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

A R T I C L E I N F O

MSC:
68T05
68T10
68T30
68T37

Keywords:
Knowledge representation
Hard-sample mining
Food recognition
Deep ensembles
Data augmentation

A B S T R A C T

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/

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.

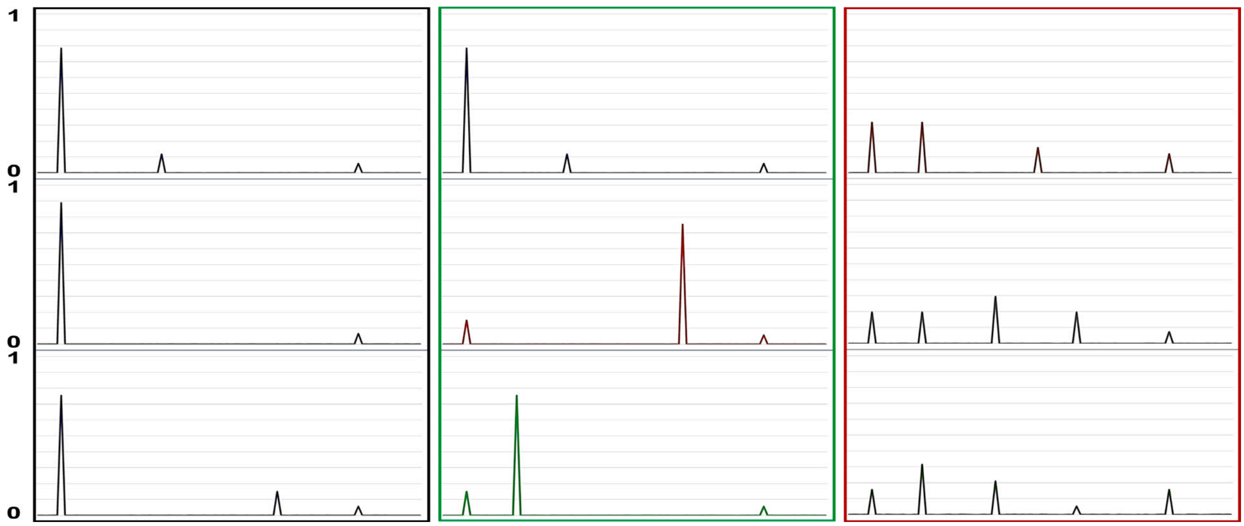


Fig. 2. Behaviour of three different models (rows) in a deep ensemble (the x-axis corresponds to the class index and the y-axis corresponds to the likelihood obtained by individual models in the ensemble). Left column: all three models predict the same class; middle column: all models predict different classes; and right column: models give low likelihood to all classes.

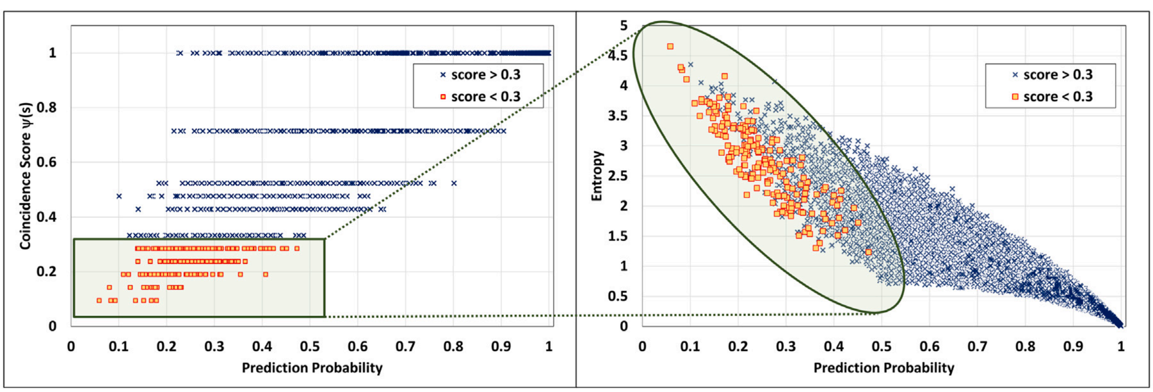


Fig. 3. Coincidence score vs. Prediction probability of samples (left plot). The relation between coincidence score and entropy is highlighted - a low coincidence score relates to high entropy.

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)).

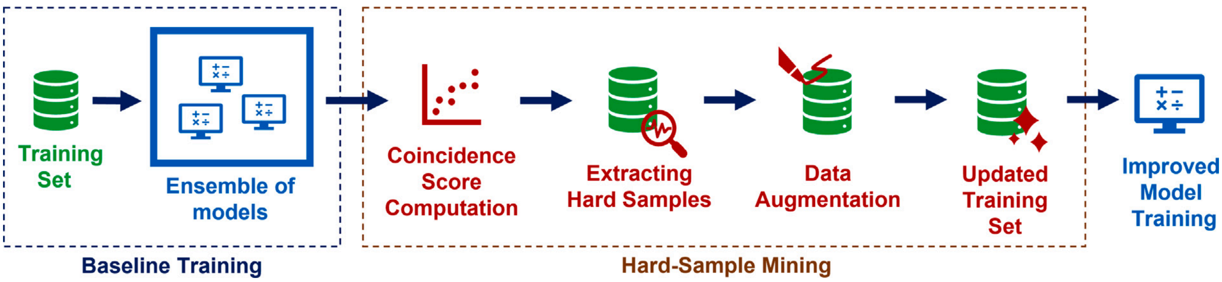


Fig. 4. Overall Training Pipeline. An ensemble is created during the baseline training; the dataset is updated using the hard samples mined using the coincidence score.

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

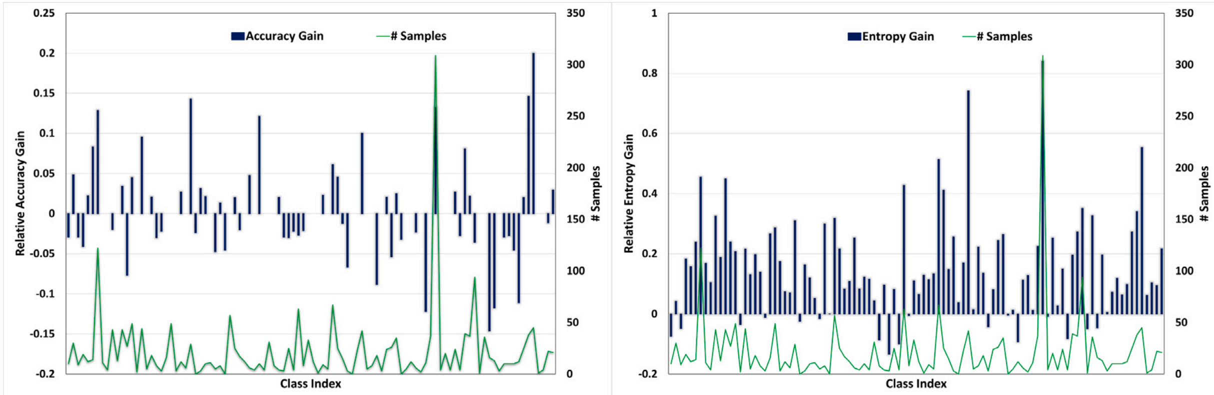
We evaluate the performance of the models using the accuracy of the validation set (or test set, in case the dataset contains only training and test splits). For all the experiments, we report the median accuracy of the individual models as a performance metric. We use median accuracy as it allows us 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.

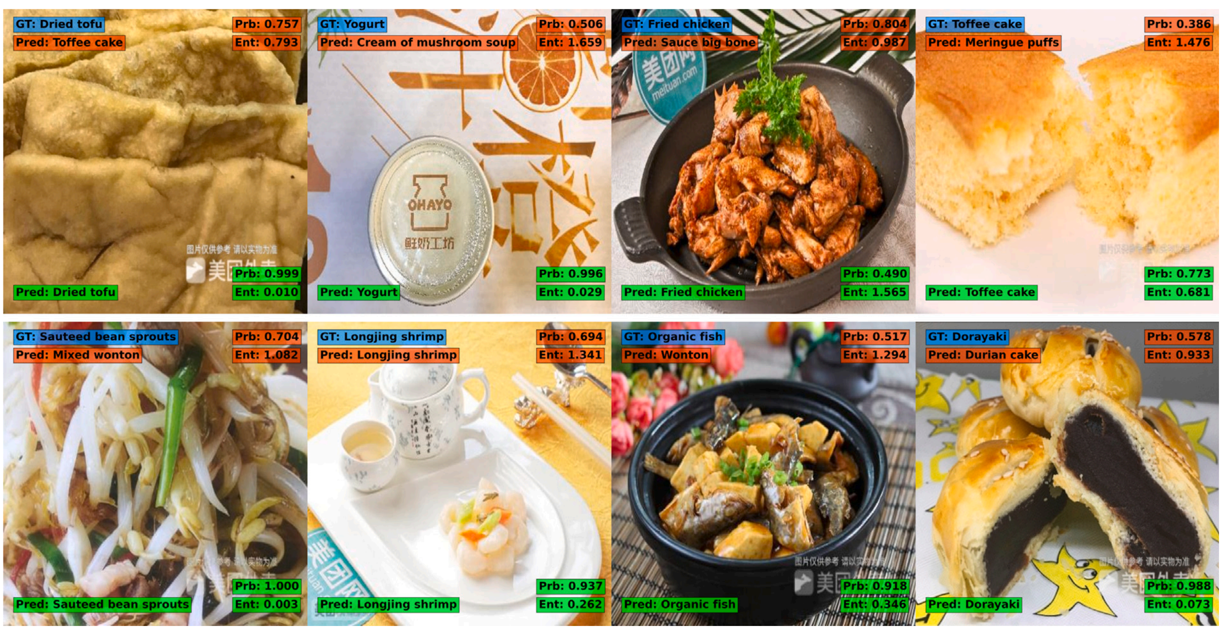


Fig. 7. Qualitative results: The ground truth is shown at the top ( ). Predictions in red ( ) correspond to the baseline model, whereas those in green correspond to the proposed approach ( ).

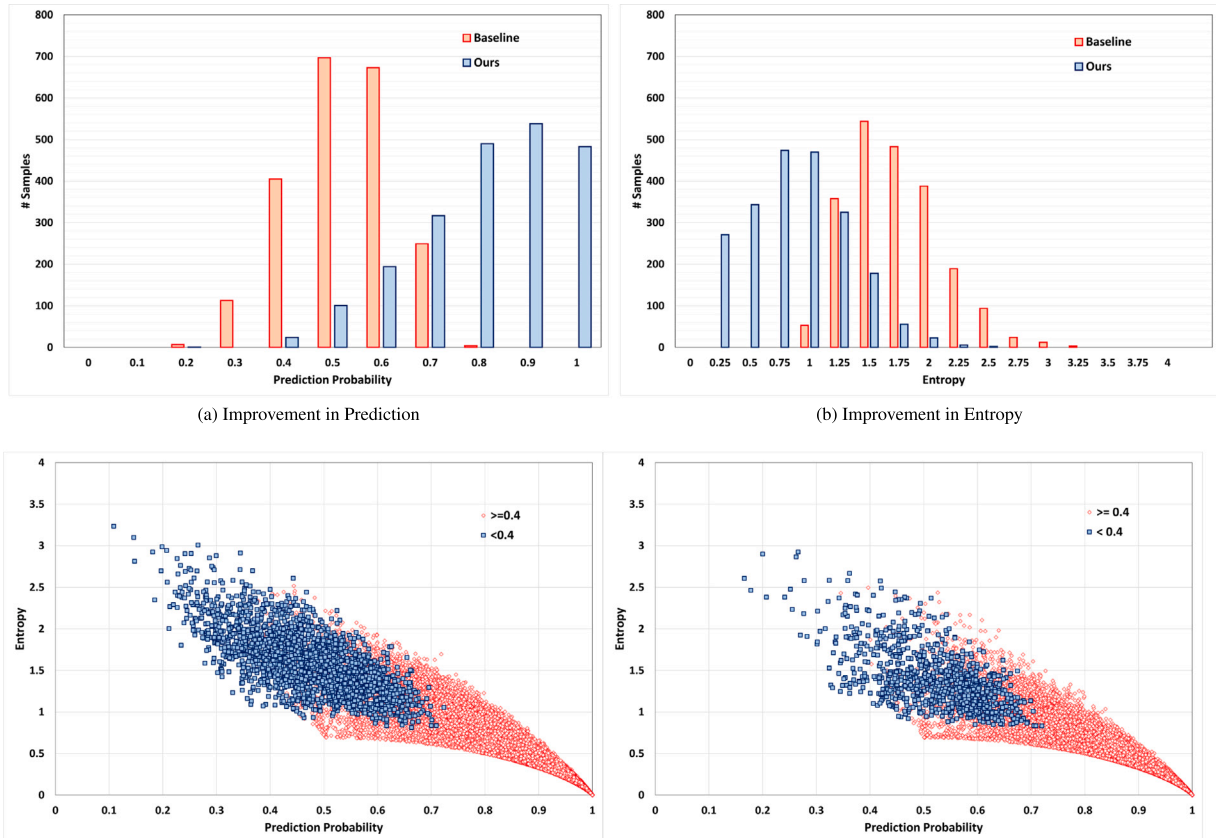
Table 2
Performance comparison of our proposed method with baseline and random selection techniques.

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.

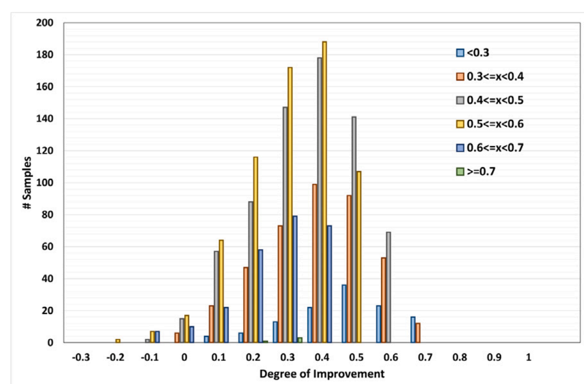


Fig. 9. Behaviour of each Coincidence Group.

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.