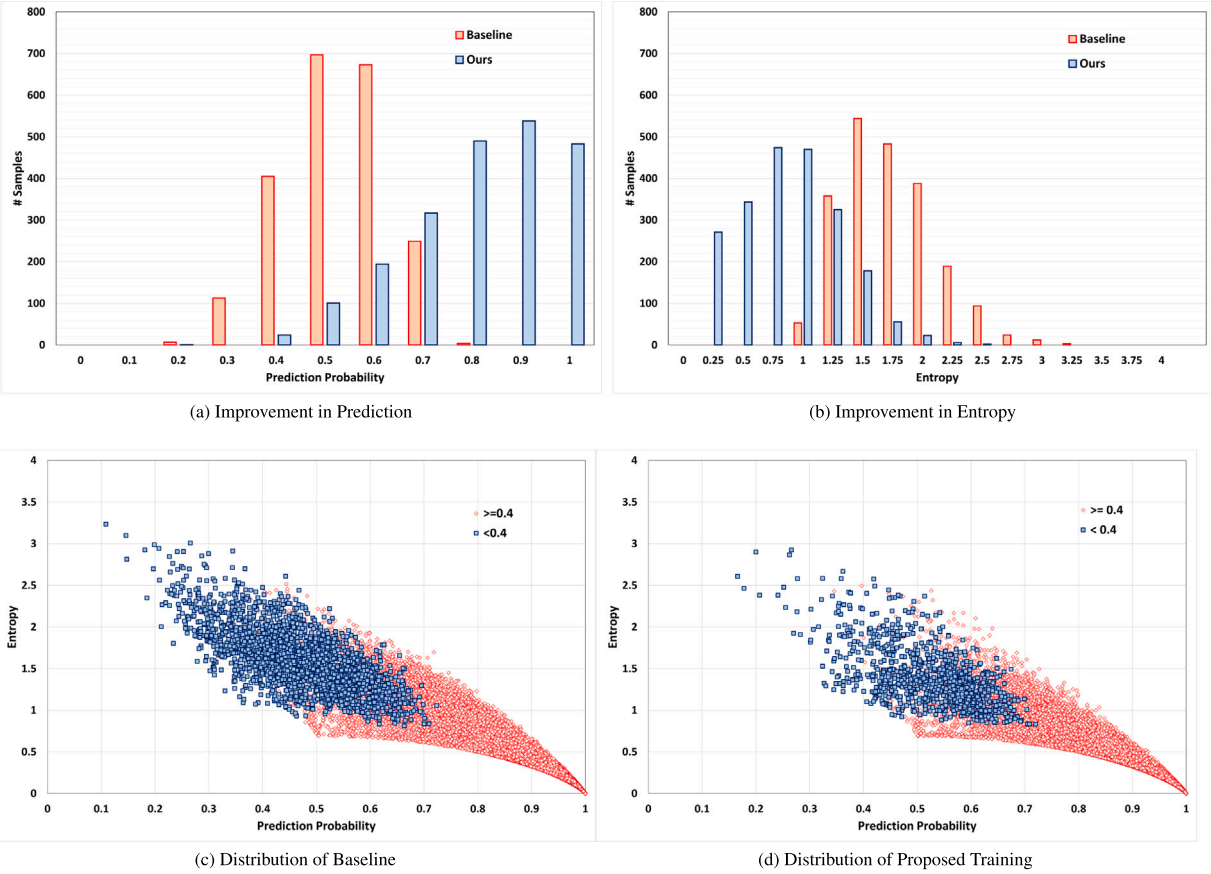
Table 2
Performance comparison of our proposed method with baseline and random selection techniques.

Dataset	Architecture	Baseline	Random selection	Coincidence score	Ours
Food-1K (100)	ResNet-50	83.33%	85.10%	<0.4	**85.93%**
Food-1K (100)	EfficientNet-B0	88.90%	89.36%	<0.5	**90.08%**
UECFood-256	EfficientNet-B0	80.66%	80.97%	<0.3	**81.18%**

learning the samples well with the presence of additional datasets. Overall there is an accuracy gain of 0.38 compared to the baseline. However, with respect to accuracy, it can be seen that there are classes which have a decrease in performance. The number of augmented samples could be one reason for this behaviour. It should be noted that in our experiments we have augmented all the identified hard samples equally (Same number of augmentations per sample). We show some of the predictions of the training samples in Fig. 7. In the cases shown, the baseline models make a wrong prediction with high entropy and comparatively lower probability. This hinders the learning of those samples. However, with the proposed strategy, the samples are learned with high accuracy and low entropy values, increasing the learnability of those particular classes.

4.3.1. *Sample behaviour analysis*

First, we analyse the behaviour of hard samples that are augmented using different plots as shown in Fig. 8. We compare the histograms of both the predicted probability (Fig. 8(a)) and entropy (Fig. 8(b)) of



(a) Improvement in Prediction

(b) Improvement in Entropy

(c) Distribution of Baseline

(d) Distribution of Proposed Training

Fig. 8. Improvement experienced on the selected hard samples before and after applying data augmentation.

the proposed training against the baseline. We use Food-1K (100) with ResNet-50 to show this analysis. The general tendency of these samples is low accuracy and high entropy values. The prediction probability of the baseline (reported in red) is more centred in the low values of the histogram bins, whereas the entropy of the samples is centred in the high bins. This relates to a substantial number of samples that are not learned by the baseline models. After augmenting the hard samples and retraining, the prediction probability histogram tends towards 1 (shown in blue) whereas the entropy moves closer towards 0. This shows that the samples are learned better with high certainty, highlighted by the increase in prediction probability and decrease in entropy values. Using both measures, it is evident that after adding an augmented sample per hard sample, the models are able to better learn those samples. We further show the accuracy versus entropy plots before and after DA. It can be seen that there are more samples which belong to the hard sample group in the baseline, whereas after learning those hard samples using the proposed technique, there are fewer samples that are still hard to learn. Note the movement of samples from a higher entropy to a lower entropy, showing the learning behaviour of the models.

We further show the behaviour of the samples with respect to each group in Fig. 9. For this analysis, we compute the improvement in accuracy against the baseline accuracy for each coincidence group of samples. We create a histogram showing the number of samples that have improved/worsened for each group. It can be observed that the performance has increased in most of the samples, even though there are a few that have gone down. An interesting observation here is that the most improved samples come from the low to mid-coincidence scores and not from the samples that belong to the higher coincidence groups. It can further substantiate the claim that the low coincidence score samples are hard samples and by treating only those samples, the performance of the models can be increased.

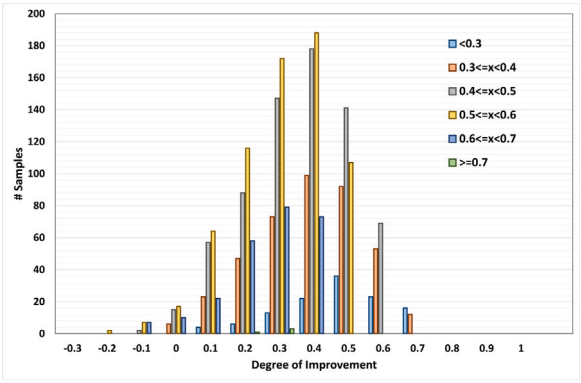


Fig. 9. Behaviour of each Coincidence Group.

4.4. Design decisions

Effect of number of DA. One of the decisions that could affect the behaviour of the proposed method is the number of augmentations that can be used for every sample. We show the results of the Food-1K (100) experiment using ResNet-50. We vary the number of augmentations for each hard sample and train the models by adding the hard sample augmentations. We train several models and report the median validation accuracy in Table 3. We used the hard samples identified using the coincidence value smaller than 0.4 ($\psi(s) < 0.4$) for this experiment. From the results, it can be seen that when the number of DA increases, the performance of the models drops. This can be attributed to the fact that more and more samples make the algorithms memorize the

Table 3
Effect of varying the number of Data augmentations of hard samples.

Experiment	Val. Accuracy
Baseline (no new samples)	83.33%
One augmentation	**85.93%**
Two augmentations	84.53%
Three augmentations	84.53%

Table 4
Performance of different Coincidence Groups. Each hard sample is augmented once.

Experiment	Val. Accuracy
Baseline	80.66%
coincidence score < 0.25	80.92%
coincidence score < 0.30	**81.18%**
coincidence score < 0.40	80.70%
coincidence score < 0.50	80.59%

training set and it loses the generalization capabilities, rather than learning the hard samples.

Behaviour of different coincidence groups. The other important factor that could affect the performance is the selection of coincidence groups. We show in Table 4 the median validation accuracy after retraining the individual models using over-sampled samples from different coincidence score groups. We keep the number of augmentations as 1 for this experiment. We show the performance using the UECFood-256 dataset on EfficientNet-B0. It can be observed that the performance starts to increase starting from lower coincidence score samples to a certain group after which the performance starts to decrease. For a system of models, if we find the groups using Eq. (1), it can be seen that the lowest groups are those where there are less number of models that agree towards a prediction. By oversampling these samples, the ensembles are able to make better decisions. However, with a higher coincidence score, there is often a case where more models agree on the prediction. Considering only those samples that are faring worse, there would always be a presence of noisy samples in the datasets. By augmenting those samples, there is no change in the performance of the models as the models are unable to learn just by augmenting those samples and require a different treatment. The performance dropping beyond the baseline can be attributed to the memorization of the training set.

4.5. Statistical significance

We check the statistical significance of the improvements to determine if the observed differences are statistically meaningful rather than occurring by chance. Following the work of [64] and considering the expensive nature of the deep learning models, we use McNemar's test [65] to validate our experiments. For this test, we construct a 2×2 contingency table based on the outcome of two tests. The diagonal elements represent the counts of correct classifications and misclassifications for both models, while the off-diagonal elements indicate the counts of classifications made exclusively by one model. We report the results of McNemar's test in Table 5. Comparing our proposed method to the baseline methods, we observe a statistically significant difference in all experiments. When comparing the random selection experiment with our proposed method, we observe that the UECFood-256 experiment does not guarantee a statistical difference. Overall, we see that our method provides a statistically significant performance in all but one case (p-value ≤ 0.2).

4.6. Limitations

We critically list the limitations of our proposed method so that they can help in future research lines.

Table 5
Statistical Significance using McNemer's Test.

Methods		Baseline vs. Ours		Random Selection vs. Ours	
Datasets	Architectures	χ^2	p-value	χ^2	p-value
Food-1K (100)	ResNet-50	24.20	***	2.72	**
Food-1K (100)	EfficientNet-B0	6.88	**	2.45	*
UECFood-256	EfficientNet-B0	1.66	*	0.25	NS

***p-value ≤ 0.05.
**p-value ≤ 0.1.
*p-value ≤ 0.2.
NS = Not Significant.

- The proposed method works in two stages: (1) First to create the ensemble and measure the coincidence score to measure the sample importance, (2) To retrain an individual model again with the augmented samples. It would be convenient to identify the samples in the early to mid stages of the training process or could follow an active learning scheme where models are continuously learned based on the sample's importance.
- There is a chance of error propagation from the first stage to the second stage, such as due to label noise, which can impact the learning process.
- In this paper, we introduce a coincidence score criterion that is based on model agreements of a deep ensemble system. The main limitation of the method is the effort it takes in building the ensembles, which are in general computationally expensive.

4.7. Broader impact

The proposed work in this paper improves the baseline performance using an ensemble-based technique. Along with performance metrics, we also study the entropy of the models. This is very important in application areas such as food recognition to have models that are more certain about their decisions. In recent years, more and more DL-based solutions get closer to regular usage and each misstep can have serious implications. Therefore, it is important to make algorithms learn well and more importantly generalize well to the target problem.

5. Conclusions and future lines

The performance of DL models is impacted directly by the training data. Deep networks are often over-parameterized and suffer from over- and under-confident estimates. It is therefore important to carefully study the behaviour of models towards different data. Of late, the study of individual samples has been well documented which affects the way the networks learn. In this paper, we propose a coincidence score that uses model agreement of deep ensembles to capture the hard samples. Once these samples are identified, we do controlled DA in order to learn these hard samples. We investigate this method in food recognition where the presence of hard samples is much higher due to the nature of images that are captured in the real world. However, the proposed method can be used in any other domain where hard samples may be present. We validated our proposal with several experiments and achieved better results compared to the baseline and random selection methods. Using single-stage pipelines and exploring active learning is a potential future direction. Our direction of future work is related to combining our approach with other decision-making criteria for identifying and improving the learning of hard samples. Another potential direction for future work is to study the influence of label noise in the learning of hard samples.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgements

This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00), CERCA Programme/Generalitat de Catalunya, PID2022-141566NB-I00 (AEI-MICINN), and Agencia Nacional de Investigación *y* Desarrollo de Chile (ANID) (Grant No. FONDECYT INICIACIÓN 11230262). B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

[1] W. Min, S. Jiang, L. Liu, Y. Rui, R. Jain, A survey on food computing, ACM Comput. Surv. 52 (5) (2019) 1–36.
[2] B. Nagarajan, R. Khatun, M. Bolaños, E. Aguilar, L. Angelini, M. El Kamali, E. Mugellini, O.A. Khaled, N. Boqué, L. Tarro, et al., Nutritional monitoring in older people prevention services, in: Digital Health Technology for Better Aging: A Multidisciplinary Approach, Springer, 2021, pp. 77–102.
[3] W. Wang, W. Min, T. Li, X. Dong, H. Li, S. Jiang, A review on vision-based analysis for automatic dietary assessment, Trends Food Sci. Technol. (2022).
[4] Q. Thames, A. Karpur, W. Norris, F. Xia, L. Panait, T. Weyand, J. Sim, Nutrition5k: Towards automatic nutritional understanding of generic food, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8903–8911.
[5] G.A. Tahir, C.K. Loo, A comprehensive survey of image-based food recognition and volume estimation methods for dietary assessment, in: Healthcare, Vol. 9, no. 12, Multidisciplinary Digital Publishing Institute, 2021, p. 1676.
[6] L.M. Amugongo, A. Kriebitz, A. Boch, C. Lütge, Mobile computer vision-based applications for food recognition and volume and calorific estimation: A systematic review, in: Healthcare, Vol. 11, no. 1, Multidisciplinary Digital Publishing Institute, 2023, p. 59.
[7] W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, S. Jiang, Large scale visual food recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2023).
[8] Y. Wang, J.-j. Chen, C.-W. Ngo, T.-S. Chua, W. Zuo, Z. Ming, Mixed dish recognition through multi-label learning, in: Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities, 2019, pp. 1–8.
[9] W. Min, L. Liu, Z. Luo, S. Jiang, Ingredient-guided cascaded multi-attention network for food recognition, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 1331–1339.
[10] J. Ródenas, B. Nagarajan, M. Bolaños, P. Radeva, Learning multi-subset of classes for fine-grained food recognition, in: Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management, 2022, pp. 17–26.
[11] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[12] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 843–852.
[13] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2017) 1452–1464.
[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[15] Y.-H. Liao, A. Kar, S. Fidler, Towards good practices for efficiently annotating large-scale image classification datasets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4350–4359.
[16] T. Tommasi, N. Patricia, B. Caputo, T. Tuytelaars, A deeper look at dataset bias, in: Domain Adaptation in Computer Vision Applications, Springer, 2017, pp. 37–55.
[17] S. Fabbrizzi, S. Papadopoulos, E. Ntoutsi, I. Kompatsiaris, A survey on bias in visual datasets, Comput. Vis. Image Underst. 223 (2022) 103552.
[18] R. Reed, R.J. MarksII, Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, Mit Press, 1999.
[19] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, Front. Comput. Sci. 14 (2) (2020) 241–258.
[20] M. Ganaie, M. Hu, et al., Ensemble deep learning: A review, 2021, arXiv preprint arXiv:2104.02395.
[21] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Advances in Neural Information Processing Systems, Vol. 30, 2017.
[22] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.
[23] J. Moon, J. Kim, Y. Shin, S. Hwang, Confidence-aware learning for deep neural networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 7034–7044.
[24] D. Csiba, P. Richtárik, Importance sampling for minibatches, J. Mach. Learn. Res. 19 (1) (2018) 962–982.
[25] G. Alain, A. Lamb, C. Sankar, A. Courville, Y. Bengio, Variance reduction in SGD by distributed importance sampling, 2015, arXiv preprint arXiv:1511.06481.
[26] T.B. Johnson, C. Guestrin, Training deep models faster with robust, approximate importance sampling, Adv. Neural Inf. Process. Syst. 31 (2018).
[27] D. Meng, Q. Zhao, L. Jiang, A theoretical understanding of self-paced learning, Inform. Sci. 414 (2017) 319–328.
[28] G. Hacohen, D. Weinshall, On the power of curriculum learning in training deep networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 2535–2544.
[29] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 41–48.
[30] B. Wang, M. Qiu, X. Wang, Y. Li, Y. Gong, X. Zeng, J. Huang, B. Zheng, D. Cai, J. Zhou, A minimax game for instance based selective transfer learning, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 34–43.
[31] A. Katharopoulos, F. Fleuret, Not all samples are created equal: Deep learning with importance sampling, in: International Conference on Machine Learning, PMLR, 2018, pp. 2525–2534.
[32] R. Baldock, H. Maennel, B. Neyshabur, Deep learning through the lens of example difficulty, Adv. Neural Inf. Process. Syst. 34 (2021).
[33] S. Hooker, A. Courville, G. Clark, Y. Dauphin, A. Frome, What do compressed deep neural networks forget? 2019, arXiv preprint arXiv:1911.05248.
[34] Z. Jiang, C. Zhang, K. Talwar, M.C. Mozer, Characterizing structural regularities of labeled data in overparameterized models, in: International Conference on Machine Learning, PMLR, 2021, pp. 5034–5044.
[35] C. Agarwal, D. D'souza, S. Hooker, Estimating example difficulty using variance of gradients, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10368–10378.
[36] E. Aguilar, B. Nagarajan, R. Khantun, M. Bolaños, P. Radeva, Uncertainty-aware data augmentation for food recognition, in: 2020 25th International Conference on Pattern Recognition, ICPR, IEEE, 2021, pp. 4017–4024.
[37] M.A. Bansal, D.R. Sharma, D.M. Kathuria, A systematic review on data scarcity problem in deep learning: solution and applications, ACM Comput. Surv. 54 (10s) (2022) 1–29.
[38] N.E. Khalifa, M. Loey, S. Mirjalili, A comprehensive survey of recent trends in deep learning for digital images augmentation, Artif. Intell. Rev. (2021) 1–27.
[39] X. Chao, L. Zhang, Few-shot imbalanced classification based on data augmentation, Multimedia Syst. (2021) 1–9.
[40] S.S. Mullick, S. Datta, S. Das, Generative adversarial minority oversampling, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1695–1704.
[41] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, Autoaugment: Learning augmentation strategies from data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 113–123.
[42] S. Lim, I. Kim, T. Kim, C. Kim, S. Kim, Fast autoaugment, Adv. Neural Inf. Process. Syst. 32 (2019).
[43] R. Hataya, J. Zdenek, K. Yoshizoe, H. Nakayama, Faster autoaugment: Learning augmentation strategies using backpropagation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, Springer, 2020, pp. 1–16.
[44] E. Aguilar, P. Radeva, Class-conditional data augmentation applied to image classification, in: International Conference on Computer Analysis of Images and Patterns, Springer, 2019, pp. 182–192.
[45] R. Rahaman, et al., Uncertainty quantification and deep ensembles, Adv. Neural Inf. Process. Syst. 34 (2021).
[46] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, S. Jiang, Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 393–401.

[47] D. Sahoo, W. Hao, S. Ke, W. Xiongwei, H. Le, P. Achananuparp, E.-P. Lim, S.C. Hoi, FoodAI: Food image recognition via deep learning for smart food logging, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2260–2268.

[48] Z. Zahisham, C.P. Lee, K.M. Lim, Food recognition with resnet-50, in: 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology, IICAIET, IEEE, 2020, pp. 1–5.

[49] B. Arslan, S. Memis, E. Battinisonmez, O.Z. Batur, Fine-grained food classification methods on the UEC food-100 database, IEEE Trans. Artif. Intell. (2021).

[50] N. Martinel, G.L. Foresti, C. Micheloni, Wide-slice residual networks for food recognition, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, pp. 567–576.

[51] L. Deng, J. Chen, Q. Sun, X. He, S. Tang, Z. Ming, Y. Zhang, T.S. Chua, Mixed-dish recognition with contextual relation networks, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 112–120.

[52] L. Meng, L. Chen, X. Yang, D. Tao, H. Zhang, C. Miao, T.-S. Chua, Learning using privileged information for food recognition, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 557–565.

[53] J. He, F. Zhu, Online continual learning for visual food classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2337–2346.

[54] H. Zhao, K.-H. Yap, A.C. Kot, Fusion learning using semantics and graph convolutional network for visual food recognition, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1711–1720.

[55] E. Aguilar, B. Nagarajan, R. Khatun, M. Bolaños, P. Radeva, Uncertainty modeling and deep learning applied to food image analysis, in: International Joint Conference on Biomedical Engineering Systems and Technologies, Springer, 2020, pp. 3–16.

[56] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15.

[57] Q.A. Wang, Probability distribution and entropy as a measure of uncertainty, J. Phys. A 41 (6) (2008) 065004.

[58] Y. Kawano, K. Yanai, Automatic expansion of a food image dataset leveraging existing categories with domain adaptation, in: Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13, Springer, 2015, pp. 3–17.

[59] J. He, F. Zhu, Exemplar-free online continual learning, 2022, arXiv preprint arXiv:2202.05491.

[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[61] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[62] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q.V. Le, Learning data augmentation strategies for object detection, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16, Springer, 2020, pp. 566–583.

[63] E.D. Cubuk, B. Zoph, J. Shlens, Q.V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.

[64] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (7) (1998) 1895–1923.

[65] Q. McNemar, Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika 12 (2) (1947) 153–157.

CHAPTER 5

Task Understanding: Applications to Food Recognition

Introduction

DL applications thrive on the fundamental understanding of the domain and the specific tasks at hand. This section explores this concept through the challenges and solutions encountered in food recognition, a complex computer vision task. While food recognition shares common challenges with other computer vision tasks, it also presents unique domain-specific hurdles. Two significant difficulties exist in adapting food recognition for real-world applications - addressing visually similar food categories and handling multi-label classification. We address these challenges using specific strategies tailored to the food recognition domain. We propose a multi-stage transfer learning framework for multi-label recognition and a subset learning-based method for fine-grained recognition.

By investigating the following questions, we emphasize the importance of understanding the task and domain in achieving successful food recognition (Numbers in brackets represent the corresponding paper).

- *What strategies can we employ to enable models to handle fine-grained food classes? (#1, #2)*
- *How can we effectively learn to classify images containing multiple food classes? (#3)*

By addressing these domain-specific challenges and exploring the role of task understanding, we aim to develop more robust and accurate DL models for food recognition applications.

Works Published Within This Chapter

1. *Javier Ródenas*, **Bhalaji Nagarajan***, Marc Bolaños, and Petia Radeva. "Learning Multi-Subset of Classes for Fine-Grained Recognition." ACMMM-W, 2022.*
2. *Jesús M. Rodríguez-de-Vera, Pablo Villacorta, Imanol G. Estepa, Marc Bolaños, Ignacio Sarasúa, **Bhalaji Nagarajan**, and Petia Radeva. "Dining on Details: LLM-Guided Expert Networks for Fine-Grained Food Recognition." ACMMM-W, 2023.*
3. ***Bhalaji Nagarajan**, Eduardo Aguilar, and Petia Radeva. "S2ML-TL Framework for Multi-label Food Recognition." ICPR-W, 2021.*

Learning Multi-Subset of Classes for Fine-Grained Food Recognition

Javier Ródenas*
Dept. de Matemàtiques i Informàtica
Universitat de Barcelona
Barcelona, Spain
javierrodenascumplido@gmail.com

Bhalaji Nagarajan*
Dept. de Matemàtiques i Informàtica
Universitat de Barcelona
Barcelona, Spain
bhalaji.nagarajan@ub.edu

Marc Bolaños
AIGecko Technologies SL
Barcelona, Spain
marc.bolanos@aigecko.com

Petia Radeva
Dept. de Matemàtiques i Informàtica
Universitat de Barcelona
Barcelona, Spain
Computer Vision Center
Cerdanyola (Barcelona), Spain
petia.ivanova@ub.edu

ABSTRACT

Food image recognition is a complex computer vision task, because of the large number of fine-grained food classes. Fine-grained recognition tasks focus on learning subtle discriminative details to distinguish similar classes. In this paper, we introduce a new method to improve the classification of classes that are more difficult to discriminate based on *Multi-Subsets learning*. Using a pre-trained network, we organize classes in multiple subsets using a clustering technique. Later, we embed these subsets in a multi-head model structure. This structure has three distinguishable parts. First, we use several shared blocks to learn the generalized representation of the data. Second, we use multiple specialized blocks focusing on specific subsets that are difficult to distinguish. Lastly, we use a fully connected layer to weight the different subsets in an end-to-end manner by combining the neuron outputs. We validated our proposed method using two recent state-of-the-art vision transformers on three public food recognition datasets. Our method was successful in learning the confused classes better and we outperformed the state-of-the-art on the three datasets.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**.

KEYWORDS

Fine-grained food recognition, Hierarchical learning, Learning subsets of classes

*Joint first authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MADiMa '22, October 10, 2022, Lisboa, Portugal
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9502-1/22/10…$15.00
https://doi.org/10.1145/3552484.3555754

ACM Reference Format:
Javier Ródenas, Bhalaji Nagarajan, Marc Bolaños, and Petia Radeva. 2022. Learning Multi-Subset of Classes for Fine-Grained Food Recognition. In *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management (MADiMa '22), October 10, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3552484.3555754

1 INTRODUCTION

Human beings are inherently skillful in fine-grained reasoning making them capable of easily distinguishing different kinds of similar objects such as identifying different varieties of dogs or cars. Fine-grained image analysis aims to do the same - to teach computers to "see" in a fine-grained manner. Classical image recognition addresses all the target classes in a similar way and has achieved impressive results on par with human performances [17]. However, the fine-grained approach establishes a more detailed methodology focusing on classifying a subset of categories within the entire set of classes [59]. It tackles the problem of image classification by focusing on differentiating between hard-to-distinguish object classes [3, 36]. Fine-grained data have high intra-class and small inter-class variances making it harder for the recognition algorithm to distinguish very similar categories.

The key to fine-grained recognition is to create automatic methods to identify the discriminative regions from the images. Transfer learning on fine-grained data relies on the networks to discover the subtle discriminant details that differentiate the subordinate classes [14]. However, this approach has two main shortcomings: on one hand, the network focuses on the most obvious distinction between classes overlooking more subtle inter-class variations, and on the other hand, all classes are treated in the same way, without applying any hierarchical or similarity-based distinction. Since misclassifications tend to occur only in the most similar classes, we cannot treat all classes equally [29].

Human cognition relies on identifying the discriminating areas in a gradual way. Several hierarchical approaches rely on this principle, where the local features are gradually learned from coarse to fine [11]. Another approach is to use part-based localization methods, which extract the differentiating local features instead of relying

Figure 1: Example of fine-grained food classes from Food1K dataset.

on global features [48, 64]. Discriminative region identification and feature learning are intertwined and reinforce each other [21].

Food image recognition is an ideal fine-grained problem. Food appears differently with different cooking methods and cuisines. Food such as *carpaccio & tomato salad* are visually quite similar; however, are entirely different set of food classes. On the other hand, it is also possible to have classes that are semantically very similar to each other. Fig. 1 shows an example of food classes that are very similar to each other but differ in the ingredients present in them. All classes shown in the image are a variation of *fried beans* with some images having different meat in them. It is very important for the algorithms to learn this differentiation.

In this work, we tackle the problem of fine-grained food recognition using the Multi-Subset learning framework. The idea of our proposed method is inspired from the Multi-Task Learning (MTL) [15, 23, 46, 51, 66]. MTL models usually have multiple classification heads that arise naturally by design intending to solve multiple tasks at the same time. Following a similar line of thought, we combine different classification heads or outputs, each of them targeting a different subset of similar classes in order to help the main task of fine-grained image classification. We differ from the MTL algorithms, where we use different classifiers that specialize in separating similar classes. Since different classifiers analyze data from the same domain, we force them to share part of their networks similar to that of an MTL model.

We further address two important design questions in this work: (1) How do we select or group similar classes? We use a clustering algorithm on the output confusion of a pre-trained network in order to create a subset of similar classes. (2) When generating models with multiple outputs or classifiers, the main question is how to combine them to make the final decision. Voting procedures [12], heuristics [9] or estimating the average likelihood of the classifiers [47] are widely used in literature. We integrate the output of classifiers in an end-to-end framework by using a fully connected layer. The layer is used to learn the optimal weights in order to combine all the outputs provided by the classifiers. We combine the output of subset-specific classifiers along with the other classes that do not necessarily belong to a hierarchy of subsets. We identify the problem of inter-subset and intra-subset data imbalance and propose a technique to address this problem.

Our contributions can be summarized as follows:

- We introduce a new method that extracts class similarities and structures the dataset in a hierarchical fashion for applying fine-grained recognition,
- We use Multi-Subset and generic classifiers that are trained and integrated into an end-to-end framework in order to determine the proper weighting of the class outputs,
- We propose a technique for solving the imbalance problem of fine-grained subsets, making our method highly scalable and applicable to datasets of different sizes,
- Our results outperform the state-of-the-art methods on three public fine-grained (in particular, food) datasets with high complexity, and
- An extensive ablation study proves the benefits of each of the components of our method.

The rest of the paper is structured as follows. In Section 2, we discuss the most recent and relevant literature. In Section 3, we define our proposed method and its different components. We brief the evaluation pipeline in Section 4 followed by detailed discussions on Section 5. Finally, we provide the concluding remarks and future works in Section 6.

2 RELATED WORK

We briefly discuss the most recent and relevant work on fine-grained classification and food recognition.

2.1 Fine-Grained Image Recognition

Most work on fine-grained image analysis is split in fine-grained image recognition [60] and fine-grained image retrieval [58]. Fine-grained image recognition focuses on learning detailed features from images to support the final classification. There are two main research strategies [55] for fine-grained recognition: (1) recognition by localization-classification subnetworks and (2) recognition by end-to-end feature encoding. First class of algorithms includes detection or segmentation methods [38], utilizing deep filters [30] and leverage attention mechanisms [31]. Algorithms in the later family focus on performing high-order feature interactions [42], designing specific loss functions [22] and progressive multi-granularity training of Jigsaw patches [19]. Models integrating both the object-attention and object-part attention discrimination [48] allowed better supervision in the classification of images into subcategories. Similarly, a CNN scheme to classify the images based on feature fusion from different parts of the image improved the model performance [61]. Combining the vision model that extracts image features using a convolutional neural network with a language model that adds semantic representations helped in better image classifications [29]. Several detailed and comprehensive summaries on fine-grained image analysis can be found in [26, 54–56, 67, 68].

2.2 Hierarchical and Multi-Task Classifiers

Hierarchical methods have been very successful in dealing with fine-grained tasks. Subsets identified from visually similar features using a similarity tree [24] were used in fine-grained classification. The hierarchical classification was done using a support vector machine classifier for each subset. In the same way, CNN was used to identify the subsets from the visual similarities [25]. Both the above mentioned techniques used a subset selector to identify the

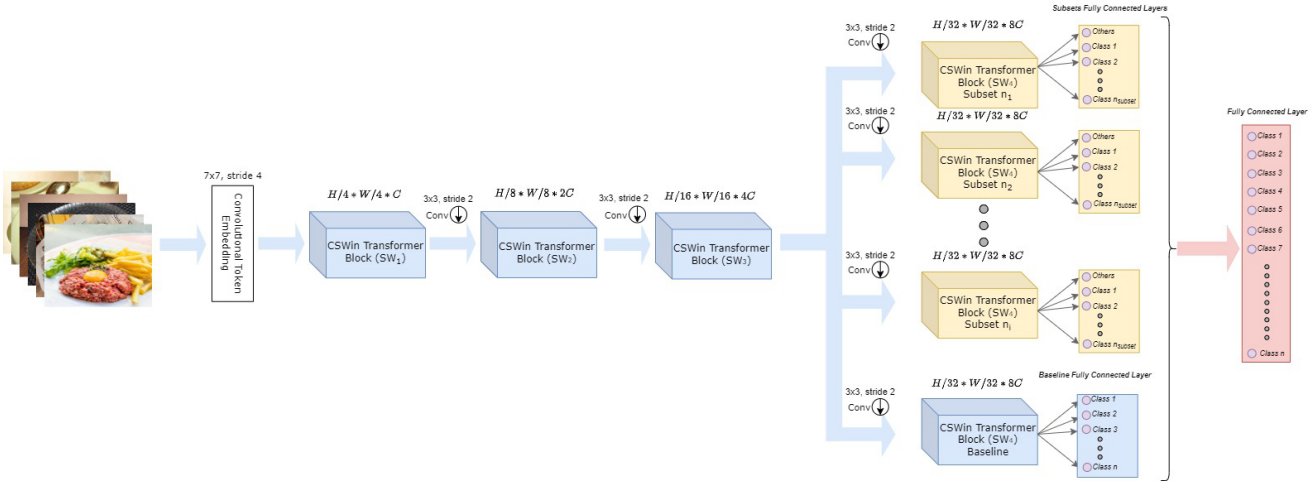


Figure 2: Overview of our proposed architecture: the first part of the architecture (in blue) contains several blocks that capture the general representation of the data; the second part has multiple branches, containing a block and a FC layer for each of the subsets identified in the dataset (in yellow) as well as a generic last branch that classifies all the classes (in blue); the last part (in red) is a FC layer joining all the knowledge extracted from the previous layers giving the final classification. Here, CSWin architecture was used as a backbone for visualization purposes.

most appropriate classifier. In contrast, our method does not focus on visual features, but on the model errors, and the final decision is taken without using a subset selector. We add a FC layer to the model so that it learns how to combine the different classifiers. Several methods were used to identify the subsets. Object-attention [24, 25, 48] and merging features from different parts of the images [61] are common methods. We differ from these methods as we rely on the confusion matrix to focus on those classes that are most confused with each other.

The Multi-Subset classifier which is the backbone of our proposed method is inspired by the multi-task learning frameworks. The MTL approaches deal with multiple attributes associated with a certain sample and build models that allow predicting all attributes (or tasks) at the same time. Normally, in deep learning approaches, a single network is trained that learns features from less to more specific as it progresses in depth. In the case of MTL, the network is trained with data from multiple tasks simultaneously. MTL can be realized using a network that shares blocks along multiple tasks where each task has its own final output and the different tasks share features from the same blocks [66]. MTL networks based on hierarchies where the input of each task depends on the output of the previous task [15] have been successful. Other structures where each task has its own complete model, but each of the tasks takes information from the other models [23, 46, 51] are also prevalent. Complete MTL summary can be seen in [13, 53, 65]. We propose our model based on MTL architecture by replicating the last block as many times as the number of subsets identified where each last block learns only features from the classes belonging to that subset.

2.3 Food Recognition

Food recognition has been an active area of research for several years [50] and it has picked up with the advent of deep learning algorithms [43]. With a more digital footprint, larger datasets are possible by web-scrapping and the recent food datasets involve a large number of images [10, 34]. This has also been a reason for more powerful food recognition algorithms. Transfer learning has been successfully used in food recognition using different architectures [4, 63]. Owing to the complexity of food images, it is necessary for the algorithms to understand the semantics in food images to make better classifications [33, 40]. Attention networks have been widely used to learn both the local and global features [44]. Apart from classification, food images are used in recognizing ingredients as well [7] and this information has helped in learning better representations of food images [16, 41]. Recently self-supervised learning algorithms are used in food recognition [5] to tackle the problem of data dependency and also to learn better image representations. Food localization and food segmentation are also widely used in understanding the nature of food images [1, 2]. An online continuous learning framework [27] has been adapted to food classification where the data from data streams is learned without forgetting the old data. With larger and more complex food datasets, several literature focuses on fine-grained classification [4, 37, 45, 57, 69]. In this proposed work, we introduce a multi-subset learning framework to tackle the fine-grained nature of food images.

3 MULTI-SUBSET METHOD FOR FINE-GRAINED FOOD RECOGNITION

The main objective of our proposed method is to identify and integrate different classifiers which specialize in discriminating classes that share some common traits and organize target classes into groups of similar nature. The highlight of our method is that it is capable of generating subsets and hierarchies from the data itself without the need of working with datasets that are structured in this way. The pipeline of our model can be seen in Fig. 2.

Figure 3: Example of an identified subset discovered by the hierarchical clustering. The grouped class names are "radish fried egg", "savory" and "egg pie". Note that the original class names are in Chinese and have been converted to English using Google translate.

In general lines, our proposal works as follows: first of all, a model baseline is trained on the whole set of classes. Later, even if the dataset at hand does not contain class groupings or hierarchies, our method allows the creation of these subsets. To do so, similar classes are detected based on the confusion matrix of the baseline classifier and organized into subsets of similar classes. Then, the new model is constructed by expanding the baseline with blocks specialized in classifying the subsets of classes. In this way, the shared blocks are used for the general features and thus optimize the training of the network. Finally, the output of the different blocks in charge of the subsets is combined using a Fully Connected (FC) layer. We show each component of our proposed method below.

3.1 Classes Subset Extraction

The first step of our method is to generate the subsets of data. For this purpose, we train a model for the whole set of classes. We refer to this as the **baseline model**. We use the inference of the baseline model to compute the confusion matrix, M, that gives us the groups of classes that are confused. Confused classes are those that tend to be the most similar. These classes are clustered using a hierarchical clustering technique (implementation details are provided in Section 4.3.1), where the distance of class c_i and c_j is defined in terms of the confusion between them:

$$d(c_i, c_j) = 1 - \frac{1}{2}(M_{ij} + M_{ji}).$$

The confusion matrix M is normalized so that $d(c_i, c_j)$ is bounded between 0 and 1. We use hierarchical clustering directly on the normalized confusion matrix to convert the class confusions into distances. After clustering the classes, we apply a threshold value to group only those classes that really have clear resemblances. All other classes are left ungrouped. We illustrate a subset of three classes detected by the hierarchical clustering in Fig. 3. The subset comprises only very similar classes highlighting the significance of our method.

We further show the clustering using a t-SNE representation in Fig. 4. We highlight two groups of classes in the t-SNE plot showing how they are very close to each other.

3.2 Multi-Subset Classifier Construction

Given the baseline model and the detected Multi-Subsets of classes, we have to create our multi-head model architecture. First, the last block of the baseline model is replicated as many times as the number of subsets detected plus one. The extra replicated block will be frozen during the learning procedure and will keep the outputs and weights that were already learned during the baseline training, thus, providing an output for each of the individual classes in the dataset. All the newly created blocks correspond to the yellow part and the frozen extra block corresponds to the last blue block of the proposed method in Fig. 2.

To train each subsets classifier, a dataset is created with samples from only the classes within the identified cluster along with an additional class named as *Others*. Class *Others* includes all those classes that do not belong to the specific subset. This allows the classifier to learn to discriminate between the classes in the group and the rest of the classes in the dataset.

In order to emphasize the contribution from the newly added Multi-Subset classifiers and at the same time leverage most of the network knowledge acquired on the baseline training, only newly replicated blocks are trained in this step. The first part of the architecture, which is shared, as well as the general baseline block, remains frozen during this training stage. In this way, each subset classifier computes a cross-entropy loss separately to generate the model loss as follows:

$$CE_i(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{j=1}^{N_{subset_i}} y_j \log(\hat{y}_j), \quad \mathcal{L} = \frac{1}{N_{subsets} + 1} \sum_{i=1}^{N_{subsets}+1} CE_i,$$

where y_j is the ground-truth, $\hat{y}_j$ is the corresponding classifier prediction, CE_i is the cross-entropy of head i, CE_0 is the cross-entropy of the Baseline head, and $N_{subsets}$ is the number of subsets.

3.3 Data Balancing for Multi-Subset Heads Training

Class imbalance raises naturally from fine-grained classification problems. Additionally, considering the structure of each of the proposed subset-specific classifiers, which contains an *Others* class for integrating any class not belonging to the subset, the imbalance issue is even more pronounced. These facts show that our model could suffer from both intra-subset and inter-subset imbalance; intra-subset being the balancing of samples between each of the classes inside the subset as well as with the *Others* class; and inter-subset being the balancing of samples provided between each of the defined subsets as well as with the baseline general classifier.

In order to deal with the inter-subset problem, we propose to balance the number of samples per epoch independently for each subset or classifier. Based on the number of classes n and the median number of samples ms per subset, the epoch data subsampling L is defined as follows:

$$L_{subset} = \frac{ms * n}{k},$$

where k is a factor to avoid memory overload. In our case, we set $k = 8$. Note that proper balancing is crucial so that the Multi-Subset heads are trained properly; heads with a higher number of classes should be trained with a higher number of samples and vice-versa in order to avoid overfitting and achieve equal training of the different heads. In this way, in each epoch the corresponding classifier considers the same number of samples based on the median of the total samples and the number of classes, solving the imbalance

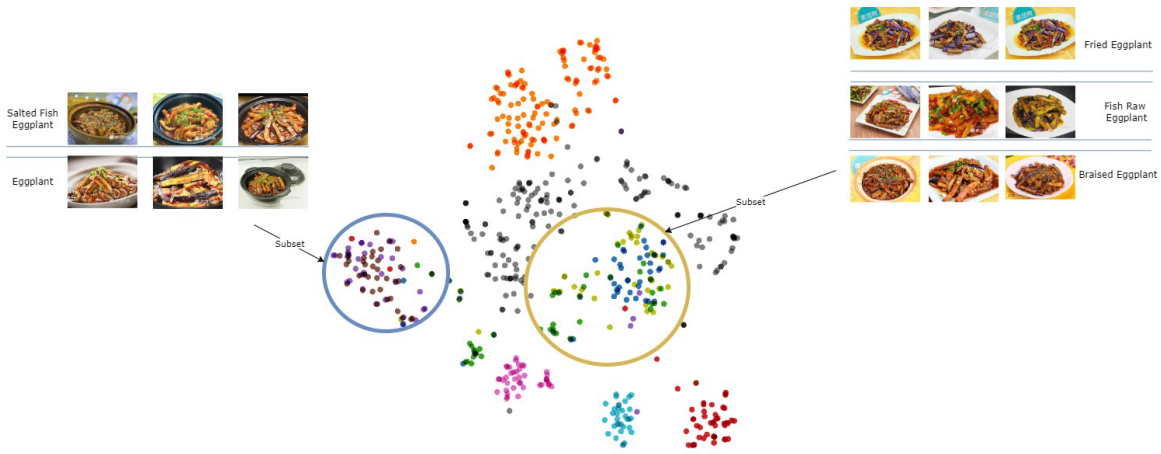


Figure 4: t-SNE sample-wise subsets visualization.

between subsets. If there are more samples or more classes, the number of sampled data is greater and vice versa.

On the other hand, regarding the intra-subset imbalance problem, the randomly selected samples at every epoch are chosen based on the probability of their occurrence in the subset defined in terms of the size of the class in the subset.

3.4 Combining Multi-Head Outputs

The last step of the algorithm focuses on how to combine the information from the different subsets heads and the baseline classifier. In order to avoid heuristics or voting systems, we propose a fully connected (FC) layer to join the different outputs coming from the different heads. Note that for every subclass classifier, the output of the class *Others* has to be connected to all the last FC inputs that do not belong to the corresponding subset block.

To assure that the baseline has more weight in the final decision, we initialize the weights that come from the baseline with an identity matrix. Let $w_{i,j}$ connects output i from the baseline classifier with neuron j. We initialize $w_{i,j}$ to 1 if $i = j$, and 0, otherwise. Note that if there are no other heads that offer a prediction on a certain class because it was not selected inside any of the subsets, the last FC layer will give exactly the same classification as that of the baseline generic classifier. We connect the FC layer with the subsets heads, where the weights are initialized randomly. In this way, the final model (FC layer) learns the weights corresponding to the subsets and becomes the sole-decision maker. The network is trained to use the information from the subsets supporting the predictions of the baseline classifier (see Fig. 5).

4 EXPERIMENTS

In this section, we give a brief overview of the datasets, evaluation metrics, and the evaluation pipeline. We also discuss the different techniques that were used to further boost the performance.

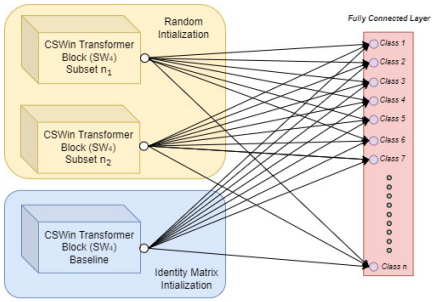


Figure 5: Combination of the Baseline head and the subsets heads connected to the FC layer. The baseline is initialized with the identity matrix and the subsets are initialized randomly.

4.1 Datasets

We use three fine-grained public food datasets to validate our proposed method. For all the datasets, we use the standard data splits.

- **Food101** [8] consists of 101 food categories with 1000 images per category, split into 750 training and 250 validation images.
- **FoodX-251** [34] consists of 251 food categories with 118K training images and 28K validation images.
- **Food1K**, a recent dataset introduced in the ICCV-2021 LargeFoodAI[1] workshop consists of 1000 food categories (a subset from the Food2K [45] dataset). Food1K contains 317K training and 53K validation images.

4.2 Evaluation Metrics

We use Top-1 accuracy in order to validate the proposed hypothesis. Note that two of the three datasets are from food challenges and therefore, they do not have public test sets (only validation sets are provided). Therefore, we report the Top-1 validation performance.

[1]https://foodai-workshop.meituan.com/foodai2021.html#index

Figure 6: Subsets identified from the Food1K dataset. Note that the original class names are in Chinese and have been converted to English using Google translate.

Comparing Methods: We use the fine-tuned backbone architectures as baseline experiments and compare them against the proposed method. We also show the SoTA performance on each dataset and show our improvements.

4.3 Implementation Details

We evaluate our proposed method using two recent and popular vision transformers that achieved excellent results on several computer vision tasks:

- **CSWin** (Cross-Shaped Window) [18] is a transformer that uses self-attention in both the horizontal and vertical strips in order to handle both the global self-attention and the local self-attention challenges. Different variants of CSWin are available; we use the CSWin-Large architecture in our experiments.
- **VOLO** (Vision Outlooker) [62] uses outlook attention to improve the encoding of fine-level features. We use the VOLO-D5 variant in our experiments.

First, we train the baseline blocks using the backbone architectures. We initialize the training using ImageNet-21k [49] pre-trained weights. Next, we train the subset blocks, followed by the FC layer that combines the results of the subsets to give the final classification prediction. All three blocks of the proposed method were trained with a batch size of 128. For all our experiments, we use an image size of 224 x 224. We use Cosine Annealing [39] learning rate scheduler varying from *lr=1e-5* to *lr=1e-6*. All our experiments were done using the PyTorch framework and were carried out on a NVIDIA GeForce RTX 3090 GPU.

4.3.1 Subset Construction. In order to construct the subsets, we used hierarchical clustering with a single linkage. In order to determine automatically the number of subsets of classes in the dendrogram, the largest vertical difference between nodes was located, and a threshold was determined corresponding to the middle of this vertical difference (as used in the sklearn library). The number of subsets is directly proportional to the size of the dataset; the algorithm detected 23 subsets for Food1K. The clustering grouped classes that are related like "tofu", "fried tofu" and "sinister tofu" in the same subset. Similarly, we constructed subsets for Food101 and FoodX-251 experiments and obtained 6 and 13 subsets, respectively. We show the subsets detected on Food1K in Fig. 6.

4.3.2 Building Multi-Subset Classifier. One of the important components of our proposed method is building the multi-Subset classifier heads. For the CSWin architecture, the model input data of size H x W x 3 is fed to a *Convolutional Token Embedding Block* to get $\frac{H}{4}$ x $\frac{W}{4}$ patch tokens where the dimension of each token is C. Next, a *convolution layer* is used to merge CSWin [18] blocks. In the same way, the next blocks are connected having as output size $\frac{H}{2^i+1}$ x $\frac{W}{2^i+1}$. Then, the last CSWin block is replicated as many times as the number of subsets identified plus the Baseline block where all have the same output size ($\frac{H}{32}$ x $\frac{W}{32} \times C$) connected to a *FC layer* which gives the output of every head. Finally, the outputs of the heads will be combined with another *FC layer* that will give the final decision (see Section 3.4).

Similarly for VOLO, given an input of size H x W x 3, VOLO tokenizes the image on smaller patches and uses Outlookers to encode token representation at the fine level (e.g., 28 × 28) extracting the features at each spatial location to generate attention weights for locally aggregating its neighboring features. The outlook attention module computes for each spatial location (i, j) the similarity to all the neighbors within a local window of size K x K centered at (i, j). Then, the outlook module aggregates the projected value representations, which is replicated by N outlook attention (Multi-Head Outlook Attention). In the next step, class-attention layers separate the information from transformers layers and information from image patches. This last block is replicated as many times as the number of subsets where a FC layer gives the output of every head. Finally, a FC layer takes the final decision combining the outputs from the different heads.

4.3.3 Gradient Accumulation and Test Time Augmentation. Along with the above settings, we also applied Gradient Accumulation (GA) in *steps=10* in all the three blocks of our method. In GA, a mini-batch is split and computed through a given number of steps while accumulating gradients. Parameters are updated after processing all of the steps. The sum of batch sizes that are processed in all steps is equal to the original batch size. Since a mini-batch is split into smaller pieces, memory usage is reduced. Hence, GA allows running configured number of steps before the model variables are updated and the gradients are accumulated in order to compute the variable updates. We applied Test Time Augmentation (TTA) [35] consisting of Random Rotation of 15 degrees, Random Horizontal Flip, and Random Resize Crop (0.8 width, 0.8 height). We aggregated the prediction score of different outputs (original image, rotated image, flipped image, and cropped image) to make the final decision.

5 RESULTS AND DISCUSSIONS

Here, we show the experimental results of our proposal in comparison with state-of-the-art methods and then give a detailed analysis on the obtained results.

5.1 Performance Comparison to the SoTA

We show the Top-1 validation accuracy of different experiments in Table 1. We compare our proposed method (Note that experiments of FoodX-251 using Volo-D5 and Food1k are reported without gradient accumulation, still achieving better performances) against the baseline models and show that the proposed method improves in all the datasets and architectures. We also compare our proposed method against the current SoTA literature on the respective datasets. For all the datasets, we show a higher result compared to the existing SoTA models. We obtain a **0.11%** improvement in Food101 and **4.67%** in FoodX-251. As per the leaderboard of Food-101 in *Papers with Code*[2], we would classify on the third position. However, the first two positions [20, 32] use additional data. Hence, we obtained the highest top-1 accuracy for Food101 on image recognition without additional training data. We also achieve higher single model performance on Food1K. Being a challenging dataset, several of the leaderboard models use ensembles of high numbers to produce better results.

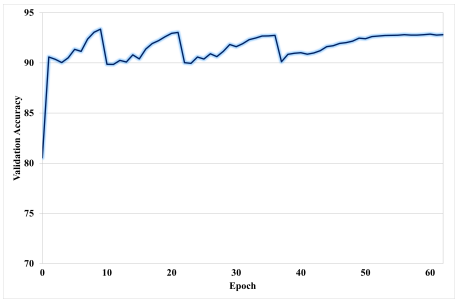
We also compare different architectures for Food101 and FoodX-251 and show that CSWin-L outperforms VOLO-D5 in both experiments. Although VOLO-D5 does not achieve SoTA performance, it is still second compared to the existing performance. We also compare the number of subsets for Food1K experiment using CSWin-L. By using a lower subset of size 10, the validation performance decreases by about 0.03%.

We show the performance curves of the baseline training and also the subsets and subsets+FC training in Fig. 7. We show the validation accuracy against every epoch. We can see that by using the subset block along with the FC layer, the performance of the Baseline improves. We show some of the visual predictions in Fig. 8. These samples were wrongly classified by the baseline model and by using the proposed method, the samples were classified correctly.

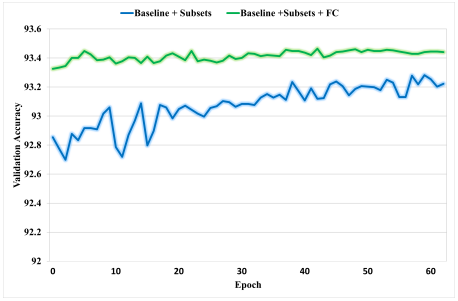
5.2 Ablation study

We study the importance of individual components of our method in Table 2. First, we study the behavior of adding only the subset block (B+S). In this case, we compute the performance by taking the average of the outputs of all subsets to get the final prediction. It can be seen that in all the experiments, the performance at this stage decreases. However, this stage is important as it enables the individual subset heads to learn. By adding the FC as the next step (B+S+FC), the performance of the models increases above the Baseline performance. The performance is further boosted by adding TTA except for VOLO-D5, where the performance dropped by 0.18%. We also show the importance of GA in the experiments by removing GA in the training pipeline and show that in all Food101 experiments and CSWin-L of FoodX-251, the performance is higher with GA. This increase in performance validates our proposed method of using B+S+FC+TTA+GA as the optimal configuration for fine-grained object recognition.

[2]https://paperswithcode.com/sota/fine-grained-image-classification-on-food-101



(a) Baseline Training



(b) Subsets and Subsets+FC Training

Figure 7: Training curves of CSWin-L using Food101 Dataset

Figure 8: Correctly classified samples of tofu (first row), steak (second row), and ham fried rice (third row) using our Multi-Subset Fine-Grained recognition method. The Baseline classified these samples wrongly.

5.3 Limitations

Our method introduces a fine-grained classification that led to substantial food recognition improvement. It consists of three training steps: Baseline training, multiple subsets generations and training of a multi-head architecture, and training of a final FC layer that combines the output of each subset in an end-to-end framework. At this moment, these three steps require to be performed sequentially for the model to be trained properly. Another limitation is the

Table 1: Top-1 Validation Accuracy of the Baseline, our proposed method and SoTA

Dataset	SoTA	Backbone	Subsets Number	Validation Accuracy	
				Baseline	Ours (Rank†)
Food101	93.70 [52]	CSWin-L	6	93.36	**93.81** (1)
		VOLO-D5	6	92.91	93.66 (2)
FoodX-251	75.23 [6]	CSWin-L	13	73.63	**79.90** (1)
		VOLO-D5	13	78.76	79.15 (1)
Food1K	61.20†† [28]	CSWin-L	10	89.90	90.13 (1)
		CSWin-L	23	89.90	**90.16** (1)

†The *Rank* column compares the results of our proposal against the SoTA and the Baseline.
††SoTA was obtained with an online-training framework, hence the low numbers. Better performance is available in the LargeFineFoodAI challenge leaderboard. However, the results are not directly comparable, because the methods in the challenge are based on ensembles.

Table 2: Ablation study (B-Baseline, S-Subsets block, FC-Fully Connected layer, TTA-Test time augmentation, GA-Gradient accumulation)

Dataset	Backbone	Subsets Number	Validation Accuracy				
			B	B + S	B + S + FC	B + S + FC + TTA	GA
Food101	CSWin-L	6	93.36	93.25	93.46	**93.66**	
	CSWin-L	6	93.67	93.46	93.69	**93.81**	✓
	VOLO-D5	6	92.91	91.92	93.18	**93.61**	
	VOLO-D5	6	93.39	93.31	93.34	**93.68**	✓
FoodX-251	CSWin-L	13	73.63	72.84	74.21	**75.20**	
	CSWin-L	13	79.79	78.24	79.48	**79.90**	✓
	VOLO-D5	13	78.76	75.85	**79.37**	79.15	
Food1K	CSWin-L	10	89.90	89.71	90.07	**90.13**	
	CSWin-L	23	89.90	89.74	90.04	**90.16**	

complexity of the model that although showing promising results, training the entire method is time-consuming.

5.4 Societal Impact

Food recognition has multiple applications that determine its high societal value: from food intake monitoring of patients to automatizing self-checking in canteen restaurants. Food intake monitoring is of high importance for many diseases like diabetes, obesity, cardiovascular diseases, osteopathy, Sarcopenia, and other chronic diseases. Still, the food recognition application should be applied in practice always with the supervision of the final user; imagine if the food recognition algorithm confuses and recommend food that a person is allergic to. As in many other Deep learning applications to real problems, technology is very powerful and can provide highly valuable tools used for people and society, but always the final decision of its output should be in charge of human beings.

6 CONCLUSIONS AND FUTURE LINES

Fine-grained recognition is a challenging computer vision problem where the inter-class differences between similar classes are very small. In this paper, we proposed a new method based on Multi-Subsets learning for better separating the fine-grained food classes. Our method efficiently integrates several heads specialized in classifying subsets of similar classes to assure class-specific treatment instead of generic class learning. We validated our proposed method using CSWin and VOLO architectures on three public food datasets and outperformed state-of-the-art methods for fine-grained food recognition without any extra data during the training. Our method is generic and scalable, achieving high performance on datasets containing hundreds of classes. Finally, we identify potential future directions as follows: Focusing on multi-label fine-grained recognition and retrieval, which is a common setting for food images, is an important direction of research. Also, exploring different clustering techniques, such as using unsupervised approaches, and exploring the role of shared blocks in the final performance are some of the interesting architectural questions that need to be answered.

ACKNOWLEDGMENTS

This work was partially funded from the European Union's Horizon 2020 Research and Innovation programme under Open Call budget of the grant agreement No. 857159 (SHAPES), TIN2018-095232-B-C21, SGR-2017 1742 and CERCA Programme / Generalitat de Catalunya. B. Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

REFERENCES

[1] Eduardo Aguilar, Marc Bolaños, and Petia Radeva. 2017. Exploring food detection using CNNs. In *International Conference on Computer Aided Systems Theory*. Springer, 339–347.

[2] Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva. 2018. Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia* 20, 12 (2018), 3266–3275.

[3] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. 2015. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2927–2936.

[4] Berker Arslan, Sefer Memis, Elena Battinisonmez, and Okan Zafer Batur. 2021. Fine-Grained Food Classification Methods on the UEC Food-100 Database. *IEEE Transactions on Artificial Intelligence* (2021).

[5] Nil Ballús, Bhalaji Nagarajan, and Petia Radeva. 2022. Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 655–666.

[6] Emanuel Ben-Baruch, Matan Karklinsky, Yossi Biton, Avi Ben-Cohen, Hussam Lawen, and Nadav Zamir. 2022. It's All in the Head: Representation Knowledge Distillation through Classifier Sharing. *arXiv preprint arXiv:2201.06945* (2022).

[7] Marc Bolaños, Aina Ferrà, and Petia Radeva. 2017. Food ingredients recognition through multi-label learning. In *International Conference on Image Analysis and Processing*. Springer, 394–402.

[8] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European conference on computer vision*. Springer, 446–461.

[9] K. E. Brown and D. A. Talbert. 2019. Heuristically reducing the cost of correlation-based feature selection. In *In Proceedings of the 2019 ACM Southeast Conference*. 24–30.

[10] Xin Chen, Hua Zhou, Yu Zhu, and Liang Diao. 2017. ChineseFoodNet: A large-scale Image Dataset for Chinese Food Recognition. *arXiv preprint arXiv:1705.02743* (2017).

[11] Ying Chen, Jie Song, and Mingli Song. 2022. Hierarchical gate network for fine-grained visual recognition. *Neurocomputing* 470 (2022), 170–181.

[12] W. Chu, Y. Liu, C. Shen, D. Cai, and X. S. Hua. 2017. Multi-task vehicle detection with region-of-interest voting. *IEEE Transactions on Image Processing* 27, 1 (2017), 432–441.

[13] Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796* (2020).

[14] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. 2018. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4109–4118.

[15] Jifeng Dai, Kaiming He, and Jian Sun. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[16] Lixi Deng, Jingjing Chen, Qianru Sun, Xiangnan He, Sheng Tang, Zhaoyan Ming, Yongdong Zhang, and Tat Seng Chua. 2019. Mixed-dish recognition with contextual relation networks. In *Proceedings of the 27th ACM International Conference on Multimedia*. 112–120.

[17] Samuel Dodge and Lina Karam. 2017. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*. IEEE, 1–7.

[18] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. 2022. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12124–12134.

[19] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. 2020. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In *European Conference on Computer Vision*. Springer, 153–168.

[20] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412* (2020).

[21] Jianlong Fu, Heliang Zheng, and Tao Mei. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4438–4446.

[22] Yu Gao, Xintong Han, Xun Wang, Weilin Huang, and Matthew Scott. 2020. Channel interaction networks for fine-grained image categorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34(07). 10818–10825.

[23] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L. Yuille. 2019. NDDR-CNN: Layerwise Feature Fusing in Multi-Task CNNs by Neural Discriminative Dimensionality Reduction, arxiv:1801.08297. arXiv:1801.08297 [cs.CV]

[24] Zongyuan Ge, Chris Mccool, Conrad Sanderson, Alex Bewley, Zetao Chen, and Peter Corke. 2015. Fine-grained bird species recognition via hierarchical subset learning. In *2015 IEEE International Conference on Image Processing (ICIP)*. 561–565. https://doi.org/10.1109/ICIP.2015.7350861

[25] Zongyuan Ge, Christopher Mccool, Conrad Sanderson, and Peter Corke. 2015. Subset Feature Learning for Fine-Grained Category Classification, arxiv:1505.02269. arXiv:1505.02269 [cs.CV]

[26] Chun-feng Guo, Hai-rong Cui, Yu Kun, and Xin-ping Mo. 2019. A Survey of Fine-Grained Image Classification Based on Deep Learning. *DEStech Transactions on Computer Science and Engineering* ica (2019).

[27] Jiangpeng He and Fengqing Zhu. 2021. Online continual learning for visual food classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2337–2346.

[28] Jiangpeng He and Fengqing Maggie Zhu. 2021. Online Continual Learning For Visual Food Classification. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (2021), 2337–2346.

[29] Xiangteng He and Yuxin Peng. 2017. Fine-grained image classification via combining vision and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5994–6002.

[30] Zixuan Huang and Yin Li. 2020. Interpretable and accurate fine-grained recognition via region grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8662–8672.

[31] Ruyi Ji, Longyin Wen, Libo Zhang, Dawei Du, Yanjun Wu, Chen Zhao, Xianglong Liu, and Feiyue Huang. 2020. Attention convolutional binary neural tree for fine-grained visual categorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10468–10477.

[32] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*. PMLR, 4904–4916.

[33] Shuqiang Jiang, Weiqing Min, Linhu Liu, and Zhengdong Luo. 2019. Multi-scale multi-view deep feature aggregation for food recognition. *IEEE Transactions on Image Processing* 29 (2019), 265–276.

[34] Parneet Kaur, Karan Sikka, Weijun Wang, Serge Belongie, and Ajay Divakaran. 2019. Foodx-251: a dataset for fine-grained food classification. *arXiv preprint arXiv:1907.06167* (2019).

[35] Masanari Kimura. 2021. Understanding Test-Time Augmentation. In *International Conference on Neural Information Processing*. Springer, 558–569.

[36] Zhongqi Lin, Shaomin Mu, Feng Huang, Khattak Abdul Mateen, Minjuan Wang, Wanlin Gao, and Jingdun Jia. 2019. A unified matrix-based convolutional neural network for fine-grained image classification of wheat leaf diseases. *IEEE Access* 7 (2019), 11570–11590.

[37] Chengxu Liu, Yuanzhi Liang, Yao Xue, Xueming Qian, and Jianlong Fu. 2020. Food and ingredient joint learning for fine-grained recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 6 (2020), 2480–2493.

[38] Chuanbin Liu, Hongtao Xie, Zheng-Jun Zha, Lingfeng Ma, Lingyun Yu, and Yongdong Zhang. 2020. Filtration and distillation: Enhancing region attention for fine-grained visual categorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34 (07). 11555–11562.

[39] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[40] Niki Martinel, Gian Luca Foresti, and Christian Micheloni. 2018. Wide-slice residual networks for food recognition. In *2018 IEEE Winter Conference on applications of computer vision (WACV)*. IEEE, 567–576.

[41] Lei Meng, Long Chen, Xun Yang, Dacheng Tao, Hanwang Zhang, Chunyan Miao, and Tat-Seng Chua. 2019. Learning using privileged information for food recognition. In *Proceedings of the 27th ACM International Conference on Multimedia*. 557–565.

[42] Shaobo Min, Hantao Yao, Hongtao Xie, Zheng-Jun Zha, and Yongdong Zhang. 2020. Multi-objective matrix normalization for fine-grained visual recognition. *IEEE Transactions on Image Processing* 29 (2020), 4996–5009.

[43] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. 2019. A survey on food computing. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–36.

[44] Weiqing Min, Linhu Liu, Zhiling Wang, Zhengdong Luo, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2020. Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network. In *Proceedings of the 28th ACM International Conference on Multimedia*. 393–401.

[45] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2021. Large scale visual food recognition. *arXiv preprint arXiv:2103.16107* (2021).

[46] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Arxiv:1604.03539 Martial Hebert. 2016. Cross-stitch Networks for Multi-task Learning. arXiv:1604.03539 [cs.CV]

[47] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos, arxiv:1906.06876. arXiv:1906.06876 [cs.CV]

[48] Yuxin Peng, Xiangteng He, and Junjie Zhao. 2017. Object-part attention model for fine-grained image classification. *IEEE Transactions on Image Processing* 27, 3 (2017), 1487–1500.

[49] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972* (2021).

[50] Mohammed Ahmed Subhi, Sawal Hamid Ali, and Mohammed Abulameer Mohammed. 2019. Vision-based approaches for automatic food recognition and dietary assessment: A survey. *IEEE Access* 7 (2019), 35370–35381.

[51] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. 2020. AdaShare: Learning What To Share For Efficient Deep Multi-Task Learning, arxiv:1911.12423. arXiv:1911.12423 [cs.CV]

[52] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. 2021. Grafit: Learning fine-grained image representations with coarse labels. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 854–864.

[53] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence* (2021).

[54] Yafei Wang and Zepeng Wang. 2019. A survey of recent work on fine-grained image classification techniques. *Journal of Visual Communication and Image Representation* 59 (2019), 210–214.

[55] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. 2021. Fine-Grained Image Analysis with Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[56] Xiu-Shen Wei, Jianxin Wu, and Quan Cui. 2019. Deep learning for fine-grained image analysis: A survey. *arXiv preprint arXiv:1907.03069* (2019).

[57] Chee Sun Won. 2020. Multi-scale CNN for fine-grained image recognition. *IEEE Access* 8 (2020), 116663–116674.

[58] Peng Xu, Qiyue Yin, Yongye Huang, Yi-Zhe Song, Zhanyu Ma, Liang Wang, Tao Xiang, W Bastiaan Kleijn, and Jun Guo. 2018. Cross-modal subspace learning for fine-grained sketch-based image retrieval. *Neurocomputing* 278 (2018), 75–86.

[59] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. 2018. Learning to navigate for fine-grained classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 420–435.

[60] Jun Yu, Min Tan, Hongyuan Zhang, Dacheng Tao, and Yong Rui. 2019. Hierarchical deep click feature prediction for fine-grained image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[61] Ye Yu, Qiang Jin, and Chang-Wen Chen. 2018. FF-CMnet: A CNN-Based Model for Fine-Grained Classification of Car Models Based on Feature Fusion. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. https://doi.org/10.1109/ICME.2018.8486443

[62] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. 2021. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112* (2021).

[63] Zharfan Zahisham, Chin Poo Lee, and Kian Ming Lim. 2020. Food recognition with resnet-50. In *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE, 1–5.

[64] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. 2014. Part-based R-CNNs for fine-grained category detection. In *European conference on computer vision*. Springer, 834–849.

[65] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[66] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2016. Learning Deep Representation for Face Alignment with Auxiliary Attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 5 (May 2016), 918–930. https://doi.org/10.1109/tpami.2015.2469286

[67] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. 2017. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing* 14, 2 (2017), 119–135.

[68] Min Zheng, Qingyong Li, Yangli-ao Geng, Haomin Yu, Jianzhu Wang, Jinrui Gan, and Wenyuan Xue. 2018. A survey of fine-grained image categorization. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*. IEEE, 533–538.

[69] Feng Zhou and Yuanqing Lin. 2016. Fine-grained image classification by exploring bipartite-graph labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1124–1133.

Dining on Details: LLM-Guided Expert Networks for Fine-Grained Food Recognition

Jesús M. Rodríguez-de-Vera
Universitat de Barcelona
Barcelona, Spain
jmolinro33@alumnes.ub.edu

Pablo Villacorta
Universitat de Barcelona
Barcelona, Spain
pvillabe7@alumnes.ub.edu

Imanol G. Estepa
Universitat de Barcelona
Barcelona, Spain
igonzaes42@alumnes.ub.edu

Marc Bolaños
AIGecko Technologies SL
Barcelona, Spain
marc.bolanos@aigecko.com

Ignacio Sarasúa
NVIDIA Corporation
Barcelona, Spain
isarasua@nvidia.com

Bhalaji Nagarajan
Universitat de Barcelona
Barcelona, Spain
bhalaji.nagarajan@ub.edu

Petia Radeva
Universitat de Barcelona
Barcelona, Spain
Computer Vision Center
Cerdanyola (Barcelona), Spain
petia.ivanova@ub.edu

ABSTRACT

In the field of fine-grained food recognition, subset learning-based methods offer a strategic approach that groups classes into subsets to guide the training process. Our study introduces a novel approach, referred to as the Dining on Details (DoD), an innovative expert learning framework for food classification. This method ingeniously harnesses the power of large language models to construct subsets of classes within the dataset. The Dining on Details's efficacy is rooted in the robustness of the ImageBind multi-modality embedding space, which can identify meaningful similarities across varied categories. Trained through an end-to-end multi-task learning process, this method enhances performance in the fine-grained food recognition task, showing exceptional prowess with highly similar classes. A key advantage of DoD is its universal compatibility, allowing it to be applied seamlessly to any existing classification architecture. Our comprehensive validation of this method on various food datasets and backbones, both convolutional and transformer-based, reveals competitive results with significant performance gains ranging from 0.5% to 1.61%. Notably, it achieves state-of-the-art results on the Food-101 dataset.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**.

KEYWORDS

Food Image Recognition; Fine-grained Classification; Subset Learning; Large Language Models; Multi-modal Learning

ACM Reference Format:
Jesús M. Rodríguez-de-Vera, Pablo Villacorta, Imanol G. Estepa, Marc Bolaños, Ignacio Sarasúa, Bhalaji Nagarajan, and Petia Radeva. 2023. Dining on Details: LLM-Guided Expert Networks for Fine-Grained Food Recognition. In *Proceedings of the 8th International Workshop on Multimedia Assisted Dietary Management (MADiMa '23), October 29, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3607828.3617797

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MADiMa '23, October 29, 2023, Ottawa, ON, Canada.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0284-6/23/10…$15.00
https://doi.org/10.1145/3607828.3617797

1 INTRODUCTION

Amidst the vast expanse of AI in several tasks, food recognition stands out as a domain intrinsically intertwined with human life. The escalating global population and the necessity for inventive healthcare solutions have propelled food recognition to a position of incremental interest from the point of view of scientific advances as well as their applications to society. Datasets such as Food-101 [5] and FoodX-251 [30] have allowed deep learning-based methods to make notable improvements in food recognition tasks. More recently, the Food-2K dataset [46] offers an extensive collection of labelled food samples, facilitating advancements in this domain. Deep learning-based food recognition finds its application in healthcare nutrition [29], nutrient identification [31] and dietary monitoring [33]. GoCARB [10] and NESTORE [47] are exemplary food recognition applications that have positively impacted the lifestyles of numerous patients. However, despite these achievements, food image data pose several challenges [65] such as high deformability [46], low inter-class and high intra-class variance [73] that hinder the full realization of trained models' potential.

One such challenge is the fine-grained nature [38, 62] of food classes, where subtle variations in ingredients and cooking can lead to semantic differences among very similar food classes. Linguine and Fettuccine, for example, are two different pasta types

Figure 1: Examples of fine-grained food classes.

while the most relevant visual difference is a slight variation in their shape (See Figure 1). We can find numerous such instances amongst food classes that have made researchers ponder further on fine-grained solutions [12, 26, 53, 67]. Current state-of-the-art (SoTA) fine-grained approaches can be summarized as: localization-classification sub-networks [20, 27, 66–68], focus on localizing the important parts of the elements on an image; end-to-end feature encoding [3, 13, 14], emphasizes feature learning; and subset learning [15], approaches that focus on dividing the training data into subsets and training several expert models on each of those subsets. Subset learning-based approaches are the least explored in the literature. However, they have proved to be successful specifically in food-related fine-grained tasks [53].

Previous subset learning methods explore the construction of model-agnostic multi-level hierarchies [8, 15, 24]. DeepME [24] combined the outputs from distinct subset-specific networks, where each subset is derived from a hierarchy of classes extracted from an external knowledge base. FGFR [53] adopted a per-class sub-network expert, compelling the network to concentrate on the most challenging classes based on the confusion matrix of a baseline model. Notably, subset learning methods have exhibited robust classification performance in intricate real-world domains, such as food recognition [53]. However, these studies lack sufficient reliability when it comes to achieving accurate generalization. DeepME [24] depends on external knowledge bases, which limit the approach to only those classes or dishes present in such knowledge bases, which is commonly not the case. FGFR [53], meanwhile, leans on the confusion matrix extracted from a pre-trained model, which requires an additional training stage that doubles the training time and resources needed.

In this work, we explore an alternative novel Large Language Model (LLM) based strategy for fine-grained subset learning. We leverage the capacity of LLMs and multi-modal approaches to obtain, for a given food class or food image, a set of ingredients that could act as priors to create the subsets. LLMs are characterised not only by their ability to act as useful assistants but also by the large amount of data they have been trained on. Falcon-40B [2] has been trained with RefinedWeb [49], which is a huge curated corpus that contains filtered information from the web. As a result, they are able to provide accurate descriptions for most dishes, including less common ones. Models such as BLIP-2 [37] adapt LLMs by allowing prompting visual inputs, further enhancing their capacity to provide better food information. In this paper, we investigate both directions and generate subsets using the obtained lexical information, avoiding the use of external information or extra training while also improving the model performance. On this front, we outline the main contributions of our work as follows: (1) We **propose Dining on Details (DoD)**, an expert-based method to address the challenge of fine-grained food recognition based on lexical priors. DoD modifies the structure of any given baseline to improve its capabilities in distinguishing very similar dishes. (2) We **explore different lexical cues** to characterize the dishes present in a food dataset. We propose a method to leverage SoTA multi-modal methods to compute the (visual) similarity of different dishes based on their textual descriptors. (3) We perform **an extensive quantitative and qualitative analysis** of different food datasets with various backbones (both CNNs and transformer-based architectures). The results highlight the benefit of incorporating verbal cues extracted using LLMs into fine-grained classification.

2 RELATED WORKS

In this section, we present a comprehensive review of the existing literature that closely aligns with the scope of our study.

Fine-grained Visual Recognition (FGVR): The field of FGVR can be broadly categorized based on the learning approach it adopts. *Localization-classification sub-networks* are characterized by a localization submodule in charge of localizing the key parts of the object to generate a local part-level feature vector for each of these parts. Local features are then used as priors for the final classification network [71]. The modelling of the localization sub-network can differ among various approaches, such as relying on existing bounding boxes to identify the parts within [25, 35], leveraging activations derived from convolutional layers [55], applying spatially weighted Fisher vector encoding to the activations [70], and finding peaks in the class response maps to detect sparse areas of interest [12]. *End-to-end feature encoding* is a paradigm that is centred on feature learning [67]. A variety of strategies, such as harnessing high-order feature interactions [13] and crafting specific loss functions [14] have been formulated to cultivate discriminative representations. In discerning fine-grained subcategories, subpixel gradients are implemented to identify minute variations as leveraged by Context-aware Attentional Pooling [3]. SnapMix [26] offers a proportionate semantic mixture that applies Class Activation Mapping [74] to limit label noise throughout the training process. *Subset learning* is a less explored family of methods in which the problem of FGVR is tackled by extracting subsets of similar classes and training an expert model for each subset. A hierarchical subset learning approach is presented in [16], where images are clustered based on deep convolutional features. In addition, they train local SVM classifiers that discriminate between the classes of each subset during inference, as well as classifiers that act as *subset selectors*. FGFR [53] used confusion matrices of baseline models to create subsets. Progressive Transfer Learning [17] presented a similar approach in which subset-specific CNNs are used to learn features specific to each subset. The subset learning approach is particularly beneficial in challenging domains such as food recognition.

Knowledge Distillation and Mutual Learning: Knowledge distillation is a burgeoning subfield of deep learning. It has attracted significant attention from researchers and scholars [21]. In recent developments, mutual learning has risen as an extension of knowledge distillation, where multiple models collaboratively learn from one another [22, 72]. OKDDip [9] executed a two-level distillation during training, involving multiple auxiliary peers and a group leader which guides the learning process. Mutual learning has a

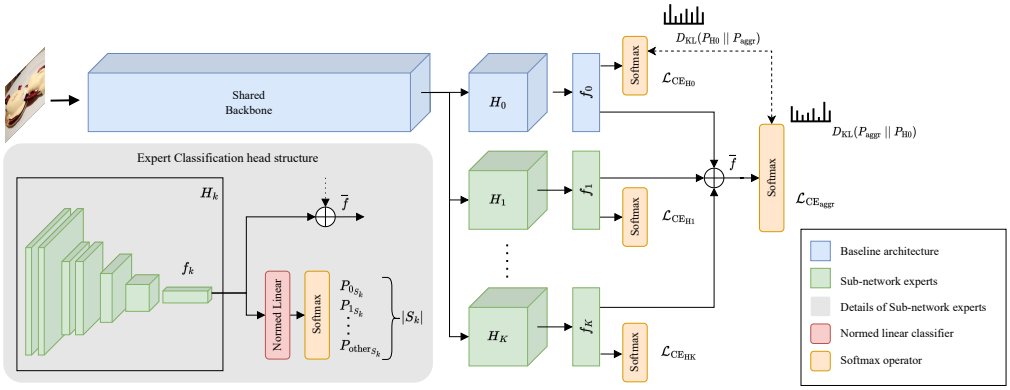


Figure 2: Overview of the architecture modifications of Dining on Details (DoD).

positive impact when used amongst various experts trained on data subsets [48]. This framework employed Kullback-Leibler divergence loss between the distinct probability distributions of the expert networks, resulting in more robust minima and enhanced representations. In GDSMP-Net [32], the authors present a framework that adopts a multi-granularity hierarchical feature fusion learning strategy. This method uses cross-layer self-distillation to increase the robustness of the learned multi-granularity features.

Multi-modality in Image Recognition and LLMs: CLIP (Contrastive Language-Image Pre-training) [51] trains a model which generates embeddings for both images and text in a shared space. This enabled open-vocabulary zero-shot classification, where the model is able to classify images into categories that were not seen during training by measuring the similarity between the image and the text. More recently, ImageBind [19] a multimodal model which provides a joint embedding space for six different data modalities, which include image and text was proposed. According to the reported results [19], the multi-modality nature of ImageBind enables it to excel and achieve SoTA in many cross-modality tasks, such as zero-shot text-based retrieval. The shared embedding space is useful for our task because it helps to close the gap between the prior information modality (text) and the target modality (image). In recent times, large language models (LLMs) have gained momentum in the landscape of AI with the appearance of closed LLMs such as GPT-4 [6] and open-source alternatives like LlaMa [60], or Falcon [50]. Thanks to their ability to generate meaningful text from a prompt, they enable the acquisition of rich textual priors for different tasks. Differing from CLIP and ImageBind, BLIP-2 [37] is an efficient pre-training strategy that bridges the modality gap between frozen image encoders and frozen LLMs using a lightweight Querying Transformer. This approach enables zero-shot tasks such as *instructed* image-to-text generation or captioning.

Food Image Analysis: Food computing [44] has attracted wide attention during the last years for its application to healthcare [34] and self-service restaurants [1]. Automated food image analysis is a sub-field of food computing that has paramount significance thanks to its critical applications in healthcare and well-being, such as nutritional intake analysis [40]. The success of deep learning has advanced food computing methods, as evidenced by recent and notable reviews on the field [34, 40, 58]. Thanks to the increasing digitalization of daily life, the amount of available online data has enabled the creation of large-scale food image datasets such as Food-2k [46] and FoodX-251 [30]. PAR-Net [28] proposed a solution to find discriminative food areas in order to improve the classifier. Other works include additional information or priors such as ingredients information [45] and relationships between the different ingredients [75] to the classification task to enhance the performance. In terms of multi-modality in food image analysis, we can mainly differentiate two types of tasks [46]: cross-modal food image-recipe text retrieval [41] and cross-modal food image or cooking recipe generation [63]. Differently, in food menu recognition [4], the authors used web data to create a language model that is capable of predicting the names of never-seen dishes. In the context of fine-grained food classification, FGFR [53] extracts subsets of classes based on the model confusion matrix and the different experts are modelled as parallel prediction heads within a single model. This method requires an expensive 3-stage training pipeline and a sophisticated re-sampling method that severely affects training time. FGFR can be identified as the closest method to our proposed Dining on Details. In contrast to FGFR, DoD does not require a multistep training pipeline and addresses the creation of subsets from a lexical and model-independent point of view, making it suitable to work directly on top of any baseline method. Instead of having to train twice, DoD relies on the capabilities of LLMs. On the other hand, the usage of mutual learning for FGVR is mainly unexplored. In this work, we explore the adoption of two-way knowledge distillation to enhance the performance of both the main classifier and the aggregation layer, restricting the distillation to the category level.

3 DoD: LLM-GUIDED EXPERT NETWORKS

Dining on Details (DoD) is developed as a universal add-on architecture that can enhance fine-grained food recognition for any base architecture. The main components of the method are outlined in Figure 2. Our proposal is built around the idea of exploiting cross-modality data to seamlessly identify sets of food classes (called *subsets*) that are similar to each other and, consequently, harder to distinguish using simple neural networks. More concretely, we

leverage textual representations of different categories to cluster food classes in the lexical space. This lexical information can be directly obtained from the class labels or by using LLMs (Figure 3), providing richer details for the models. Once the subsets are identified, we add as many sub-network experts ($H_1 ... H_K$) as the number of identified subsets (shown in green in Figure 2) along with the original classification head (H_0 in the diagram). Each sub-network expert consists of a series of layers that are trained to be specialized in separating the classes belonging to that specific subset. Since these classes belong to the same cluster, they are close in the cross-modal latent space, i.e., we enable the network to learn to distinguish among very similar (fine-grained) categories. The subset-specific information is combined with the output of the generic classifier H_0 through a final aggregation layer to produce the final prediction of the network. The models are trained end-to-end in a multi-task learning style, resulting in improved performance in the fine-grained food classification task with respect to the baselines. In order to provide additional supervision to the learning process, we employ a mutual distillation loss between H_0 and the final output of the model. In the subsequent sections, we briefly explain each of the components of the proposed DoD.

3.1 Baseline Architecture

Given any network architecture that is aimed at classification tasks, the model can be divided into a backbone and a classification head, H_0. The input to the backbone is the original image, and the output is a feature map F, which in turn acts as the input to H_0. At the end of the classification head, a standard linear layer outputs n logits (where n is the number of classes), which are passed through a softmax function to obtain the final prediction. The input to the final linear layer is $f_0 \in \mathbb{R}^d$, and it is the representative feature representation of the image within the baseline network.

3.2 Subset Construction

The process of subset construction can be divided into three main steps. First, a textual description of the elements in the dataset is obtained. After that, a lexical encoder is used to translate the textual description into an embedding in the lexical feature space. Lastly, a clustering algorithm is applied to the generated embeddings to create the groups of categories (subsets). In this work, we explore the potential of using different lexical cues to describe the categories of the food dataset. Particularly, we explore three methods to leverage lexical information: (1) class names, (2) ingredients, and (3) image captions. We summarize each of the methods in Figure 3.

Class Names: Class names are the most straightforward way to obtain lexical information. We use the class labels from the dataset directly, preprocess them and pass them to the lexical encoder.

Ingredients: We leverage LLMs to obtain a list of ingredients for a given class name. The specific ingredients of the food class would provide additional information regarding its similarity with other recipes. The class name is used as a prompt to generate the list of ingredients. A text composed of the class name and comma-separated ingredients is used as the textual representation of each class, which is the input to the lexical encoder.

Image Captions: We exploit cross-modality models to obtain the textual representation of images in the training set. Particularly,

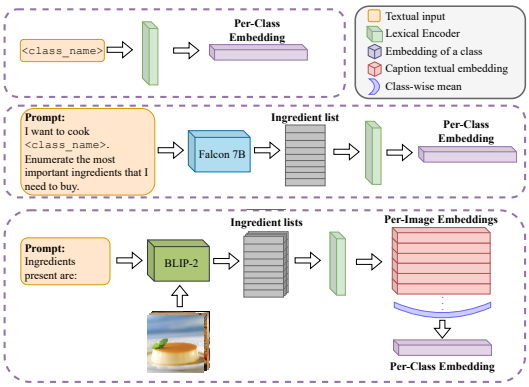


Figure 3: Methods used to obtain the lexical embeddings.

the model is fed with the food images and a textual prompt asking for the visible ingredients in the image (e.g. "Ingredients present in the image are:") A querying transformer is used to query the input (image) using the prompt. This enables the model to produce a list of ingredients, in this case only the visible ones, which is used as a textual representation of each image. The captions are produced per image, obtaining several lists of ingredients for each class (as many as images), which are further processed through the lexical encoder.

In all three cases, the textual representations are processed by a neural lexical encoder which transforms the input sentence into a fixed-length embedding in the lexical feature space. Specifically, for the image captions, the lexical embedding obtained from the image-level captions are grouped by computing the mean of all embeddings of all images of the same class obtaining a final class embedding (shown in Figure 3). Once we obtain a vector representation for each class, we apply clustering techniques to group similar classes into subsets. The process is summarized in Figure 4.

First, we compute the similarity between a pair of classes. This is performed by calculating the cosine similarity between the corresponding embedding vectors. Formally, given two classes c_i and c_j, and their representation in the lexical space l_i and l_j, we compute the cosine similarity as:

$$\text{sim}(c_i, c_j) = \frac{l_i^\top l_j}{|l_i||l_j|} = \cos(l_i, l_j) = \cos(\theta_{ij}).$$

Next, we apply Hierarchical Agglomerative Clustering (HAC) with single-linkage to build a dendrogram and finally group the classes into clusters (subsets). The constructed subsets are non-overlapping i.e. classes do not appear in more than one subset.

3.3 Subset Heads

For each subset of classes, S_k, we introduce a new sub-network expert, H_k, at the end of the backbone. H_k is composed of a series of convolutional or transformer blocks, followed by a cosine similarity classifier (also known as the normed linear layer) [64]. The former is responsible for producing a feature representation $f_k \in \mathbb{R}^d$ that better describes the differences between the classes of that subset. The latter classifies every image into $|\mathcal{S}_k| + 1$, that is,

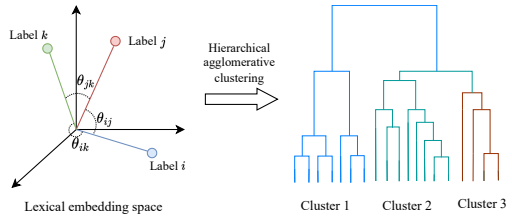


Figure 4: Cluster construction.

the specific class within its corresponding subset or "other" whenever the concept of the picture belongs to a different subset. The newly added sub-network experts are siblings of the original H_0 in the computational graph of the model. Thus, the backbone is a shared image encoder for the different heads (both original head and sub-network experts), and the output of the shared backbone F is the input to all the heads.

It is possible for each sub-network expert head to have a significant imbalance, with most examples during the training falling into the "other" category. To address this, we replace the conventional dot product of linear layers with the following formula to calculate the logit value for the i-th class of the k-th head:

$$s_i^k = \frac{f_k^\top w_i}{|f_k||w_i|},$$

where w_i represents the i-th column of the weight matrix of the normalized linear layer. This approach is inspired by object detection where the "background" class also accounts for most examples [64]. This method avoids the need for complicated weighting or balancing techniques. It should be noted that the i-th class of H_k does not necessarily match the i-th class in the original dataset vocabulary. Except for the final normed linear layer (final classification layer), the architecture of the original head H_0 and the sub-network experts remain the same.

3.4 Aggregating General and Expert Predictions

The final prediction is obtained by aggregating the predictions of the different heads H_k, $0 \leq k \leq K$. We use only the intermediate embeddings $f_0, f_1, \ldots, f_K$ during the aggregation. Particularly, the averaged representation $\overline{f}$ is computed as follows:

$$\overline{f} = \frac{1}{K+1} \sum_{0 \leq k \leq K+1} f_k.$$

$\overline{f}$ encodes relevant information that is capable of distinguishing the elements within the same cluster. The average operation acts as a regularizer. The original H_0, which is trained to predict the specific class of the image like in the baseline model, produces a feature vector f_0 which is refined with specific per-cluster information from the different sub-network experts. $\overline{f}$ is then passed into a linear layer with n outputs, which is used to produce the final output.

3.5 Mutual Learning to Improve Aggregation

Besides the conventional cross-entropy loss, we incorporate a mutual learning loss component that optimizes the mutual information between the probability distributions produced by H_0 and the aggregation layer. This is done by minimizing the Kullback-Leibler divergence [36] between both probability distributions for every input image:

$$\begin{aligned} \mathcal{L}_{\mathrm{ML_{H0}}} &= D_{\mathrm{KL}}(P_{\mathrm{H0}} \| P_{\mathrm{aggr}}) \\ \mathcal{L}_{\mathrm{ML_{aggr}}} &= D_{\mathrm{KL}}(P_{\mathrm{aggr}} \| P_{\mathrm{H0}}). \end{aligned} \tag{1}$$

The reason for adding $D_{\mathrm{KL}}(P_{\mathrm{H0}} \| P_{\mathrm{aggr}})$ is to distill the improved performance of the different heads' aggregation to the original head H_0: The better the input of the aggregation layer, the better the results would be. On the other hand, we also include $D_{\mathrm{KL}}(P_{\mathrm{aggr}} \| P_{\mathrm{H0}})$ to provide additional guidance to the learning process of the aggregation layer and ensure that the performance of the aggregation is at least as good as that of H_0, as well as to mitigate the potentially detrimental effect of noisy signals.

3.6 Overall Training

The backbone, the original classification head, the experts, and the aggregation layer are jointly trained end-to-end. The optimization criterion is a loss function composed of $2 + K + 2$ terms organized into 3 main components:

$$\mathcal{L} = \lambda_1(\mathcal{L}_{\mathrm{CE_{H0}}} + \mathcal{L}_{\mathrm{ML_{H0}}}) + \lambda_2 \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\mathrm{CE_{Hk}}} + \lambda_3(\mathcal{L}_{\mathrm{CE_{aggr}}} + \mathcal{L}_{\mathrm{ML_{aggr}}}),$$

where $\mathcal{L}_{\mathrm{CE}}$ represents the cross-entropy loss, $\mathcal{L}_{\mathrm{ML}}$ are the losses described in (1), and $\lambda_1, \lambda_2, \lambda_3 > 0$ weigh the contribution of the loss of the original classification head, the experts, and the aggregation layer, respectively. We empirically set the values $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 1$ for all the experiments. The gradients flow through the whole network, which forces the experts to learn not only to produce features that are useful for the sub-problem of classifying the classes of a single subset but also to learn to produce features that are useful for the final aggregation. Moreover, this multi-task setting also enables additional information to flow to different heads and the backbone, which enriches the training process.

4 EXPERIMENTS

We first show the implementation details used to validate our proposed method and then present different quantitative and qualitative results to highlight the benefits of DoD.

4.1 Implementation Details

Datasets: We use three fine-grained food classification datasets to evaluate DoD: UECFood-100 [42] consists of 12,740 images grouped into 100 categories; Food-101 [5], with 101 food categories and 101k images; and FoodX-251 [30], having 118k training, 12k validation and 28k test images belonging to 251 fine-grained classes. For all datasets, we use the publicly available data splits (for UECFood-100, we use the first cross-validation fold as a train-test split).

Large Language Models: We use LLMs to produce two different kinds of textual descriptors: retrieving ingredients using the class name and retrieving ingredients directly from the image. For the former, we make use of an instruction-oriented (prompt-based) version of Falcon-7B [2], a *small* open-source LLM speeded up with 8-bit quantization. For the image captioning (image-to-ingredients) task, we use BLIP-2 [37]. BLIP-2 uses a combination of LLM (to

Table 1: Test accuracy (%) of baseline and proposed DoD.

Dataset	Backbone	Baseline	DoD (Ours)	Gain
UECFood-100	EfficientNet-B0	78.43	**79.58**	+1.15
UECFood-100	ResNet-50	77.24	**78.85**	+1.61
UECFood-100	SwinV2-T	77.94	**78.44**	+0.50
Food-101	SwinV2-T	89.96	**90.70**	+0.74
FoodX-251	SwinV2-T	72.89	**74.25**	+1.36

process the text) and a Vision Transformer (to process the images) to generate queries and values for a querying transformer, which outputs the image caption. We use the BLIP-2 with OPT-6.7B [69] as the LLM and COCO-trained ViT-g as the vision model.

Lexical Encoders: We evaluate three different lexical encoders to obtain the embeddings of the different lexical sources: (1) Universal Sentence Encoder (USE) [7], which was one of the earliest deep models that allowed seamless encoding of sentences into vectors; (2) ImageBind-H [19], the biggest available multi-modality model in the ImageBind family, which is capable of mapping the textual representations to a feature space that is shared with images, and (3) a sentence transformer [52] based on MPNet-B [57] (MPNet-B-v2), which is optimized for language understanding tasks (making it suitable for tasks such as text similarity assessment).

Training Details: We use three different backbones to validate the generalization ability of our proposed method: ResNet-50 [23], EfficientNet-B0 [59] (convolutional architectures) and SwinV2-T [39] (transformer architecture). All backbones are initialized with pre-trained ImageNet weights [11]. For all experiments, the configuration of the baseline and the proposed method is kept the same. The only difference is the usage of the sub-network experts and the mutual distillation loss introduced by DoD. We use a one-cycle learning rate scheduler [56], with a cosine annealing cycle of 40 epochs and an annihilation phase of 160 epochs. We set early stopping with a patience of 15 epochs to prevent overfitting. The maximum learning rate was tuned for each architecture to optimize the baseline results on UECFood-100. We use SnapMix [26], a SoTA augmentation technique for fine-grained recognition. The image size is 384×384 for the convolutional networks, and 256×256 for SwinV2-T. Unless specified otherwise, we use ImageBind-H as the lexical encoder and the ingredient list obtained with image captioning (BLIP-2) as the text source for DoD. All experiments are created using PyTorch. We run the UECFood-100 and Food-101 experiments using a single NVIDIA RTX 3090 GPU and the FoodX-251 using a single NVIDIA A100 GPU.

4.2 Results

We compare the results obtained by our proposed method, DoD with the baseline results. Baselines are constructed using normal transfer learning on specific food datasets. We show the test accuracy of different experiments in Table 1 along with the gain in performance comparing the proposed method with the baseline. We validate our proposed method using different experiments to emphasize the independence against both datasets and architectures. We use UECFood-100 against the different backbones and use SwinV2-T against the three datasets. The obtained results clearly highlight that our proposed DoD improves the baseline in every combination of dataset and backbone, with substantial gains ranging from 0.5% (for UECFood-100 and SwinV2-T) to 1.61% (for UECFood-100 and ResNet-50). This proves the effectiveness and generalization of the proposed method for dealing with different backbones and datasets.

Table 2: Comparison of DoD with SoTA methods in Food-101. † =bigger image size. § =subset-based method.

Method	Test Accuracy (%)
Grafit [61] (ICCV'21)	93.7
EffNet-B7 [59] (ICML'19)†	93.0
PMG [8] (CVPR'21)†§	87.5
FGFR [53] (Madima'22)§	93.8
DoD + SwinV2-B§	**94.9**

Comparison with SoTA Results: We compare the results of our proposed DoD with SwinV2-B backbone with other SoTA methods available for Food-101 in Table 2 and achieve substantial improvements over the compared methods. According to the results in *paperswithcode*[1], the obtained test accuracy is the best amongst the methods that do not use additional data during training. Interestingly DoD+SwinV2-T still achieves competitive results amongst most methods which include a much larger backbone. According to the same benchmark, our lightweight configuration would be placed at the 9^{th} position, just below [18] (-0.77%) which uses a much bigger backbone (ViT-L). Regarding the results obtained with FoodX-251, DoD+SwinV2-T outperforms FGFR+CSWin-L [53] (74.25% vs 74.21%) without using gradient accumulation and test-time augmentation (Ablation study results), which uses a much larger backbone. These results validate the benefits of our proposed DoD, where the sub-network experts help in better learning the fine-grained food classes.

4.3 Quantitative Analysis

We study the different lexical sources and their impact on the model performance and also assess the role of mutual learning. We use EfficientNet-B0 trained on UECFood-100 dataset for this analysis.

Exploring Different Lexical Sources: We perform an exhaustive evaluation of the different lexical sources and encoders and report the test accuracy of DoD using the respective combination in Table 3. On all three lexical sources, ImageBind-H is the best-performing lexical encoder. USE is slightly better (+0.01% in accuracy) only for the ingredients list. However, in the case of the other two sources, ImageBind-H achieves clearly superior performance. This suggests that the cross-modal nature of ImageBind's embeddings is highly beneficial to close the gap between our prior information modality (text) and our target modality (image). In addition, for this selected lexical encoder, image captioning is identified as the best text source, followed by the class names and the ingredients list. This behaviour can be explained by the fact that captions are more informative and detailed than the class names, which are general in nature. The ingredients list obtained with LLMs is also more detailed, but it is not as useful as the captions

[1]https://paperswithcode.com/sota/fine-grained-image-classification-on-food-101

Table 3: Comparison of different text sources and language encoders on UECFood-100 using EfficientNet-B0.

Text Source	Language Encoder	Test Accuracy (%)
Class names	USE	78.83
	All MPNet-B-v2	78.92
	ImageBind-H	79.27
Ingredient list	USE	79.15
	All MPNet-B-v2	78.94
	ImageBind-H	79.14
Ingredient list (captioning)	USE	78.90
	All MPNet-B-v2	78.94
	ImageBind-H	**79.58**

in ImageBind, because they can contain non-visible ingredients, which might introduce noise in the embeddings. The obtained results highlight the potential of the proposed DoD. The use of lexical cues has proven to be positive for all combinations studied in Table 3, which supports the potential to exploit textual priors through cross-modality to boost fine-grained food recognition.

Role of Mutual Learning: In Table 4, we analyze the impact of using mutual information on the performance of DoD. We show the results of using both one-way distillation and two-way distillation. It can be seen that mutual learning using two-way distillation is more beneficial than one-way distillation. This is because the whole model is trained together in an end-to-end manner, forcing H_0 to mimic a sub-optimal aggregation layer, which will cause the errors to backpropagate and become larger as the iterations progress. This proves that by training in an end-to-end fashion, the aggregation layer helps improve model learning thanks to the additional supervision task introduced.

4.4 Qualitative Analysis

We use UMAPs [43] and GradCAMs [54] to provide a deeper insight into how DoD successfully differentiates between similar classes.

UMAPs: UMAP is applied to the image embeddings generated by (1) the classification head of the baseline and (2) the sub-network expert of the proposal corresponding to that specific cluster of classes. We compare the UMAP representations obtained from images that belong to the classes of two subsets on the Food-101 dataset and also on the FoodX-251 dataset in Figure 5. As can be seen, our proposed method, DoD generates embeddings that makes it easier to separate similar classes in the feature space. The clouds of points that correspond to different categories of the subset are more separated, and they present a "purer" composition: the number of points belonging to a different class in the wrong point cloud is clearly reduced after applying the proposed method.

GradCams: We show the GradCAMs computed from the classification head of the baseline and the sub-network expert's head in Figure 6 (first four examples for Food-101 and the rest for FoodX-251). All examples shown here are correctly classified by DoD, meanwhile are misclassified by the baseline. As can be seen, the classes predicted by the baseline are conceptually similar to the ground truth and often belong to the same cluster. Thanks to the proposed sub-network experts, the network learns to focus on the parts of the images that are more helpful in identifying the different classes of the subset. For instance, in the second column of Figure 6, the baseline focuses on the rice and predicts the class as "risotto". However, the proposed method focuses on other ingredients to detect the proper class as "Bibimbap". Similarly, in the fourth example, it is very difficult to distinguish the kind of cake by simply focusing on the cake. DoD, however, successfully uses the neighbouring information to make the right prediction as "Strawberry shortcake". Similarly in the fifth example, the baseline focuses on the top of the vegetable, which is the main difference between tomatoes and peppers. In the sixth column, the baseline struggles to focus on useful information. However, the sub-network expert head successfully focuses on the pasta, as its shape is the main difference between fettuccine and other types of pasta. Finally, in the last column, the expert pays attention to the top of the flan, which is the only part visually different from panna cotta (caramel instead of jam). This qualitative analysis points out that the performance gains obtained by our proposal are indeed caused by a better ability of the model to differentiate similar classes.

Table 4: Impact of Mutual Learning.

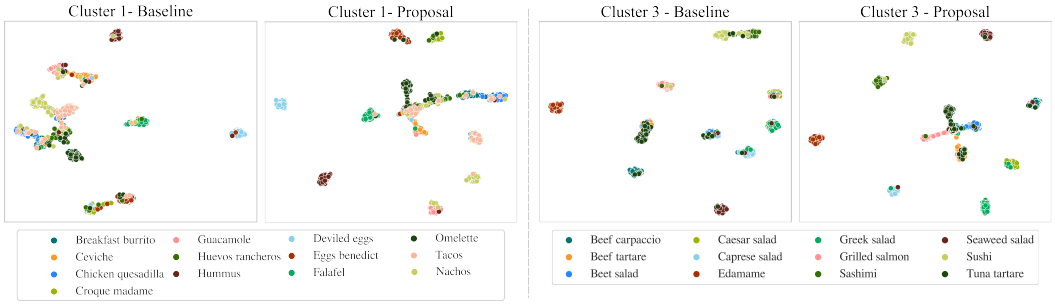
$KL(P_{\text{H0}} \| P_{\text{aggr}})$	$KL(P_{\text{aggr}} \| P_{\text{H0}})$	Test Accuracy (%)
		79.18
✓		79.05
✓	✓	**79.58**

4.5 Limitations

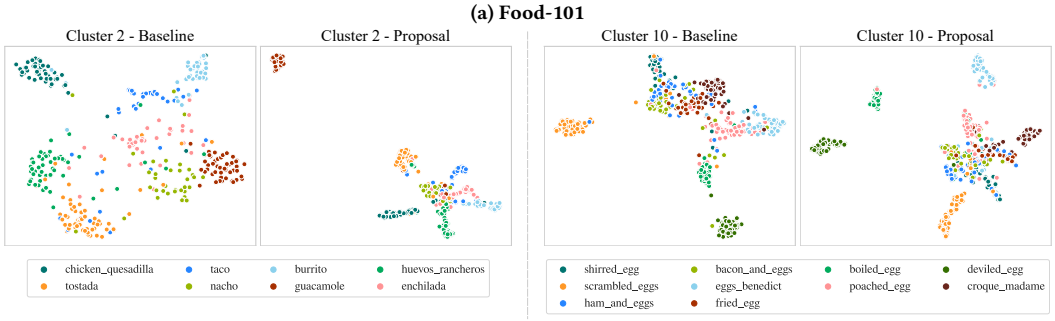
We carefully elucidate the limitations that can serve as future research directions: (1) In this work, we explored a wide variety of lexical encoders and sources. It must be noted that we have used **only modestly sized LLMs**. However, it is possible to obtain better results by using larger and higher-performing LLMs. (2) The addition of expert heads involves some **computational overhead** with respect to the baseline. This might affect its applicability to time-critical tasks. (3) Although the training method of DoD is not computationally demanding, the execution of LLMs on inference to compute the lexical descriptions are computationally expensive. Though this is done once for a given dataset, it might prevent its **applicability in low-resource scenarios**.

5 CONCLUSIONS AND FUTURE LINES

In this work, we introduced Dining on Details (DoD), an expert learning framework for fine-grained food recognition that leveraged LLMs to build subsets of classes. The method relies on the robustness of the embedding space of the multi-modality model ImageBind to find meaningful similarities between different categories. As a result, DoD acts as a universal add-on that enhances the classification ability of any deep neural network, improving the classification ability of classes that are close to each other in the multi-modal space. The obtained results showed the great versatility of DoD, as well as its capability to obtain competitive results. Our thorough analysis verified that the proposed method enables the learning of more useful features to discriminate highly similar



(a) Food-101



(b) FoodX-251

Figure 5: UMAPs obtained from embeddings of the baseline classification head and sub-network expert head of DoD.

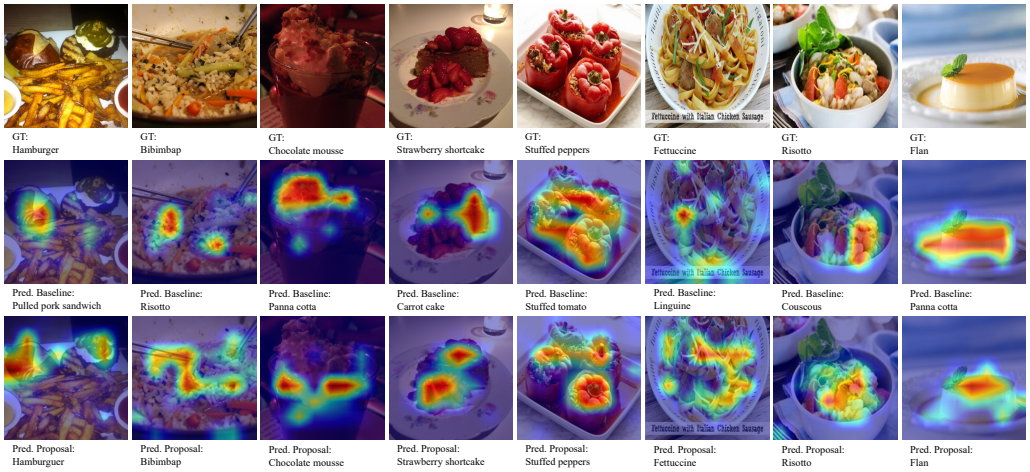


Figure 6: GradCAM examples for Food-101 and FoodX-251. The first row is the input image, the second row is the GradCAM of the baseline and the third row is the GradCAM of DoD.

classes. Regarding future lines of work, we will deepen in the use of crafted prompts to obtain textual priors. Another interesting line of research is the adaptation of the proposed grouping technique to build multi-level hierarchies and integration of this additional information to further enhance the capabilities of DoD.

ACKNOWLEDGMENTS

This work was partially funded by the Horizon EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), DeepFoodVol (AEI-MICINN, PDC2022-133642-I00), PID2022-141566NB-I00 (AEI-MICINN), and CERCA Programme / Generalitat de Catalunya. B. Nagarajan acknowledges the support of FPI Becas, MICINN. The authors thankfully acknowledge the computer resources at FinisTerrae III and the technical support provided by the Galician Supercomputing Center (CESGA) (RES-IM-2023-2-0025).

REFERENCES

[1] Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva. 2018. Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia* 20, 12 (2018), 3266–3275.

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance. (2023).

[3] Ardhendu Behera, Zachary Wharton, Pradeep RPG Hewage, and Asish Bera. 2021. Context-Aware Attentional Pooling (CAP) for Fine-Grained Visual Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 929–937.

[4] Marc Bolaños, Marc Valdivia, and Petia Radeva. 2019. Where and what am i eating? image-based food menu recognition. In *Computer Vision–ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part VI 15*. Springer, 590–605.

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision*.

[6] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712* (2023).

[7] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*. 169–174.

[8] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. 2021. Your" Flamingo" is my" Bird": fine-grained, or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11476–11485.

[9] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. 2020. Online Knowledge Distillation with Diverse Peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3430–3437.

[10] Joachim Dehais, Marios Anthimopoulos, and Stavroula Mougiakakou. 2016. Go-carb: A smartphone application for automatic assessment of carbohydrate intake. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. 91–91.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 248–255.

[12] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. 2019. Selective Sparse Sampling for Fine-Grained Image Recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 6598–6607. https://doi.org/10.1109/ICCV.2019.00670

[13] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik. 2018. Pairwise Confusion for Fine-Grained Visual Classification. In *Proceedings of the European conference on computer vision (ECCV)*. 70–86.

[14] Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. 2018. Maximum-Entropy Fine Grained Classification. *Advances in neural information processing systems* 31 (2018).

[15] ZongYuan Ge, Chris McCool, Conrad Sanderson, Alex Bewley, Zetao Chen, and Peter Corke. 2015. Fine-grained bird species recognition via hierarchical subset learning. In *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 561–565.

[16] ZongYuan Ge, Chris McCool, Conrad Sanderson, Alex Bewley, Zetao Chen, and Peter Corke. 2015. Fine-Grained Bird Species Recognition via Hierarchical Subset Learning. In *2015 IEEE International Conference on Image Processing (ICIP)*. 561–565. https://doi.org/10.1109/ICIP.2015.7350861

[17] ZongYuan Ge, Christopher McCool, Conrad Sanderson, and Peter Corke. 2015. Subset Feature Learning for Fine-Grained Category Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 46–52.

[18] Andrea Gesmundo. 2022. A continual development methodology for large-scale multitask dynamic ml systems. *arXiv preprint arXiv:2209.07326* (2022).

[19] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15180–15190.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[21] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision* 129 (2021), 1789–1819.

[22] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. 2020. Online Knowledge Distillation via Collaborative Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11020–11029.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[24] Ming He, Guangyi Lv, Weidong He, Jianping Fan, and Guihua Zeng. 2021. DeepME: Deep Mixture Experts for Large-scale Image Classification.. In *IJCAI*. 722–728.

[25] Xiangteng He and Yuxin Peng. 2017. Weakly Supervised Learning of Part Selection Model with Spatial Constraints for Fine-Grained Image Classification. *Proceedings of the AAAI Conference on Artificial Intelligence* 31, 1 (Feb. 2017). https://doi.org/10.1609/aaai.v31i1.11223

[26] Shaoli Huang, Xinchao Wang, and Dacheng Tao. 2021. SnapMix: Semantically Proportional Mixing for Augmenting Fine-Grained Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 1628–1636.

[27] Zixuan Huang and Yin Li. 2020. Interpretable and Accurate Fine-Grained Recognition via Region Grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8662–8672.

[28] Qiu Jianing, Frank Po Wen Lo, Yingnan Sun, Siyao Wang, and Benny Lo. 2019. Mining discriminative food regions for accurate food recognition. In *Proceedings of the British Machine Vision Conference. Cardiff, UK*, Vol. 158.

[29] Salaki Reynaldo Joshua, Seungheon Shin, Je-Hoon Lee, and Seong Kun Kim. 2023. Health to Eat: A Smart Plate with Food Recognition, Classification, and Weight Measurement for Type-2 Diabetic Mellitus Patients' Nutrition Control. *Sensors* 23, 3 (2023), 1656.

[30] Parneet Kaur, , Karan Sikka, Weijun Wang, serge Belongie, and Ajay Divakaran. 2019. FoodX-251: A Dataset for Fine-grained Food Classification. *arXiv preprint arXiv:1907.06167* (2019).

[31] Rajdeep Kaur, Rakesh Kumar, and Meenu Gupta. 2023. Deep neural network for food image classification and nutrient identification: A systematic review. *Reviews in Endocrine and Metabolic Disorders* (2023), 1–21.

[32] Xiao Ke, Yuhang Cai, Baitao Chen, Hao Liu, and Wenzhong Guo. 2023. Granularity-Aware Distillation and Structure Modeling Region Proposal Network for Fine-Grained Image Classification. *Pattern Recognition* 137 (2023), 109305. https://doi.org/10.1016/j.patcog.2023.109305

[33] Nadine Alvina Kong, Foong Ming Moy, Shu Hwa Ong, Ghalib Ahmed Tahir, and Choo Kiong Loo. 2023. MyDietCam: Development and usability study of a food recognition integrated dietary monitoring smartphone application. *Digital Health* 9 (2023), 20552076221149320.

[34] Fotios S Konstantakopoulos, Eleni I Georga, and Dimitrios I Fotiadis. 2023. A Review of Image-based Food Recognition and Volume Estimation Artificial Intelligence Systems. *IEEE Reviews in Biomedical Engineering* (2023).

[35] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. 2015. Fine-Grained Recognition without Part Annotations. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5546–5555. https://doi.org/10.1109/CVPR.2015.7299194

[36] Solomon Kullback and Richard A Leibler. 1951. On Information and Sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[37] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023).

[38] Zijia Lin, Guiguang Ding, Jungong Han, and Ling Shao. 2017. End-to-End Feature-Aware Label Space Encoding for Multilabel Classification with Many Classes. *IEEE transactions on neural networks and learning systems* 29, 6 (2017), 2472–2487.

[39] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. 2022. Swin Transformer V2: Scaling Up Capacity and Resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12009–12019.

[40] Merieme Mansouri, Samia Benabdellah Chaouni, Said Jai Andaloussi, and Ouail Ouchetto. 2023. Deep Learning for Food Image Recognition and Nutrition Analysis Towards Chronic Diseases Monitoring: A Systematic Review. *SN Computer Science* 4, 5 (2023), 513.

[41] Javier Marın, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2021. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 1 (2021), 187–203.

[42] Y. Matsuda, H. Hoashi, and K. Yanai. 2012. Recognition of Multiple-Food Images by Detecting Candidate Regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*.

[43] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software* 3, 29 (2018), 861.

[44] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. 2019. A survey on food computing. *ACM Computing Surveys (CSUR)* 52, 5 (2019), 1–36.

[45] Weiqing Min, Linhu Liu, Zhengdong Luo, and Shuqiang Jiang. 2019. Ingredient-guided cascaded multi-attention network for food recognition. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1331–1339.

[46] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. 2023. Large scale visual food recognition.

IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[47] Bhalaji Nagarajan, Rupali Khatun, Marc Bolaños, Eduardo Aguilar, Leonardo Angelini, Mira El Kamali, Elena Mugellini, Omar Abou Khaled, Noemi Boqué, Lucia Tarro, et al. 2021. Nutritional monitoring in older people prevention services. In *Digital Health Technology for Better Aging: A Multidisciplinary Approach*. Springer, 77–102.

[48] Changhwa Park, Junho Yim, and Eunji Jun. 2023. Mutual Learning for Long-Tailed Recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2675–2684.

[49] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116* (2023). arXiv:2306.01116 https://arxiv.org/abs/2306.01116

[50] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116* (2023).

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[52] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[53] Javier Ródenas, Bhalaji Nagarajan, Marc Bolaños, and Petia Radeva. 2022. Learning Multi-Subset of Classes for Fine-Grained Food Recognition. In *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management*. 17–26.

[54] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[55] Marcel Simon and Erik Rodner. 2015. Neural Activation Constellations: Unsupervised Part Model Discovery with Convolutional Networks. In *Proceedings of the IEEE international conference on computer vision*. 1143–1151.

[56] Leslie N Smith and Nicholay Topin. 2019. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, Vol. 11006. SPIE, 369–386.

[57] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems* 33 (2020), 16857–16867.

[58] Mohammed Ahmed Subhi, Sawal Hamid Ali, and Mohammed Abulameer Mohammed. 2019. Vision-based approaches for automatic food recognition and dietary assessment: A survey. *IEEE Access* 7 (2019), 35370–35381.

[59] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International conference on machine learning*. PMLR, 6105–6114.

[60] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[61] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. 2021. Grafit: Learning Fine-Grained Image Representations with Coarse Labels. In *Proceedings of the IEEE/CVF international conference on computer vision*. 874–884.

[62] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2008. Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, Vol. 21. 53–59.

[63] Hao Wang, Guosheng Lin, Steven CH Hoi, and Chunyan Miao. 2020. Structure-aware generation network for recipe generation from images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*. Springer, 359–374.

[64] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. 2020. Frustratingly Simple Few-Shot Object Detection. In *Proceedings of the 37th International Conference on Machine Learning*. 9919–9928.

[65] Yunan Wang, Jing-jing Chen, Chong-Wah Ngo, Tat-Seng Chua, Wanli Zuo, and Zhaoyan Ming. 2019. Mixed Dish Recognition through Multi-Label Learning. In *Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities*. 1–8.

[66] Yaming Wang, Vlad I Morariu, and Larry S Davis. 2018. Learning a Discriminative Filter Bank Within a CNN for Fine-Grained Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4148–4157.

[67] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. 2021. Fine-Grained Image Analysis with Deep Learning: A Survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 12 (2021), 8927–8948.

[68] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *European conference on computer vision*. Springer, 818–833.

[69] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[70] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. 2016. Picking Deep Filter Responses for Fine-Grained Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1134–1142. https://doi.org/10.1109/CVPR.2016.128

[71] Yu Zhang, Xiu-Shen Wei, Jianxin Wu, Jianfei Cai, Jiangbo Lu, Viet-Anh Nguyen, and Minh N. Do. 2016. Weakly Supervised Fine-Grained Categorization With Part-Based Image Representation. *IEEE Transactions on Image Processing* 25, 4 (2016), 1713–1725. https://doi.org/10.1109/TIP.2016.2531289

[72] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4320–4328.

[73] Jiannan Zheng, Liang Zou, and Z Jane Wang. 2018. Mid-level deep Food Part mining for food image recognition. *IET Computer Vision* 12, 3 (2018), 298–304.

[74] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.

[75] Feng Zhou and Yuanqing Lin. 2016. Fine-grained image classification by exploring bipartite-graph labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1124–1133.

S2ML-TL Framework for Multi-label Food Recognition

Bhalaji Nagarajan[1(✉)], Eduardo Aguilar[1,2], and Petia Radeva[1,3]

[1] Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona, Spain
{bhalaji.nagarajan,petia.ivanova}@ub.edu
[2] Departamento de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Antofagasta, Chile
eaguilar02@ucn.cl
[3] Computer Vision Center, Cerdanyola, Barcelona, Spain

Abstract. Transfer learning can be attributed to several recent breakthroughs in deep learning. It has shown upbeat performance improvements, but most of the transfer learning applications are confined towards fine-tuning. Transfer learning facilitates the learnability of the networks on domains with less data. However, learning becomes a difficult task with complex domains, such as multi-label food recognition, owing to the shear number of food classes as well as to the fine-grained nature of food images. For this purpose, we propose S2ML-TL, a new transfer learning framework to leverage the knowledge learnt on a simpler single-label food recognition task onto multi-label food recognition. The framework is further enhanced using class priors to tackle the dataset bias that exists between single-label and multi-label food domains. We validate the proposed scheme with two multi-label datasets on different backbone architectures and the results show improved performance compared to the conventional transfer learning approach.

Keywords: Food recognition · Multi-label · Class priors

1 Introduction

Food is not just a source of energy, but is a cultural expression. It is central to the regular and pleasure activities of day-to-day life. Food habits directly affect the lifestyle and healthcare of individuals. Consuming a well-balanced healthy diet averts a wide range of noncommunicable diseases and conditions such as diabetes and obesity [43]. Growing emphasis on well-maintained food habits has given rise to a large number of automated food intake monitoring systems [42,48,66] that help in regulating food intake. Food computing is possible through different forms of data - images, texts, tables, etc. of which images are more prevalent due to the large number of social media posts. Automated food recognition is vital to any food journal because of the ease it gives to the users and is integral to several modern dietary applications [15]. However, food recognition is a very complex computer vision task, due to the nature of images [60]. Food images

© Springer Nature Switzerland AG 2021
A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12665, pp. 629–646, 2021.
https://doi.org/10.1007/978-3-030-68821-9_50

are often mixed or composed of many items in a single platter. Moreover, high intra-class and low inter-class variance make it difficult to work for any food recognition model.

Deep Learning has been vital in improving the performance of several tasks. However, a well-developed model requires large volumes of training data. Collecting huge volumes of data and annotating them are both costly and time intensive. This data dependency is addressed using Transfer Learning (TL), the ability of deep networks to transfer knowledge from one domain or task to another [54]. Automated food recognition using deep networks takes advantage of TL, as there are insufficient large-scale food datasets [38]. The common food recognition datasets, such as ETH Food101 [8] and VireoFood172 [11] are quite small and do not represent wide range of diverse classes. Food recognition falls under fine-grained recognition task, where the latent features are less separable, making it difficult to establish a well-trained network. These factors make food recognition a challenging research problem, but most of the existing literature until today, are often done on single-label datasets. However, the food images in real world are often not single-labelled by nature [60]. Single-label recognition is an easier task, because the final activation function (softmax) is bound to get exactly one output whereas in a multi-label scenario, possible solutions are larger (allowing to activate 0, 1 or many final labels), thus making the probability error much higher. In spite of all the challenges it presents, tacking food recognition in a multi-label perspective would bring the problem closer to the real-world.

High success of TL in visual recognition is due to its capacity to extract shared low-level features such as edges and curves across different image domains [58]. However, as the layers go deeper, they become more task-specific. Multi-stage TL has been widely used in classification tasks of high complexity [16]. Using this approach, an ImageNet-trained network is often used to fine-tune on common datasets, which in turn, would be used to fine-tune on more complex datasets. This forces the deeper layers to learn intermediate representations that would allow better learning of the complex task. It has to be also noted that TL does not always result in improvements. When weights are transferred from similar domains, the learning of layers is faster and more accurate, whereas, when the two domains are dissimilar, the learning could give inaccurate results, known as Negative transfer [10,62]. However, there are a lot of open questions regarding when negative transfer appears and how to avoid it [44]. A typical heuristic solution to avoid negative transfer is to train only on samples that are nearer to the target domain and avoid those samples that are far away from the source domain [46]. Choice of positive samples thus becomes an important aspect. However, there is no existing mechanism to determine if the particular sample would result in positive or negative transfer [44].

The recent success in single-label food classifiers can be largely attributed to the ease in training such models. We therefore hypothesise to utilize the learning ability of single-label classifiers into a more complex task: multi-label recognition. In this work, we propose Single to Multi-label Transfer Learning (S2ML-TL) framework, using single-label recognition as an intermediate task to achieve multi-label recognition. As we will show, this allows the model to

generalize better and improve its learnability. We further enhance the framework by using class priors that helps in selection of positive samples for better transferability. We organize the paper as follows: In Sect. 2, we review the recent literature on food recognition. We provide details of the proposed scheme in Sect. 3 and show the experimental details and results in Sect. 4. We present the concluding remarks in the last section.

2 Related Works

Food recognition is increasingly popular because of the advancements in intelligent health applications in dietary management, food perception, recommendation and health analysis [36]. With the development of deep networks, food recognition has become on par with human performance and have out-performed hand-crafted feature based classifiers [21,23,65]. Different architectures have been applied on the food recognition tasks: GoogLeNet [28,63], Network-In-Networks [55], Inception V3 [18], ResNet50 [39], Ensemble NN [40], Wide Residual Networks [31], and CleanNet [27]. The base architectures were also modified to enhance the learning task, such as modifying Alexnet in NutriNet [34]. However, most of the popular benchmark datasets [8,11,13,25] do not offer a diverse representation of food classes. They are confined to either a certain region or to a certain representation. With the explosion of images in social platforms, recent datasets involve a large number of food images [12,24]. With increased computational capabilities, ensemble of various models is a common approach [2,45,56]. An integration of local and flat classifiers [4] and fusion of slice network and residual network [31] have also been effective. Preceding the classification with a detection network is another common approach [35]. Faster R-CNN based detection was used to find the candidate regions before classification [22]. Recent architecture advances include using teacher-student learning [67], knowledge distillation [20], attention network [37] and incremental learning [53].

Real world food images are often multi-labelled in nature, ie. more than one food class is present in an image. Compared with single-label food recognition, there have been much fewer works in multi-label food recognition. A general approach is to detect the location of food and then recognize each class in the image [6]. Multi-task based learning methods [3,68] and multi-scale region-based classifier [61] have also improved the performance. Using segmentation networks before recognition [49] has also been effective. Region proposal based candidate region detection [5,32,50] has helped in improving the recall rate. Using additional information such as ontology [14] has also assisted in improving the multi-label performance. However, there is still a big difference in the performance between single-label and multi-label recognition tasks, owing to the complexity of the data and also to the difficulty in learning such tasks.

In the proposed S2ML-TL framework, we use single-label models to improve the learning of multi-label models for multi-label food recognition, which up to our knowledge is the first such approach. Additionally, we propose to enhance the model with specially designed class priors for better recognition performance, which is used to tackle the covariate shift problem. The complexity and

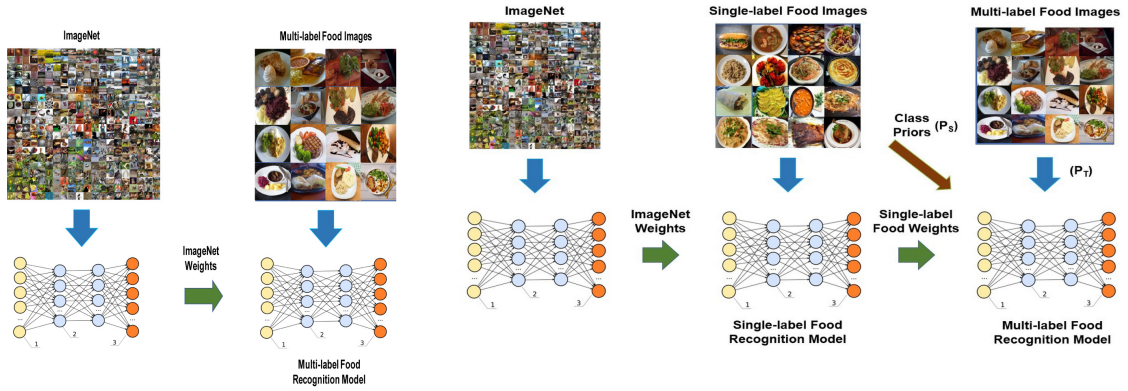


Fig. 1. Traditional TL (fine-tuning) (a) vs. S2ML-TL framework (b)

fine-grained nature of the food images, coupled with increased performance in multi-stage frameworks in other domains are primary reasons in our proposed methodology.

3 Proposed Methodology

Supervised fine-tuning in two-stages, using datasets similar to the target domain in the first stage, yields better results than a single stage fine-tuning [41]. The advantage of using such a framework is that the models are generalized as they go deeper in the stages and also the learning is made possible with smaller number of more complex images compared to the earlier stages. With this underlying premise, we propose the S2ML-TL framework. Figure 1 shows the proposed scheme against the traditional TL methodology.

3.1 S2ML-TL Framework

Let us consider a domain, $D = \{x, P(X)\}$ consisting of feature space x and marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in x$. A task, $T = \{y, f(x)\}$ consists of a label space y and a target prediction function $f(x)$. $f(x)$ determines the conditional probability function $P(y|x)$. Given a learning task T_t based on D_t, transfer learning aims at increasing the learning performance by leveraging the knowledge used in another learning task T_s based on domain D_s. The deep network learns the latent features using the transferred knowledge from D_s and T_s, where $D_s \neq D_t$ and/or $T_s \neq T_t$. In addition, in most cases, the size of D_s is much larger than the size of D_t.

As discussed previously, TL has been successfully applied on various food datasets. However, when multi-label food recognition is considered, knowledge transfer becomes much harder. To overcome such a scenario, we hypothesise S2ML-TL framework. Let us consider two tasks T_{t1} and T_{t2} on domains D_{t1} and D_{t2}, where D_{t1} and D_{t2} are similar domains, but the task T_{t2} (e.g. multi-label recognition) is more complex than T_{t1} (e.g. single-label recognition).

In this scenario, transferring knowledge from the source task, T_s on domain D_s to the target task T_{t2} (multi-label) on domain D_{t2} could be achieved using the task T_{t1} (single-label) on domain D_{t1} as an intermediate transfer. The deep network trained on the simpler domain D_{t1} learns the latent space better from the source and then when it is used on the more complex domain D_{t2}, the network is better equipped to learn the more complex multi-label space.

The impact of TL can be seen in one or combination of the following three aspects: a higher performance at the very beginning of learning, a steeper slope in the learning curve, and a higher asymptote [57]. With the first two cases, it is evident that the learning is happening faster and with the last one, it is known that the performance of TL is better. However, transferring knowledge between two completely different tasks might also result in a negative transfer, that is, decreasing in performance of the target deep network, which makes the selection of task/domain a very important factor. In most of the tasks, transfer of knowledge is performed through mapping of characteristics of the source and target tasks [57]. The correspondence between tasks is automatically performed during TL or is induced by a mapping. In this work, the performance of the TL framework is further enhanced using class priors, which acts as mapping between the simpler and the complex task.

3.2 Class Priors for Improved Transferability of Food Recognition

Covariate shift [1,47] is a common phenomenon in TL where the input and output relationships are the same, but the distributions of training and test sets are different. The dataset bias or the domain shift makes the generalization of deep networks a difficult task [59]. Importance-weighting [9,17,51] has been a common solution to this problem. The distribution of samples in a domain is a direct representation of the nature of the domain. In this work, we use this information as a prior knowledge, that would enable the algorithms to be robust to shifts.

The priors of both domains would be different due to overlapping classes, under-sampled classes, unevenly distributed classes or classes that are more similar to each other. All these situations give rise to an uncertainty in the model which makes prior-induced learning important. Class priors are computed for the simpler (the single-label) domain D_{t1} and are used in the training of the more complex (the multi-label) domain D_{t2}. The class priors of the target domain (D_{t2}), P_T and the source domain (D_{t1}), P_S are computed as a function of the samples that represent a certain class. The ratio of the class priors, P_T/P_S is thus used to establish the mapping between domains. However, it has to be noted that it is not always necessary that the classes present in both domains need to be the same. In this particular food recognition problem, the information of the ingredients present in classes of both domains are used when the classes are not matching.

In the multi-label setting, for the domain $D = \{x, P(X)\}$, with the number of classes as C, we would have $P(X) = [y_1, y_2, ..., y_C]$. The most commonly used loss function to train a multi-label classification model is the Binary Cross-Entropy (BCE) loss:

$$l(x, y) = \frac{1}{C} \sum_{c=1}^{C} [\ y_i log(p(y_i)) + (1 - y_i) log(1 - p(y_i))\ ] \tag{1}$$

where y_i is the label for the class i and $p(y_i)$ is the corresponding predicted probability.

The ratio of priors, P_T/P_S, is added to the loss computation to enhance the transferability of the domain knowledge. This means that when P_S is smaller, the classes has to be learnt from the data which is available with the target domain. However, when P_S is larger, the learning can be achieved from the source model. During TL, the training of the target domain starts with initialization of weights from the source domain. At this point, the weights of the network are more suitable for the source domain than the target domain. Over iterations, the network learns the underlying target domain distribution until convergence so that the learning of the target domain is complete. In this way the learning would gradually forget the source domain and adapt better to the target domain. This gradual learning process is replicated in the impact of source distribution over the target distribution using the prior impact factor, r_i:

$$r_i = \alpha \frac{P_T}{P_S} + (1 - \alpha) P_T \tag{2}$$

where $0 \leq \alpha \leq 1$. As we can see, higher target priors denote the importance of that particular class in the domain. During the beginning of the training, α is set to maximum and is gradually decayed along the training process. However, the value of weights may scale the loss value, which in turn can increase the difficulty of optimization. In order to make the optimization process stable, a correction constant, β is used, which will ensure that the prior impact factor r_i, falls under proper range of values. The balanced prior impact factor, r_i^b, is given as follows:

$$r_i^b = \beta \left[\alpha \frac{P_T}{P_S} + (1 - \alpha) P_T \right] \tag{3}$$

where $\beta \geq 0$. Finally, the prior-induced loss function is as in Eq. (4):

$$l_p(x, y) = \frac{1}{C} \sum_{c=1}^{C} [\ y_i.log(p(y_i)) + (1 - y_i).log(1 - p(y_i))\ ] * r_i^b \tag{4}$$

This prior-induced loss function has shown optimal performance in transferring knowledge from the source to target domain, as it will be validated in the next section.

Prior Computation. The next objective is to compute the ratio of priors, $P(T)/P(S)$, between the single-label (source) and multi-label datasets (target). $P(T_i)$ for the target datasets defines the probability of each class, i, in the multi-label datasets:

$$P(T_i) = \frac{1}{N_t} \sum_{n=1}^{N_t} y_i^n \tag{5}$$

where N_t is the total number of training samples in the target domain.

It is also possible that there could be more than one class of the source dataset where the target class is present. In this case, $P(S)$ is computed as the sum of probabilities of all classes, where the target class is present. For the classes, i present in the target domain, the mapping in the source domain is computed as follows:

$$P(S_i) = \frac{1}{N_s} \sum_{i} \sum_{j=1}^{N_s} y_j,\ j \in \{all\ source\ classes\ containing\ i\} \tag{6}$$

where N_s is the total number of the training samples in the source domain.

4 Validation

In this section, we brief the datasets, evaluation metrics and the implementation details of the deep networks. We also show the computation of priors and the selection of hyper-parameters. Finally, we show the experimental results and the inferences made from those results.

4.1 Datasets

In order to validate the proposed framework, we use two multi-label food datasets, Food201 [33] and Combo-plates. Food201 is a public dataset, whereas Combo-plates is an in-house dataset. We use Food101 [8] as the single-label dataset. Figure 2 shows sample images of each of the dataset used in the validation.

Food101. Food101 is one of the most popular single-label food datasets. It contains 101 food classes of international cuisines with 1000 images each. The dataset is divided into 75% training and the remaining as test images. The dataset is annotated for the class names only.

Food201. Food201 is a multi-label dataset that can be considered as an extension to the Food101 class labels. The images from Food101, although, single-labelled, contain images that have more than one class. This behaviour was leveraged to create the Food201 dataset and resulted in around 35k training images and 15k test images. These images were re-annotated for a new 100 classes in addition to the ones in Food101.

(a) (b) (c)

Fig. 2. Sample images from Food101 (a), Food201 (b) Combo-plates (c)

Combo-Plates. There are a very few public multi-label food datasets, of which, the multi-label datasets are primarily targeted towards ingredient detection [11] or category detection [14]. One popular multi-label food recognition dataset is UNIMIB2016 [13], which is actually a tray-based dataset, where each tray would again contain spatially separated classes in different plates inside the tray. However, these images are confined to a very particular type of presentation. With very few public datasets available and moreover the aim of having resemblance with the real-world food representations, we decided to annotate a new dataset (Combo-plates).

We created the Combo-plates dataset from web-scrapped images. Initially, we constructed a list of 75 commonly consumed European foods and created search queries using different combination of the classes and crawled the candidate images from different search engines. The candidate images were cleaned using a food detector network, so that the non-food images are removed and then we annotated the filtered food images. We further filtered the images during annotation, in such a way that all the images are multi-labelled in nature, placed in a single platter and the classes had a distinguishable boundary between them. The annotation was done independent of the search queries and there is no fixed combination or fixed number of the classes in each image. The dataset is split into 75% train, 10% validation and 15% test images.

Prior Computation. For both Food201 and Combo-plates, we compute $P(T_i)$ using Eq. (5), which is the probability of each class. As we explained previously, not all classes are present in both the source and target and we use the ingredient information to create the associated mapping. The ingredients information is already available for Food101 dataset via the Ingredients101 [7] dataset. For a single class in Food201 and Combo-plates, there are more than one class in Food101. For example, for the class, *avocado* present in Combo-plates, it is present in *guacamole* and *ceviche* classes of Food101. In this case, we compute $P(S)$ using Eq. (6). Since Food101 is a balanced dataset, the probabilities of each source class would be the same. Calculating priors of Food201 from Food101 is a simpler case, as the 101 primary classes are a direct overlapping information. The priors are computed only for the classes that are present in both datasets.

We then fit the computed source and target priors into Eq. (3) to compute the balanced prior impact factor r_i^b.

4.2 Evaluation Metrics

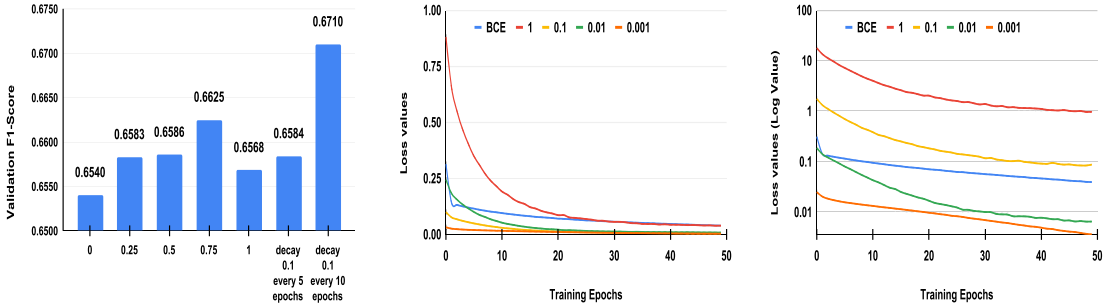
We use Precision, Recall and F1-Score to evaluate the proposed hypothesis. Accuracy is not a good measure for the essence of the problem, due to the multi-label and multi-class nature of the dataset and also due to the class imbalance in both datasets. We perform 'Micro' averaging to compute the overall F1-Score for each model.

4.3 Implementation Details

We use Resnet50 [19] and InceptionResnetV2 [52] as backbone architectures. For single-label training, we used the pre-trained ImageNet models and trained with the standard data splits available in Food101. For multi-stage transfer learning, we initialized the weights of single-label models. For all the models, we removed the classification layer and the fully connected layer from the base models and replaced with a fully connected layer of 2048 neurons, followed by the classification layer of n neurons, where n is the number of classes in each dataset. For single-label training, we used a softmax activation, whereas for multi-label training, we used a sigmoid activation. For all the models, we trained the entire network. As pre-processing, we resized all the images to 224×224. As far as the training parameters are concerned, we used Adadelta optimizer with an initial learning rate of 1 and we used a batch size of 24. We trained all the models in Keras framework with Tensorflow backend.

Hyper-parameter Selection. There are two parameters that we define in our proposed scheme, α and β, which affect the impact of source distribution on the target distribution. We show the selection of these parameters using the experiments performed using InceptionResnetV2 trained on Combo-plates using Food101 as the source domain.

In order to evaluate the impact of α, we considered α values at 0, 0.25, 0.5, 0.75 and 1. When $\alpha = 1$, the first term of r_i^b is only active, ie. the loss function is impacted by the ratio of priors, whereas, when $\alpha = 0$, it is similar to applying class weights during the model training. The values in Fig. 3(a) show that the models in between the limits outperform the models at the extremes. This prompted us to use a decaying function on α based on the number of epochs. We investigate this, by using a decay of 0.1 every 5 epochs and 0.1 every 10 epochs. It can be inferred from the Fig. 3(a) that the decayed α works better rather than a fixed α. We attribute this to the learning of the model, where the model has minimum or no influence from the source domain during the later stages of the model training. Out of the two decay networks, the one with larger decay performs better. This also shows the importance of target distribution weights on the training of the models.



(a) (b) (c)

Fig. 3. Performance of models with different α values (a); loss curves of models with varying β values at $\alpha = 0$ (b); and $\alpha = 0.75$ (c).

We use the training loss curves to study the smoothing parameter, β using the optimal value of α. Figure 3(b) shows the curves with fixed $\alpha = 0$ and the Fig. 3(c) shows the loss curves with fixed $\alpha = 0.75$. The loss curves of BCE is taken as a reference curve and β is set to bring the curves closer to the reference loss curve. This is performed in order to induce higher performance during early training steps and also to have steeper slope in the learning curve. We empirically select the values of β as 0.1 and 0.003 for transferring Food101 to Combo-plates and Food201 respectively.

Comparing Methods. We establish standard BCE loss with single transfer learning scheme as the baseline. For this, we use an ImageNet pre-trained model directly on the multi-label datasets. We compare this baseline with different strategies: (1) Empirical risk minimization (ERM): The plain multi-stage framework with standard BCE loss. (2) Kullback–Leibler Divergence (KL): We replace the prior computation with the KL entropy values. (3) Priors: We compute the ratio of priors as in Eq. (3) and use Eq. (4) for the loss computation. KL-Divergence is a relative entropy score that defines how one probability distribution is different from another distribution [29]. KL-divergence based loss function is used to improve the similarity of the latent distributions to the original distribution. KL-divergence has been used in generative adversarial networks [64], Prior Networks [30] and Density fixing Regularization methods [26]. The main intuition in using KL-divergence as a comparing method is its effectiveness in providing the entropy between distributions. Since there is no prior results on Food201 multi-label recognition, we compare it against the above methods only.

4.4 Results and Discussion

We show the overall precision, recall and F1-score for the validation and test splits of both multi-label datasets to study the effectiveness of the proposed

Table 1. Model performance of InceptionResnetV2 on Combo-plates

Model	Validation data			Test data		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Standard TL	0.7209	0.5865	0.6468	0.7152	0.5840	0.6430
ERM	0.6991	0.5667	0.6260	0.6900	0.5700	0.6200
KL	0.7030	0.6212	0.6596	0.6984	0.6173	0.6553
Priors	0.7045	0.6229	**0.6612**	0.7000	0.6200	**0.6600**

framework. We show the performance of InceptionResnetV2 and Resnet50 models on Combo-plates dataset in Table 1 and Table 2 respectively, while the performance of InceptionResnetV2 and Resnet50 models on Food201 dataset are shown in Table 3 and Table 4.

Comparison with SoA Methods. From the tables showing the overall performance of the experiments, it can be seen that except for the InceptionResnetV2 model on Food201 (decreased by 1.6% on the test set), all the other experiments show improvements with simple multi-stage transfer learning using BCE loss. However, it can be seen in all experiments that the recall has increased compared to the baseline method. Higher recall is particularly noted because it would result in lesser false negatives. When we use Food101 dataset in computing the KL-Divergence and in Priors to enhance the loss function further, the performance is boosted further. Since the priors are a direct representation of the classes, it is also necessary to infer the class-wise performance of the models. We compute the difference in F1-score of BCE loss and prior-induced loss for each class and denote the positive or negative transfer of the model, which we call it as degree of transfer. Figure 4 shows the histogram of the degree of transfer for all the four experiments. It can be inferred that in all the models, most of the classes have a positive transfer. However, the histogram also shows that classes have a negative transfer.

What Are the Possible Reasons for Negative Transfer in Some Classes? In order to answer this question, we study the relationship between the validation F1-score and the number of training samples of each class where priors were applied. We study this relationship using the experiments with InceptionResnetV2 as shown in Fig. 5. For the experiments using combo-plates, it is clearly evident that the classes having fewer training examples (<100) were not enhanced in the training. Also, with the corresponding scatter plot, it is evident that the validation F1-score is very low in these classes. Lacking of training data can be attributed to this performance. However, the performance of Food201 is showing mixed results. Classes that are showing a negative trend are classes that are difficult to classify (such as *ketchup* and *mustard*) or that has a larger variation inside the same class (such as *potatoes*, where there is also a *boiled*

Table 2. Model performance of Resnet50 on Combo-plates

Model	Validation data			Test data		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Standard TL	0.7250	0.5581	0.6307	0.7200	0.5582	0.6289
ERM	0.7268	0.5590	**0.6320**	0.7223	0.5616	0.6319
KL	0.6956	0.5473	0.6126	0.6933	0.5491	0.6128
Priors	0.6861	0.5783	0.6276	0.6882	0.5886	**0.6345**

Table 3. Model performance of InceptionResnetV2 on Food201

Model	Validation data			Test data		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Standard TL	0.6800	0.4595	0.5485	0.6997	0.5001	0.5833
ERM	0.7936	0.5354	0.6394	0.7895	0.5563	**0.6527**
KL	0.7521	0.4755	0.5826	0.7567	0.5044	0.6053
Priors	0.8189	0.6176	**0.7041**	0.7464	0.5550	0.6366

Table 4. Model performance of Resnet50 on Food201

Model	Validation data			Test data		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Standard TL	0.7204	0.4215	0.5319	0.7322	0.4636	0.5678
ERM	0.7518	0.4546	0.5666	0.7493	0.4800	0.5852
KL	0.7918	0.4317	0.5587	0.7740	0.4370	0.5586
Priors	0.7767	0.5877	**0.6691**	0.7313	0.5400	**0.6213**

potatoes class present in the dataset). The performance of such classes may be due to the fine-grained nature of the dataset and also due to the close inter-class variance with other classes. In general, with the histograms in Fig. 4, it can be seen that there is a general positive trend in the performance.

4.5 Ablation Study

Finally we show the importance of each term in the proposed framework using Table 5. When $\alpha = 1$, the ratio of class probabilities is constant throughout the training process and the decay curve proves to improve the performance compared to the constant α. P_T uses only the target probabilities as weighting factor. Results using source distribution alone and using target distribution alone show both are equally important and the final proposed term gives a cumulative performance improvement. In all the experiments using β, the performance increases, showing the importance of scaling of the loss value.

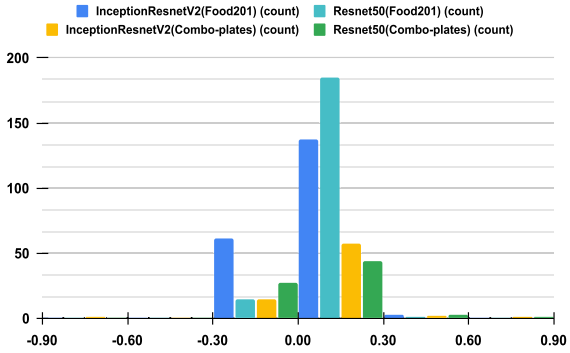


Fig. 4. Histogram showing the degree of transfer.

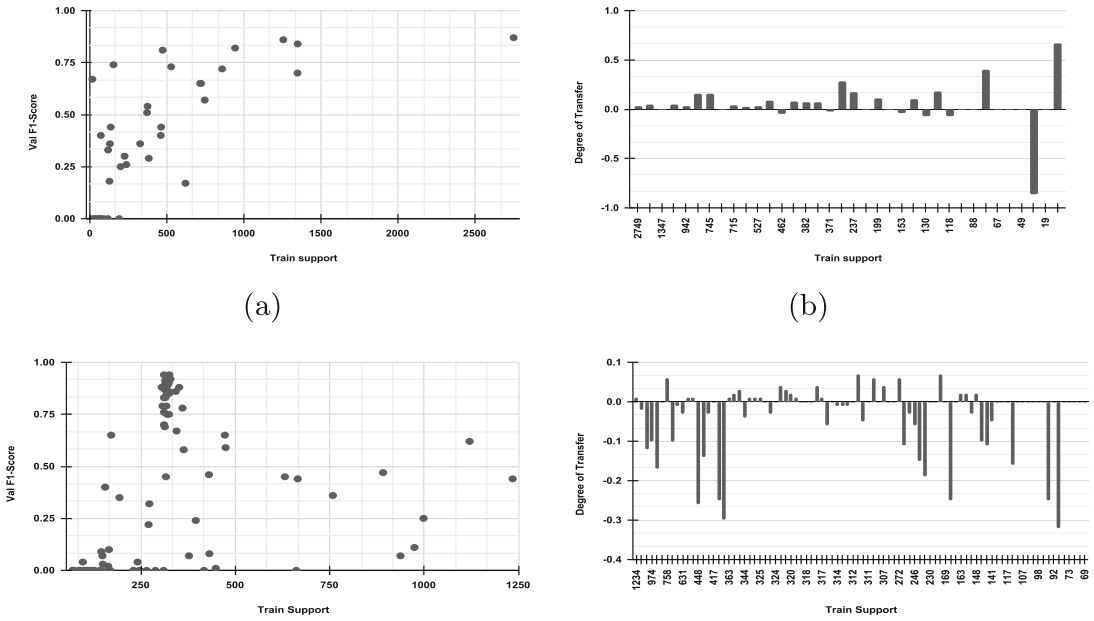


Fig. 5. Degree of transfer with respect to the train support. Train support vs. Val F1-score of InceptionResnetV2 on Combo-plates (a), Degree of transfer (b), InceptionResnetV2 on Food201 (c), Degree of transfer (d)

Table 5. Ablation study

Model	$\alpha * \frac{P_T}{P_S}$	P_T	β	Validation data			Test data		
				Prec.	Recall	F1 Sc.	Prec.	Recall	F1 Sc.
$\alpha = 1$ (w/o β)	x	-	-	0.7537	0.5685	0.6481	0.7394	0.5666	0.6415
$\alpha = 1$ (w β)	x	-	x	0.7237	0.6012	0.6568	0.7200	0.6000	0.6500
Decayed α (w/o β)	x	-	-	0.7281	0.5958	0.6553	0.7112	0.5932	0.6469
Decayed α (w β)	x	-	x	0.7118	0.6094	0.6567	0.6963	0.6086	0.6495
Target priors (w/o β)	-	x	-	0.6972	0.6238	0.6585	0.6811	0.6140	0.6458
Target priors (w β)	-	x	x	0.7092	0.6068	0.6540	0.7050	0.5994	0.6479
Proposal (w/o β)	x	x	-	0.6996	0.6034	0.6480	0.7045	0.6069	0.6521
Proposal (w β)	x	x	x	0.7114	0.6349	**0.6710**	0.7011	0.6127	**0.6539**

5 Conclusions

Transfer learning has been an active means of transferring knowledge from tasks of one domain to another. In this work, we use S2ML-TL framework for multi-label food recognition by transferring knowledge from an intermediate simpler domain to a more complex domain. We validated the proposed framework on two multi-label food datasets, using single-label food dataset as an intermediate in two backbone architectures. We showed that utilizing single-label recognition tasks will further increase the learning ability of the multi-label recognition tasks. We also showed that the recognition performance could be further boosted by using class priors. We further analyzed the results using class-wise performance analysis and also studied the impact of transfer based on the number of training samples. We have validated the selection of the hyper-parameters used in the proposed approach. We observed that the proposed approach provides a good performance in most of the classes across different experiments. Finally, as a future work, we will advance the proposed framework to tackle the long-tail data problem, which is common for any multi-label dataset, thereby easing the process of annotating large multi-label datasets.

Acknowledgements. This work was partially funded by TIN2018-095232-B-C21, SGR-2017 1742, Nestore project of the European Commission Horizon 2020 programme (Grant no. 769643), Validithi EIT Health program and CERCA Programme/Generalitat de Catalunya. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

1. Adel, T., Zhao, H., Wong, A.: Unsupervised domain adaptation with a relaxed covariate shift assumption. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
2. Aguilar, E., Bolaños, M., Radeva, P.: Food recognition using fusion of classifiers based on CNNs. In: Battiato, S., Gallo, G., Schettini, R., Stanco, F. (eds.) ICIAP 2017. LNCS, vol. 10485, pp. 213–224. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68548-9_20
3. Aguilar, E., Bolaños, M., Radeva, P.: Regularized uncertainty-based multi-task learning model for food analysis. J. Vis. Commun. Image Represent. **60**, 360–370 (2019)
4. Aguilar, E., Radeva, P.: Uncertainty-aware integration of local and flat classifiers for food recognition. Pattern Recogn. Lett. **136**, 237–243 (2020)
5. Aguilar, E., Remeseiro, B., Bolaños, M., Radeva, P.: Grab, pay, and eat: semantic food detection for smart restaurants. IEEE Trans. Multimedia **20**(12), 3266–3275 (2018)
6. Anzawa, M., Amano, S., Yamakata, Y., Motonaga, K., Kamei, A., Aizawa, K.: Recognition of multiple food items in a single photo for use in a buffet-style restaurant. IEICE Trans. Inf. Syst. **102**(2), 410–414 (2019)

7. Bolaños, M., Ferrà, A., Radeva, P.: Food ingredients recognition through multi-label learning. In: Battiato, S., Farinella, G.M., Leo, M., Gallo, G. (eds.) ICIAP 2017. LNCS, vol. 10590, pp. 394–402. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70742-6_37
8. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 446–461. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_29
9. Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: International Conference on Machine Learning, pp. 872–881 (2019)
10. Cao, Z., Long, M., Wang, J., Jordan, M.I.: Partial transfer learning with selective adversarial networks. In: CVPR, pp. 2724–2732 (2018)
11. Chen, J., Ngo, C.W.: Deep-based ingredient recognition for cooking recipe retrieval. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 32–41 (2016)
12. Chen, X., Zhou, H., Zhu, Y., Diao, L.: ChineseFoodNet: a large-scale image dataset for Chinese food recognition. arXiv preprint arXiv:1705.02743 (2017)
13. Ciocca, G., Napoletano, P., Schettini, R.: Food recognition: a new dataset, experiments, and results. IEEE J. Biomed. Health Inform. **21**(3), 588–598 (2016)
14. Donadello, I., Dragoni, M.: Ontology-driven food category classification in images. In: Ricci, E., Rota Bulò, S., Snoek, C., Lanz, O., Messelodi, S., Sebe, N. (eds.) ICIAP 2019. LNCS, vol. 11752, pp. 607–617. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30645-8_55
15. El Khoury, C.F., Karavetian, M., Halfens, R.J., Crutzen, R., Khoja, L., Schols, J.M.: The effects of dietary mobile apps on nutritional outcomes in adults with chronic diseases: a systematic review and meta-analysis. J. Acad. Nutr. Diet. **119**(4), 626–651 (2019)
16. Godasu, R., El-Gayar, O., Sutrave, K.: Multi-stage transfer learning system with light-weight architectures in medical image classification (2020)
17. Grover, A., et al.: Bias correction of learned generative models using likelihood-free importance weighting. In: NIPS, pp. 11058–11070 (2019)
18. Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., Cagnoni, S.: Food image recognition using very deep convolutional networks. In: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, pp. 41–49 (2016)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. corr abs/1512.03385 (2015)
20. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
21. Horiguchi, S., Amano, S., Ogawa, M., Aizawa, K.: Personalized classifier for food image recognition. IEEE Trans. Multimedia **20**(10), 2836–2848 (2018)
22. Jiang, L., Qiu, B., Liu, X., Huang, C., Lin, K.: DeepFood: food image analysis and dietary assessment via deep model. IEEE Access **8**, 47477–47489 (2020)
23. Kagaya, H., Aizawa, K., Ogawa, M.: Food detection and recognition using convolutional neural network. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 1085–1088 (2014)
24. Kaur, P., Sikka, K., Wang, W., Belongie, S., Divakaran, A.: FoodX-251: a dataset for fine-grained food classification. arXiv preprint arXiv:1907.06167 (2019)

25. Kawano, Y., Yanai, K.: Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8927, pp. 3–17. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16199-0_1
26. Kimura, M., Izawa, R.: Density fixing: simple yet effective regularization method based on the class prior. arXiv preprint arXiv:2007.03899 (2020)
27. Lee, K.H., He, X., Zhang, L., Yang, L.: CleanNet: transfer learning for scalable image classifier training with label noise. In: CVPR, pp. 5447–5456 (2018)
28. Liu, C., Cao, Yu., Luo, Y., Chen, G., Vokkarane, V., Ma, Y.: DeepFood: deep learning-based food image recognition for computer-aided dietary assessment. In: Chang, C.K., Chiari, L., Cao, Yu., Jin, H., Mokhtari, M., Aloulou, H. (eds.) ICOST 2016. LNCS, vol. 9677, pp. 37–48. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39601-9_4
29. MacKay, D.J., Mac Kay, D.J.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
30. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. In: Advances in Neural Information Processing Systems, pp. 7047–7058 (2018)
31. Martinel, N., Foresti, G.L., Micheloni, C.: Wide-slice residual networks for food recognition. In: 2018 IEEE WACV, pp. 567–576. IEEE (2018)
32. Matsuda, Y., Yanai, K.: Multiple-food recognition considering co-occurrence employing manifold ranking. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR 2012), pp. 2017–2020. IEEE (2012)
33. Meyers, A., et al.: Im2Calories: towards an automated mobile vision food diary. In: ICCV, pp. 1233–1241 (2015)
34. Mezgec, S., Seljak, B.K.: Using deep learning for food and beverage image recognition. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 5149–5151. IEEE (2019)
35. Miasnikov, E., Savchenko, A.: Detection and recognition of food in photo galleries for analysis of user preferences. In: Campilho, A., Karray, F., Wang, Z. (eds.) ICIAR 2020. LNCS, vol. 12131, pp. 83–94. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50347-5_9
36. Min, W., Jiang, S., Liu, L., Rui, Y., Jain, R.: A survey on food computing. ACM Comput. Surv. (CSUR) **52**(5), 1–36 (2019)
37. Min, W., Liu, L., Luo, Z., Jiang, S.: Ingredient-guided cascaded multi-a ention network for food recognition (2019)
38. Min, W., et al.: ISIA Food-500: a dataset for large-scale food recognition via stacked global-local attention network. arXiv preprint arXiv:2008.05655 (2020)
39. Ming, Z.-Y., Chen, J., Cao, Yu., Forde, C., Ngo, C.-W., Chua, T.S.: Food photo recognition for dietary tracking: system and experiment. In: Schoeffmann, K., et al. (eds.) MMM 2018. LNCS, vol. 10705, pp. 129–141. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73600-6_12
40. Nag, N., Pandey, V., Jain, R.: Health multimedia: lifestyle recommendations based on diverse observations. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, pp. 99–106 (2017)
41. Ng, H.W., Nguyen, V.D., Vonikakis, V., Winkler, S.: Deep learning for emotion recognition on small datasets using transfer learning. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pp. 443–449 (2015)
42. Oka, R., et al.: Study protocol for the effects of artificial intelligence (AI)-supported automated nutritional intervention on glycemic control in patients with type 2 diabetes mellitus. Diabetes Ther. **10**(3), 1151–1161 (2019)

43. World Health Organization, et al.: Healthy diet. Technical report, World Health Organization. Regional Office for the Eastern Mediterranean (2019)
44. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2009)
45. Pandey, P., Deepthi, A., Mandal, B., Puhan, N.B.: FoodNet: recognizing foods using ensemble of deep networks. IEEE Sig. Process. Lett. **24**(12), 1758–1762 (2017)
46. Peng, Z., Zhang, W., Han, N., Fang, X., Kang, P., Teng, L.: Active transfer learning. IEEE Trans. Circ. Syst. Video Tech. **30**(4), 1022–1036 (2019)
47. Sakai, T., Shimizu, N.: Covariate shift adaptation on learning from positive and unlabeled data. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4838–4845 (2019)
48. Selamat, N.A., Ali, S.H.M.: Automatic food intake monitoring based on chewing activity: a survey. IEEE Access **8**, 48846–48869 (2020)
49. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0026683
50. Shimoda, W., Yanai, K.: Foodness proposal for multiple food detection by training of single food images. In: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, pp. 13–21 (2016)
51. Stojanov, P., Gong, M., Carbonell, J.G., Zhang, K.: Low-dimensional density ratio estimation for covariate shift correction. In: Proceedings of Machine Learning Research, vol. 89, p. 3449 (2019)
52. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
53. Tahir, G.A., Loo, C.K.: An open-ended continual learning for food recognition using class incremental extreme learning machines. IEEE Access **8**, 82328–82346 (2020)
54. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11141, pp. 270–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01424-7_27
55. Tanno, R., Okamoto, K., Yanai, K.: DeepFoodCam: a DCNN-based real-time mobile food recognition system. In: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, pp. 89–89 (2016)
56. Tasci, E.: Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition. Multimed. Tools Appl. **79**, 30397–30418 (2020). https://doi.org/10.1007/s11042-020-09486-1
57. Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, pp. 242–264. IGI Global (2010)
58. Wang, M., Deng, W.: Deep visual domain adaptation: a survey. Neurocomputing **312**, 135–153 (2018)
59. Wang, X., Li, L., Ye, W., Long, M., Wang, J.: Transferable attention for domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5345–5352 (2019)
60. Wang, Y., Chen, J.J., Ngo, C.W., Chua, T.S., Zuo, W., Ming, Z.: Mixed dish recognition through multi-label learning. In: Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities, pp. 1–8 (2019)

61. Wang, Y., Chen, J.J., Ngo, C.W., Chua, T.S., Zuo, W., Ming, Z.: Mixed dish recognition through multi-label learning. In: Proceedings of the 11th Workshop on Multimedia for Cooking and Eating Activities, CEA 2019, p. 1–8. Association for Computing Machinery, New York (2019)
62. Wang, Z., Dai, Z., Póczos, B., Carbonell, J.: Characterizing and avoiding negative transfer. In: CVPR, pp. 11293–11302 (2019)
63. Wu, H., Merler, M., Uceda-Sosa, R., Smith, J.R.: Learning to make better mistakes: semantics-aware visual food recognition. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 172–176 (2016)
64. Wu, J., Huang, Z., Thoma, J., Acharya, D., Van Gool, L.: Wasserstein divergence for GANs. In: Proceedings of ECCV, pp. 653–668 (2018)
65. Yanai, K., Kawano, Y.: Food image recognition using deep convolutional network with pre-training and fine-tuning. In: 2015 IEEE ICMEW, pp. 1–6. IEEE (2015)
66. Zhang, Y., Parker, A.G.: Eat4Thought: a design of food journaling. In: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–8 (2020)
67. Zhao, H., Yap, K.H., Kot, A.C., Duan, L.: JDNet: a joint-learning distilled network for mobile visual food recognition. IEEE J. Sel. Top. Sig. Process. **14**(4), 665–675 (2020)
68. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR, pp. 2921–2929 (2016)

CHAPTER 6

Results and Discussions

In this chapter, we provide a concise overview of the most relevant and impactful results achieved for each component of the data-models-tasks triad. For a deeper dive into the specifics of each topic and its associated results, we encourage the readers to refer to the corresponding research papers presented in the preceding chapters. In each section, we reiterate the questions and illustrate their corresponding solutions.

6.1 Data

This section delves into specific challenges encountered within the data itself. By identifying these challenges, we formulated a series of focused research questions aimed at effective solutions. Each research paper(s) presented tackles the raised question and proposes a corresponding solution. In the following sections, we will provide an overview of the key data challenges we addressed, along with the innovative solutions we developed to overcome them.

6.1.1 Learning with Noisy Labels

LNL methods are designed to address the challenges caused by inaccuracies and inconsistencies in annotations. Building upon the work presented in Chapter 2, our research aimed at significantly improving the sample-selection LNL algorithms. We conducted an in-depth investigation into the core components of DivideMix [LSH20], a benchmark algorithm in this domain, and identified several areas of improvement. Specifically, we addressed the critical challenges of:

- **How to mitigate the impact of DNNs memorization?** We investigated the importance of memorization in LNL algorithms and proposed improvements through DivideMix++. We highlight the importance of this study in Figure 6.1.
- **How does label uncertainty influence the learning process in LNL?** Effectively incorporating label uncertainty into the decision-making process for sample selection using Bayesian Approaches in LNL algorithm Figure 6.2.
- **How to effectively determine the clean and noisy samples?** Achieving a more accurate separation between clean and noisy data points using a Class-conditional Approach. The benefit of this study is highlighted in Figure 6.3.

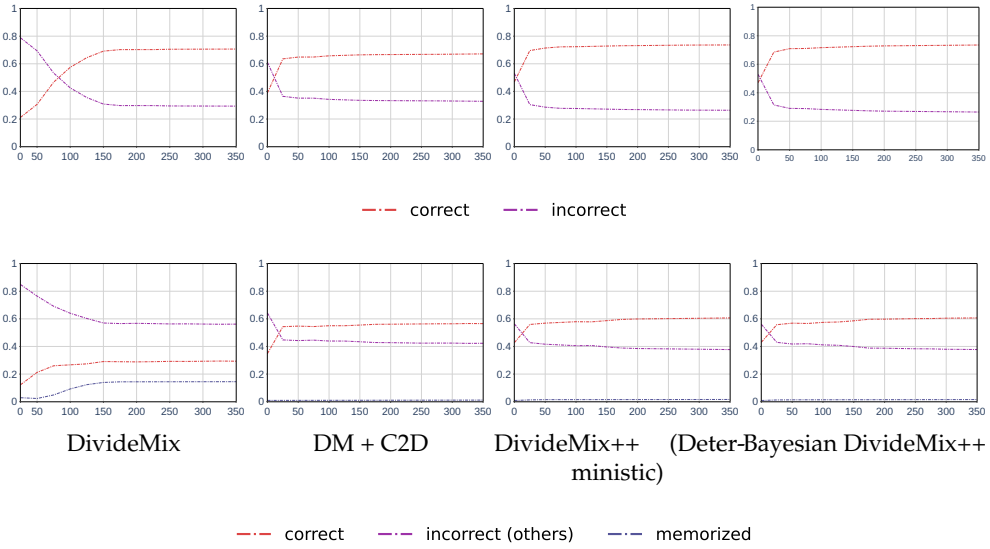


FIGURE 6.1: **Robustness towards Memorization** (CIFAR-100 Sym. 90%). The top row shows the fraction of clean samples that are predicted correctly and incorrectly. The second row shows the fraction of noisy samples that are predicted correctly, memorized and predicted incorrectly. The x-axis corresponds to the epochs and the y-axis corresponds to the fraction of samples.

Method / Noise Ratio		20%	50%	80%	90%
DivideMix	Best	77.3	74.6	60.2	31.5
(ICLR, 2020) [LSH20]	Last	76.9	74.2	59.6	31.0
DM + AugDesc	Best	79.5	77.2	66.4	41.2
(CVPR, 2021) [Nis+21]	Last	79.2	77.0	66.1	40.9
DM + C2D	Best	78.69±0.17	76.43±0.25	67.78±0.30	58.7±0.31
(WACV, 2022) [Zhe+22]	Last	78.32±0.35	76.07±0.41	67.43±0.30	58.45±0.30
Sel-CL+	Best	76.5	72.4	59.6	48.8
(CVPR, 2022) [Li+22]	Last	-	-	-	-
UNICON	Best	78.9	77.6	63.9	44.8
(CVPR, 2022) [Kar+22]	Last	-	-	-	-
ULC	Best	77.3	74.9	61.2	34.5
(AAAI, 2022) [Hua+22]	Last	77.1	74.3	60.8	34.1
LongReMix	Best	77.9±0.2	75.5±0.2	62.3±0.5	34.7±0.3
(PR, 2023) [Cor+23]	Last	77.5±0.2	74.9±0.2	61.7±0.5	30.7±5.9
Lipschitz Reg.	Best	78.8	75.3	62.6	37.5
(PR, 2023) [Mia+23]	Last	77.4	74.3	61.0	34.7
Bayesian	Best	**80.02±0.03**	**78.31±0.14**	**70.01±0.23**	**61.15±0.34**
DivideMix++	Last	79.56±0.13	77.71±0.13	69.55±0.22	60.70±0.42

TABLE 6.1: **CIFAR-100 - Performance comparison** (test accuracy (%, mean ± std over five runs)) with SoTA methods on different symmetric noise settings.

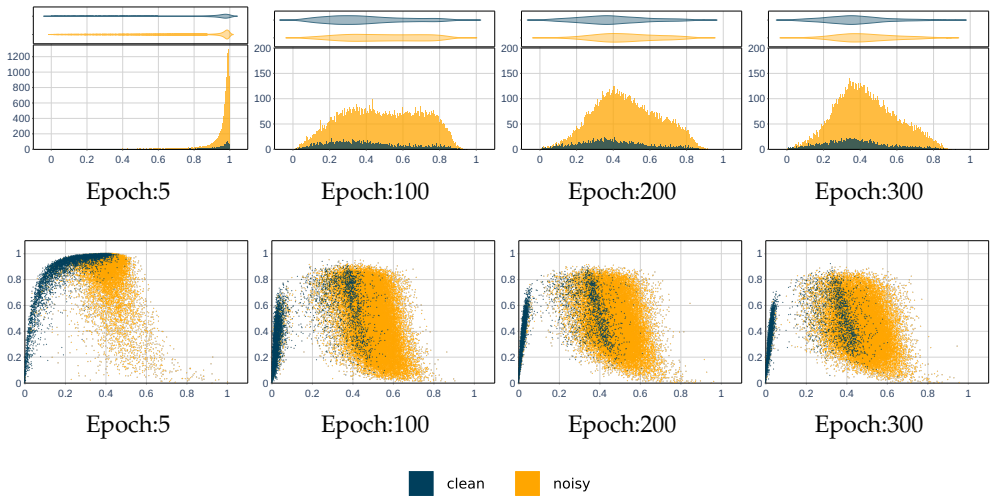


FIGURE 6.2: **Uncertainty of clean and noisy samples for CIFAR-100 90% sym. noise.** The first row corresponds to the histogram of uncertainty values of clean and noisy samples. Note the change in the y-axis for plots other than Epoch:5. The second row corresponds to the uncertainty against the loss values.

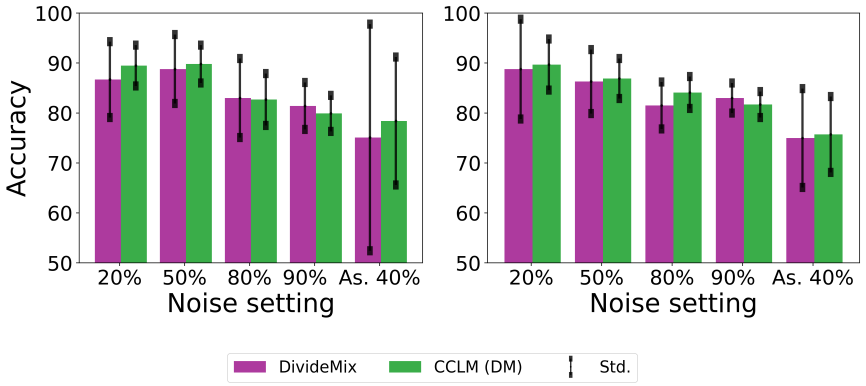


FIGURE 6.3: **Clean/Noisy split accuracy** on CIFAR-10 (left) and CIFAR-100 (right) for all the synthetic noise settings.

Detailed results and discussions are provided in the corresponding papers.

Our proposed approach, Bayesian DivideMix++ *(Nagarajan, Bhalaji, et al., 2024)*, demonstrates significant performance improvements as shown in Table 6.1 (For a broader range of datasets and detailed discussions, refer to the corresponding paper). Through these targeted improvements, we achieved state-of-the-art results in label noise handling. CCLM *(Tatjer, Albert, et al., 2023 and Tatjer, Albert, et al., 2024)* showed the importance of class-conditional approaches in achieving better clean/noisy split. All the advancements enhanced the robustness of the algorithms and opened new avenues for further advancements in this field.

6.1.2 Self-Supervised Learning

This section explores advancements in SSL, particularly focusing on improvements to contrastive learning methods (Detailed discussions can be found in Chapter 3).

How to effectively use neighbours in the Neighbour contrastive learning methods? Our research concentrated on refining the core mechanisms of contrastive learning. We **introduced a novel contrastive approach utilizing "centroids"** along with neighbour contrast and feature contrast and achieved state-of-the-art performances *(Estepa, Imanol G., et al., 2023)*. This approach helped better identify neighbours (visualized in Figure 6.4).

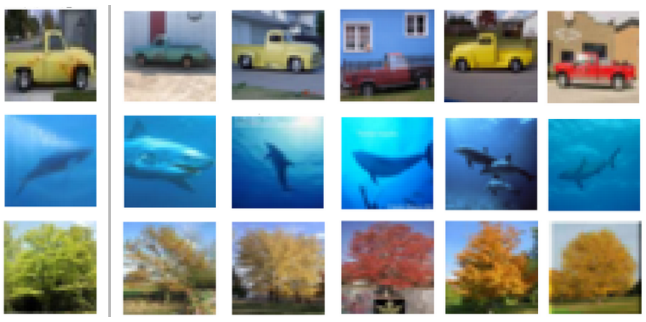


FIGURE 6.4: **Nearest Neighbour extractions performed by All4One.** Query is shown on the left, whereas the Top-5 Neighbours (in order) for each Query are shown in the other columns.

Are all selected neighbours "good" and informative? Superior neighbours directly translated to enhanced performance on downstream tasks. **However, not all neighbours provide meaningful representations.** We investigated the importance of "good" neighbours in effectively selecting neighbours for contrast *(Estepa, Imanol G., et al., 2023b)*. By strategically **incorporating "bridge points",** we improved the performance of benchmark NNCLR [Dwi+21] (illustrated through Figure 6.5 and Figure 6.6). *Thorough discussions are presented in the corresponding papers.*

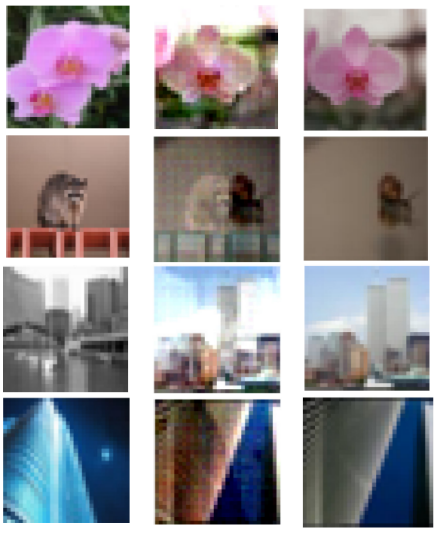


FIGURE 6.5: **Decoded Bridge Point visualizations** of Query (Left), Bridge Points (Middle) and Nearest Neighbour (Right) using encoder-decoder image reconstruction.

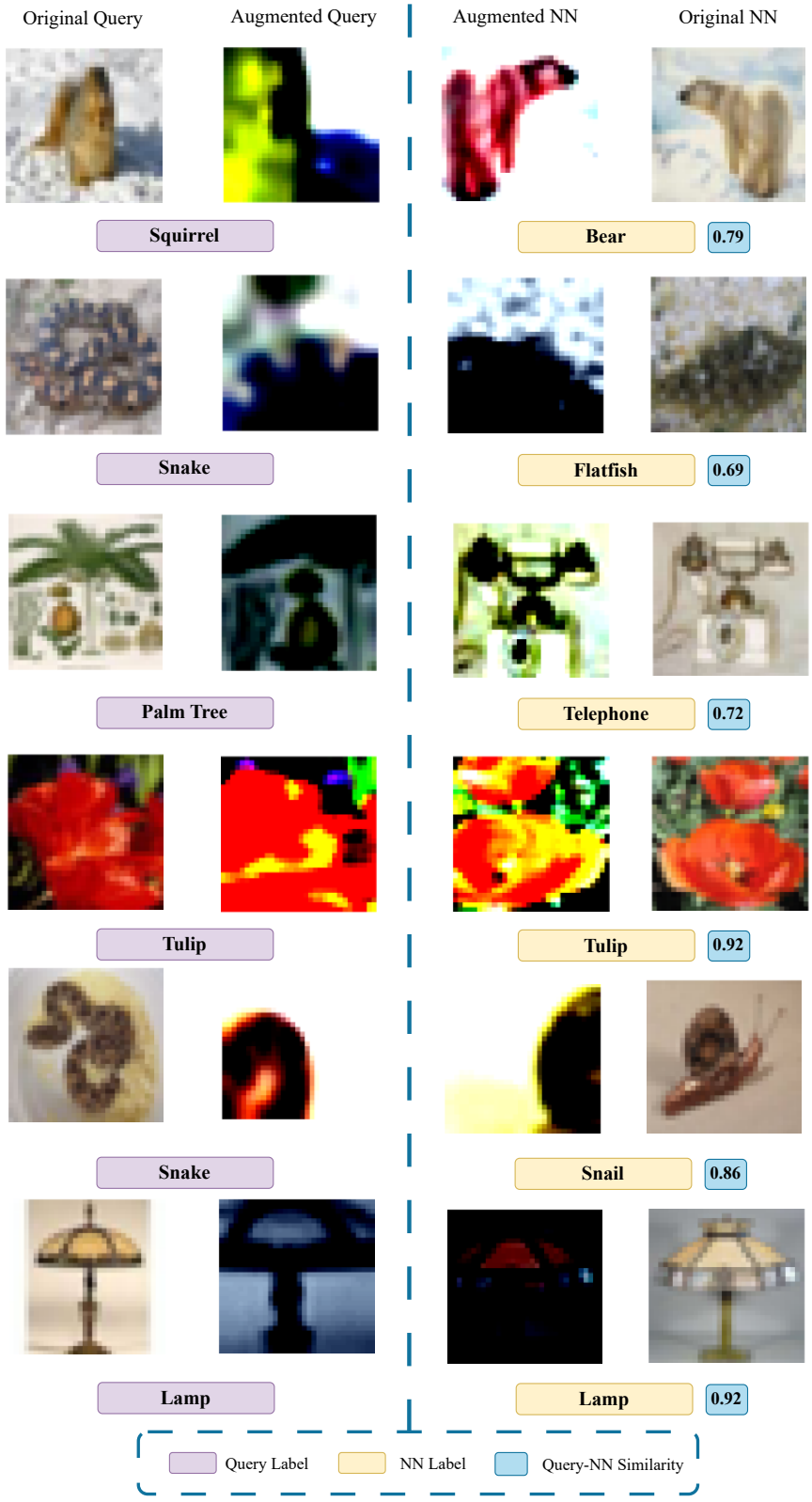
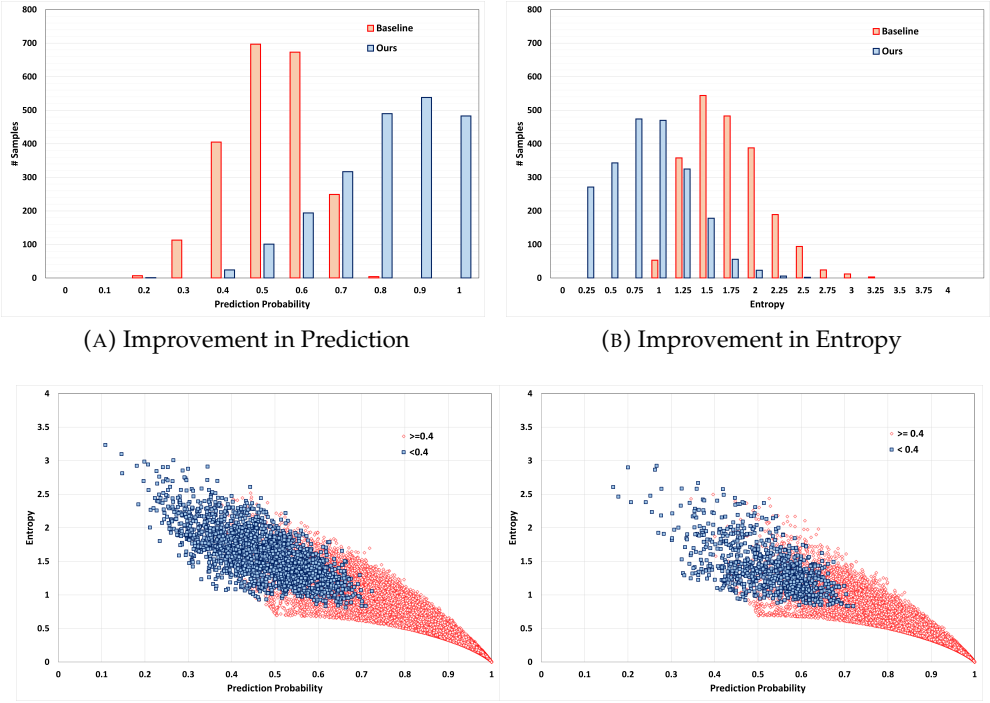


FIGURE 6.6: **Replaced Nearest Neighbour Visualization.**

6.1.3 Learning with Hard Samples

Dataset	Architecture	Baseline	Random Selection	Ours
Food-1K (100)	ResNet-50	83.33%	85.10%	**85.93%**
Food-1K (100)	EfficientNet-B0	88.90%	89.36%	**90.08%**
UECFood-256	EfficientNet-B0	80.66%	80.97%	**81.18%**

TABLE 6.2: Performance comparison of our proposed hard-sample mining method with baseline and random selection techniques



(A) Improvement in Prediction

(B) Improvement in Entropy

(C) Distribution of Baseline

(D) Distribution of Proposed Training

FIGURE 6.7: Improvement experienced on the selected hard samples before and after applying data augmentation.

Real-world datasets inevitably contain "hard samples" that pose significant learning difficulties. We proposed a novel metric called the **"coincidence score"** derived from deep ensembles to pinpoint these hard samples *(Nagarajan, Bhalaji, et al., 2023)*. Once identified, we employed a controlled oversampling strategy specifically for these hard samples allowing the model to focus its learning efforts on these crucial examples. The effectiveness of this approach is shown in Table 6.2. We conducted a series of analyses (Highlighted in Figure 6.7) to provide a deeper understanding of how handling hard samples contributed to improved model learning. A comprehensive discussion on the coincidence score, oversampling strategy, and detailed analysis can be seen in the corresponding paper.

6.2 Models

This section explores the importance of understanding the model behaviour through uncertainty quantification. We focus on measuring epistemic uncertainty, which reflects the model's inherent uncertainty about its predictions. In the previous sections, we utilized uncertainty to improve the proposed LNL method (Refer *Bayesian DivideMix++*), and also used uncertainty to evaluate the effectiveness of our hard sample mining strategy (Refer *Hard-Sample Mining*).

How the quantified uncertainty be effectively leveraged to enhance models? We explore the applicability of uncertainty in various contexts beyond the examples discussed earlier. We highlight two specific applications: (1) Semantic segmentation of food images, and, (2) Model selection for Ensembles. We presented **Bayesian versions** of two state-of-the-art semantic segmentation methods *(Aguilar, Eduardo, et al., 2022)*. We highlighted that the proposed Bayesian methods improved the baseline methods and provided information to improve the decision-making (Figure 6.8 and Figure 6.9).

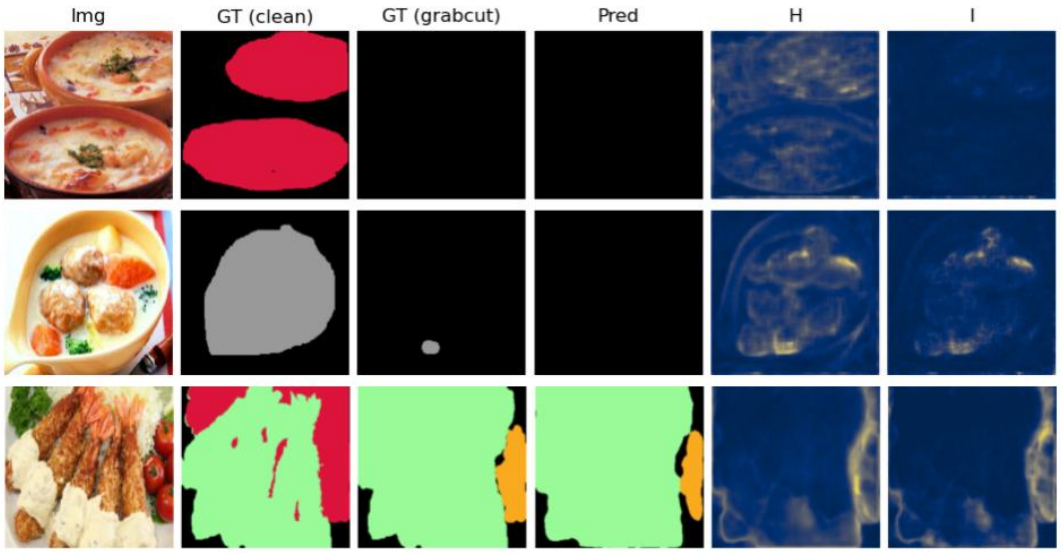


FIGURE 6.8: Qualitative results obtained in the training set of UECFOODPIX [ESY19]. The first, second, and third rows represent images selected using the criteria EM_1, EM_2, and EM_3, respectively. Note that Img, GT (clean), GT (GrabCut), Pred, H, and I correspond to the target image, the clean labels provided by UECFOODPIXComplete [OY21], the GrabCut labels provided by UECFOODPIX, the predicted mask, the uncertainty measured with the entropy, and the uncertainty measured with the mutual information.

How insights gained from uncertainty quantification be used to improve the learning process? Second, we explored how epistemic uncertainty can be leveraged to improve the performance of Deep Ensembles. We proposed a **model selection metric called Forward Step-wise Uncertainty-Aware Model Selection** to select the optimal individual models to incorporate into an ensemble *(Aguilar, Eduardo, et al., 2022b)*. This approach leveraged epistemic uncertainty to guide the model selection. By incorporating uncertainty information, the computational cost involved in an exhaustive search of all possible combinations was reduced. It also helped to reduce the bias on the validation set. Our uncertainty-aware selection strategy successfully identifies high-performing individual models for the ensemble as evidenced from Table 6.3. For a detailed discussion of the proposed method, including the specific uncertainty metric employed, refer to the corresponding paper.

# Models	UECFood256			VIREO172		
	T	N_to_P	P_to_N	T	N_to_P	P_to_N
2	376	0.6862	0.3138	1388	0.6571	0.3429
3	243	0.5844	0.4156	907	0.5976	0.4024
4	145	0.5448	0.4552	626	0.5799	0.4201
5	115	0.5565	0.4435	490	0.5755	0.4245

TABLE 6.3: Data showing the percentage of images that change their predicted class positively (N_to_P) and negatively (P_to_N) after adding a new ResNet50 model to the ensemble by mean of the FS_UAMS approach. *T* denotes the total of images that change their predicted class with respect to the previous ensemble.

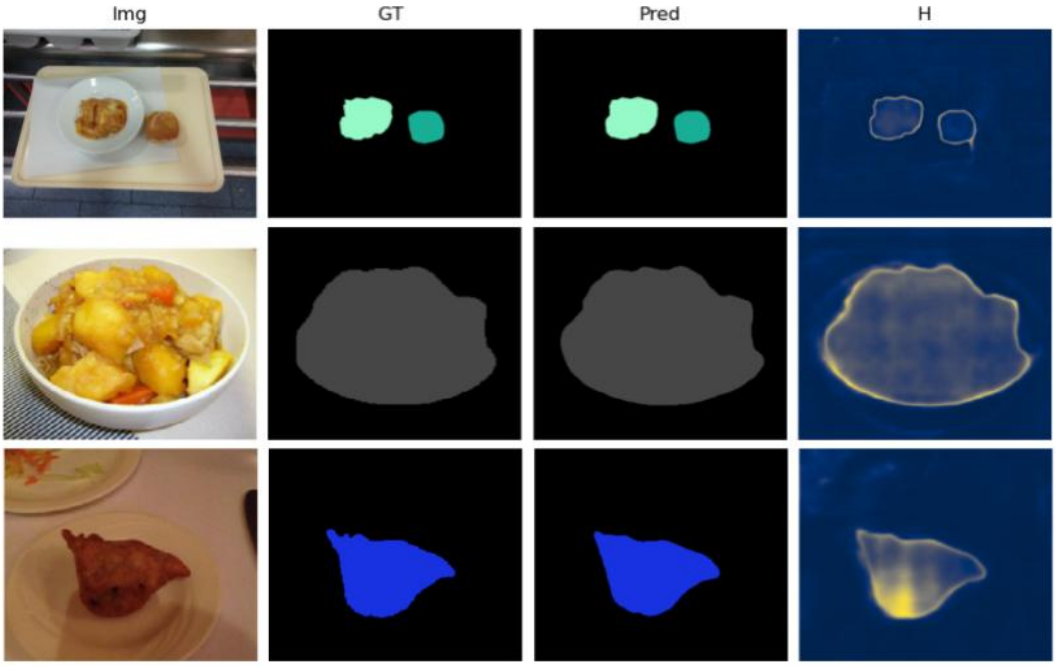


FIGURE 6.9: Images in the test set for the UNIMIB2016 [CNS16] (top), UECFOODPIXComplete [OY21] (middle), and Food201 [Mey+15] (bottom) datasets with the highest performance.

6.3 Tasks

This section emphasizes the critical role of understanding the domain and task in achieving successful DL applications (Refer Chapter 5). We showcase this concept through the specific challenges encountered in food recognition.

What strategies can we employ to enable models to handle fine-grained food classes? One of the key hurdles in food recognition is the fine-grained nature of food classes. To overcome this, we proposed **expert networks** *(Ródenas, Javier, et al., 2022 and Rodríguez-de-Vera, Jesús M., et al., 2023)* that are tasked to learn only clusters of classes, rather than all classes at once. Detailed discussions on this approach can be found in the corresponding papers. The LLM-guided DoD method *(Rodríguez-de-Vera, Jesús M., et al., 2023)* achieved state-of-the-art performance on the Food-101 dataset [BGVG14] as shown in Table 6.4. Expert networks showcase improved classification performance (evidenced by Figure 6.10).

Method	Test Accuracy (%)
Grafit [Tou+21] (ICCV'21)	93.7
EffNet-B7 [TL19] (ICML'19)†	93.0
PMG [Cha+21] (CVPR'21)†§	87.5
FGFR [Ród+22] (Madima'22)§	93.8
DoD + SwinV2-B§	**94.9**

TABLE 6.4: Comparison of DoD with SoTA methods in Food-101. † =bigger image size. § =subset-based method.

FIGURE 6.10: Correctly classified samples of tofu (first row), steak (second row), and ham fried rice (third row) using our Multi-Subset Fine-Grained recognition method. The Baseline classified these samples wrongly.

How can we effectively learn to classify images containing multiple food classes? Real-world food images contain multiple food items on a single plate. We addressed this challenge **by proposing S2ML-TL** *(Nagarajan, Bhalaji, et al., 2021)*, a multi-stage transfer learning framework. S2ML-TL leveraged single-label recognition as an intermediate task, enabling robust and accurate multi-label classification. *Further details and discussions on this framework can be found in the corresponding paper.*

CHAPTER 7

Conclusions

The thesis has revolved around the intricate relationship between data, models and tasks, all of which play critical roles in DL model development for image understanding. Here, we revisit key advancements explored throughout this work, offering a consolidated perspective on the interplay between these essential elements.

7.1 Conclusions

The field of DL has witnessed remarkable growth and continuous evolution. Each element of the triad plays an essential role, and their effective interaction is paramount for achieving success in DL applications. While data, models, and tasks may be considered distinct entities, the synergistic relationship is responsible for the true potential of DL. High-quality data provides the foundation for training powerful models. Models, in turn, learn from the data and refine their ability to solve specific tasks. The choice of task itself can influence the type of data and model architecture required. By recognizing the challenges associated with each, we can develop more robust and effective DL solutions.

The initial stage of this thesis delved into the fundamental challenges associated with data in DL models. It is well established that DL models require vast amounts of labelled data, but acquiring such data can be a significant hurdle. The process of annotation, while crucial, can introduce label errors that ultimately hinder model performance. To address these issues, this research explored the development of noise-robust models. We focused on sample-selection methods, investigating techniques to select clean and noisy data used for training LNL methods. By formulating research questions and tackling them systematically, we enhanced the ability of LNL models to learn effectively even in the presence of noisy labels.

Shifting focus, we explored scenarios with limited or absence of labelled data. We leveraged the power of contrastive SSL algorithms. While these algorithms offer significant capabilities, they also present their unique challenges. We addressed these challenges by investigating and developing improved contrastive methods, ultimately achieving enhanced feature representations from unlabeled data. Furthermore, we explored the inherent heterogeneity within the data itself. To address the challenge of "hard samples", we proposed a sample importance strategy that

prioritizes the training process on these "hard samples," leading to improved overall model performance.

The second part of the thesis focused on the critical role of uncertainty quantification in understanding DL model behaviour. The knowledge of uncertainty is crucial for achieving transparency and improving model performance. Our research investigated the measurement of epistemic uncertainty, which reflects the inherent limitations of the model and the task itself. We explored various tasks, including semantic segmentation, to understand how uncertainty quantification could be applied effectively. We explored how the quantified uncertainty could be leveraged to improve model performance. We investigated methods to utilize uncertainty metrics for tasks like model selection within ensembles. We also used uncertainty measurements to drive and validate decision-making in LNL algorithms and "hard sample" mining methods.

The final part of the thesis explored the importance of task understanding in achieving optimal DL model performance. We used the challenging domain of food recognition as a case study to illustrate this concept. Food recognition presents unique challenges due to the inherent visual complexity of food images. We focussed on solving the tasks of fine-grained recognition and multi-label classification. By using our understanding of these domain-specific challenges, we designed targeted modifications to the DL models. These modifications aimed to address the specific complexities encountered in food recognition tasks.

This thesis has been a journey to advance the frontiers of DL research for image understanding. The core principle guiding this exploration has been the intricate interplay between three critical components: data, models and tasks. Throughout this work, we have visited different challenges associated with each element and explored innovative approaches to address them.

7.2 Thoughts for the Future

Scientific progress thrives on the continuous exchange of knowledge and the pursuit of new questions. As Mario Benedetti aptly stated, "When we believed that we had all the answers, suddenly, all the questions changed." This sentiment holds for our research. While this thesis has addressed key challenges and presented significant advancements in the realm of DL, it also opens doors to exciting new research directions.

Leveraging Generative Models A substantial portion of this thesis delved deeply into the intricate realm of data and its pivotal role in DL models. Recent advancements in generative models have opened doors to synthetic data creation. Synthetic data can mitigate the limitations of real-world data scarcity. Recent research suggests that DL models can be developed to learn informative representations within the latent space directly and can be used to generate realistic data points. This capability of generative models holds immense promise for SSL techniques.

Rise of Large Models Recent breakthroughs in high-performance computing have paved the way for the development of massive foundation models and pre-trained models. These powerful models exhibit remarkable versatility, demonstrating the ability to transfer learned knowledge across diverse downstream tasks. Large models

with their inherent capacity for robust feature extraction could potentially be leveraged to improve noise detection in LNL algorithms and enhance feature learning in SSL methods.

Make Models Explainable This thesis was written at the juncture of the EU AI Act. As AI applications become increasingly sophisticated and integrated into our daily lives, legal frameworks are rapidly evolving to address ethical considerations, transparency, and accountability. It is therefore paramount to develop models capable of informed decision-making with more explainable tools.

Multi-modal Learning The recent surge in multi-modal learning and large language models can empower models to learn from a rich tapestry of data modalities. These diverse forms of information allow us to create richer and more informative feature representations. By enabling models to capture the relationships and complementary information from different modalities, we can enhance the model's ability to perform complex tasks.

"AI is fast evolving, and it will continue to be!"

The field of AI has witnessed remarkable growth, particularly in recent years.
This rapid progress has been marked by increasingly powerful models and the widespread adoption of DL techniques across diverse domains.
We are at a pivotal point, where it is the time to embrace bold thinking and set ambitious goals that can leverage the power of AI to
create a more impactful and positive future.

APPENDIX A

Research Activities

Journals

» Albert Tatjer[‡], **Bhalaji Nagarajan[‡],** Ricardo Marques[‡‡], and Petia Radeva[‡‡]. "Decoding Class Dynamics in Learning with Noisy Labels." Pattern Recognition Letters, 2024. **(JCR: Q2, IF: 5.1)**.

» **Bhalaji Nagarajan**, Ricardo Marques, Eduardo Aguilar, and Petia Radeva. "Bayesian DivideMix++ for Enhanced Learning with Noisy Labels." Neural Networks, 2024. **(JCR: Q1, IF: 7.8)**.

» **Bhalaji Nagarajan**, Marc Bolaños, Eduardo Aguilar, and Petia Radeva. "Deep Ensemble-Based Hard Sample Mining for Food Recognition." Journal of Visual Communication and Image Representation, 2023. **(JCR: Q2, IF: 2.6)**.

» Eduardo Aguilar, **Bhalaji Nagarajan**, Beatriz Remeseiroc, Petia Radeva. "Bayesian Deep Learning for Semantic Segmentation of Food Images." Computers and Electrical Engineering, 2022. **(JCR: Q2, IF: 4.3)**.

» Eduardo Aguilar, **Bhalaji Nagarajan**, and Petia Radeva. "Uncertainty-Aware Selecting for an Ensemble of Deep Food Recognition Models." Computers in Biology and Medicine, 2022. **(JCR: Q1, IF: 7.7)**.

International Conferences and Workshops

» Jesús M. Rodríguez-de-Vera, Pablo Villacorta, Imanol G. Estepa, Marc Bolaños, Ignacio Sarasúa, **Bhalaji Nagarajan**, and Petia Radeva. "Dining on Details: LLM-Guided Expert Networks for Fine-Grained Food Recognition." ACMMM-W, 2023. **(Oral and Poster).**

» Imanol G. Estepa, Ignacio Sarasúa, **Bhalaji Nagarajan**, and Petia Radeva. "All4One: Symbiotic Neighbour Contrastive Learning via Self-Attention and Redundancy Reduction." ICCV, 2023. **(GGS Rating: A++)**.

» Imanol G. Estepa, Jesús M. Rodríguez-de-Vera, **Bhalaji Nagarajan**, and Petia Radeva. "Good Fences Makes Good Neighbours." ICCV-W, 2023. **(Poster).**

» Albert Tatjer[‡], **Bhalaji Nagarajan[‡]**, Ricardo Marques, and Petia Radeva. "CCLM: Class-conditional Label Noise Modelling." IbPRIA, 2023. **(Oral, Best Paper Award)**.

[‡]Equal Contributions
[‡‡]Equal Supervision

- Javier Ródenas[‡], **Bhalaji Nagarajan**[‡], Marc Bolaños, and Petia Radeva. "Learning Multi-Subset of Classes for Fine-Grained Recognition." ACMMM-W, 2022. **(Oral and Poster).**
- **Bhalaji Nagarajan**, Ricardo Marques, Marcos Mejia, and Petia Radeva. "Class-conditional Importance Weighting for Deep Learning with Noisy Labels." VISAPP, 2022. **(Oral).**
- Nil Ballús, **Bhalaji Nagarajan**, and Petia Radeva. "Opt-SSL: An Enhanced Self-Supervised Framework for Food Recognition." IbPRIA, 2022. **(Poster).**
- **Bhalaji Nagarajan**, Eduardo Aguilar, and Petia Radeva. "S2ML-TL Framework for Multi-label Food Recognition." ICPR-W, 2021. **(Oral).**

Book Chapters **(Not Included in this Thesis)**

- **Bhalaji Nagarajan**, Rupali Khatun, Marc Bolaños, Eduardo Aguilar, Leonardo Angelini, Mira El Kamali, Elena Mugellini, Omar Abou Khaled, Noemi Boqué, Lucia Tarro, and Petia Radeva. "Nutritional Monitoring in Older People Prevention Services." Digital Health Technology for Better Aging, Springer, 2021.

Other Contributions **(Not Included in this Thesis)**

- Jesús M. Rodríguez-de-Vera, Imanol G. Estepa, Ignacio Sarasúa, **Bhalaji Nagarajan**, and Petia Radeva. "Precision at Scale: Domain-Specific Datasets On-Demand." *(Under Review).*
- Changhui Hu[‡], **Bhalaji Nagarajan**[‡], Ricardo Marques, and Petia Radeva. "Adaptive Vision-Language Prompt Learners for Learning with Noisy Labels" *(Under Review).*
- Jesús M. Rodríguez-de-Vera, Imanol G. Estepa, Marc Bolaños, **Bhalaji Nagarajan**, and Petia Radeva. "LOFI: LOng-tailed FIne-Grained Network for Food Recognition." CVPR-W, 2024. **(Oral and Poster).**
- Guillem Martinez, Maya Aghaei, Martin Dijkstra, **Bhalaji Nagarajan**, Femke Jaarsma, Jaap van de Loosdrecht, Petia Radeva, and Klaas Dijkstra. "Hyper-Spectral Imaging for Overlapping Plastic Flakes Segmentation." ICIP, 2022. **(GGS Rating: A-)**. **(Oral).**
- Eduardo Aguilar, **Bhalaji Nagarajan**, Rupali Khantun, Marc Bolaños, and Petia Radeva. "Uncertainty-Aware Data Augmentation for Food Recognition." ICPR, 2021. **(GGS Rating: A-)**. **(Oral).**
- Eduardo Aguilar, **Bhalaji Nagarajan,** Rupali Khatun, Marc Bolaños, and Petia Radeva. "Uncertainty Modeling and Deep Learning Applied to Food Image Analysis." BIOSTEC, 2020. **(Oral).**

Co-Supervisions

TFM **(Trabajo de Fin de Máster)**

- Sergio Rodriguez Llana, "Oropharyngeal Squamous Cell Carcinoma Classification of HPV Status through CT Scans", *Jan 2024 - Present.*
- Imanol Gonzalez Estepa, "Analysis and Improvement Proposal on Self-supervised Deep Learning", *Sep 2022 - Feb 2023.* **(Matrícula de honor).**
- Jesús Molina Rodríguez De Vera, "Deep Learning for Food Instance Segmentation", *Sep 2022 - Feb 2023.*

TFG (Trabajo Final de Grado)

- Oscar Blazquez Jimenez, "Exploring Evidential Uncertainty in Learning with Noisy Labels", *Sep 2023 - Present.*
- Nil Ballús Riu, "Exploring Self-supervised Learning in Deep Learning for Food Recognition", *Sep 2021 - Jan 2022.*
- Ling Zhu, "Using Deep Learning for Food Recognition", *Feb 2020 - Sep 2020.*

Research Funds

Doctoral Funding

- FPI (MICINN), *Feb 2021 - Jul 2023.*
- FI-AGAUR (Resigned), *Jul 2020 - Jan 2021.*
- ICREA Academic 2014 Research Grant, *Feb 2019 - Mar 2020.*

Computational Resources Funding

- EuroHPC Development Access Call, EHPC-DEV-2023D12-059, *Jan 2024 - Present.*
- RED Española de Supercomputación (RES), IM-2023-2-0025 *(Jul 2023 - Oct 2023)* and IM-2023-3-0019 *(Nov 2023 - Feb 2024)*.

Invited Talks

- "From Pixels to Plates - Exploring the Open Challenges in Visual Food Analysis", HPAI Annual Meeting, Barcelona Supercomputing Center, *Dec 01, 2023.*
- "Self-supervised Fine-grained Food Recognition", GRAPES Learning Week and Advanced Topics in ML Workshop, Universitat de Barcelona, *Sept 07, 2023.*
- "From Pixels to Plates: Convergence of Machine Learning and Statistics in Food Recognition", Computer Science Meets Mathematics Workshop, Universitat de Barcelona, *Sept 06, 2023.*

Participation in Challenges

- AIcrowd Food Recognition Challenge, 7^{th} *position*, 2022.
- AIcrowd Food Recognition Challenge, 8^{th} *position*, 2021.
- iFood Challenge (FGVC6 Workshop), CVPR, 8^{th} *position*, 2019.

Presentations

- Deep Learning Barcelona Symposium (DLBCN), 2022 (Oral and Poster), 2023 (Poster).
- 2^{nd} ELLIS Doctoral School (Poster), 2022.
- CVC Workshop on Computer Vision Trends and Challenges, (Oral and Poster), 2022.

Other Activities

- Teaching Activities: Computational Vision (Laboratory), *Masters in Artificial Intelligence*, University of Barcelona, 2019-2022 (4 semesters).
- Seasonal School Instructor: IEEE SPS Summer School, MIT-WPU, Pune, 2021.
- Student Volunteer: ACM Multimedia, 2022 and ICCV, 2023.
- Organizing Weekly Seminars (since 2021), AIBA, University of Barcelona.
- Server and System Maintenance (GPU Systems), AIBA, University of Barcelona.

Bibliography

[Tou+23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023).

[Alm+23] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. "Falcon-40B: an open large language model with state-of-the-art performance". In: (2023).

[Zha+22a] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. "Scaling vision transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 12104–12113.

[Bal+23] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. "A cookbook of self-supervised learning". In: *arXiv preprint arXiv:2304.12210* (2023).

[SP97] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[SMH11] Ilya Sutskever, James Martens, and Geoffrey E Hinton. "Generating text with recurrent neural networks". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 1017–1024.

[Zhu+24] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. "Toolqa: A dataset for llm question answering with external tools". In: *Advances in Neural Information Processing Systems* 36 (2024).

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[Cro+23] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. "Diffusion models in vision: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[Ram+21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation". In: *International conference on machine learning*. Pmlr. 2021, pp. 8821–8831.

[Ess+24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. "Scaling rectified flow transformers for high-resolution image synthesis". In: *arXiv preprint arXiv:2403.03206* (2024).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[Dos+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks". In: *science* 313.5786 (2006), pp. 504–507.

[Wan+23] Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. "Large-scale multi-modal pre-trained models: A comprehensive survey". In: *Machine Intelligence Research* 20.4 (2023), pp. 447–482.

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3 (2016), pp. 1–40.

[Ray+23] Abhisek Ray, Maheshkumar H Kolekar, Raman Balasubramanian, and Adel Hafiane. "Transfer learning enhanced vision-based human activity recognition: A decade-long analysis". In: *International Journal of Information Management Data Insights* 3.1 (2023), p. 100142.

[The+23] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. "Transfer learning enables predictions in network biology". In: *Nature* 618.7965 (2023), pp. 616–624.

[Zha+22b] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. "A survey on negative transfer". In: *IEEE/CAA Journal of Automatica Sinica* 10.2 (2022), pp. 305–329.

[Wan+19b] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. "Characterizing and avoiding negative transfer". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 11293–11302.

[Guo+16] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. "Deep learning for visual understanding: A review". In: *Neurocomputing* 187 (2016), pp. 27–48.

[LLA22] Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. "An introduction to deep learning in natural language processing: Models, techniques, and tools". In: *Neurocomputing* 470 (2022), pp. 443–456.

[Pra+23] Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe. "End-to-end speech recognition: A survey". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).

[Cho+24] Joseph Cho, Fachrina Dewi Puspitasari, Sheng Zheng, Jingyao Zheng, Lik-Hang Lee, Tae-Ho Kim, Choong Seon Hong, and Chaoning Zhang. "Sora as an AGI World Model? A Complete Survey on Text-to-Video Generation". In: *arXiv preprint arXiv:2403.05131* (2024).

[Rom+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.

[Don+18] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[Vya+23] Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. "Audiobox: Unified audio generation with natural language prompts". In: *arXiv preprint arXiv:2312.15821* (2023).

[Foo22] Keith D. Foote. *A Brief History of Deep Learning*. 2022. URL: https://www.dataversity.net/brief-history-deep-learning/ (visited on 04/18/2024).

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[Rus+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[SZ14] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[TL19] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[TL21] Mingxing Tan and Quoc Le. "Efficientnetv2: Smaller models and faster training". In: *International conference on machine learning*. PMLR. 2021, pp. 10096–10106.

[She+23] Li Shen, Yan Sun, Zhiyuan Yu, Liang Ding, Xinmei Tian, and Dacheng Tao. "On efficient training of large-scale deep learning models: A literature review". In: *arXiv preprint arXiv:2304.03589* (2023).

[Zha+23] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. "Data-centric ai: Perspectives and challenges". In: *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM. 2023, pp. 945–948.

[Rid+21] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. "Imagenet-21k pretraining for the masses". In: *arXiv preprint arXiv:2104.10972* (2021).

[SKL19] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. "Selfie: Refurbishing unclean samples for robust deep learning". In: *International conference on machine learning*. PMLR. 2019, pp. 5907–5915.

[Yan+14] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. "Learning from multiple annotators with varying expertise". In: *Machine learning* 95 (2014), pp. 291–327.

[FBQSM16] C Fabian Benitez-Quiroz, Ramprakash Srinivasan, and Aleix M Martinez. "Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 5562–5570.

[NAM21] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. "Pervasive label errors in test sets destabilize machine learning benchmarks". In: *arXiv preprint arXiv:2103.14749* (2021).

[CS21] Adam Conner-Simons. *Major ML datasets have tens of thousands of errors*. 2021. URL: https://www.csail.mit.edu/news/major-ml-datasets-have-tens-thousands-errors (visited on 04/18/2024).

[Son+22] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. "Learning from noisy labels with deep neural networks: A survey". In: *IEEE transactions on neural networks and learning systems* (2022).

[AZLL19] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. "Learning and generalization in overparameterized neural networks, going beyond two layers". In: *Advances in neural information processing systems* 32 (2019).

[Zha+21a] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[AL88] Dana Angluin and Philip Laird. "Learning from noisy examples". In: *Machine Learning* 2 (1988), pp. 343–370.

[ZNS21] Yivan Zhang, Gang Niu, and Masashi Sugiyama. "Learning noise transition matrix from only noisy labels via total variation regularization". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12501–12512.

[Wu+21] Songhua Wu, Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Nannan Wang, Haifeng Liu, and Gang Niu. "Class2simi: A noise reduction perspective on learning with noisy labels". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 11285–11295.

[Yao+20] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. "Dual t: Reducing estimation error for transition matrix in label-noise learning". In: *Advances in neural information processing systems* 33 (2020), pp. 7260–7271.

[Zha+17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[Che+22] Yingyi Chen, Shell Xu Hu, Xi Shen, Chunrong Ai, and Johan AK Suykens. "Compressing features for learning with noisy labels". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[WSF22] Yikai Wang, Xinwei Sun, and Yanwei Fu. "Scalable penalized regression for noise detection in learning with noisy labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 346–355.

[Wan+19a] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. "Symmetric cross entropy for robust learning with noisy labels". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 322–330.

[Ma+20] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. "Normalized loss functions for deep learning with noisy labels". In: *International conference on machine learning*. PMLR. 2020, pp. 6543–6553.

[Li+22] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. "Selective-supervised contrastive learning with noisy labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 316–325.

[LSH20] Junnan Li, Richard Socher, and Steven CH Hoi. "Dividemix: Learning with noisy labels as semi-supervised learning". In: *arXiv preprint arXiv:2002.07394* (2020).

[Kar+22] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. "Unicon: Combating label noise through uniform selection and contrastive learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9676–9686.

[Cai+23] Kecan Cai, Hongyun Zhang, Witold Pedrycz, and Duoqian Miao. "SSS-Net: A shadowed-sets-based semi-supervised sample selection network for classification on noise labeled images". In: *Knowledge-Based Systems* (2023), p. 110732.

[Ozb+23] Utku Ozbulak, Hyun Jung Lee, Beril Boga, Esla Timothy Anzaku, Homin Park, Arnout Van Messem, Wesley De Neve, and Joris Vankerschaver. "Know your self-supervised learning: A survey on image-based generative and discriminative training". In: *arXiv preprint arXiv:2305.13689* (2023).

[Liu+21] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. "Self-supervised learning: Generative or contrastive". In: *IEEE transactions on knowledge and data engineering* 35.1 (2021), pp. 857–876.

[Bao+21] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. "Beit: Bert pre-training of image transformers". In: *arXiv preprint arXiv:2106.08254* (2021).

[He+21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. "MAE: Masked Autoencoders Are Scalable Vision Learners". In: *arXiv preprint arXiv:2111.06377* (2021).

[Li+23] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. "Mage: Masked generative encoder to unify representation learning and image synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2142–2152.

[Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[Zbo+21] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. "Barlow twins: Self-supervised learning via redundancy reduction". In: *International conference on machine learning*. PMLR. 2021, pp. 12310–12320.

[Dwi+21] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9588–9597.

[Bro+93] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems* 6 (1993).

[OS+16] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. "Deep metric learning via lifted structured feature embedding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4004–4012.

[He+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[Gri+20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in neural information processing systems* 33 (2020), pp. 21271–21284.

[CH21] Xinlei Chen and Kaiming He. "Exploring simple siamese representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15750–15758.

[KTP21] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. "Mean shift for self-supervised learning". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10326–10335.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.

[Sun+17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. "Revisiting unreasonable effectiveness of data in deep learning era". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

[Jia+20] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. "Characterizing structural regularities of labeled data in overparameterized models". In: *arXiv preprint arXiv:2002.03206* (2020).

[ADH22] Chirag Agarwal, Daniel D'souza, and Sara Hooker. "Estimating example difficulty using variance of gradients". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10368–10378.

[Agu+21] Eduardo Aguilar, Bhalaji Nagarajan, Rupali Khantun, Marc Bolaños, and Petia Radeva. "Uncertainty-aware data augmentation for food recognition". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 4017–4024.

[LPK20] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. "Explainable ai: A review of machine learning interpretability methods". In: *Entropy* 23.1 (2020), p. 18.

[Pan+23] Cecilia Panigutti, Ronan Hamon, Isabelle Hupont, David Fernandez Llorca, Delia Fano Yela, Henrik Junklewitz, Salvatore Scalzo, Gabriele Mazzini, Ignacio Sanchez, Josep Soler Garrido, et al. "The role of explainable AI in the context of the AI Act". In: *Proceedings of the 2023 ACM conference on fairness, accountability, and transparency*. 2023, pp. 1139–1150.

[SO23] Waddah Saeed and Christian Omlin. "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities". In: *Knowledge-Based Systems* 263 (2023), p. 110273.

[DBH18] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. "Explainable artificial intelligence: A survey". In: *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE. 2018, pp. 0210–0215.

[Zha+21b] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742.

[Gal+16] Yarin Gal et al. "Uncertainty in deep learning". In: (2016).

[DKD09] Armen Der Kiureghian and Ove Ditlevsen. "Aleatory or epistemic? Does it matter?" In: *Structural safety* 31.2 (2009), pp. 105–112.

[Soi17] Christian Soize. *Uncertainty quantification*. Springer, 2017.

[Abd+21] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information fusion* 76 (2021), pp. 243–297.

[Gaw+23] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. "A survey of uncertainty in deep neural networks". In: *Artificial Intelligence Review* 56.Suppl 1 (2023), pp. 1513–1589.

[Mac92] David JC MacKay. "A practical Bayesian framework for backpropagation networks". In: *Neural computation* 4.3 (1992), pp. 448–472.

[Nea12] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[KG17] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* 30 (2017).

[Gri11] Justin Grimmer. "An introduction to Bayesian inference via variational approximations". In: *Political Analysis* 19.1 (2011), pp. 32–47.

[GG16] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[Sal+21] Nareen OM Salim, Subhi RM Zeebaree, Mohammed AM Sadeeq, AH Radie, Hanan M Shukur, and Zryan Najat Rashid. "Study for food recognition system using deep learning". In: *Journal of Physics: Conference Series*. Vol. 1963. 1. IOP Publishing. 2021, p. 012014.

[KPM20] Chairi Kiourt, George Pavlidis, and Stella Markantonatou. "Deep learning approaches in food recognition". In: *Machine Learning Paradigms: Advances in Deep Learning-based Technological Applications* (2020), pp. 83–108.

[Min+19] Weiqing Min, Shuqiang Jiang, Linhu Liu, Yong Rui, and Ramesh Jain. "A survey on food computing". In: *ACM Computing Surveys (CSUR)* 52.5 (2019), pp. 1–36.

[Yun+18] Raza Yunus, Omar Arif, Hammad Afzal, Muhammad Faisal Amjad, Haider Abbas, Hira Noor Bokhari, Syeda Tazeen Haider, Nauman Zafar, and Raheel Nawaz. "A framework to estimate the nutritional value of food in real time using deep learning techniques". In: *IEEE Access* 7 (2018), pp. 2643–2652.

[Min+23] Weiqing Min, Zhiling Wang, Yuxin Liu, Mengjiang Luo, Liping Kang, Xiaoming Wei, Xiaolin Wei, and Shuqiang Jiang. "Large scale visual food recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[Nis+21] Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. "Augmentation strategies for learning with noisy labels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8022–8031.

[Zhe+22] Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. "Contrast to divide: Self-supervised pre-training for learning with noisy labels". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 1657–1667.

[Hua+22] Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. "Uncertainty-aware learning against label noise on imbalanced datasets". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 6. 2022, pp. 6960–6969.

[Cor+23] Filipe R Cordeiro, Ragav Sachdeva, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. "Longremix: Robust learning with high confidence samples in a noisy label environment". In: *Pattern Recognition* 133 (2023), p. 109013.

[Mia+23] Qing Miao, Xiaohe Wu, Chao Xu, Wangmeng Zuo, and Zhaopeng Meng. "On better detecting and leveraging noisy samples for learning with severe label noise". In: *Pattern Recognition* 136 (2023), p. 109210.

[ESY19] Takumi Ege, Wataru Shimoda, and Keiji Yanai. "A new large-scale food image segmentation dataset and its application to food calorie estimation based on grains of rice". In: *Proceedings of the 5th international workshop on multimedia assisted dietary management*. 2019, pp. 82–87.

[OY21] Kaimu Okamoto and Keiji Yanai. "UEC-FoodPIX Complete: A large-scale food image segmentation dataset". In: *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part V*. Springer. 2021, pp. 647–659.

[CNS16] Gianluigi Ciocca, Paolo Napoletano, and Raimondo Schettini. "Food recognition: a new dataset, experiments, and results". In: *IEEE journal of biomedical and health informatics* 21.3 (2016), pp. 588–598.

[Mey+15] Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P Murphy. "Im2Calories: towards an automated mobile vision food diary". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1233–1241.

[BGVG14] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. "Food-101–mining discriminative components with random forests". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*. Springer. 2014, pp. 446–461.

[Tou+21] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. "Grafit: Learning Fine-Grained Image Representations with Coarse Labels". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 874–884.

[Cha+21] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. "Your" Flamingo" is my" Bird": fine-grained, or not". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11476–11485.

[Ród+22] Javier Ródenas, Bhalaji Nagarajan, Marc Bolaños, and Petia Radeva. "Learning Multi-Subset of Classes for Fine-Grained Food Recognition". In: *Proceedings of the 7th International Workshop on Multimedia Assisted Dietary Management*. 2022, pp. 17–26.

This work was partially funded by the EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR) (previously SGR-2017 1742), Icrea Academia'2014 and Icrea Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET-000089434, Erasmus+ EU), DeepSense (ACE053/22/000029, ACCIÓ), TIN2018-095232-B-C21, European Union's Horizon 2020 Research and Innovation Programme under the Open Call budget of the grant agreement No. 857159 (SHAPES), Nestore project of the European Commission Horizon 2020 programme (Grant no. 769643), Greenhabit EIT Digital program, Validithi EIT Health program, CERCA Programme/Generalitat de Catalunya, and Grants PID2022-141566NB-I00 (IDEATE), PDC2022-133642-I00 (DeepFoodVol), and CNS2022-135480 (A-BMC) funded by MICIU/AEI/10.13039/501100011033, by FEDER (UE), and by European Union NextGenerationEU/PRTR. We acknowledge the support of FI-AGAUR Becas and FPI Becas, MICINN, Spain. We thankfully acknowledge the computer resources at FinisTerrae III and the technical support provided by the Galician Supercomputing Center (CESGA) (RES-IM-2023-2-0025, RES-IM-2023-3-0019). We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.