Recognizing Action and Activities from Egocentric Images

Alejandro Cartas Ayala

Recognizing Action and Activities from Egocentric Images

Alejandro Cartas Ayala

Doctor of Philosophy

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

2020

Director	Dr. Petia Radeva Facultat de Matemàtiques i Informàtica Universitat de Barcelona
Co-director	Dr. Mariella Dimiccoli Institut de Robòtica i Informàtica Industrial CSIC-UPC

Abstract

Egocentric action recognition consists in determining what a wearable camera user is doing from his perspective. Its defining characteristic is that the person himself is only partially visible in the images through his hands. As a result, the recognition of actions can rely solely on user interactions with objects, other people, and the scene. Egocentric action recognition has numerous assistive technology applications, in particular in the field of rehabilitation and preventive medicine.

The type of egocentric camera determines the *activities* or *actions* that can be predicted. There are roughly two kinds: lifelogging and video cameras. The former can continuously take pictures every 20-30 seconds during day-long periods. The sequences of pictures produced by them are called *visual lifelogs* or *photo-streams*. In comparison with video, they lack of motion that typically has been used to disambiguate actions. We present several egocentric action recognition approaches for both settings.

We first introduce an approach that classifies still-images from lifelogs by combining a convolutional network and a random forest. Since lifelogs show temporal coherence within consecutive images, we also present two architectures that are based on the long short-term memory (LSTM) network. In order to thoroughly measure their generalization performance, we introduce the largest photo-streams dataset for activity recognition. These tests not only consider hidden days and multiple users but also the effect of time boundaries from events. We finally present domain adaptation strategies for dealing with unknown domain images in a real-world scenario.

Our work on egocentric action recognition from videos is primarily focused on object-interactions. We present a deep network that in the first level models person-to-object interactions, and in the second level models sequences of actions as part of a single activity. The spatial relationship between hands and objects is modeled using a region-based network, whereas the actions and activities are modeled using a hierarchical LSTM. Our last approach explores the importance of audio produced by the egocentric manipulations of objects. It combines a sparse temporal sampling strategy with a late fusion of audio, RGB, and temporal streams. Experimental results on the EPIC-Kitchen dataset show that multimodal integration leads to better performance than unimodal approaches.

Acknowledgements

First of all, I would like to thank my parents Rodolfo and Graciela. Thank you for being with me all these years through thick and thin. I am also grateful for the support of my three brothers Rodolfo, Marco, and Daniel. Please know that your help was vital and that my gratitude goes beyond words. I must also express my gratitude towards my aunt Rosa for her loving and for taking care of my grandma alongside my cousins.

I would like to thank my supervisors Petia Radeva and Mariella Dimiccoli for the chance of working with them and, most importantly, for their guidance and advice. I not only learned a lot from you but had a great lifetime experience, thank you.

I want to thank my teammates Maya Aghaei, Marc Bolaños, Estefania Talavera, and Eduardo Aguilar for their assistance and collaboration. All those egocentric pictures were not, by all means, taken by themselves. By the way Maya, the UB did not feel that empty with you. I would also like to honorary thank Eduard Roure, you know you were more than an honorary member to all of us. Additionally, I like to thank Cosmin Bercea for his many coffee talks at Philology. I must thank all the participants involved in the collection of the lifelog sequences. I especially would like to acknowledge Juan Marín for his initial work on egocentric still-images.

I would like to thank Jordi Luque and Carlos Segura for the opportunity to work with them at Telefónica R+D. I learned a lot in my brief time with them and it made possible our multimodal approach. I also had the chance to meet great friends like Martin Kocour and Sebastián García.

Sometimes you unexpectedly meet great people in random places that care about you and move you forward, thanks for your support Gustavo Garduño and Verónica García.

I am really grateful to Gabriel de Oliveira, you made me not only your friend but also a member of your family. Thanks to you I met great people which I am also grateful. First, it came Oriol Secall, that proved to be another brother to me time after time. A few days later was the caring master, Stephanie Loomans. Finally, a few months after, was your family Sofia, María, Aurora, and your parents.

I must thank the friendship of my Catalan gang. Thank you Pedro Herruzo for your wise advice and optimism. Laura Portell, despite your animosity towards me, you have always been a sweet friend, thanks. I do not how many times I interrupted you in your deep work and never said anything, thank you, Guillem Pascual.

I am also grateful to all my labmates at Altell for sharing more than lunches at the German place, Ciprian Corneanu, Marc Oliu, Carles Riera, José Mena, Camilo Rey, Javier Selva, Alexa Monseguí, Luis Hernández, and Sorina Smeureanu. Cippy, you opened my eyes innumerable times, thanks. José, thank you for setting the example. I will always remember that the best days of the week are Mondays, Javi. Luisito, I do not know how many times you lent me your ears.

I would like to thank my friends from Mexico that have always giving me their support, Gabriel Busto, Waldo Salud, Maritere Meza, Antonio Fragoso, Sonia Segura, Francisco Mendez, Juan Pablo Alanis, Javier Medina, Manuel Ruiz, José Incera, Alton Macdonald, and Cipriano Hernández. Waldo, thanks for your weekly messages. Toño, I know I always can count on you, thank you, brother. Chonita, thank you for being there from the beginning. Pepe, your advice was fundamental to me to keep going, thank you. Alton, thank you for siding with me no matter what.

I would like to thank the Mexican National Science and Technology Council (CONACYT), which covered the stipends of the first four years of PhD through the grant 366596. I have taken a lot from my country and I hope to be able to return it in the future, thanks!

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Alejandro Cartas Ayala*)

A mi abuela

Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Egocentric Vision 3

1.3 Egocentric Action Recognition and Its Challenges 4

1.4 Goals 6

1.5 Contributions 7

1.6 Thesis outline 9

2 Related Work **11**

2.1 Action Recognition 12

2.2 Deep Learning 13

2.3 Third-Person Action Recognition Methods 15

2.4 Egocentric Action Recognition 17

2.5 First-Person Action Recognition Methods 18

2.6 Egocentric Action Recognition Datasets 23

2.7 Domain Adaptation 24

I Egocentric Activity Recognition From Visual Lifelogs **27**

3 Activity Recognition at Image-Level **29**

3.1 Approach 30

3.2 NTCIR-12 Dataset 31

3.2.1 Random partition 32

3.2.2 Temporal partition 32

3.3 Implementation 33

3.3.1 CNN Training 34

3.3.2 Random Forests training 35

3.4 Evaluation Metrics 37
3.5 Results 37
3.5.1 Random Data Partition Results 37
3.5.2 Temporal Data Partition Results 39
3.6 Conclusion 41

4 Activity Recognition at Sequence-Level 43
4.1 Temporal Approaches 46
4.2 Event-Based Approach 48
4.3 ADLEgoDataset 49
4.3.1 Data Collection 50
4.3.2 Annotation Process 51
4.3.3 Dataset Details 51
4.3.4 Dataset split 53
4.4 Experimental Settings 55
4.4.1 Temporal Methods Implementation 55
4.4.2 Event-based Methods Implementation 58
4.5 Evaluation Metrics 58
4.6 Results 59
4.6.1 Temporal Methods over NTCIR-12 59
4.6.2 Temporal Methods over ADLEgoDataset 60
4.6.3 Event-based Methods 63
4.7 Conclusions 64

5 Domain Adaptation Applications for Activity Recognition on Lifelogs 67
5.1 Domain Adaptation Using a Regularization Loss 69
5.2 Source and Target Datasets 70
5.3 Implementation 72
5.4 Results 75
5.5 Conclusions 78

II Egocentric Action Recognition From Videos 79

6 Region-Based Action Recognition From Egocentric Videos 81
6.1 Proposed approach 83
6.1.1 Frame-Level Modeling 84

6.1.2 Temporal modeling 87
6.1.3 Data augmentation 88
6.2 Datasets 91
6.3 Implementation details 92
6.4 Evaluation 93
6.5 Results 95
6.5.1 Ablation Studies 95
6.5.2 Comparison with the state of the art 97
6.6 Conclusions 99

7 Multimodal Action Recognition From Egocentric Videos 101
7.1 Proposed approach 102
7.2 EPIC Kitchens Dataset 105
7.3 Experimental Setup 107
7.3.1 Implementation 107
7.4 Evaluation metrics 108
7.5 Results 109
7.5.1 Discussions 114
7.6 Conclusions 115

8 Conclusions 117
8.1 Summary of Contributions 117
8.2 Future Work 119

A Publications 121

Bibliography 123

List of Figures

1.1 Evolution of wearable cameras 2

1.2 Example of an action seen on a third and first-person perspectives . . 4

1.3 Example of consecutive frames from an egocentric video and a visual lifelog. 5

2.1 Early action recognition pipeline for low-level features 13

2.2 Example of convolutional neural network 14

3.1 Activity recognition at image-level 30

3.2 Examples of all the *activity* categories randomly selected from the annotated NTCIR-12 dataset 31

3.3 Distribution of activity categories per user of the NTCIR-12 dataset . 32

3.4 NTCIR-12 temporal splits summary 33

3.5 Mean accuracy of the validation split for each random forest with respect to different number of trees 36

3.6 Normalized confusion matrices of the best combination of layers for each baseline convolutional neural network 39

3.7 Classification activity examples 40

4.1 Temporal coherence of concepts in lifelogging *activity* sequences . . . 44

4.2 Initial frames of five consecutive events segmented from a lifelog sequence 45

4.3 Thirteen *activity* day sequences of annotated photos from two people of the NTCIR-12 dataset 46

4.4 Sliding window training mechanism for lifelog sequences 47

4.5 Proposed architectures for *activity* recognition using temporal contextual information 48

4.6 Proposed architecture for activity recognition using events segmentation 49

4.7 Sampled pictures from the ADLEgoDataset 51
4.8 Distribution of *activity* categories in the ADLEgoDataset 52
4.9 ADLEgoDataset splits summary 54
4.10 Normalized confusion matrices of all models for CNN+LSTM and CNN Piggyback LSTM for the NTCIR-12 dataset 60
4.11 Normalized confusion matrices of the best models for the seen and unseen test sets for ADLEgoDataset 62

5.1 Examples of people performing the same *activity* from different domain datasets 68
5.2 Domain adaptation training pipeline 69
5.3 Categories mapping between the source and target domains with their data distributions 71
5.4 Sensitivity of the CORAL distance due to different learning rates . . . 74
5.5 Validation accuracy for the ResNet-50 and Xception on the transfer learning to the Castro's dataset and the NTCIR-12 dataset 75
5.6 Maximum mean discrepancy (MMD) between the categories from the source and target datasets 76
5.7 Mean test accuracy with respect to different percentages of added target labeled images to the training/validation source data using ResNet-50 78

6.2 Examples of primary and secondary regions 84
6.3 Examples of primary regions extraction taken from random frames from the GTEA and GAZE+ datasets 85
6.4 Overview of our proposed region-based architecture 87
6.5 Two different action sequences for all users from the GTEA dataset . 88
6.6 Visual data augmentation process 89
6.7 Examples of sequence data augmentation 90
6.8 Examples of people performing the same *action* (*put cheese on bread*) from the evaluated datasets 91
6.9 Diagram of the test architecture 94
6.10 GTEA 61 test curves for different values of β. 96
6.11 Examples of true predictions on all evaluated datasets using our proposed model 98
6.12 Examples of false predictions on all evaluated datasets using our proposed model 99

7.1 Audio and vision are complementary sources of information for recognizing egocentric object interactions 102
7.2 Pipeline of our multimodal proposed approach 103
7.3 Heatmap of the number of *action* instances in the EPIC Kitchens dataset training split 105
7.4 Statistics and histogram of the time duration of the video segments in the training partition of the EPIC Kitchens 106
7.5 Accuracy difference of the *noun* split for the unweighted test predictions 109
7.6 Accuracy difference of the *verb* split for the unweighted test predictions 110
7.7 Accuracy difference of the *action* split for the unweighted test predictions 111
7.8 Qualitative results for the unweighted multimodal *action* classification experiment 112

List of Tables

3.1 Comparison of the ensembles of CNN+Random forest for different combinations of layers on the random data partition 38

3.2 Comparison of the ensembles of CNN+Random forest for different combinations of layers on the temporal data partition 41

4.1 Comparison between existing egocentric lifelogs datasets and the ADLEgoDataset 50

4.2 Comparison of all temporal methods on the NTCIR-12 dataset 59

4.3 Comparison of all temporal methods on the ADLEgoDataset 61

4.4 Comparison between full-day sequences and event segmentation on the ADLEgoDataset 63

5.1 Activity classification performance results obtained by adding the Castro's and NTCIR-12 datasets for training without domain adaptation . 76

5.2 Action recognition accuracy for the domain shifts from the ADLEgoDataset dataset using ResNet-50.Best result per measure is shown in bold. 77

6.1 State of the art egocentric object-interaction recognition methods and their characteristics 83

6.2 Action recognition accuracy for different methods on the GTEA, Gaze, and Gaze+ datasets 94

6.3 Similarity between sequences of actions in videos for each dataset . . 95

6.4 Classification accuracy on each split of the GTEA 71 dataset. 96

6.5 Classification accuracy on each split of the GAZE+ dataset. 97

6.6 Classification performance comparison using visual data augmentation on the GAZE+ dataset for 44 categories 97

6.7 Classification performance comparison using sequence data augmentation on the GAZE+ dataset for 44 categories 98

7.1 Architecture of our proposed traditional dilated network for audio classification 104
7.2 Classification performance for the equally stratified *action* data split for the verb+noun fusion 109
7.3 Classification performance for the equally stratified *action* data split for the *action* classifier 111
7.4 Performance comparison with EPIC Kitchens challenge baseline results for the seen test partition 112
7.5 Performance comparison with EPIC Kitchens challenge baseline results for the unseen test partition 113
7.6 Classification performance results on the comparison *verb* data split . 114

Acronyms

ADL Activity of daily living.

AI Artificial intelligence.

AP Average pooling.

BLSTM Bidirectional long short-term memory.

CNN Convolutional neural network.

CORAL Correlation alignment function.

CV Computer vision.

DA Domain adaptation.

FC Fully-connected layer.

GAP Global average pooling.

GMM Gaussian mixture model.

HLSTM Hierarchical long short-term memory.

HMM Hidden Markov model.

LFE Late fusion ensemble.

LSTA Long short-term attention.

LSTM Long short-term memory.

MCG Multiscale Combinatorial Grouping.

MFCC Mel frequency cepstral coefficient.

MMD Maximum mean discrepancy.

NN Nearest neighbor.

ORN Object reasoning network.

PAF Parts affinity field.

RF Random forest.

RGB Reg, green, blue color channels.

RNN Recurrent neural networks.

SGD Stochastic gradient descent.

STFT Short-time Fourier transform.

STIP Space-time interest points.

SVM Support vector machine.

TSN Temporal segments network.

Chapter 1

Introduction

Contents

1.1 Background and Motivation 1
1.2 Egocentric Vision 3
1.3 Egocentric Action Recognition and Its Challenges 4
1.4 Goals 6
1.5 Contributions 7
1.6 Thesis outline 9

1.1 Background and Motivation

The goal of ubiquitous computing is that computers, as a technology, *blend* into the physical environment, making them imperceptible to people, as formulated by Weiser [1993]. For example, writing can be considered an invisible *literacy technology* to store and transmit knowledge, lacking the shortcomings of individual-memory. It not only encompasses printing symbols, but also grammar and syntactic rules for structuring ideas [Foer, 2011]. The modern world cannot be understood without it, printing symbols are everywhere, not only in books or magazines but also in small tea bags and street signs.

Ubiquitous computing was described as the third wave of computing [Weiser, 1991], being the first and second the mainframes and personal computers, respectively. An important role for this idea to become closer to reality has been played by cheaper and smaller hardware. This has allowed that a broader array of sensors have been

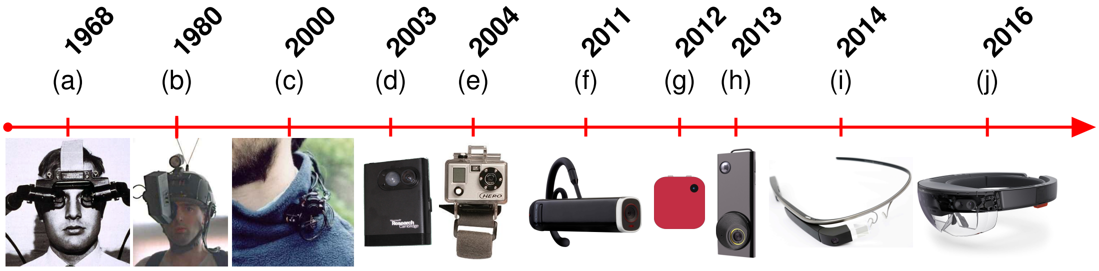


Figure 1.1: Evolution of wearable cameras. (a) Prototype of augmented reality camera (adapted from Sutherland [1968]), (b) first wearable camera by Mann [1997], (c) prototype wearable robot (adapted from Mayol-Cuevas et al. [2002]), (d) Microsoft SenseCam chest-mounted camera, (e) GoPro head-mounted video camera, (f) Looxcie ear camera, (g) Narrative Clip chest-mounted camera, (h) Autographer, (i) Glass video camera, and (j) Microsoft HoloLens.

integrated into different and more sophisticated types of portable devices that can be worn, i.e. *wearable devices*, ranging from motion to radar sensors.

The sensor diversity is due in part to more promising consumer applications from wearable devices. Initially, the applications were limited to only relay data like movement and heart rate tracking. Nowadays, these devices can measure a more complex array of things like stress [Muaremi et al., 2013], muscle fatigue [B. Ribas Manero et al., 2016], and mood [Budner et al., 2017]. Currently, their goal is also set to detect and mitigate problems like spotting seizures [Vandecasteele et al., 2017].

A special kind of wearable devices are cameras, that have also been benefited by advances on camera technologies and miniaturization trends [Zarepour et al., 2017]. Despite the fact that the wearable video cameras had already been idealized by Bush [1945], prototyped by Sutherland [1968] and invented by Mann [1997] in the early eighties, they become commercially available in the mid-two-thousands in the niche of first-person sports filming. Different prototypes and consumer wearable cameras throughout the years representing their evolution are shown in Fig. 1.1.

Wearable cameras share the following three characteristics with wearable devices. First, their size and weight allowed them to be carried to different locations. Second, they can non-invasively monitor and record data without human intervention. Third, depending on the device specification, they can continuously collect data during long periods of time. They have been successfully used for its recording capabilities on applications for health [Albrecht et al., 2014], journalism [Cao et al., 2014], and police [Palmer, 2016].

Unlike most wearable sensors, they capture external and directly interpretable information by a person; such as places, objects, and people around the user. In comparison with fixed cameras, wearable ones can daily gather large amounts of human-centric data in a naturalistic setting, hence offering rich contextual information about the activities of the user. As a result, activity recognition from wearable cameras has several important applications as assistive technology, in particular in the field of rehabilitation and preventive medicine. Examples include self-monitoring of ambulatory activities of elderly people [Abebe et al., 2016; Zhan et al., 2015], monitoring patients suffering dementia [Karaman et al., 2010, 2014], determining sedentary behavior of a user based on their spent time watching TV [Zhang and Rehg, 2018].

Perhaps, the most ambitious consumer application for action recognition from wearable cameras is lifelogging [Blum et al., 2006]. In rough terms, the goal of lifelogging is to create a multimedia diary of everything we do and was initially glimpsed by Bush [1945]. Its importance lies in the fact that it has sparked the creation of new hardware and software technology. And this in turn has propelled more research on the topic.

Although Weiser [1991] considered that ubiquitous computing did not need a revolution in artificial intelligence, time has proved that sophisticated applications need more than just more powerful wearable sensors. It is clear that these applications required egocentric vision to interpret the deluge of visual information. Specifically, new egocentric action recognition methods are essential for its further development.

1.2 Egocentric Vision

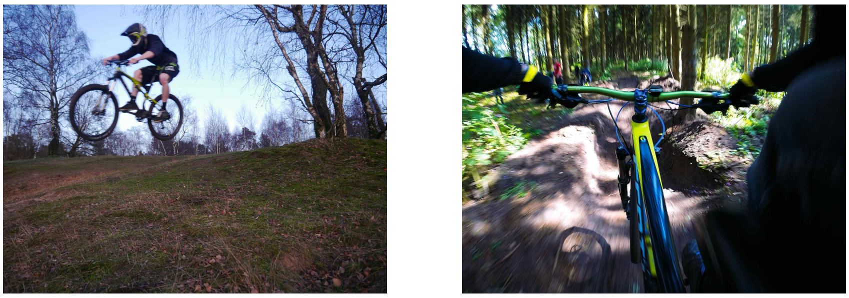
Egocentric vision is a sub-field of computer vision that comprises the extraction, analysis, and understanding of useful information from images or video captured by a wearable camera. The camera wearer cannot be only human, but it can also be an animal [Iwashita et al., 2014] or a robot [Xia et al., 2015]. Since the camera is usually worn in the head or the chest, it approximates the visual field or perspective of the camera wearer. Consequently, the term has been interchangeably used with *first-person* vision. Nevertheless, research works under this definition can not include cameras worn on places likes wrists [Ohnishi et al., 2016], thighs [Watanabe et al., 2011], or waist [Ozcan et al., 2013].

1.3 Egocentric Action Recognition and Its Challenges

A broad definition of egocentric action recognition consists in determining what a person carrying wearable devices is doing itself. The kind of *actions* that can be predicted using wearable devices depends on its type of sensor and where it is worn. Early research on the area starts around twenty years ago [Foerster et al., 1999; Seon-Woo Lee and Mase, 2002]. In both works, they use an accelerometer to predict ambulatory *activities* such as sitting, standing, or walking. However, our focus in this work is concerned only with wearable cameras.

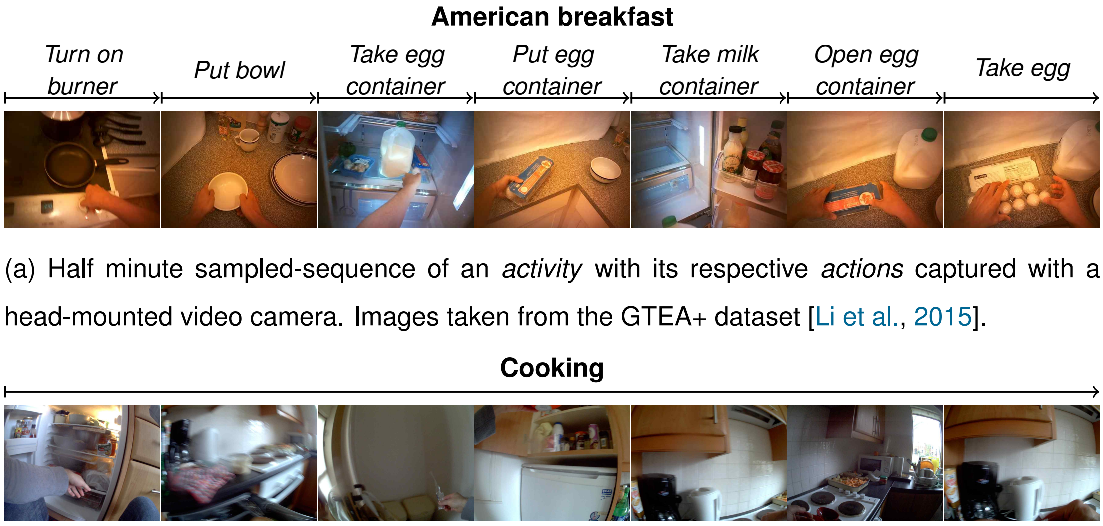
A specific definition for egocentric action recognition from wearable cameras consists of determining what its user is doing from his perspective. The characteristic problem is that the person itself is only partially visible in the images through his hands. As a consequence, egocentric action recognition relies solely on the user context: the objects he is manipulating, other persons around he is interacting with, and the environment itself. For instance, the difference between doing an action from the third and first-person perspectives can be appreciated in Fig. 1.2. Both perspectives show a person riding a mountain bike in a hill, but much of the context is reduced in the first-person perspective as only the arms holding the handlebar and the motion blur can be seen.

Like any other kind of wearable device, the type of camera sensor also determines the egocentric *activities* or *actions* that can be predicted. A defining characteristic is its temporal resolution, which refers to how many pictures can take in seconds or minutes. Wearable cameras can be roughly divided into high and low temporal resolution. The former kind can record videos for short periods of time. The latter kind can continuously take pictures at regular intervals of 20-30 seconds during day-long periods. The



(a) Third-person perspective (b) First-person perspective

Figure 1.2: Example of an action seen on a third and first-person perspectives. Original photos of Maurice Müller.



(a) Half minute sampled-sequence of an *activity* with its respective *actions* captured with a head-mounted video camera. Images taken from the GTEA+ dataset [Li et al., 2015].

(b) Four minutes full-sequence of an *activity* captured by a chest-mounted photo camera. Images taken from the NTCIR-2012 dataset [Gurrin et al., 2016].

Figure 1.3: Example of consecutive frames from an egocentric video and a visual lifelog.

sequences of pictures produced by them are called *visual lifelogs* or *photo-streams*. For example, Fig. 1.3 shows two frame sequences from wearable video (on top) and photo (on bottom) cameras. Video sequences have enough contextual information that the frames can explicitly depict an action. In the case of the sampled video sequence, it contains *actions* such as *turn on burner*, *put bowl*, etc. And this sequence of *actions* defines an *activity* , in this case, they describe *preparing American breakfast*. On the other hand, a lifelog sequence hardly contains representative frames of all *actions* performed by the person. Furthermore, the *actions* on its captured frames might be difficult to identify, as the objects or hands are occluded or partially visible. From this example, we can only guess that the person in the lifelog sequence is *cooking*. Consequently, while the focus on lifelogs has been classifying *activities* at large temporal scale such as *cycling* or *eating*, in egocentric videos the focus has been classifying short temporal scale *actions* like *put oil*, *cut potato*, *dry hand*, etc.

Moreover, this task inherits the same challenges as traditional third-person action recognition, namely:

- First, egocentric images suffer a large variation of appearance that leads to intra-class variation. Viewpoint changes are the result of the camera user not being static and also being in a wide variety of scenarios. Even more, the lighting conditions are not fixed since the camera can be worn in indoor and outdoor settings at different times of the day.

- Second, manual collection of training data is difficult. Although large amounts of videos/photographs can be gathered from a single user, their manual annotation is challenging. Furthermore, rich training data required to be collected from multiple users. Another issue is that some categories rarely occurred and produce a class imbalance. A final issue in this regard is that third-party privacy concerns exist in gathering training data.

- Third, action/activity vocabulary is not well defined and it depends on the context. The granularity of activities into actions is not properly defined because of their temporality and compositionality. Moreover, determining what a person is doing could be formulated as a multi-label classification problem.

- Fourth, lifelog pictures present additional challenges with respect to egocentric videos. As a result of its low temporal resolution, temporally adjacent images present abrupt appearance changes making motion features infeasible to be estimated. These features have been played a fundamental role in action recognition as they help to disambiguate actions/activities. Furthermore, crucial actions to identify the activity being performed can be missed and due to the non-intentional occlusions of the photos.

1.4 Goals

The goals of this dissertation can be divided by our two venues of research exploration:

- **Egocentric activity recognition from lifelogs**. We consider that the task of action recognition from lifelogs can be modeled using different time scales. It can be modeled as *fleeting instants* or single images, as a *successive moments* or sequences of images, and as *contextual events* or sub-grouped sequences of images. As this task has received little attention in the literature, we not only present different approaches for each time scale but also gathering enough data to conduct generalization experiments. Additionally, their application on real scenarios faces challenges like the performance drop of using pretrained models on new visual unseen data during training. We consider different domain adaptation to tackle this issue.

- **Egocentric action recognition from videos**. We consider two previously unexplored characteristics of egocentric object interactions. First, sequences of *actions* are logically performed in order to make an *activity* . Second, the interaction with objects produce sounds with distinguishable features for classification. We propose a model for each characteristic and study its impact on action recognition.

1.5 Contributions

Our contributions to the activity recognition from lifelogs can be summarized as follows:

- We propose an ensemble approach for still images extracted from lifelog sequences. This approach is a late fusion method that combines the output of the last fully connected layers of a convolutional network using a random forest. In order to extensively test its generalization capabilities, we annotated a subset of images from a multiuser dataset (NTCIR-12) and present two different test splits. Thus, making the performance evaluation more rigorous in comparison with previous studies. Our method was tested on several deep architectures, also making its evaluation more robust. This approach was published in [Cartas et al., 2017b].

- We introduce two different models and one training strategy that exploits the *temporal coherence* of lifelogs. Temporal coherence is the tendency of similar concepts to appear in neighboring frames of a sequence. Our first model is an extension in time of the ensemble approach outlined in the previous paragraph, and it uses as a temporal mechanism long short-term memory (LSTM) units. The second approach is a deep architecture that models the temporal relation between adjacent overlapping frames in subsequent input batches of an LSTM. Both approaches make use of our proposed *sliding window* training strategy. This strategy splits a lifelog into overlapping frame segments of fixed length to be used as input batches. We also present an approach that uses temporal boundaries from segmented events of a lifelog sequence.

  Our initial evaluation over the NTCIR-12 dataset demonstrates that it is possible to capture the temporal evolution of features over time from lifelogs. We perform a more extensive comparison to measure the generalization capabilities in a bigger dataset. To this end, we introduce the largest lifelog dataset of Activity of Daily Living (ADLEgoDataset) consisting of 102,227 images from 15 users, and 35 different categories. We propose a robust benchmark that considers different metrics in not only unseen full day sequences but also unseen users during training. These approaches are published in [Cartas et al., 2017a, 2018b, 2019].

- With the aim of measuring and improving models trained and tested on data acquired by different cameras and people, we present domain adaptation strategies to cope with this problem. Explicitly, we experimentally measure the discrepancy between source and target datasets and their diminishing of classification performance. We use the deep correlation alignment (CORAL) adaptation method to deal with this problem and show that a good performance is not always achieved on different target datasets. We propose to use different amounts of target data during the transfer learning training and show that competitive results can be achieved with a little amount of target labeled data. This approach is published in [Cartas et al., 2020]

Our contributions to action recognition from videos can be summarized as follows:

- We propose a novel hierarchical deep learning architecture that models object-interactions and their logical steps sequences to perform an *activity* . This architecture models the relationship between the hands and the interacting objects using a region-based model. The hands are the primary region obtained using skin segmentation, whereas the interacting object region is inferred from a set of region proposals. From the *action* classification of our region-based model, our architecture also captures the sequence of *actions* using a hierarchical LSTM (HLSTM). We perform a detailed ablation analysis of the various components of our model. We demonstrate that the proposed architecture achieves significant improvement with respect to competitive methods modeling the temporal structure and/or object interactions. This approach is presented in [Cartas et al., 2018a, 2017c].

- We introduce an audiovisual deep learning architecture that models the object-interactions in an egocentric kitchen context. Specifically, our the visual sources of our architecture optical flow and RGB frames and are modeled using a temporal segments network (TSN). The audio input is the raw sound wave from the video and a new convolutional network architecture is proposed for its classification. Both architectures are joined together using a late fusion approach that models the action, verb, and noun independently. We provide an extensive evaluation and comparison with published methods of the proposed multimodal architecture on the EPIC-Kitchens dataset. In addition to the action performance, we provide for the first time detailed results on the object and verb components. This approach is introduced in [Cartas et al., 2019; Cartas et al., 2019].

1.6 Thesis outline

We first review related work of action recognition to ours in chapter 2. This review starts with a broad context on third-person methods and ends with a detailed focus on first-person methods. Moreover, the terms *action* and *activity* used throughout this thesis are precisely defined.

The first part of the thesis is devoted to our work on activity recognition from lifelogs and is divided into three chapters. Chapter 3 describes our approach for single images in lifelogs. Appealing to the temporal coherence of activities, in chapter 4 we present our temporal methods for lifelog sequences. Considering a real-case scenario, we detail a domain adaptation application in chapter 5.

The second part of the thesis presents methods for action recognition from egocentric videos. In chapter 6 we model egocentric object-interaction and the sequence of action that describes whole activities. Next in chapter 7, we present a multimodal model that takes into account the different sounds produced by egocentric object interactions.

Finally, in chapter 8 we draw main conclusions and outline possible future lines of work, and a list of publications from this thesis is presented in appendix A.

Chapter 2

Related Work

Contents

2.1 Action Recognition 12

2.2 Deep Learning 13

2.3 Third-Person Action Recognition Methods 15

2.4 Egocentric Action Recognition 17

2.5 First-Person Action Recognition Methods 18

2.6 Egocentric Action Recognition Datasets 23

2.7 Domain Adaptation 24

In this chapter we present previous work on the problems of egocentric action recognition. We first provide a brief overview of action recognition and define the terms *action* and *activity* (Section 2.1). Then, we introduce two deep learning models that form the basis of modern action recognition methods (Section 2.2). Next, we present third-person approaches that have influenced first-person (Section 2.3). Afterwards, we present general egocentric action recognition (Section 2.4) as an introduction to first-person vision approaches (Section 2.5). Then, we describe the available video and lifelogging datasets (Section 2.6). Finally, we give an overview of domain adaptation methods (Section 2.7).

Since a growing number of papers on action recognition are published each year, we mostly covered recent works that we considered relevant. We recommend to the avid reader consulting action recognition surveys on historical methods [Poppe, 2010; Weinland et al., 2011], deep learning methods [Herath et al., 2017], and wearable sensors [Cornacchia et al., 2017; Lara and Labrador, 2013]. There are also surveys that cover egocentric vision and have special sections on first-person action recognition methods [Bambach, 2015; Bolaños et al., 2017].

2.1 Action Recognition

The objective of action recognition is to determine what is *happening* in a scene [Bobick, 1996]. Although this is a vague definition for a computer vision task, it encompasses the multiple works that have been published since the word *action* is an ambiguous term [Herath et al., 2017]. The lack of a vocabulary to describe the granularity and composition of actions is one of its difficulties as stated by Schmid [2016]. For example, sometimes an *action* is interchangeably used with the word *activity* [Pirsiavash and Ramanan, 2012], other times a set of *actions* are part of an *activity* [Fanti, 2008] or vice versa [Bobick, 1997], and some other new words are defined such as *actoms* [Gaidon et al., 2011].

The concept of motion has played a fundamental role in solving the action recognition task. Historically, research on action recognition starts as part of a broader area on machine perception of motion during the late seventies [Bobick, 1997]. While the information provided by motion is important and helps to disambiguate actions such as opening/closing a door, it is derived from time. Therefore, our taxonomy of what is happening on a scene is in terms of its length of duration. Similarly to Fanti [2008], we consider that an *action* can be as short as one millisecond or as long as a few minutes. Moreover, we consider that an *activity* is composed of two or more actions and its duration starts at the seconds scale.

Action recognition is a classification problem and, as such, it has a more technical definition. As defined by Poppe [2010], it consists in assigning *action* labels to image sequences, but this definition could be extended to any kind of temporal data. Initial efforts are part of the called 'model-based' vision approaches started by Binford [1971]. They try to reconstruct the body of a person using volumetric shapes as human parts and describe movements such as walking [Hogg, 1983] or jumping [Horowitz and Pentland, 1991]. Consequently, they are also categorized as top-down approaches.

The bottom-up approaches classify an *action* based on low-level features and become the predominant methods in the field [Herath et al., 2017; Poppe, 2010]. The general classification pipeline for low-level features is presented in the seminal work of Yamato et al. [1992], and it is shown in Fig. 2.1. The pipeline labels a sequence of images in three steps [Oneață, 2015, p. 10]. First, it extracts a feature vector for each

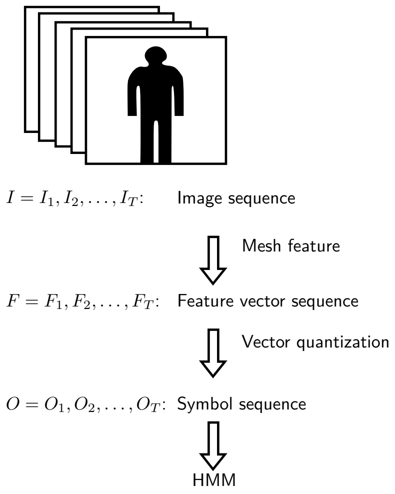

$I = I_1, I_2, \ldots, I_T$: Image sequence
Mesh feature
$F = F_1, F_2, \ldots, F_T$: Feature vector sequence
Vector quantization
$O = O_1, O_2, \ldots, O_T$: Symbol sequence
HMM

Figure 2.1: Early action recognition pipeline for low-level features. First, a feature vector is extracted for each frame in an image sequence. Then, the features are clustered (*encoded*) into a set of symbols using vector quantization. Finally, the classification is performed by a hidden Markov model (HMM). Adapted with permission from Yamato et al. [1992].

image in the sequence. Second, it performs feature encoding on the resulting vectors; sometimes, a learned *dictionary* or *codebook* is used for vector quantization. Third, it classifies each image based on the encoded vectors. Several approaches have used this pipeline by varying the type of features and classifiers. For instance, Laptev et al. [2007] classifies six different types of *actions* using a combination of hand-crafted features named space-time interest points (STIPs), support-vector machines (SVMs) and nearest neighbor (NN) as classifiers.

2.2 Deep Learning

Modern action recognition methods are bottom-up approaches that are based on the connectionist paradigm of Artificial Intelligence (AI). New methods under this resurgent paradigm have been categorized under the umbrella of deep learning methods. In this section, we describe basic deep learning architectures that form the basis for more specialized methods for current action recognition methods.

Convolutional neural networks (CNNs) Inspired by the physiological model of the mammalian visual system by Hubel and Wiesel [1962], several attempts for automati-

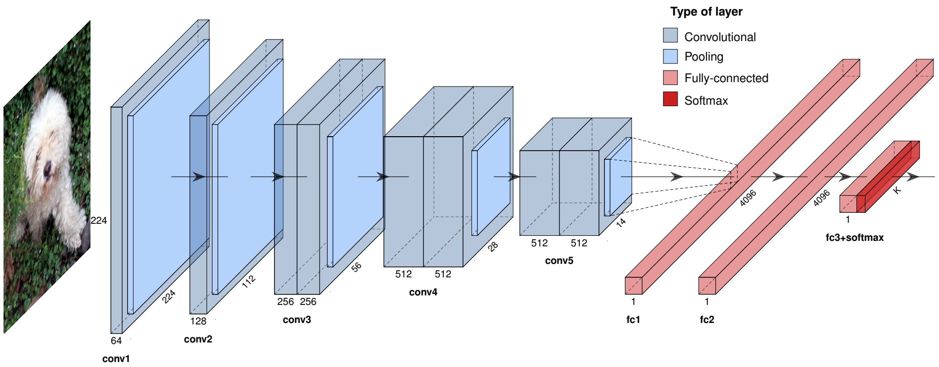


Figure 2.2: Example of convolutional neural network. This architecture corresponds to the VGG-11 network introduced by Simonyan and Zisserman [2014b].

cally learning an image representation for classification purposes have been introduced [Fukushima, 1988; LeCun et al., 1990; Serre et al., 2005; Fidler et al., 2006]. These approaches build a hierarchical representation on top of low-level features learned from the edges of objects. Although initially they were not widely adopted, the most successful of them so far are the convolutional neural networks (CNNs), or simply convolutional networks, introduced by LeCun et al. [1990, 1998]. In comparison with the rest of the cited models, they were designed to even learn the visual filters in an end-to-end fashion.

Convolutional networks were designed to have as input grid-structure data. They are multi-layer networks that can be divided into two main consecutive parts. The first part works as a feature extractor and the second one as a classifier. The former is composed of a stack of convolutional and pooling layers. Convolutional layers create feature maps using learned filters from the data, whereas pooling layers downsample them. The representation of bi-dimensional information is hierarchically captured on each convolutional layer. For instance, when the input is image data, the first convolutional layers represent filters similar to Gabor filters, and further layers form lines and arcs until subsequent layers capture objects. The classifier part comprises one or more Fully-Connected (FC) neuron layers followed by a softmax layer. Fig. 2.2 shows a CNN with the layers corresponding to the feature extractor and classifier parts in blue and red colors.

Many computer vision tasks have made great strides since Krizhevsky et al. [2012] won the ImageNet [Russakovsky et al., 2015] classification challenge using a convolutional network, namely AlexNet. In comparison with the LeNet architecture [LeCun et al., 1998], this model implements a newer activation function named rectified lin-ear unit (ReLU) between the convolution and pooling layers. This activation function allows it to have more layers with more neurons, making it a *deeper* network. On the other hand, this architecture is possible thanks to more powerful computing hardware. The success of AlexNet span research to improve the efficiency and accuracy of CNNs [Simonyan and Zisserman, 2014b; Szegedy et al., 2015; He et al., 2016; Chollet, 2017b; Hu et al., 2018].

Long short-term memory Recurrent neural networks (RNNs) are another type of neural network that is resurgent in the field. A particular kind of them has been of special interest to the research community since they can model sequential information. Specifically, the first model to effectively uses the backpropagation algorithm to learn a sequence was the long short-term memory (LSTM) units proposed by Hochreiter and Schmidhuber [1997]. They go through a sequence state by state, and possess a gate mechanism that determines at each state what information to keep and discard and by how much. Another extension of this network named bidirectional LSTM (BLSTM) [Graves and Schmidhuber, 2005] evaluates a sequence in forward and backward order and merges the result. In contrast with its unidirectional version, it captures patterns that could have been missed and potentially obtains more robust representations. New recurrent architectures have been proposed [Cho et al., 2014] and new mechanisms have been incorporated such as attention [Xu et al., 2015].

2.3 Third-Person Action Recognition Methods

Still images Convolutional networks can straightforwardly be used to classify actions from still images. Although they cannot make abstractions of concepts like motion from single frames, they can capture spatial information for a general *activity* category. For instance, the picture of a person dribbling a basketball ball next to another can be classified as *playing basketball*, but not as specific as *dribbling a ball*.

With the purpose of exploiting contextual relationships between persons and objects in a picture, Gkioxari et al. [2015a,b] studied deep region-based models using an object detector, i.e. R-CNN [Girshick et al., 2014; Girshick, 2015]. They proposed the R*CNN model that takes as input one primary region and a set of candidate secondary regions. The primary region is manually selected and contains the person whose action has to be predicted. The candidate secondary regions are supposed to capture contextual information such as objects the person is interacting with, the pose of the person, or what other people in the image are doing. In their architecture, all the regions and the whole image are processed together, but each region is individually processed in FC layers. The classification score is the addition of the primary region score plus the maximum score from all candidate regions.

Video Although convolutional networks were used on video data [Ning et al., 2005] before they became popular, they did not deal with motion. Initial works sought to add temporal mechanisms to CNNs and be able to predict action. They could roughly be divided into three basic temporal architectures that led to more complex designs with multiple inputs of different kinds. The first of them are three-dimensional convolutional layers (3D-ConvNets) to handle volumetric spatio-temporal data [Kim et al., 2007; Baccouche et al., 2011; Ji et al., 2013; Tran et al., 2015]. Another architecture consisted in putting LSTM units on top of convolutional layers [Baccouche et al., 2011; Donahue et al., 2015, 2017; Ng et al., 2015]. The last architecture performs early, late, or hybrid fusion mechanisms on different layers of a CNN [Karpathy et al., 2014].

The first deep architecture to explicitly use motion derived data was proposed by Simonyan and Zisserman [2014a]. Besides having as input a single color image (spatial stream), their architecture adds another convolutional branch for handling several optical flow frames (motion or temporal stream) using late fusion, hence the name two-stream model. Wang et al. [2016, 2019] introduces the Temporal Segment Networks (TSN) that splits a video into K sequential segments of an equal number of frames. Simultaneously from each segment, spatial and temporal frames are sparsely sampled and processed by a CNN stream. Later approaches [Carreira and Zisserman, 2017; Tran et al., 2018] aim to treat an input video as a volume but reducing the number of parameters of 3D convolutional layers, thus increasing the speed and number of input frames. Girdhar et al. [2019] adapt the attention mechanism [Vaswani et al., 2017] for language translation for an action localization task.

Audiovisual modalities Another source of information for discriminating *actions* in a scene corresponds to audio input. Audio classification has been traditionally associated with speech recognition, an AI task that has the objective of converting acoustic speech into its textual component words [Nilsson, 2009]. Standard acoustic models used Gaussian Mixture Models (GMMs) and HMMs before deep neural networks were considered by Hinton et al. [2012], although they were previously modeled with neural networks designed for audio by Waibel et al. [1989] and then convolutional networks by LeCun and Bengio [1998]. Convolutional networks are suitable for audio information since its spectral information can be represented in matrix form and required fewer parameters than neural networks. Deep convolutional networks for speech recognition were first used by Abdel-Hamid et al. [2012] and GMMs were then substituted with LSTMs by Sak et al. [2014]. Initial cross-modalities works from audiovisual inputs focus on unsupervised tasks. For example, Harwath et al. [2016] proposed a two-stream neural network that learns semantically meaningful words and phrases at the spectral feature level from a natural input image and its spoken captions without relying on speech recognition or text transcriptions.

The first deep multimodal work on action recognition was presented by Wu et al. [2016]. Their architecture considered three streams for spatial, motion, and audio data. Each stream was separately processed by CNN followed by an LSTM. All the streams were joined by a late fusion mechanism. Long et al. [2018b] proposed a multimodal attention mechanism on top of a BLSTM that had as inputs audiovisual extracted features. Another multimodal architecture that considers irrelevant the order of the multimodal features was introduced by Long et al. [2018a]. This architecture considered each modality branch as an attention cluster. Inside each cluster, the features of each frame were processed by a single attention unit. Several other cross-modalities works have been reported in the context action recognition challenges [Fabian Caba Heilbron and Niebles, 2015; Ghanem et al., 2017, 2018] and have been in the first places, but they lack of details. Among the reported submissions, Wenhao Wu [2018]; Zhang et al. [2018] extended the TSN architecture for audio streams, and Zhao et al. [2017]; Yao and Li [2018] used another stream for pose.

2.4 Egocentric Action Recognition

The objective of egocentric action recognition is to determine what a person is doing through the use of wearable devices. As stated by Lara and Labrador [2013], the integrated kind of sensors to these devices can be grouped into four categories: acceleration, environmental, location, and physiological. An example of each one of them used for action recognition are accelerometers and gyroscopes [Ermes et al., 2008], microphones and thermometers [Parkka et al., 2006], global positioning systems (GPS) devices [Hao Tian et al., 2009; Reddy et al., 2010], and heart and breathing rate monitors [Cheng et al., 2013], correspondingly. This distinction is important as the set of predictable *actions* depends on the sensor type, but also where the device is worn. For instance, although accelerometers are traditional associated with ambulatory activities [Arif et al., 2014; Foerster et al., 1999; Seon-Woo Lee and Mase, 2002], they also have been used as wrist bands to predict more precise *actions* like *reading* [Bao and Intille, 2004].

Contrary to other types of wearable sensors, cameras capture external and interpretable information of the user surroundings such as places, objects, and people. According to Lara and Labrador [2013], wearable *actions* can be grouped into seven categories: ambulation [Arif et al., 2014], transportation [Siirtola and Röning, 2012], phone usage [Berchtold et al., 2010], daily activities [Kao et al., 2009], exercise/fitness [Munguia Tapia et al., 2007], military [Minnen et al., 2007], and upper body [Cheng et al., 2010]. Wearable cameras provide more visual information to disambiguate these kinds of classes and, depending if they are video or lifelog cameras, they can be used on several categories from the seven groups. For instance, arm motion might not provide enough information to distinguish between *brushing teeth* and *eating/drinking* [Maurer et al., 2006], but cameras can capture the person holding a toothbrush with his hand or looking at himself brushing his teeth in a mirror.

2.5 First-Person Action Recognition Methods

The goal of first-person action recognition is to determine what the camera wearer is doing himself. Since only his/her hands are sometimes visible throughout the images, the recognition methods rely only on the scene context like the manipulating objects, other people, and the environment. The methods also depend on whether the images come from a video or lifelog camera. The latter has low temporal resolution and motion features like optical flow cannot be obtained from it. Thus, most of third-person methods cannot be adapted to lifelog cameras. Additionally, the methods depend where the camera is worn, it has been reported to be used mounted in the forehead [Price and Damen, 2019], as glasses [Bambach et al., 2015], in a necklace [Yu et al., 2019b], or even on the wrists [Ohnishi et al., 2016]. In this regard, most of the work has been focused on head-mounted cameras; as its perspective allows it to capture more informative regions or to have less severe occlusions than chest-mounted cameras.

Still images Little research has been done on still egocentric images. Castro et al. [2015] collected a dataset of 40,103 images from a single person using a chest-mounted camera taken during a six-month period. They classified nineteen daily living *activ-*

ities using a late fusion approach. Specifically, given an input image, they compute the softmax probabilities vector from a CNN (i.e. AlexNet) and combined it with its timestamp and color histogram using a random forest (RF). Their approach exploits the fact that a person usually performs *activities* like *working* in the same place and time of the day, so it can presumably work on people sharing the same lifestyle.

Lifelogs Most of the work on lifelogs has been focused on their segmentation into events [Talavera et al., 2015; Dimiccoli et al., 2017; Garcia del Molino et al., 2018] and their summarization [Bolaños et al., 2018; Fan et al., 2018]. Yu et al. [2019a] collected acceleration, location, time, and image data from two users having the same lifestyle. They proposed a multimodal method that combines all these data using a late fusion approach. This method has three independent branches for time, movement & location, and semantic information. The time branch provides a fixed probability of the activities that the person might be doing with respect to the time of the day. The movement & location predicts an activity using an SVM from a feature vector formed by movement features and GPS coordinates. The semantic branch predicts an activity based on an object classification made by a CNN. These branches are combined using the Dezert-Smarandache probability framework [Dezert, 2002]. Another multimodal approach using acceleration features and images was presented by Yu et al. [2019b]. They defined a taxonomy of *activities* that depends on *motion state* categories. For instance, the *activity reading* belongs to the motion states *standing* or *seated*, but not to *walking*. Their proposed model is a two-level deep hierarchical architecture. The first level classifies the acceleration features into motion states using an LSTM. Depending on the predicted motion state, then a CNN is used to predict only the *activities* corresponding to that motion state using as input an image.

Video Much of the research on first-person action recognition from videos has been focused on exploiting egocentric features: hands, interaction with active/passive objects, head motion (also called ego-motion), the gaze, their temporal structure, or a combination of them. The following paragraphs detail each of these categories.

Hands. Mayol and Murray [2005] proposed a probabilistic method to detect *activities* performed on a table using office objects like keyboard, calculator, or a ball. Their method calculates the probability of each *activity* event based on the probabilities of the area of the hands using skin color segmentation, the class of objects using template color histograms, and the spatial distribution of events around the field of view (FOV).

To evaluate the importance of the location of hands in egocentric activities, Bambach et al. [2015] first built a hand detector based on R-CNN [Girshick et al., 2014]. Their hand proposal algorithm uses a probabilistic model of bounding boxes where the hands might appear using a pixel skin model. They evaluated the location of hands in four board game *activities* by using hand masks on full frames. García Hernando et al. [2018] collected RGB-D video sequences dataset of 45 different *actions* involving the hands. They found that methods relying on only hand pose like LSTMs and Gram matrix [Zhang et al., 2016] outperform methods relying only on color and optical flow like Two-Stream network.

Interaction with active/passive objects. Surie et al. [2007] created a virtual world to experiment on three domains of the egocentric perspective first proposed by [Pederson, 2003]. At their core of egocentric domains is the manipulated object, that might be seen in the *observable space* or it is present but not visible in the *manipulable space* of the person. Pirsiavash and Ramanan [2012] model the semantic context of the objects that appeared on the scene into active and passive. For instance, a fridge is considered active when it is opened, and passive when it is closed. Their model incorporates this information as different parts-based models object detectors [Felzenszwalb et al., 2010]. Additionally, they adapted spatial pyramid matching method [Lazebnik et al., 2006] to be also temporal with respect to the video length. McCandless and Grauman [2013] continue exploring the spatio-temporal pyramids by not sampling the frames uniformly, but by randomly selecting the input frames and choosing the ones with active objects. González Díaz et al. [2013] argued that activities involve sequences of active objects and places where they are performed. For example, baking a cake is an activity done in a kitchen using different utensils, while doing house chores requires the user to move around the house. They extended the spatio-temporal pyramids by sampling temporal neighborhoods of selected frames in a sliding window fashion. Matsuo et al. [2014] extended the spatio-temporal pyramids model by calculating a visual attention on active objects based on the work of Yamada et al. [2012]. Sudhakaran and Lanz [2018] devised an unsupervised method for predicting interacting objects and *actions* from raw video. Their method is based on Class Activation Mapping (CAM) [Zhou et al., 2016] and convolutional LSTMs (ConvLSTMs) [Sainath et al., 2015]. An activation map is first extracted using a preliminary classification using a CNN. This activation map is element-wise multiplied by the las convolutional layer and feed to a ConvLSTM. Sudhakaran et al. [2019b] continue their work and proposed a novel attention mechanism named Long Short-Term Attention (LSTAs) based on LSTMs. They added another stream for dealing with optical flow and not only predicted the interacting object but also the verbal action on the activation map.

Head motion. Kitani et al. [2011] proposed an unsupervised approach for action detection on sports videos based on the head motion. Their approach encodes the head motion in a vector that is grouped by a stacked Dirichlet process mixture model. They encode egocentric head motion using directional and frequency components arguing that *actions* like *turning* the head have strong instantaneous component, whereas *running* has strong periodic components. Poleg et al. [2014] modeled the head motion splitting the frames into a grid. For each cell in the grid, they calculated a cumulative displacement curve that smooths the optical flow over time.

Gaze. Bulling et al. [2011] described three different eye movements that describe the gaze of a person: fixation points, blinks, and *saccades* (rapid eye movements that scan the scene). These movements were later investigated on egocentric action recognition methods. Fathi et al. [2012] proposed a generative probabilistic model that could predict *actions* using the gaze. Specifically, they used three features obtained regions near the fixation points: the scores of detected objects, colors, and texture histograms of current and future neighborhood areas. Hipiny and Mayol-Cuevas [2012] modeled the gaze using gradient regions around a grid near the fixation point. An *action* for them is a sequence of gradient region templates and it can be predicted using a visual bag-of-words method. Instead of using fixation points to predict actions, Ogaki et al. [2012] use the saccades. Along the optical flow produced by the head motion, they encode it using frequency analysis into a wordbook. Shiga et al. [2014] created a late fusion method that combines gaze and spatial features. In the case of gaze features, they combine fixation points and saccades into a codebook sequence that is classified using a multi-class SVM. For the spatial features, they extract scale-invariant feature transform (SIFT) [Lowe, 2004] around the gaze area and also classified them using a multi-class SVM. Using the I3D architecture, Li et al. [2018] created a deep model that predicts the gaze and action using as input RGB and optical flow frames. They first learn an attention map using the gaze during training, and it is subsequently used to pool visual features for action classification.

Temporal structure. Fathi et al. [2011b,a] investigated the relationships between consecutive *actions* using a probabilistic model, although not their logical temporal order. An *action* is modeled after *object interactions*, i.e. features from hands and objects previously extracted; and an *activity* is modeled after sequences of *actions* . Further work by Fathi and Rehg [2013] modeled the state changes of objects by first discovering changing regions across the frame sequences. An *action* is predicted by concatenating the features of the detected changed regions from initial and final frames into an SVM. Ryoo and Matthies [2013, 2016] investigated two methods based on global and local descriptors, i.e. histograms of optical flow and cuboids, respectively. Their first method relies on SVM having a kernel that looks for similarities in both kinds of descriptors. Their second method proposed another kernel that works at a hierarchical structure of atomic *activities* . With the goal of modeling atomic *actions* , Zaki et al. [2017] devised a method that extracts features from video data in a sliding window fashion, and it later encodes this spatial features using temporal pyramid based on the work of [Lazebnik et al., 2006]. Bokhari and Kitani [2016] devised a deep reinforcement learning method to forecast *actions* of a person inside an office. Their method models the camera wearer as an agent and rewards it for future *actions* with respect to the state of interacting objects, i.e. forecasting the clean a cup if it is dirty.

Combination. Yu and Ballard [2002] presented pioneering work that incorporated the eye velocity movement, segmented objects based on fixation points, and hand movement. Their model predicted three *activities* using the parallel HMM. Behera et al. [2012] presented a four-level hierarchical architecture in which an *action* is modeled as a series of atomic *actions* . These atomic *actions* are modeled as bag-of-words of accelerometer and visual features. Following the work on dense trajectories [Wang and Schmid, 2013; Wang et al., 2013], Li et al. [2015] proposed to encode all described egocentric cues into a Fisher vector obtaining competitive results to CNNs. Ma et al. [2016] proposed an architecture that models an *action* as the composite of a *verb* plus an *object*. Their architecture is similar to two-stream, but the spatial stream first performs hand segmentation and location of *objects of interest* using skin masks and two-dimensional Gaussian distributions, respectively. Additionally, both streams are joint using late fusion to predict the *action* , *verb*, and the *object*.

Other. Singh et al. [2016] extended the two-stream architecture adding another stream called EgoConvNet. The input of EgoConvNet has four channels containing a skin binary mask, the horizontal and vertical head motion, and a saliency map.

Multimodal Song et al. [2016b,a] presented a multimodal dataset from sensor data and proposed two methods, one using crafted features and the other using deep learning. The first method not only uses dense trajectories on video, but they also adapted it for one-dimensional sensors signals like acceleration. They joined video and sensors dense trajectories using Fisher vector encoding in a sliding window fashion. The second approach used two-stream architecture for video data and LSTMs for the sensor signals. Their approach performed late fusion over the softmax scores. They found that the crafted-features performed better than deep methods, arguing that the dataset was not big enough. Using heart rate and acceleration data along with video data, Nakamura et al. [2017] jointly predicts physical activities and energy expenditure. Their deep model extracts features from video and raw acceleration data that are concatenated in an early fusion. Later on, they are processed by an LSTM with classification and a regression loss. Arabaci et al. [2018] presented an audiovisual approach on egocentric video using several crafted features. The audio is processed using Mel-frequency cepstrum coefficients (MFCCs) and a Gaussian mixture model (GMM). The video data is processed using several optical flow features designed for egocentric motion [Abebe et al., 2016] on cameras worn on different body parts. Kazakos et al. [2019] introduce the Temporal Binding Network (TBN) that adds temporal frame sampling boundaries to the TSN architecture. As in the TSN architecture, the order of the input frames is irrelevant. One of their input modalities is a spectral audio image, but it just covers a small fraction of the video sequence.

2.6 Egocentric Action Recognition Datasets

Video Datasets The vast majority of egocentric datasets for action recognition were captured using video cameras [Bolaños et al., 2017; Damen et al., 2018]. Given the high energy consumption of video cameras, this kind of datasets only covers *actions* spanning up to a few hours and not whole days, as lifelog datasets. Another obstacle for these devices is its obtrusiveness, as they are being typically mounted on the head. This causes that the *actions* in these datasets are specific to one environment, usually indoor. Examples of datasets indoor environments include tasks like cooking [Damen et al., 2018; de la Torre et al., 2008; Fathi et al., 2011a, 2012; Li et al., 2018], interactions with toys [Ryoo and Matthies, 2016], working [Abebe et al., 2019; Damen et al., 2014], or daily indoor activities [Pirsiavash and Ramanan, 2012; Sigurdsson et al., 2018]. Just a few datasets have outdoor *actions* such as basketball [Abebe et al., 2016] or ambulatory activities [Poleg et al., 2016].

Lifelogging Datasets A reduced number of egocentric lifelogs datasets have been presented during the last five years. In comparison with the video datasets detailed above, these kinds of datasets cover full-day *activities* performed in a larger variety of settings. Both characteristics made them more difficult to acquire. First, the process is more expensive because the recording time is longer. Second, indoor recording settings do not have as many privacy issues as recording several locations and people. Castro et al. [2015] introduced one of the first datasets that describe the life of one person using 19 *activities* , but it did allow to test generalization capabilities on several people. Most of the lifelogging datasets were introduced in the context of image retrieval challenges [Gurrin et al., 2016, 2017; Dang-Nguyen et al., 2018; Gurrin et al., 2019]. They cover several weeks, but their annotated categories and images are low and only describe ambulation and transportation *activities* . The life of three people was captured in the NTCIR-12 challenge dataset [Gurrin et al., 2016]. It was independently annotated with 21 *activities* by [Cartas et al., 2017a,b]. Gurrin et al. [2017] presented another dataset for image retrieval containing the pictures of two people labeled with four distinct categories. It was later used in another challenge contest by Dang-Nguyen et al. [2018], but only took into account one person and two categories. Gurrin et al. [2019] introduced one last dataset consisting of pictures from two subjects performing two *activities* .

2.7 Domain Adaptation

Domain adaptation (DA), also called *dataset shift problem* [Storkey, 2009], deals with the learning setting where the training and testing sets are considered to be sampled from different distributions. It is also assumed that training data is labeled and it comes from the *source* domain while testing data is unlabeled and it is called the *target* domain. The task consists in finding a function that bridges both domains and generalizes well on the target domain. For example, the training pictures may come from a different camera than the testing pictures, and the pictures might show different locations and taken by different users. The mathematical formalization of this learning task is first established by Daumé and Marcu [2006]. The first deep DA work is presented by Glorot et al. [2011], they propose an auto-encoder for sentiment analysis. Another approach based on CNNs that became predominant is later introduced by Long et al. [2015]. They presented a two-stream deep architecture in which each stream represents the source and target model, respectively. In this approach, a domain regularization loss is used on both streams to adapt the source to the target domain. The purpose of this loss is the shift between domains using a discrepancy metric such as the maximum mean discrepancy (MMD) [Long et al., 2015, 2016; Yan et al., 2017;

Long et al., 2017], the central moment discrepancy [Zellinger et al., 2017, 2019], the correlation alignment (CORAL) function [Sun et al., 2016; Sun and Saenko, 2016], and the Wasserstein metric [Lee et al., 2019]. Inspired by adversarial training Goodfellow et al. [2014], another popular approach consists in finding a common feature space using adversarial training Ganin and Lempitsky [2015]; Tzeng et al. [2017]; Saito et al. [2018a,b]. For instance, Tzeng et al. [2017] trains a source encoder CNN and its weights are subsequently fixed to train a target encoder. The objective of the target encoder training is to deceive a domain discriminator between samples of both domains. Another method is presented by Ganin and Lempitsky [2015], they simultaneously train a generator and a discriminator by inverting the gradients using a special layer.

Part I

Egocentric Activity Recognition From Visual Lifelogs

Chapter 3

Activity Recognition at Image-Level

Contents

3.1 Approach 30

3.2 NTCIR-12 Dataset 31

3.3 Implementation 33

3.4 Evaluation Metrics 37

3.5 Results 37

3.6 Conclusion 41

In this chapter, we present our method on *activity* classification from still lifelog images. As stated in previous chapters, this problem has received little attention in the literature in comparison with *action* classification from egocentric videos. This might be partially due to the lack of available datasets, but also to the fact that motion features cannot be reliably extracted from them, because of its low frame rate. Both things are essential for action classification. Our method combines the outputs of different layers from a convolutional network as the input to a random decision forest.

This chapter is organized as follows. In section 3.1 we present our ensemble approach based on convolutional networks and random forests. Subsequently, in section 3.2 we describe the dataset we used in our classification experiments, namely the NTCIR-12 [Gurrin et al., 2016]. We also described its annotation process and training/testing partitions. Then we present the implementation details in section 3.3. In Section 3.4 we detail the experimental evaluation metrics that take into account the dataset imbalance from random and temporal partitions. We then present the results in Section 3.5. Finally, in Section 3.6 we present the concluding remarks of this chapter.

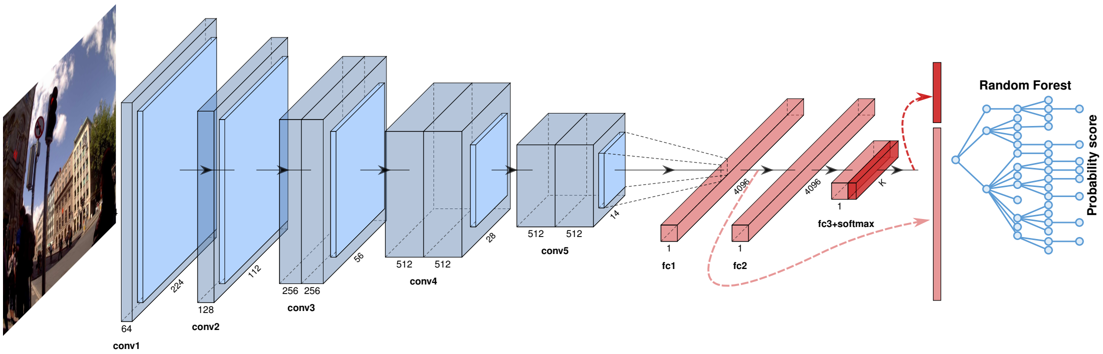


Figure 3.1: Activity recognition at image-level. After fine-tuning a CNN, we combine the softmax probabilities and other fully-connected layers into a single vector. Later on, we train a random forest using this vector as its input.

3.1 Approach

Our method is an ensemble classifier composed of a convolutional network (CNN) and a random decision forest. It concatenates the outputs from distinct layers of a convolutional backbone network into a single vector. Depending on the backbone architecture, the input for the random forest can be extracted from the last convolutional layer, a fully-Connected (FC) layer, or the softmax layer. This vector is later used as the input of the random forest (RF). For instance, an RF that takes as input the output of the first FC layer and the softmax layer from a VGG-11 is shown in Fig. 3.1. The ensemble is trained by first fine-tunning the backbone network on the data. Subsequently, the RF is trained over the selected concatenated outputs from the CNN.

Our approach can be seen as the shallow version of Deep Neural Decision Forests [Kontschieder et al., 2015], but on multiple layers. Moreover, it can be also seen as the generalized version of the method proposed by Castro et al. [2015]. In comparison with them, our approach does not rely on color histograms and time features, and it combines different output layers from a CNN rather than just the final prediction. Both characteristics made their method depend on specific context settings of the persons from the training dataset, i.e. it might not generalize well on certain cases. For instance, their classification might be affected by the lifestyles of different persons having the same activity at distinct times of the day. The insight underlying our method is that different output layers could help on the generalization of characterizing features of a given activity among distinct users.

In comparison with the work presented by Castro et al. [2015], we performed our

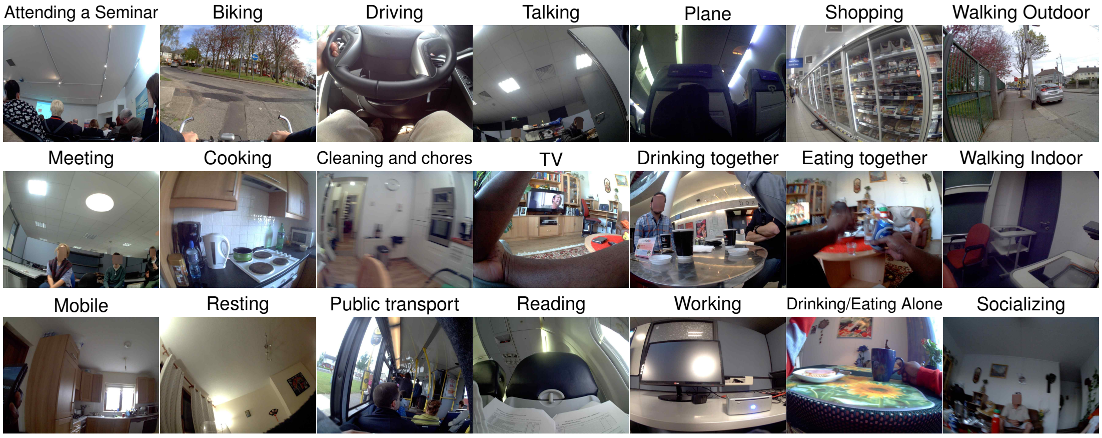


Figure 3.2: Examples of all the *activity* categories randomly selected from the annotated NTCIR-12 dataset [Gurrin et al., 2016] captured by a chest-mounted OMG Autographer camera.

tests on three people having different lifestyles, rather than just one person. Therefore, our approach cannot take advantage of their time information and color histogram. Moreover, the number of labeled images for a single person in our annotated dataset (≈ 15k) is less than half than theirs (≈ 40k). Hence, our task is more challenging because we must deal with an increased intra-class variability with a much smaller number of images.

3.2 NTCIR-12 Dataset

In our experiments, we used a subset of images from the NTCIR-12 dataset [Gurrin et al., 2016]. This dataset was introduced in the context of an image retrieval challenge. It comprises of 89,593 egocentric pictures describing the daily life of three people. The pictures were automatically captured every thirty seconds using an OMG Autograph camera. The total number of recorded days was 79, and each person wore the camera for about three weeks. This dataset originally contained 13,883 annotated images using six different *activity* categories. But these categories only described ambulation and transportation *activities*, according to the groups proposed by Lara and Labrador [2013].

We annotated a subset of 44,901 images using 21 different *activity* categories that better describe the daily activities of the people. For example, a labeled image for each category is shown in Fig. 3.2. Around 15,000 images were annotated for each

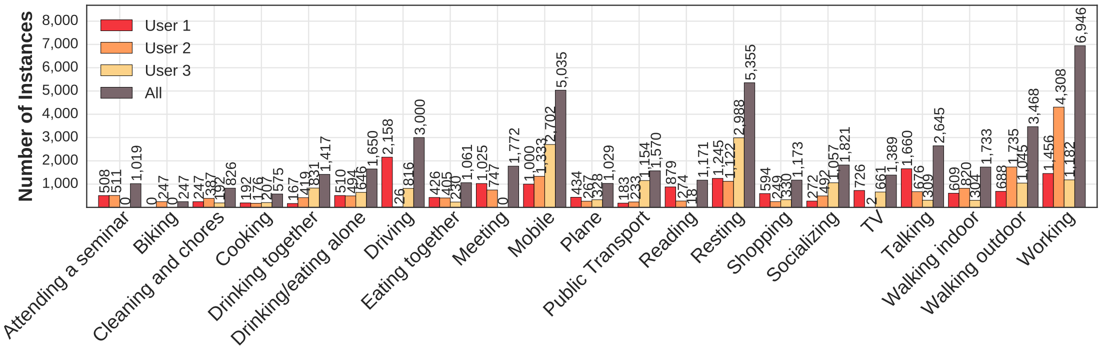


Figure 3.3: Distribution of *activity* categories per user of the NTCIR-12 dataset.

person in the dataset. Although the labeled pictures correspond to different dates and times, during the annotation process the images were not individually labeled but in a sequential fashion. In other words, the temporal context of a continuous *activity* across frames was implicitly taken into account by the annotators. The distribution of categories per user is shown in Fig. 3.3.

We tested our methods on two different kinds of data partitions: random and temporal splits. The former randomly and proportionally divide each category of the dataset into training and testing splits. The latter takes into account the time and proportionally separates full lifelogging sequences for training and testing. Since similar consecutive frames are only in one of the splits, the classification task is harder on this separation. Both data partitions are further detailed in the following subsections.

3.2.1 Random partition

This partition considered all the pictures from the three users regardless of their date and time. With the goal of maintaining the same percentage for each class, we first performed stratified 10-fold cross-validation over the images. Then a validation split for each fold containing 10% of the data was created by further making a stratified shuffle of its training split.

3.2.2 Temporal partition

We split the 78 annotated lifelogs sequences into training, validation, and test sets. In order to reflect the class distribution of the whole dataset across the splits as seen in Fig. 3.4, the full sequences were allocated as follows. Since the day sequences contain a different number of annotated pictures, they were grouped into bins containing

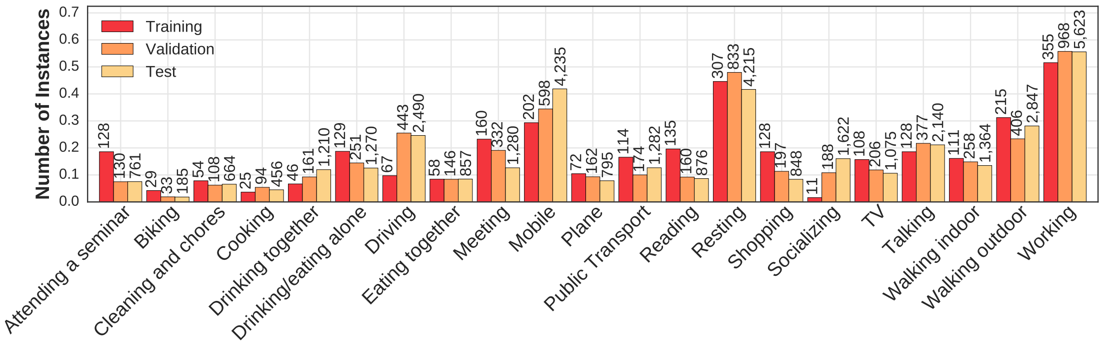


Figure 3.4: NTCIR-12 temporal splits summary. All split distributions are normalized, but the corresponding number of instances for each category is shown on top of each split.

a similar number of images using the first-fit decreasing algorithm. This resulted in 50 bins containing an average of 2 days and 1,000 images. In the second step, candidates of test splits were enumerated by considering all the possible combinations of bins using the Twiddle algorithm [Chase, 1970]. Specifically, we considered 7 out of the 50 bins for the test split, resulting in $C_7(50) = 99,884,400$ test candidate splits. In the third step, we found the most representative split by comparing the category distribution of each candidate split with respect to the whole dataset. The comparison measure between distributions was the Bhattacharyya distance. We selected the test candidate split with the shortest distance. The validation split was obtained by doing the same steps but only with the remaining training bins, i.e. by considering 3 out of the 43 left bins $C_3(43) = 12,341$. The number of day sequences for training, validation, and test splits were 59, 7, and 12, respectively.

3.3 Implementation

We used three CNN architectures as the backbone of our ensembles, namely the VGG-16 [Simonyan and Zisserman, 2014b], InceptionV3 [Szegedy et al., 2016], and ResNet-50 [He et al., 2016]. The input of the RFs in our ensembles was the output of each and all FC layers following the last convolutional layer. The following subsections provide further implementation details for all the models.

3.3.1 CNN Training

All the CNN models were pre-trained on ImageNet [Russakovsky et al., 2015] using the Keras framework [Chollet et al., 2015]. All models were fine-tuned in two phases using the stochastic gradient descent (SGD) as an optimization method. In the first phase, only the FC layers were back-propagated with the objective of initializing their weights. During the second phase, also the last convolutional layers were added to the fine-tuning process. As a mechanism for regularization, dropout layers Srivastava et al. [2014] were added after each FC and average pooling layers. The class imbalance was handle only for the random partition models using the class weighting scheme proposed by King and Zeng [2001].

VGG-16

Random Partition We fine-tuned a VGG-16 network [Simonyan and Zisserman, 2014b] as being the closest to AlexNet [Krizhevsky et al., 2012]. During the first phase, we used SGD for 10 epochs for all folds, a learning rate $\alpha = 1 \times 10^{-5}$, a batch size of 1, a momentum $\mu = 0.9$, and a weight decay equal to $5 \times 10{-}6$. In the second phase, the last three convolutional layers were also fine-tuned. Moreover, the SGD ran for another 10 epochs for each fold and set with the same parameters except the learning rate $\alpha = 4 \times 10^{-5}$.

Temporal Partition The VGG-16 was fine-tuned for 14 epochs. For the first 10 epochs, only the FC layers were optimized with a learning rate $\alpha = 1 \times 10^{-5}$, a batch size of 1, a momentum $\mu = 0.9$, and a weight decay equal to $5 \times 10{-}6$. In the last 4 epochs, the last 3 convolutional layers were added and the learning rate was modified to $\alpha = 1 \times 10^{-5}$.

InceptionV3

Random Partition During the first phase, the last FC layer of InceptionV3 [Szegedy et al., 2016] was optimized using SGD for 10 epochs for all folds, a learning rate $\alpha = 1 \times 10^{-5}$, a batch size of 32, a momentum $\mu = 0.9$, and a weight decay equal to $5 \times 10{-}6$. In the second phase, the last inception block was added to the optimization process and the network was optimized for another 10 epochs setting the learning rate $\alpha = 4 \times 10^{-5}$ and the batch size to 10.

Temporal Partition InceptionV3 was fine-tuned for 12 epochs. The first phase consisted of 10 epochs in which the only FC layer was optimized with a learning rate $\alpha = 1 \times 10^{-5}$, a batch size of 10, a momentum $\mu = 0.9$, and a weight decay equal to $5 \times 10{-}6$. In the second phase, the last inception block was also added to the fine-tuning process and the learning rate α increased to 4×10^{-5}.

ResNet-50

Random Partition In the first phase, the last FC layer of ResNet-50 [He et al., 2016] was fine-tuned for 10 epochs for all folds, a learning rate $\alpha = 1 \times 10^{-3}$, a batch size of 32, a momentum $\mu = 0.9$, and a weight decay equal to $5 \times 10{-}6$. In the last phase, the last residual block was also optimized using SGD with the same learning rate and a batch size of 10 for three additional epochs.

Temporal Partition ResNet-50 was fine-tuned for 4 epochs. In the first 2 epochs, only the FC layer was optimized with a learning rate $\alpha = 1 \times 10^{-3}$, a batch size of 10, a momentum $\mu = 0.9$, and a weight decay equal to $5 \times 10{-}6$. In the last 2 epochs, the last residual block was also fine-tuned but the training hyperparameters remained the same.

3.3.2 Random Forests training

With the goal of finding an adequate number of trees for each RF, we trained all the RF combinations from each CNN using a number of trees equal to $50, 75, 100, 200, \ldots, 900$. Their training criterion was the Gini impurity [Breiman et al., 1984] and their nodes were expanded until all the leaves were pure. The mean accuracy of all validation splits was calculated for each RF and number of trees, as depicted in Fig. 3.5. The number of trees to be used for each RF was the lowest one after which the performance did not improve significantly. The random forests (RF) in our ensembles were trained using the Scikit Learning framework [Pedregosa et al., 2011]. The training details of each ensemble configuration are given below.

VGG-16+RF

Random Partition Five different RF were trained using the distinct layers of the best epoch from the fine-tuned VGG-16 for each fold. The combinations of layers for the

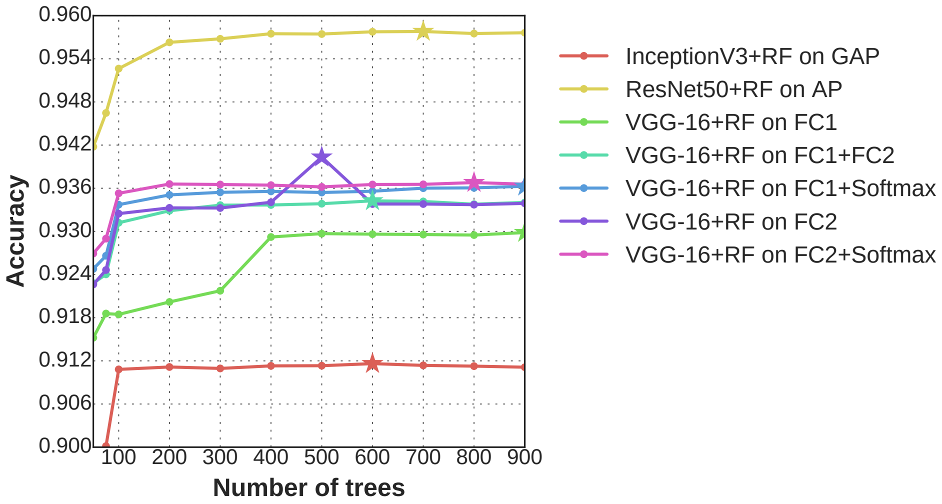


Figure 3.5: Mean accuracy of the validation split for each random forest with respect to different number of trees. The maximum accuracy values is pointed by a star mark.

ensembles were: FC1, FC2, FC1+FC2, FC1+FC3, and FC2+FC3. Their respective number of trees of each combination were 400, 500, 600, 300, and 200. The maximum depth of each combination was 49, 47, 53, 58, and 48, correspondingly.

Temporal Partition The five combinations of RFs previously described were trained for this data partition using a number of trees equal to 500. Explicitly, the combinations of layers for the ensembles were: FC1, FC1+FC2, FC1+FC3, FC2, and FC2+FC3. Their maximum depth for each combination was 55, 48, 60, 52, and 51, respectively.

For comparative purposes, the Convolutional Neural Network Late Fusion Ensemble (CNN LFE) [Castro et al., 2015] was also implemented using the VGG-16 as backbone network. This RF model received as input the softmax probability scores, the day of the week, the time of the day, and 10-bin size histogram for each color channel. Its number of trees was 500 and its maximum depth was 47.

InceptionV3+RF

Random Partition A random forest was trained using the global average pooling (GAP) layer from InceptionV3. This random forest had 100 trees as estimators and a maximum depth of 58.

Temporal Partition The GAP layer was also used as input for an RF with 100 estimators and a maximum depth of 72.

ResNet50+RF

Random Partition A random forest was trained using the average pooling (AP) layer from the previously ResNet-50 network. This random forests had 400 trees as estimators and a maximum depth of 49.

Temporal Partition A random forest was trained on the AP layer using 500 estimators and its resulting max. depth was 53.

3.4 Evaluation Metrics

Since the dataset is highly imbalanced, the classification performance is measured by not only using the accuracy, but also other macro metrics for precision, recall, and F1-score. These additional macro metrics are an unweighted average of the metrics taken separately for each class, therefore they do not consider the available number of instances for each class. Moreover, the results are cross-validated.

3.5 Results

3.5.1 Random Data Partition Results

The classification results on the random data partition are presented in Table 3.1. These results show that the CNN+RF ensembles improve the performance of all backbone CNNs achieving a similar classification performance. The improvement of the baseline accuracy for the VGG-16, InceptionV3, and ResNet-50 are 6.04%, 8.2%, and 1.31%, correspondingly. This suggests that the ResNet-50 training is more robust and that its output feature vector encodes the information better. Although the best ensemble is *VGG-16+RF on FC2+FC3*, the macro metrics indicate that the *ResNet-50+RF on AP* has a similar or even better performance. Considering that ResNet-50 is a lighter architecture than VGG-16, the best ensemble is the *ResNet-50+RF on AP*.

The recall scores in Table 3.1 show that some of the categories with fewer learning instances improved their recall significantly, i.e. *Cooking*, and *Cleaning and Choring*.

	CNN			CNN+RF						
Activity	**VGG-16**	**InceptionV3**	**ResNet-50**	**VGG-16+RF on FC1**	**VGG-16+RF on FC1+FC2**	**VGG-16+RF on FC1+FC3**	**VGG-16+RF on FC2**	**VGG-16+RF on FC2+FC3**	**InceptionV3 on GAP**	**ResNet50+RF on AP**
Attending a seminar	76.93	78.70	**80.57**	77.75	78.21	78.11	78.21	78.11	78.70	78.31
Biking	92.32	91.93	99.17	98.20	97.57	97.98	97.17	97.98	98.80	**99.20**
Cleaning and chores	61.84	57.14	92.22	92.06	92.13	92.62	92.26	**92.74**	92.61	92.39
Cooking	84.78	81.98	91.69	96.48	95.97	95.79	95.47	**96.49**	95.10	96.15
Drinking/eating alone	74.92	75.94	88.25	88.70	89.77	89.71	89.65	89.71	**90.85**	90.37
Drinking together	89.42	80.74	94.93	**96.48**	95.98	96.05	95.91	95.84	95.63	95.49
Driving	99.50	99.60	**99.90**	99.74	99.77	99.77	99.73	99.77	99.83	99.83
Eating together	88.78	80.58	96.42	96.75	97.26	97.26	96.89	97.36	96.50	**98.11**
Meeting	90.00	80.92	96.61	95.36	95.49	95.49	95.54	95.43	93.29	**96.78**
Mobile	93.32	89.59	95.27	97.55	97.46	97.36	97.30	97.34	97.22	**97.58**
Plane	84.94	**87.67**	82.61	77.21	80.66	81.15	81.54	83.58	87.16	75.70
Public Transport	89.17	89.23	**92.41**	90.87	91.14	91.53	91.27	91.59	91.46	91.59
Reading	88.23	81.40	96.83	96.49	96.67	96.67	96.59	96.59	97.35	**97.69**
Resting	92.87	87.97	91.54	94.56	94.90	**95.14**	94.90	95.11	91.07	91.02
Shopping	82.44	81.07	93.18	94.98	95.57	95.14	95.57	95.65	95.40	**96.68**
Socializing	91.60	89.13	97.47	98.47	98.46	98.46	98.57	**98.73**	97.37	98.63
Talking	76.25	72.36	89.64	93.53	93.50	93.50	93.65	93.61	**95.54**	94.10
TV	92.59	93.74	96.04	**97.21**	96.91	96.69	97.05	96.91	96.55	96.54
Walking indoor	79.63	79.17	92.73	94.81	94.92	94.75	95.04	94.87	95.15	**95.85**
Walking outdoor	90.89	90.95	94.61	96.96	97.03	96.94	97.00	96.83	97.35	**98.27**
Working	96.27	95.82	97.34	98.18	97.98	97.91	97.96	97.91	98.81	**99.38**
Accuracy	89.46	87.07	94.08	95.26	95.41	95.41	95.41	**95.50**	95.27	95.39
Macro precision	87.06	83.85	92.60	94.63	94.59	94.62	94.57	94.61	94.96	**96.14**
Macro recall	86.51	84.08	93.31	93.92	94.16	94.19	94.15	**94.39**	94.37	94.27
Macro F1-score	86.45	83.77	92.78	94.11	94.23	94.28	94.23	94.38	94.53	**95.00**

Table 3.1: Comparison of the ensembles of CNN+Random forest on different combinations of layers on the random data partition. Upper table shows the recall scores per category and the lower table shows the performance metrics.

Moreover, the confusion matrices of the convolutional networks depicted in Fig. 3.6 show that the classes *Drinking/Eating alone* and *Eating together* suffer overlapping. It also shows that CNN+RF ensembles increase their recall for all backbone networks. The category *Plane* was the only one with a decreasing recall for the *VGG-16+RF on FC2+FC3* ensemble. We consider that its decrease is a consequence of the random forest balancing the prediction error among classes, as its baseline recall is high (87.67%) with respect to its number of training instances (1,026).

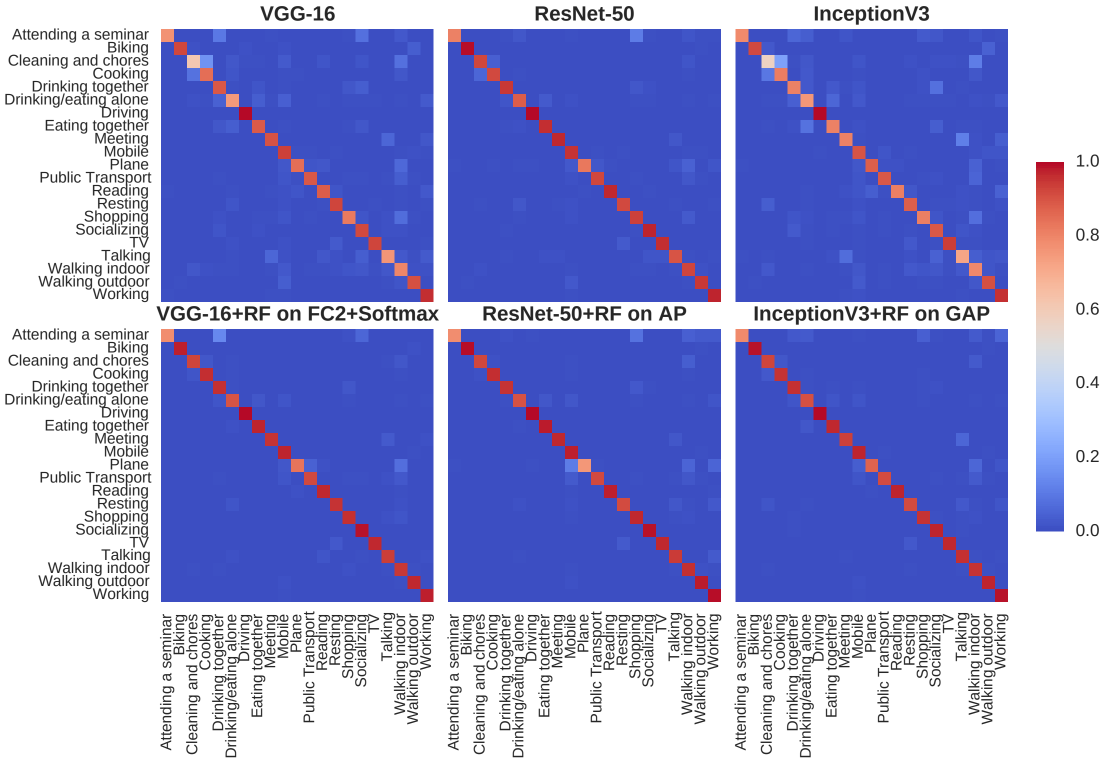
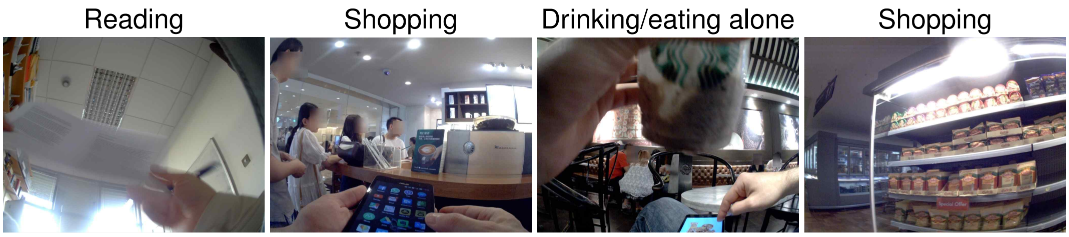


Figure 3.6: Normalized confusion matrices of the best combination of layers for each baseline convolutional neural network.

Some classification examples for the CNN and CNN+RF models are shown in Fig. 3.7. The first and second columns show an example where most of the CNN+RF improved the *activity* classification. An example where both the CNN and CNN+RF wrongly classified an *activity* is presented in the third column. The last column exemplifies an *activity* that was correctly classified by the CNN but incorrectly by the CNN+RF.

3.5.2 Temporal Data Partition Results

The classification results on the temporal data partition are shown in Table 3.2. In comparison with the results of the random data partition (Table 3.1), the classification performance dropped significantly as expected. Specifically, the accuracy was diminished on average by 17.71% for all methods. The results show that the CNN+RF ensemble only improves the prediction for one of the three CNN tested models, i.e. the VGG-16. Furthermore, the best model was a CNN model, namely InceptionV3. This suggests that newer convolutional networks encode better the information at the end of



VGG-16 Top 5

#	Activity	Score
1	Drinking/eating alone	0.3954
2	Mobile	0.1861
3	Cooking	0.0687
4	Working	0.0653
5	Socializing	0.0525
11	Reading	0.0220

#	Activity	Score
1	Mobile	0.6543
2	Drinking together	0.1352
3	Socializing	0.0967
4	Shopping	0.0912
5	Attending a seminar	0.0095

#	Activity	Score
1	Mobile	0.6646
2	Public Transport	0.1136
3	Drinking together	0.0699
4	Talking	0.0366
5	Working	0.0304
9	Shopping	0.0098

#	Activity	Score
1	Shopping	0.9981
2	Walking indoor	0.0019
3	Walking outdoor	0.0000
4	Mobile	0.00001
5	Talking	0.0000

ResNet-50 Top 5

#	Activity	Score
1	Socializing	0.5277
2	Reading	0.4666
3	Cleaning and chores	0.0030
4	Cooking	0.0010
5	Public Transport	0.0006

#	Activity	Score
1	Mobile	0.9953
2	Shopping	0.0020
3	Drinking together	0.0016
4	Socializing	0.0009
5	Meeting	0.0001
19	Walking outdoor	0.0000

#	Activity	Score
1	Cleaning and chores	0.8409
2	Cooking	0.0915
3	Drinking/eating alone	0.0498
4	Socializing	0.0112
5	Biking	0.0029
7	Shopping	0.0008

#	Activity	Score
1	Shopping	1.0000
2	Walking indoor	0.0000
3	Walking outdoor	0.0000
4	Talking	0.0000
5	TV	0.0000

InceptionV3 Top 5

#	Activity	Score
1	Cooking	0.3268
2	Cleaning and chores	0.1674
3	Talking	0.1599
4	Meeting	0.1251
5	Resting	0.0526
10	Reading	0.0236

#	Activity	Score
1	Mobile	0.8928
2	Drinking together	0.0291
3	Shopping	0.0285
4	Socializing	0.0135
5	Walking indoor	0.0095

#	Activity	Score
1	Drinking together	0.2910
2	Mobile	0.1733
3	Drinking/eating alone	0.0928
4	Talking	0.0798
5	Socializing	0.0669
12	Shopping	0.0159

#	Activity	Score
1	Shopping	0.9979
2	Eating together	0.0005
3	Cooking	0.0004
4	Plane	0.0004
5	Walking indoor	0.0003
7	Walking outdoor	0.0001

Figure 3.7: Classification activity examples. On top of each image is shown its true activity label and on bottom its top 5 predictions by VGG-16, ResNet-50 and InceptionV3. Additionally, the result of the ensembles *VGG-16+RF on FC2+FC3*, *ResNet-50+RF on AP*, and *InceptionV3+RF on GAP* is highlighted on color in its corresponding table. The green and red colors means true positive and false positive classification, respectively.

the convolutional layers, thus the random forests did not improve their performance.

The results also showed that not all combinations of final layers for CNN+RF improved the performance. In the case of the VGG-16 only the FC1 plus the FC3 layers were better than the baseline. Moreover, the CNN LFE method [Castro et al., 2015] was better than the latter. This might be due to the additional information from the color histograms and the date & time, but also because of the reduced number of evaluated people.

	CNN			CNN+RF							
Measure	**VGG-16**	**InceptionV3**	**ResNet-50**	**VGG-16+RF on FC1**	**VGG-16+RF on FC1+FC2**	**VGG-16+RF on FC1+FC3**	**VGG-16+RF on FC2**	**VGG-16+RF on FC2+FC3**	**CNN LFE on VGG-16** [Castro et al., 2015]	**InceptionV3 on GAP**	**ResNet-50+RF on AP**
Accuracy	75.97	**78.44**	78.31	74.22	74.83	76.80	75.02	75.87	77.39	74.62	77.08
Macro Precision	68.50	72.44	73.06	70.77	69.52	71.93	69.12	69.78	69.66	**76.64**	73.68
Macro Recall	67.49	**70.62**	69.94	65.06	65.91	66.86	65.60	66.08	67.79	62.61	67.24
Macro F1-score	66.80	69.85	**70.60**	65.80	65.93	67.51	65.60	66.28	66.99	64.94	68.54

Table 3.2: Comparison of the ensembles of CNN+Random forest for different combinations of layers on the temporal data partition.

3.6 Conclusion

In this chapter we introduced an ensemble approach that combines convolutional networks and random forests (CNN+RF) for image-level *activity* recognition. We presented a classification performance comparison between the CNN+RF and their respective CNN backbone models for random and temporal data partitions. The results showed that all the CNN+RF models outperformed their corresponding CNN baseline for the random partition. The best model for this partition was the *VGG-16+RF on FC2+FC3* ensemble achieving a 95.50% of accuracy. Nonetheless, a similar performance was achieved by *ResNet-50+RF on AP*, but its backbone architecture is lighter. The results for the temporal partition showed that the CNN+RF models not always improved its CNN baseline. The best tested model was the *InceptionV3* and only one ensemble for the three tested networks resulted in better performance, i.e. *VGG-16+RF on FC1+FC3*. In the next chapter, we present more robust generalization tests on the same methods on a dataset captured by a large number of people having a wide variety of lifestyles.

Chapter 4

Activity Recognition at Sequence-Level

Contents

4.1 Temporal Approaches 46

4.2 Event-Based Approach 48

4.3 ADLEgoDataset 49

4.4 Experimental Settings 55

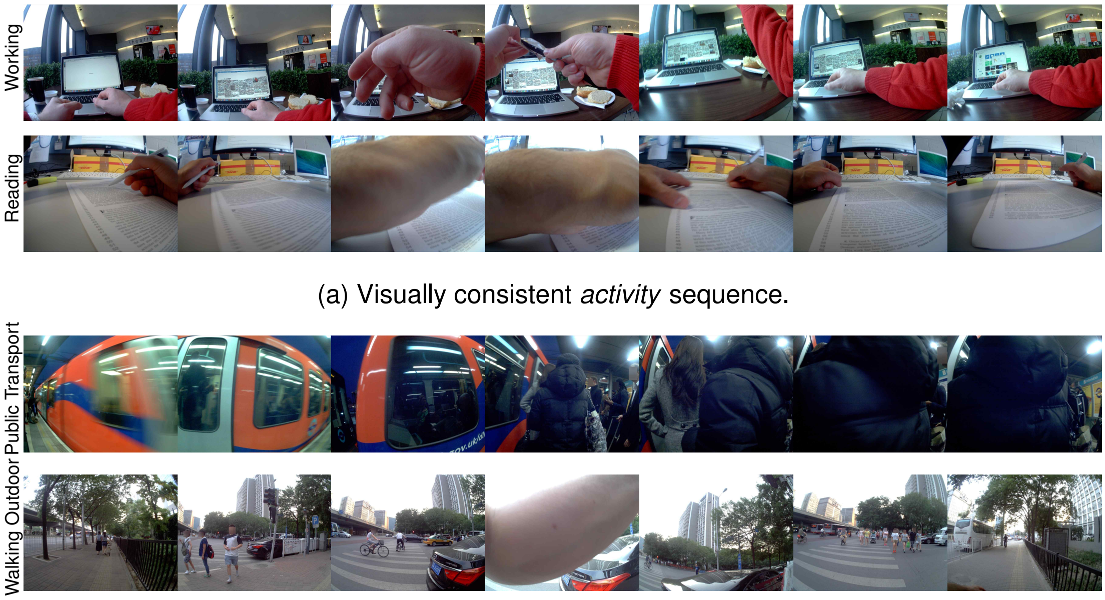
4.5 Evaluation Metrics 58

4.6 Results 59

4.7 Conclusions 64

In this chapter, we present our method for activity recognition from sequences of visual lifelogs. One of its main obstacles is the very low frame-rate, they typically have between 2 or 3 frames per minute. In consequence, motion-based features such as optical flow cannot be reliably estimated. Nevertheless, they still exhibit *temporal coherence* of concepts as presented by Byrne et al. [2010]. The temporal coherence refers to the tendency of similar concepts to appear in neighboring frames of a sequence. For example, an *activity* such as *cooking* has associated repetitive objects like *fridge*, *stove*, and the *kitchen* itself.

This tendency is usually present in visually consistent and even visually varied *activity* sequences. Both types of sequences are illustrated in Fig. 4.1. Visually consistent sequences are characterized for mostly presenting the same concepts throughout all the frames. In the case of the *working* sequence in Fig. 4.1a, the laptop computer, and the



(a) Visually consistent *activity* sequence.

(b) Visually varied *activity* sequence.

Figure 4.1: Temporal coherence of concepts in lifelogging *activity* sequences. Although each sequence was captured at 2 fps, the temporal coherence concepts is preserved in (a) visually consistent and (b) visually varied *activity* sequences. Images taken from the NTCIR-2012 dataset [Gurrin et al., 2016].

hands are present in all the sequences. In a visually varied sequence, the concepts appear only in some consecutive frames. In the case of the *walking outdoor* sequence in Fig. 4.1a, a car appears in three of the middle frames of the sequence. The temporal coherence of concepts is preserved in drastic changes in appearance.

Furthermore, temporal coherence provides valuable information for single frames that are difficult to interpret alone. One of these difficulties is occlusion of arms/hands, near objects, and other persons; as seen in all the *activity* sequences of Fig. 4.1. Another difficulty is the abrupt changes of the field of view, like the one in the first and second frames of the *walking outdoor* sequence in Fig. 4.1b. The correct *activity* prediction of these individual frames cannot be made without looking at neighboring frames.

The temporal coherence of an *action* or *activity* is defined by their starting and ending boundaries. When they are known, they are implicitly used for training temporal action recognition algorithms[1]. These boundaries in lifelogs sequences are clearly unknown but splitting them into similar temporal subsequences could provide further

[1] Otherwise, the task of determining not only the *action* but their starting and ending times are known as action detection.

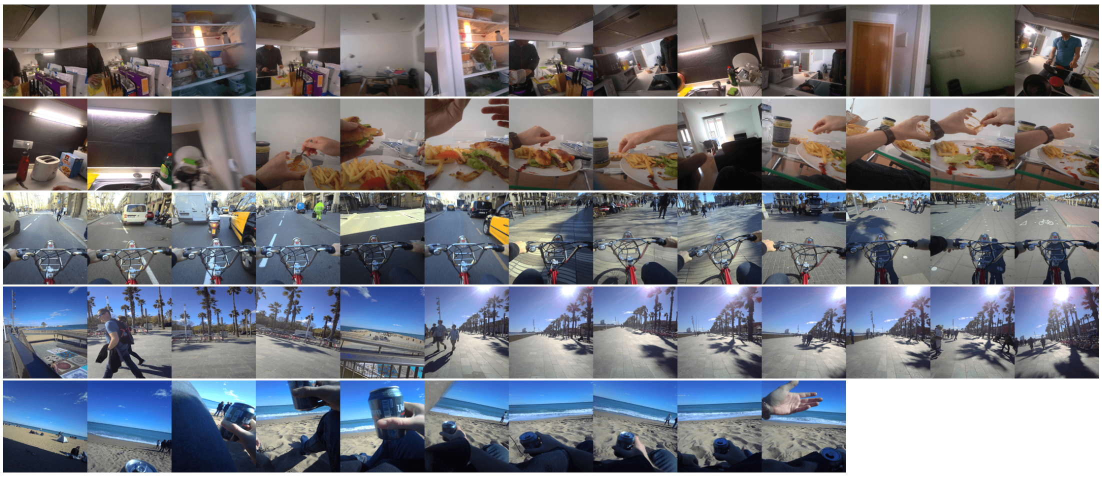


Figure 4.2: Initial frames of five consecutive events segmented from a lifelog sequence using SR-Clustering method proposed by [Dimiccoli et al., 2017]. Each row shows the first 13 frames of each event.

information for action recognition. These subsequences or *events* are obtained by clustering sequential images in homogeneous groups with respect to criteria. These criteria are specific to the application at hand. For example, Poleg et al. [2014] modeled events as temporal segments characterized by the same global motion and partitioned egocentric videos based on motion-features. As another example, Furnari et al. [2016] modeled events as groups of pictures highlighting the presence of personal locations of interest specified by the end-user. We focus on the definition provided by Talavera et al. [2015] and Dimiccoli et al. [2017], they define events as temporal semantic segments sharing semantic and contextual information. For example, Fig. 4.2 shows five consecutive subsequences segmented from a lifelog using the method proposed by Dimiccoli et al. [2017].

In this chapter, we also present our lifelogging dataset, the ADLEgoDataset. One of the main challenges of activity recognition from lifelogs is the lack of data collected from different users performing a wide set of *activities* . In consequence, robust generalization tests cannot be performed. Unlike egocentric videos, lifelogs cover full-day activities performed in a larger variety of settings. Both characteristics made the lifelogging datasets more difficult to acquire. First, it requires longer recording times that also makes the process more expensive. Second, recording several locations and people during a day has more privacy restrictions than indoor locations.

In section 4.1, we present one temporal training strategy and two temporal architectures for activity recognition that work directly over egocentric lifelogs. These meth-

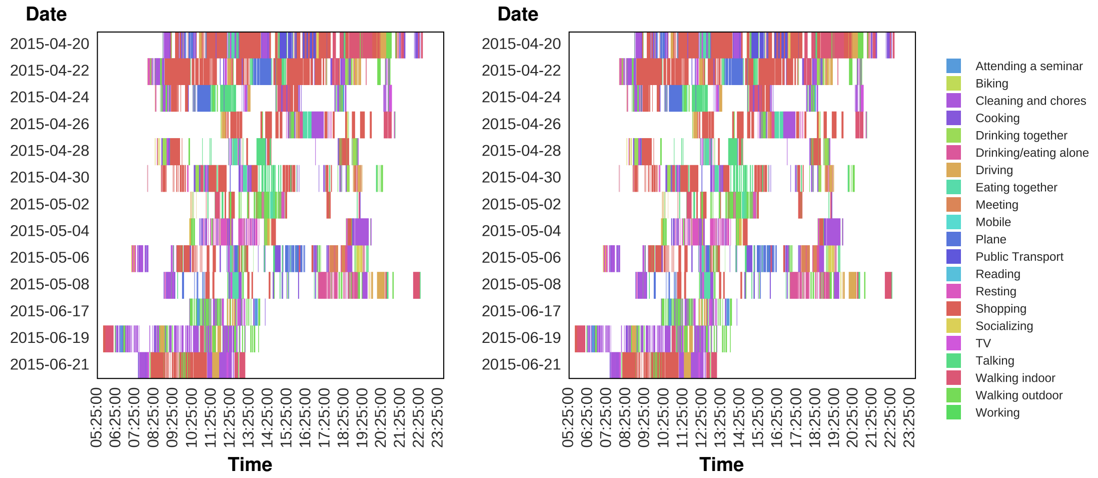


Figure 4.3: Thirteen *activity* day sequences of annotated photos from two people of the NTCIR-12 dataset [Gurrin et al., 2016].

ods use long short-term memory *(LSTM)* units [Hochreiter and Schmidhuber, 1997] on top of a CNN to discover long-range temporal relationships and to learn how to integrate information over time. Complementary to them, we present in section 4.2 another approach that uses temporal boundaries estimated from segmented events. Next, in section 4.3 we throughly describe ADLEgoDataset, consisting of 105,529 images, from 15 users, with an average of 6,682 images per user. In section 4.3.4, we describe its training and testing splits considering not only *seen users* during training, but also *unseen users*. We explain our experimental settings and implementation details on the NTCIR-12 and ADLEgoDataset in section 4.4, followed by the evaluation metrics in section 4.5. Finally, in section 4.6 we discussed the obtained results and provide concluding remarks in section 4.7.

4.1 Temporal Approaches

Encouraged by Byrne et al. [2010], our hypothesis is that successive images in lifelog sequences encode temporal information that makes more accurate *activity* predictions. Since the duration of a lifelog can be from a few hours to a whole day, it contains different *activities* performed by a person. Despite the sparseness of these annotated *activities* , temporally adjacent frames share the same label, as seen in the sequences of Fig. 4.3. Our approaches aim to exploit these characteristics by using LSTM units [Hochreiter and Schmidhuber, 1997]. They have been proved successful on third-person *action* recognition in several works [Baccouche et al., 2011; Donahue et al., 2015, 2017; Ng

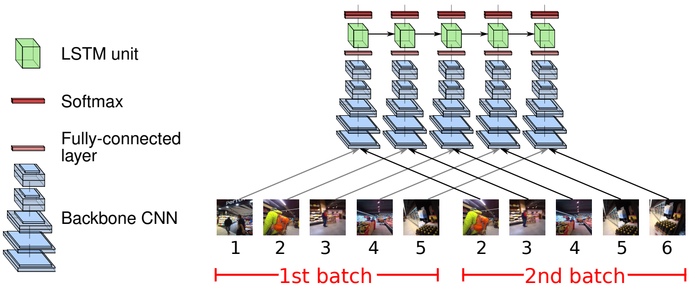


Figure 4.4: Sliding window training mechanism for lifelog sequences. A lifelog sequence is divided into overlapping subsequences that are the input of a CNN+LSTM architecture.

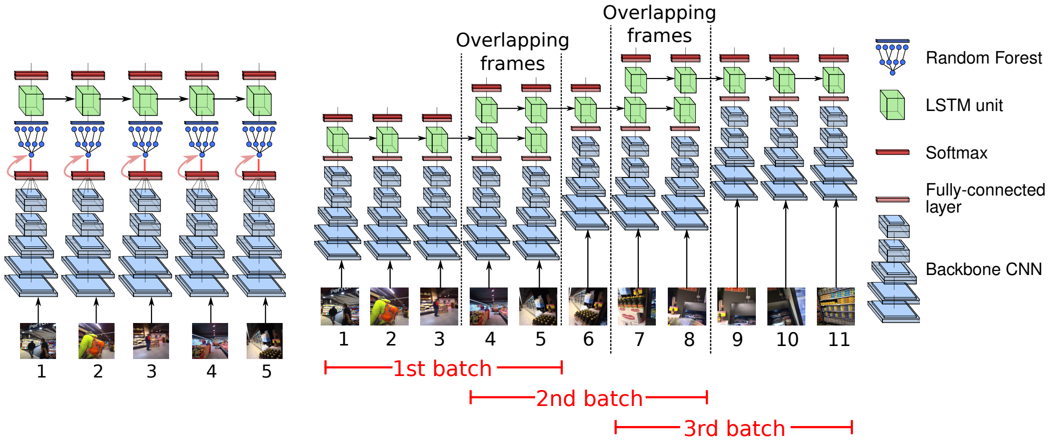
et al., 2015]. We propose a training mechanism for lifelogs called *sliding window*, and two LSTM architectures: the *CNN+RF+LSTM* and the *Piggyback LSTM*.

Sliding Window Training Mechanism Instead of feeding a full lifelog sequence to a CNN+LSTM network, this training mechanism splits the lifelog into overlapping frame segments of fixed length, as illustrated in Fig. 4.4. In comparison with previous works on third-person *action* recognition [Donahue et al., 2017; Ng et al., 2015], the model is not supposed to determine a single *activity* but to improve the prediction of neighboring frames depicting the same *activity* . In other words, we do not expect that it learns the temporal evolution of features, but to *smooth* the predictions of similar frames when not knowing the event boundaries.

CNN+RF+LSTM Architecture This architecture is the temporal version of our static egocentric image classifier CNN+RF, proposed in chapter 3. The temporal mechanism used on this architecture are LSTM units, as shown in Fig. 4.5a.

Piggyback LSTM Architecture This architecture explicitly models the temporal relation between adjacent overlapping frames in subsequent input batches of an LSTM architecture. Since a day lifelog can be composed of around two thousand frames, it is infeasible for an LSTM to learn such long-range dependencies [Gers et al., 2000]. The purpose of this model is to learn these complex long-range temporal dependencies without considering each lifelog as a single sequence. A similar model to ours was independently presented at the same time by Lee et al. [2017], by it was used for skeleton-based *action* recognition.

Our architecture consists of a backbone CNN, a fully-connected (FC) layer, an



(a) (b)

Figure 4.5: Proposed architectures for *activity* recognition using temporal contextual information: (a) CNN+RF+LSTM and (b) CNN+Piggyback LSTM with a window size and overlap of 5 and 2 frames, respectively.

LSTM unit, and a final FC layer, as shown in Fig. 4.5b. It carries the information across consecutive input batches by overlapping n frames between them. This is achieved by specifically making the output dimensions of the first FC layer and the LSTM equal. In this way, after feedforwarding the first sequence batch, the LSTMs output of the last n frames is stored. In subsequent passes, these values are the input for the first n LSTM units. For instance, Fig. 4.5b shows a configuration composed of 5 timesteps with overlapping of 2 output/input frames. For the first input sequence (frames 1-5 in Fig. 4.5b), the LSTM has as input the output of the FC layers. For subsequent sequences, the overlapping frames are 4-5 and 7-8 in Fig. 4.5b. For these frames, the LSTM input is the output of the LSTM layer from the previous sequence.

4.2 Event-Based Approach

Our approach consists of first splitting a lifelog into events. These events are defined as temporally adjacent pictures that share contextual and semantic features. These features correspond to visual concepts and features extracted from convolutional network [Dimiccoli et al., 2017].

In order to exploit the temporal boundaries determined by the event segmentation, we proposed to use an LSTM variant as a temporal learning mechanism. We combined the encoding produced by a CNN with a bidirectional LSTM (BLSTM) [Graves and

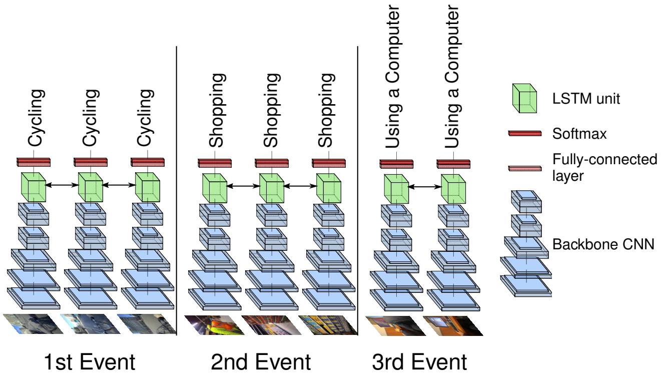


Figure 4.6: Proposed architecture for activity recognition using events segmentation. Event boundaries are determined after clustering a lifelog sequence into subsequences sharing contextual and semantic features. These boundaries are later used on a CNN+BLSTM architecture.

Schmidhuber, 2005]. This recursive neural network evaluates a sequence in forward and backward order and merges the result. Thus, it captures patterns that might have been missed by the unidirectional version and it obtains potentially more robust representations [Chollet, 2017a]. A schematic overview of our approach is illustrated in Fig. 4.6.

4.3 ADLEgoDataset

The purpose of our dataset was to tackle two main issues with existing lifelog datasets. Namely, (*i*) the reduced number of people and lifelog sequences and (*ii*) the small number and diversity of *activity* categories. Both issues prevent thoroughly generalization tests for activity recognition on lifelog sequences.

A comparison summary between the lifelogs datasets presented in section 2.6 and ours is shown in Table 4.1. Besides considering the number of people, annotated classes, and images, it also shows the number of day lifelogs. This number is relevant when performing robust generalization tests on frame-sequence data, as they must be done on full sequences rather than only frames. Moreover, Table 4.1 also highlights the diversity of *activities* of our dataset, since it has more categories belonging to more activity groups proposed by Lara and Labrador [2013].

Dataset	Camera	Body Location	#Frames	#Days	#People	#Classes	#Annotated Frames	Activity Groups
NTCIR-12 Gurrin et al. [2016] Cartas et al. [2018b]	Autographer	Chest	90k	90	3	6 Gurrin et al. [2016]	13.8k	Ambulation, Transportation
						21 Cartas et al. [2018b]	44.9k	Ambulation, Transportation, Device Usage, Daily Activities
NTCIR-13 Gurrin et al. [2017]	Narrative Clip	Chest	**110k**	90	2	4	5.8k	Ambulation, Transportation
NTCIR-14 Gurrin et al. [2019]	Autographer	Chest	81k	43	2	2	8.9k	Ambulation, Transportation
Castro, et al. Castro et al. [2015]	Looxcie	Ear	40k	182	1	17	39.1k	Transportation, Daily Activities, Exercise/Fitness
ADLEgoDataset	Narrative Clip	Chest	105k	**191**	**15**	**35**	**100k**	Ambulation, Transportation, Device Usage, Daily Activities, Exercise/Fitness

Table 4.1: Comparison between existing egocentric lifelogs datasets and the ADLEgo-Dataset. The *activities* are grouped according to the taxonomy proposed by Lara and Labrador [2013]. The highest attribute values are highlighted in bold.

4.3.1 Data Collection

The pictures were collected by 15 computer science post-graduates students the wore a necklace lifelogging camera. The participants were 3 women and 12 men. The collected pictures mainly show distinct outdoor and indoor locations across one city. The common location for all participants was the university where they study or work.

Each participant was instructed to wear the camera while performing their daily activities during whole days. The minimum period for wearing the camera was 10 days. For privacy concerns, they were allowed to cover the camera in private situations, e.g. using the restroom. Additionally, they have permission to discard sensitive pictures, even images from whole days.

The participants wore the first and second versions of the Narrative Clip camera, but only two of them used the first version. Both cameras autonomously take two pictures per minute, but the latter has a wider field of view and an 8 megapixels resolution instead of 5. Their battery charge can last for almost 12 hours, thus allowing them to collect between 1,200 and 1,900 images per day.

The *activity* categories were based on previous works [Cartas et al., 2017a,b, 2018b; Castro et al., 2015], but their purpose was not to model the student lifestyle and were indirectly filtered after the data collection. Specifically, the original cate-

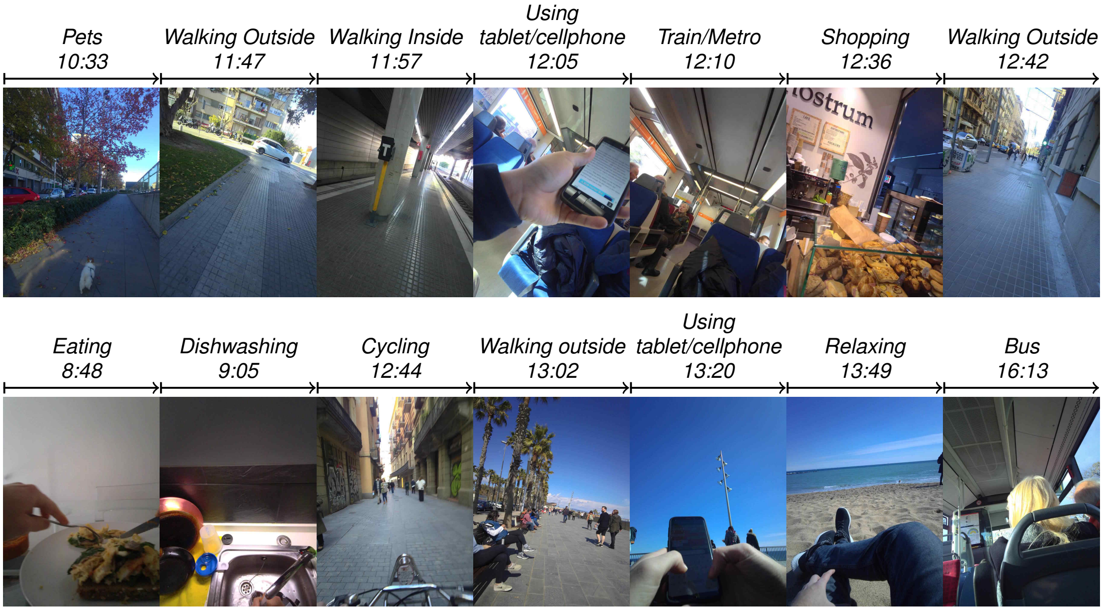


Figure 4.7: Sampled pictures from the ADLEgoDataset. Each row corresponds to annotated pictures from two different people with their respective activity and time.

gories included a broader set of *activities* like *child rearing*, *meditating*, *painting*, or *praying*. Nevertheless, the participants did not choose any of them during the annotation process. The final categories are general *activities* that belong to five distinct groups proposed by Lara and Labrador [2013], as seen in Fig. 4.8.

4.3.2 Annotation Process

Most of the participants were also involved in the annotation process. Their involvement had two goals: (*i*) determine the right *activity* label for their images, and (*ii*) establish when the *activities* started and ended. The temporal context is important when annotating the images since the correct label can be lost when inspecting single frames due to occlusions. For example, a frame in a *cycling* sequence might hide the bicycle handlebar and could lead to a mislabeling like *walking outside*. In order to meet both goals, the pictures were annotated in sequences using our annotation tool introduced by Cartas et al. [2017b].

4.3.3 Dataset Details

The ADLEgoDataset consists of 105,529 egocentric pictures taken from 15 post-graduate students, covering in total 191 days and 35 *activities* . In comparison with egocentric video datasets, its *activities* are not restricted to a specific domain like *kitchen*, and

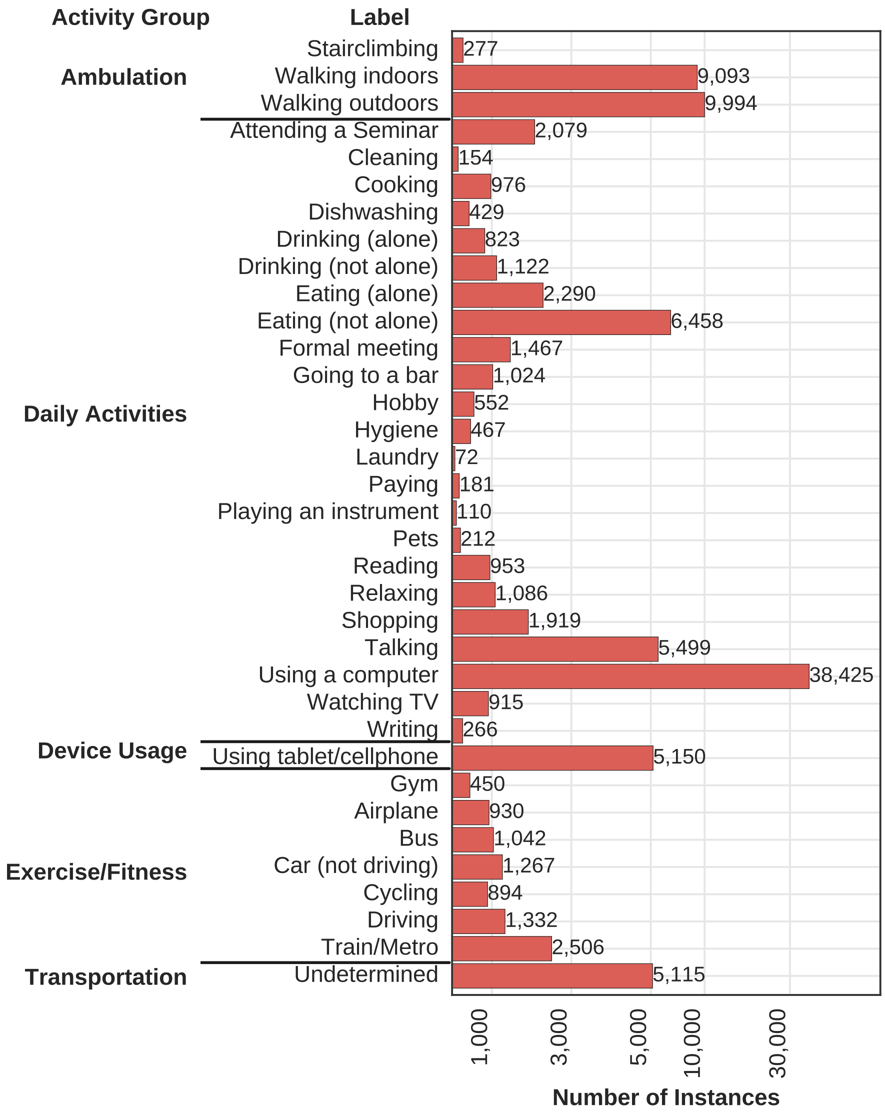


Figure 4.8: Distribution of *activity* categories in the ADLEgoDataset according to the proposed groups by Lara and Labrador [2013]. Note that the horizontal axis has a logarithm scale.

were mainly performed in different outdoor and indoor locations in a city. These *activities* are based on previous works [Cartas et al., 2017a,b, 2018b; Castro et al., 2015] and belong to five of the seven categories introduced by Lara and Labrador [2013], as seen on Fig. 4.8. All participants wore an egocentric camera during a different number of days and times, resulting in 14.6 days and 6,816.73 images per user on average. Fig. 4.7 shows two *activity* sequences of two users in different settings.

4.3.4 Dataset split

Our goal in doing the training and testing partitions was to make possible the evaluation of generalization capabilities of the methods presented in this and the previous chapter. As mentioned in section 3.2, random training/testing partitions are not reliable on sequential data since consecutive frames depicting similar information might be present in both partitions. Therefore, instead of hiding single random frames from the training split, we selected in a test split full-day sequences from *seen users* during training. This selection was made as proportional as possible with respect to the categories since it had to be representative of the dataset. Contrary to the temporal partition of the NTCIR-12 dataset in section 3.2.2, we considered that this kind of partition is not enough to assess the generalization performance, because similar days might depict similar activities in the same context of a person. Consequently, we made another test split consisting of *unseen users* during training. This test split was not constrained to be representative of the training split. The data percentage of the seen and unseen test users was around 10% and 5%, respectively. Moreover, we discarded the *activities* that had less than 200 instances or that were performed by only one user, except for four categories (*airplane*, *cleaning*, *gym*, and *pets*). These categories were also considered for further comparisons in the experiments of the next chapter.

We first created the *unseen users* split because it reduced the complexity of the *seen users* split. The procedure is detailed as follows:

Unseen users split First, we calculated all the possible combinations of unseen users from the 15 participants (i.e. 32,767) by using the Twiddle algorithm [Chase, 1970]. Then we calculated the total number of images for each combination and filtered the ones that did not have between 4.5% and 5% of images from the total amount of images in the ADLEgoDataset. Finally, we selected the combination with the lowest number of participants.

Seen users split This split is focused on separating complete days of images (or *full-day sequences*) from users, rather than separating users. A full-day sequence is composed of several images with different activity labels from one user. The objective of this test split is to separate full-day sequences from the training that maintains a similar category distribution as the whole dataset and thus being representative of what it is intended to learn. We measure the similarity between category distributions using the Bhattacharyya distance.

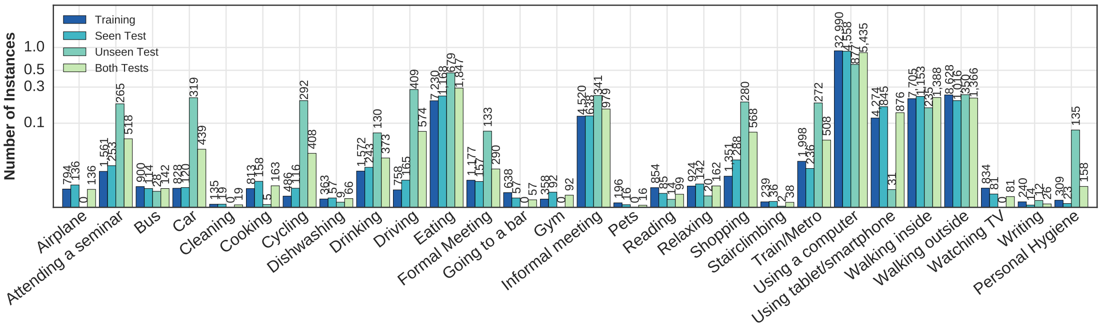


Figure 4.9: ADLEgoDataset splits summary. Our splitting method generated data splits with similar distribution shapes, except for the unseen users split. Note that the distributions are normalized and their vertical axis has a logarithm scale.

After removing the unseen users from the dataset, the remaining number of users is 9 and their number of full-day sequences is 103. By counting the number of images from each full-day sequence, 10% of the dataset for the split is obtained by selecting between 6 and 32 full-day sequences. We considered that the most representative full-day sequences are the ones with the closest category distribution with respect to the whole dataset. Consequently, finding it involves comparing the category histograms between the whole dataset and all possible combinations of full-day sequences. Although the number of test days is low, the search is prohibitively expensive as is characterized by combinatorial growth. For instance, the number of test sets considering 6 days out of the 103 is $\approx 1.42 \times 10^9$, but for 32 days out of the 103 is $\approx 4.42 \times 10^{26}$.

Finding the best full-day sequences for the split was performed as a two-step optimization search using heuristics. With the goal of reducing the search space, instead of dealing with single full-day sequences, we first grouped them into *bins* of one or more full-day sequences. This was modeled as a bin packing problem, where the objects were full-day sequences, and their weight was its number of images. To further reduce the search space by half, these bins were matched in pairs with similar category distributions. The idea is that one bin was destined for the training set and the other for the test set. The resulting number of bin pairs was 32 containing between one and two full-day sequences.

The second step evaluated all test split candidates to find the most similar to the ADLEgoDataset distribution. A test split candidate is a combination of bin pairs that contains all activity categories and its number of images is approximately 10% of the data. The distributions of all our final splits are depicted in Fig. 4.9. It shows that all split distributions have a similar shape, except for the unseen users split because it considered random users as described above.

4.4 Experimental Settings

The aim of our experiments was to measure the effectiveness of our proposed methods. Our experiments were carried out in the NTCIR-12 and the ADLEgoDataset datasets. Concretely, we used the temporal splits described in sections 3.2.2 and 4.3.4. We first measured the performance of all the temporal methods described in section 4.1 in the NTCIR-12 dataset. A more robust comparison was later done over the ADLEgoDataset, but only using the temporal methods that showed good performance. Additionally, the event approach was also tested in the ADLEgoDataset over the same split.

All the models were implemented using the Keras framework [Chollet et al., 2015]. Our optimization algorithm was stochastic gradient descent (SGD). For regularization purposes, we used dropout layers [Srivastava et al., 2014], batch normalization, and early stopping criteria.

4.4.1 Temporal Methods Implementation

The backbone network for the models trained over NCTIR-12 was the VGG-16 architecture [Simonyan and Zisserman, 2014b], whereas the backbone network for the models trained over ADLEgoDataset was the ResNet-50 He et al. [2016]. The main difference in their training is that the weights of the ResNet-50 were frozen during training of the temporal models. We considered data augmentation to randomly applied horizontal flips, translation and rotation shifts, and zoom operations. All the configuration and training details are provided below.

VGG-16 CNN We used the same trained VGG-16 network as previously described in section 3.3. This model served as the initial weights of all temporal models.

VGG-16 CNN+LSTM *Architecture.* We used the same architecture for training using the full lifelog sequence and the sliding window mechanism. Specifically, we removed the last two original FC layers from the VGG-16 network, and instead added an LSTM layer of 256 units followed by an FC layer with 21 outputs.

Training. The initial fine-tuning weights were the ones obtained for the VGG-16 baseline model. For both kinds of training, the first four blocks of convolutional layers remained frozen during the optimization process.

Full-sequence. We trained the CNN+LSTM model for 38 epochs using a timestep input of 10 frames, a learning rate $\alpha = \times 7.5^{-5}$, a momentum $\mu = 0.9$, and weight decay equal to $5 \times 10{-}6$.

Sliding window. We trained three LSTM configurations with a time step of 5, 10, and 15 frames. All the configurations had different learning rates, but the same momentum $\mu = 0.9$ and weight decay equal to $5 \times 10{-}6$. The learning rate α for the time step configuration of 5, 10, 15 was 2.5×10^{-5}, 1×10^{-4}, 1×10^{-4}, correspondingly.

VGG-16 CNN+Piggyback LSTM *Architecture*. The last two FC layers of the VGG-16 network were substituted by a FC layer with a 256 dimension output, an LSTM layer of 256 units, and a FC layer with 21 outputs. A filter layer was used to implement the feedback for the overlapping frames in the sequence. In consequence, the architecture had one additional input for the previous batch and one used as a mask.

Training. The training consisted of two phases. The objective of the first phase was to learn the high-level features from adjacent frames. In this phase, the lifelog sequences were treated without overlapping and the training followed the sliding window mechanism. The goal of the second phase was to learn temporal patterns throughout the sequence, and thus the overlapping of m frames were considered between consecutive input batches. During this phase, all the convolutional layers and the first FC layer were frozen. In order to consider all frames from a sequence, the first n frames of a lifelog were considered as the first frames of independent sequences.

We trained three different configurations of timesteps equal to 5, 10, 15 frames and overlapping equal to 2, 3, and 4, respectively. All the configurations shared the same momentum $\mu = 0.9$ and weight decay equal to $5 \times 10{-}6$, but different learning rate. The learning rate α for the timestep configuration of 5, 10, 15 was 2.5×10^{-5}, 1×10^{-4}, 1×10^{-4}, correspondingly.

VGG-16 CNN+RF+LSTM *Architecture*. Our architecture takes as input the results of a trained CNN+RF model and consists of an LSTM of 32 units followed by an FC layer. With the purpose of performing regularization, dropout layers were added between the connections of the LSTM layer. The five ensembles of CNN+RF for the VGG-16 described in section 3.3 were used. Explicitly, the ensembles combining the layers FC1, FC1+FC2, FC1+FC3, FC2, and FC2+FC3.

Training. For all the baselines ensembles, we trained three configurations with a timestep of 5, 10, and 15 frames. They were trained using the sliding window mechanism described above. The training hyperparameters were a learning rate $\alpha = 1 \times 10^{-2}$, momentum $\mu = 0.9$, and weight decay equal to $5 \times 10-6$,

ResNet-50 CNN We used ResNet-50 He et al. [2016] as CNN network and replaced the top layer with a fully-connected layer of 28 outputs. The fine-tuning considered the class-weighting scheme proposed by King and Zeng [2001]. Moreover, the last ResNet block and the only FC layer were unfrozen. The CNN initially used the weights of a pre-trained network on ImageNetRussakovsky et al. [2015]. It was trained during 7 epochs using a learning rate $\alpha = 1 \times 10^{-2}$, a learning rate decay of 5×10^{-4}, a momentum $\mu = 0.9$, and a weight decay equal to $\alpha = 1 \times 10^{-3}$.

ResNet-50 CNN+LSTM and CNN+BLSTM *Architecture*. These models removed the top layer of the ResNet-50 network and respectively added an LSTM and BLSTM layer having 256 units, followed by a fully-connected layer of 28 outputs.

Training. For both models, the learning rates of the *full sequence* and the *sliding window* training were $\alpha = 1 \times 10^{-2}$ and $\alpha = 1 \times 10^{-3}$, correspondingly. They had the same momentum $\mu = 0.9$, weight decay equal to $\alpha = 5 \times 10^{-6}$, batch size of 1, and a timestep of 5.

ResNet-50 CNN+RF Two random forests were trained using the output of different layers from the previously described ResNet-50 network. Specifically, the first RF was trained using as input the features extracted from the average pooling layer. The second RF uses the average pooling layer plus the concatenation of the FC layer. Additionally, the Convolutional Neural Network Late Fusion Ensemble (CNN LFE) [Castro et al., 2015] was also implemented using as input the softmax probability scores, the day of the week, the time of the day, and 10-bin size histogram for each color channel. The number of trees for all of them was set to 500 and used the Gini impurity criterion Breiman et al. [1984].

ResNet-50 CNN+RF+LSTM and CNN+RF+BLSTM *Architecture*. These models respectively added an LSTM and BLSTM layer having 30 units, followed by a fully-connected layer of 28 outputs.

Training. Both models were trained using as input the prediction of the best *CNN+RF* model, namely the combination of the avg. pooling and the FC layers. The learning rate for both models and types of training was $\alpha = 1 \times 10^{-3}$, a momentum $\mu = 0.9$, weight decay equal to $\alpha = 5 \times 10^{-6}$, batch size of 1, and a timestep of 5.

4.4.2 Event-based Methods Implementation

The following models take into account the event boundaries extracted using the algorithm presented by Dimiccoli et al. [2017]. They use as initial weights the trained CNN and CNN+RF ResNet-50 models. All of them have the same architecture as their counterparts described in the section above. With the purpose of making a fair comparison, their weights and outputs of the backbone models were frozen during training.

ResNet-50 CNN+LSTM and CNN+BLSTM For both models, the learning rates for training were $\alpha = 1 \times 10^{-2}$ and $\alpha = 1 \times 10^{-3}$, correspondingly. They had the same momentum $\mu = 0.9$, weight decay equal to $\alpha = 5 \times 10^{-6}$, batch size of 1, and a timestep of 5.

ResNet-50 CNN+RF+LSTM Both models were trained using as input the combination of the avg. pooling and the FC layers. The learning rate for both models and types of training was $\alpha = 1 \times 10^{-3}$, a momentum $\mu = 0.9$, weight decay equal to $\alpha = 5 \times 10^{-6}$, batch size of 1, and a timestep of 5.

4.5 Evaluation Metrics

The *activity* recognition problem from lifelogs is different from the *action* recognition from videos, as it is naturally posed as many-to-many sequence classification. Accordingly, the evaluation is carried out on the frame level and without any kind of average over the full lifelog sequence. Since the classes of both datasets are imbalanced, using the accuracy as the only classification metric might be misleading. With the purpose of offering a solid performance comparison, our evaluation takes into account the accuracy, the mean average precision (mAP), and macro metrics for precision, recall, and F1-score.

	CNN	CNN+LSTM				CNN+Piggyback		
		Sliding Window				LSTM		
	VGG-16	timestep 5	timestep 10	timestep 15	Day sequence	timestep 5	timestep 10	timestep 15
Accuracy	75.97	79.68	80.39	**81.73**	81.62	75.97	79.04	78.51
Macro Precision	68.50	72.96	75.25	**76.68**	76.03	69.74	72.98	73.00
Macro Recall	67.49	71.36	71.86	74.04	**75.38**	62.98	71.88	69.52
Macro F1-score	66.80	70.87	71.97	74.16	**74.63**	63.24	71.06	69.88

(a) Fully deep temporal methods (CNN+LSTM and CNN+Piggyback LSTM)

	CNN+RF+LSTM														
	FC1			FC1+FC2			FC1+FC3			FC2			FC2+FC3		
	timestep 5	timestep 10	timestep 15	timestep 5	timestep 10	timestep 15	timestep 5	timestep 10	timestep 15	timestep 5	timestep 10	timestep 15	timestep 5	timestep 10	timestep 15
Accuracy	85.96	86.87	85.55	75.86	82.20	81.87	85.45	**87.71**	86.24	85.73	86.78	85.38	85.89	86.85	86.35
Macro Precision	79.81	80.45	80.00	69.97	76.75	74.53	79.29	**81.55**	80.05	81.22	81.37	79.71	79.62	80.22	80.11
Macro Recall	81.36	81.36	79.34	65.19	73.85	73.99	79.11	**81.88**	80.79	80.56	80.97	79.80	80.35	81.19	80.74
Macro F1-score	80.00	80.39	78.45	65.63	74.08	73.62	78.05	**81.10**	79.56	79.96	80.37	78.63	78.82	80.23	79.51

(b) Random Forest based method (CNN+RF+LSTM)

Table 4.2: Comparison of all temporal methods on the NTCIR-12 dataset.

4.6 Results

4.6.1 Temporal Methods over NTCIR-12

The results of our first experiments over the NTCIR-12 using fully deep temporal architectures are presented in Table 4.2a. While the temporal coherence improved the CNN baseline, the models still failed to correctly predict highly correlated categories such as *attending a seminar* and *meeting*, as seen on the confusion matrices on Fig. 4.10.

The best fully deep temporal architectures were the CNN+LSTM, achieving 81.73% of accuracy. Although the best mechanism for training this architecture was the *sliding window*, it was not clear its superiority with respect to training from the full *day sequence*. The results in Table 4.2a show that, despite improving the CNN baseline for two of the three configurations, CNN+Piggyback LSTM architecture had significantly less improvement than the CNN+LSTM architecture. This might be due to not having clear temporal patterns throughout the full day sequences, as shown in Fig. 4.3. Therefore, the CNN+Piggyback LSTM was not further tested in the ADLEgoDataset.

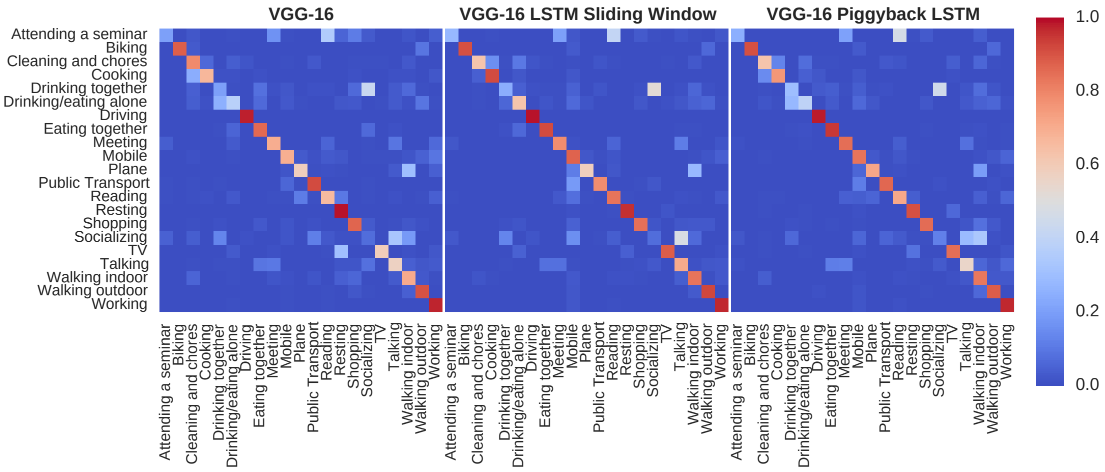


Figure 4.10: Normalized confusion matrices of all models for CNN+LSTM and CNN Piggyback LSTM for the NTCIR-12 dataset.

The results of our CNN+RF+LSTM over the NTCIR-12 are shown in Table 4.2b. In this case, they also confirmed that the temporal coherence helps the *activity* classification. Coincidentally, the best timestep for all models was 10. The best method achieved an accuracy of 87.71% and used the first and last fully connected layers, i.e. FC1+FC3. They showed a higher improvement over their fully deep counterparts methods of Table 4.2a. Although our temporal models obtained a good performance in the NTCIR-12 dataset, the results changed in a more extensive validation by using a more modern convolutional network and a larger scale dataset, as described in the next subsection.

4.6.2 Temporal Methods over ADLEgoDataset

The performance results of all the static and temporal models on the seen and unseen test partitions are presented in Table 4.3. The best models for the seen and unseen test splits were CNN+BLSTM (80.64%) and CNN+LSTM (79.87%), respectively. In both test splits, the *sliding window* training resulted in better performance. Although both models achieved a similar accuracy on the test splits, the rest of the metrics remained significantly different. This indicates that the CNN+BLSTM model suffers from overfitting on unseen users. Overall, the best model for both test splits was the CNN+LSTM achieving an 80.12% accuracy, as it had a similar performance on the seen users split, and better performance on the unseen users split.

In contrast with the results presented in the last subsection, our experiments in-

		STILL IMAGE LEVEL				IMAGE SEQUENCE LEVEL							
		CNN	CNN+RF			CNN+RF+ LSTM		CNN+RF+ BLSTM		CNN+LSTM		CNN+BLSTM	
	Measure	ResNet-50	Avg. pool	Avg. pool+pred.	CNN LFE [Castro et al., 2015]	Day sequence	Sliding window	Day sequence	Sliding window	Day sequence	Sliding window	Day sequence	Sliding window
SEEN USERS	Accuracy	79.46	78.70	78.71	78.40	79.34	78.11	77.61	79.01	79.93	80.23	79.13	**80.64**
	mAP	63.06	66.08	66.05	62.75	64.35	55.79	59.63	63.84	69.74	**70.62**	68.21	69.96
	Macro precision	67.83	67.41	67.59	66.95	66.35	56.49	**73.04**	71.67	68.56	67.91	67.17	68.16
	Macro recall	65.04	60.36	60.50	60.12	58.11	48.78	48.45	57.84	64.44	67.53	62.74	**68.55**
	Macro F1-score	64.22	61.92	62.07	61.72	59.19	49.27	53.51	60.95	64.37	65.60	63.27	**66.85**
UNSEEN USERS	Accuracy	75.44	73.71	73.79	72.51	75.40	74.29	69.00	73.44	77.88	**79.87**	76.00	78.05
	mAP	53.42	55.71	56.05	51.43	52.03	50.28	46.27	51.20	59.02	**62.01**	58.03	59.03
	Macro precision	41.24	48.78	47.98	50.87	52.58	40.41	48.51	**59.75**	55.89	53.01	52.14	48.28
	Macro recall	43.74	43.11	43.23	42.57	46.46	40.61	40.71	46.85	49.47	**49.63**	48.71	46.44
	Macro F1-score	39.52	40.97	40.99	41.22	44.09	37.70	38.28	45.76	47.07	**47.30**	44.94	43.45
ALL	Accuracy	78.30	77.26	77.29	76.71	78.21	77.01	75.13	77.41	79.34	**80.12**	78.23	79.90
	mAP	59.02	62.92	62.97	59.70	61.27	54.71	56.01	60.47	66.52	**67.45**	65.08	66.96
	Macro precision	64.74	66.32	66.39	66.04	66.63	57.36	**72.60**	71.33	67.96	67.79	66.51	67.25
	Macro recall	60.95	56.53	56.67	56.11	54.86	47.33	44.82	53.98	60.97	64.16	59.06	**64.25**
	Macro F1-score	60.80	58.89	58.97	58.62	57.91	48.65	49.75	57.91	61.84	63.71	60.83	**64.01**

Table 4.3: Comparison of all temporal methods on the ADLEgoDataset.

dicate that the CNN+RF models decreased the overall accuracy of the ResNet-50 network. Considering both test splits, the macro precision improved whereas the macro recall decreased. Thus, indicating that the CNN+RF models are confident in their predictions, but they miss a large number of class samples. Consequently, both temporal models trained on top of this configuration (CNN+RF+LSTM and CNN+RF+BLSTM) have a decreasing score in all the considered metrics with respect to the CNN baseline.

The confusion matrices of the best CNN+BLSTM and CNN+LSTM models for the seen and unseen test splits are illustrated in Fig. 4.11. A straight comparison of all classes between each test split cannot be made, as the number of test samples is different and it might be misleading. For instance, not all categories appear on the unseen test split like *airplane* or *watching tv*. Additionally, the proportion of the

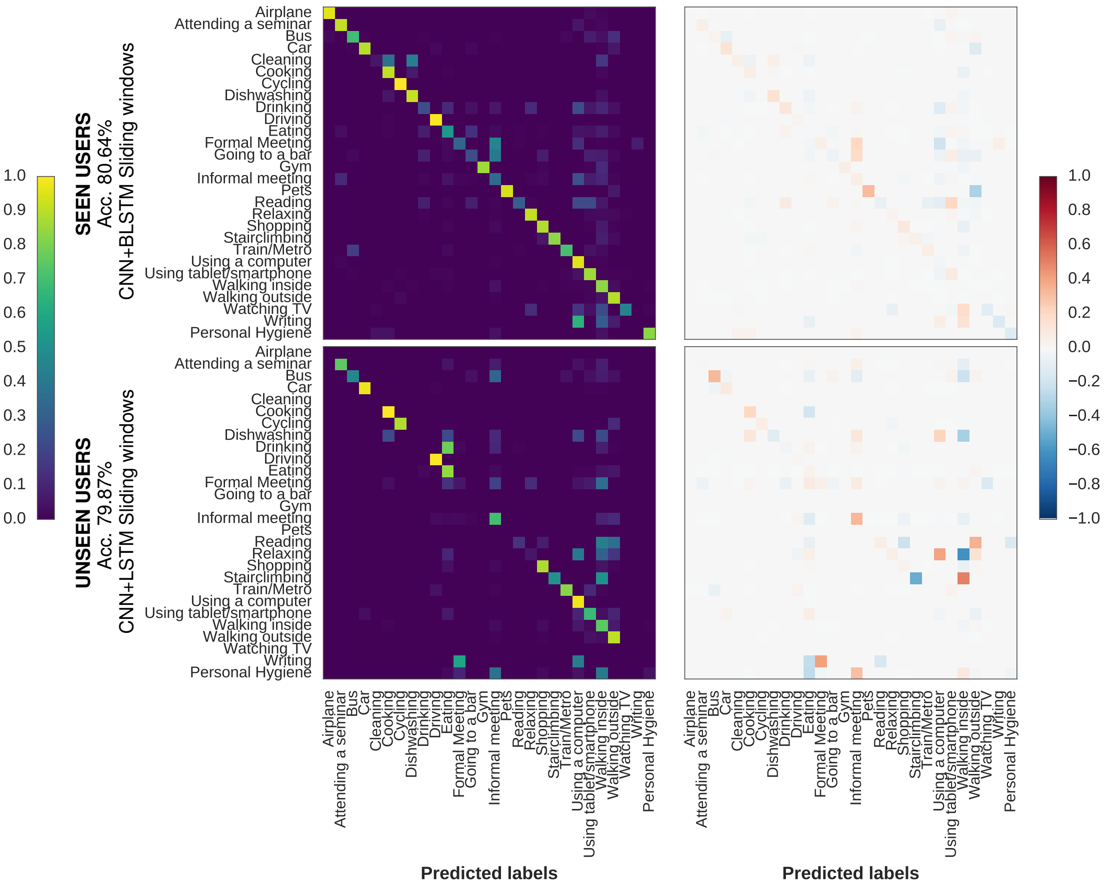


Figure 4.11: Normalized confusion matrices of the best models for the seen and unseen test sets and their difference with respect to the CNN model. The increase and decrease of confidence is represented by the intensity of red and blue colors. Note that the classes *Airplane*, *Cleaning*, *Going to a bar*, *Gym*, and *Watching TV* do not appear on the unseen users test set.

number of test samples is less in some classes, e.g. *stairclimbing*.

Nevertheless, a comparison between the results of each temporal model and the CNN model can be done by calculating their difference, as shown at the right of each confusion matrix row in Fig. 4.11. Since the accuracy improvement with respect to the baseline is higher on the unseen than on the seen test split, there are more changes in its difference. Moreover, the plots show low performance on the CNN model for the categories *Cleaning*, *Relaxing*, *Drinking*, and *Writing*. They might be due to the large intra-class variability of the category (*Relaxing*), the social context ambiguity (*Formal and Informal meeting*), and to the fact that same activities occurs on very similar places (*Cleaning*, *Cooking* and *Dishwashing*).

		STILL IMAGE LEVEL		IMAGE SEQUENCE LEVEL							
		CNN	CNN+RF	CNN+RF+ LSTM		CNN+RF+ BLSTM		CNN+LSTM		CNN+BLSTM	
	Measure	ResNet-50	Avg. pool+pred.	Day sequence	Event Segmentation	Day sequence	Event Segmentation	Day sequence	Event Segmentation	Day sequence	Event Segmentation
SEEN USERS	Accuracy	79.46	78.71	79.34	78.12	77.61	78.00	**79.93**	79.65	79.13	79.38
	mAP	63.06	66.05	64.35	61.44	59.63	60.89	**69.74**	67.47	68.21	67.73
	Macro precision	67.83	67.59	66.35	63.60	**73.04**	71.84	68.56	67.49	67.17	66.42
	Macro recall	**65.04**	60.50	58.11	56.66	48.45	51.80	64.44	62.86	62.74	62.24
	Macro F1-score	64.22	62.07	59.19	56.29	53.51	56.66	**64.37**	63.21	63.27	62.44
UNSEEN USERS	Accuracy	75.44	73.79	75.40	75.65	69.00	69.59	**77.88**	77.37	76.00	75.20
	mAP	53.42	56.05	52.03	51.13	46.27	45.21	**59.02**	58.20	58.03	58.11
	Macro precision	41.24	47.98	52.58	47.18	48.51	48.64	55.89	**57.91**	52.14	55.70
	Macro recall	43.74	43.23	46.46	47.10	40.71	41.30	49.47	49.32	48.71	**53.83**
	Macro F1-score	39.52	40.99	44.09	44.33	38.28	38.81	47.07	47.88	44.94	**48.22**
ALL	Accuracy	78.30	77.29	78.21	77.41	75.13	75.58	**79.34**	78.99	78.23	78.17
	mAP	59.02	62.97	61.27	58.67	56.01	56.99	**66.52**	64.58	65.08	64.30
	Macro precision	64.74	66.39	66.63	63.62	**72.60**	71.31	67.96	66.93	66.51	65.33
	Macro recall	60.95	56.67	54.86	53.57	44.82	47.88	**60.97**	59.05	59.06	58.10
	Macro F1-score	60.80	58.97	57.91	54.95	49.75	52.81	**61.84**	60.91	60.83	59.45

Table 4.4: Comparison between full-day sequences and event segmentation on the ADLEgoDataset.

4.6.3 Event-based Methods

The results of using the event segmentation on the temporal methods are shown in Table 4.4. Although its difference with respect to the methods trained using the full *day sequence* is small, its performance is lower. The best method overall is the CNN+LSTM using the full day sequence in both seen and unseen partitions. In comparison with the results presented using *sliding window* in Table 4.3, the performance using event segmentation is significantly lower. This might be explained for two main reasons. First, although the segmented events have consistent groups of sequential images, they still contain different *activities* . Second, the *sliding window* exploits the temporal neighborhood of frames and, thus, having a smoothing effect over the sequence.

4.7 Conclusions

In this chapter, we introduced three methods that exploit the temporal coherence of lifelogs and a method based on event segmentation. All our methods used features extracted from a CNN and LSTM units as a temporal mechanism. We first presented a training mechanism over lifelogs named *sliding window* and two architectures: CNN+RF+LSTM and CNN+Piggyback LSTM. The *sliding window* training mechanism generates consecutive training batches of a fixed size from a lifelog sequence. Continuing the work from the last chapter, the CNN+RF+LSTM architecture sets a random forest before the temporal mechanism. Our last temporal architecture is the CNN+Piggyback LSTM, that it is able to handle information of previous batches from a sequence by reprocessing a fixed number of overlapping frames. Then we presented an event-based approach that extracts temporal boundaries from temporal segments sharing semantic and contextual information from lifelogs.

Additionally, we presented the ADLEgoDataset, the so-far largest egocentric lifelog dataset of activities of daily living consisting of 105,529 annotated images. It was recorded by 15 different participants wearing a Narrative Clip camera while performing 35 activities of daily life in a naturalistic setting during a total of 191 days. With respect to other available lifelog datasets, it contains many more categories, annotated images, users, and types of activities, hence allowing to perform generalization tests on unseen users.

We first tested the three temporal approaches over the NTCIR-12 dataset. These initial tests showed that training a CNN+LSTM in a *sliding window* fashion had a better performance than considering the lifelog as a full day sequence, achieving 81.73% accuracy. In the other hand, the CNN+Piggyback LSTM showed an improvement over the CNN baseline by achieving a 79.04% accuracy, but it did not represent an improvement over the CNN+LSTM architecture. Our last temporal architecture, the CNN+RF+LSTM model, obtained the best performance overall achieving 87.71% accuracy.

Since the initial tests were done in a small dataset containing a few users, they were validated in the ADLEgoDataset. Moreover, this validation was carried out using a more modern convolutional network architecture, namely ResNet-50; because the features extracted from the VGG-16 proved not to be as robust as more modern architecture as reported in the last chapter. The results confirmed that the best training mechanism for lifelogs was the *sliding window* using a CNN+LSTM or a CNN+BLSTM.

The best overall method was the CNN+LSTM obtaining an 80.12% of accuracy, but the CNN+BLSTM had a better performance in the seen users partitions having an 80.64%. They also showed that the CNN+RF overfit when trained on a wider context having more users and images, and thus its temporal counterpart the CNN+RF+LSTM model. Finally, the event-based approach did not prove to be as effective as training with the full day sequence. This might be a consequence of the LSTM having a smoothing effect over neighbor frames in a sequence and the segmented events not defining strict *activity* boundaries.

Chapter 5

Domain Adaptation Applications for Activity Recognition on Lifelogs

Contents

5.1 Domain Adaptation Using a Regularization Loss 69

5.2 Source and Target Datasets 70

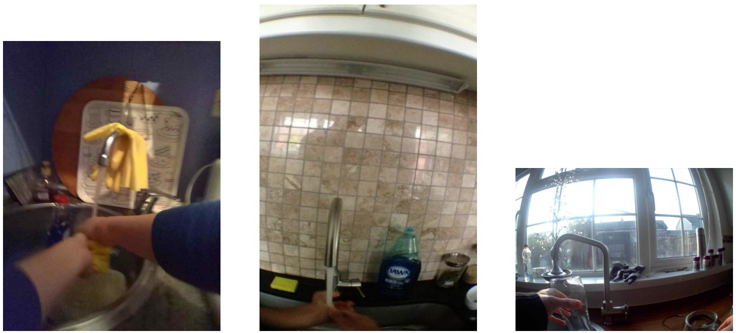
5.3 Implementation 72

5.4 Results 75

5.5 Conclusions 78

In real-world lifelogs applications, a system pretrained on a large scale dataset is commonly deployed on new unseen visual data. In these scenarios, the distribution of the training is also called *source*, and typically differs from the distribution of new data, also called *target*. For instance, this is always the case when the target data are acquired by a different wearable camera than the source data, or when the target data have been collected by people having a very different lifestyle than those who collected the source data, i.e. having different jobs/hobbies and living in different countries. In addition, new data can be unlabeled or scarcely labeled. Therefore, ensuring performance on unseen users from the same domain does not assure that the model could be employed in real-world applications. In addition, to guarantee the robustness of the method, performance should keep stable on larger and more varied datasets.

As a visual example, Fig. 5.1 shows egocentric images of different people *washing the dishes* in their respective houses captured with three different wearable cameras. Besides the visual variability of tap and sinks in different kitchens, one can notice the contrast of fields of view and the angle distortion produced by different lenses. Due to



ADLEgoDataset
Narrative Clip 2
Chest mounted

Castro et al.
[Castro et al., 2015]
Looxcie
Ear mounted

NTCIR-12
[Gurrin et al., 2016]
Autographer
Chest mounted

Figure 5.1: Examples of people performing the same *activity* from different domain datasets. Below each image is its corresponding dataset, egocentric camera type, and body wearing location.

the different nature of the source and target domains, performance on the target domain typically experiences a drop.

In this chapter, we aim at mitigating the performance drop by applying a semi-supervised learning technique, namely domain adaptation (DA). Our goal is to assess the performance between egocentric domains with and without transfer learning, rather than proposing a new adaptation method tailored at egocentric lifelog sequences. Therefore, we strictly focus on a simple image-based DA method, the Deep Correlation Alignment (CORAL) regularization loss proposed by Sun and Saenko [2016].

We perform two experiments using the ADLEgoDataset as the source domain, and the NTCIR-12 [Gurrin et al., 2016] and Castro et al. [2015] datasets as target domains. These datasets were selected as target domains, as they are the closest to our dataset in the number of activity categories and annotated images, and were recorded with different cameras, as it can be appreciated in Table 4.1. In the first experiment, we measure the performance of adding annotated images from different domains for training without using DA, and we quantify the difference between the target and the source domains. In the second experiment, we use the CORAL loss function as DA method on the target datasets and calculate the amount of labeled target data needed to achieve a good classification performance. Specifically, we consider it in a semi-supervised context, where different amounts of labeled target examples are taken into account.

This chapter is organized as follows. In Section 5.1, we detail the domain adapta-

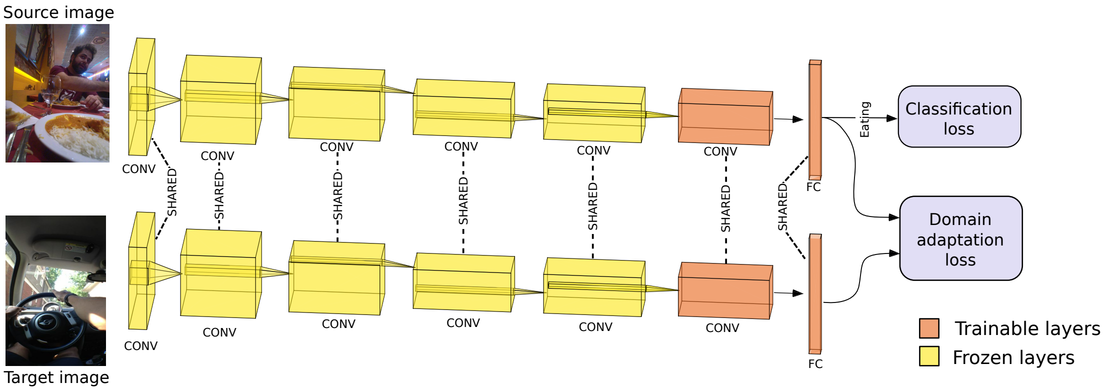


Figure 5.2: Domain adaptation training pipeline. During training, two CNNs with shared weights are used for the source and target data domains, respectively. Since the target domain labels are unknown, only the classification loss for the source CNN is evaluated. The adaptation from the source to the target domain comes from penalizing the discrepancy of their predictions using the domain adaptation loss. In this example, the discrepancy of both images should be high, because the source and target images correspond to the classes *eating* and *driving*.

tion technique we used, namely the CORAL regularization loss. Next, in Section 5.2 we outline the datasets we used and their splits in our experiments. In Section 5.3, we thoroughly describe the implemented models. The experimental results evaluation and discussion are presented in Section 5.4. Finally, we present our conclusions in Section 5.5.

5.1 Domain Adaptation Using a Regularization Loss

Let $L_S = \{y_i\}$, $i \in \{1, ..., L\}$ be the labels from the source domain, and let us assume that the target domain has only unlabeled examples. During training, both domains have their own CNN architecture with shared weights, but only the source domain has a classification loss ℓ_{CLASS}. In order to adapt the learned model from the source to the target domain, a regularization loss ℓ_{DA} is used. This domain regularization loss penalizes the discrepancy between the output distributions from two single feature layers having a dimension d. This is a common setting used by Long et al. [2015]; Sun and Saenko [2016]; Zellinger et al. [2019] and it is illustrated in Fig. 5.2, where a single DA loss is penalizing the output of the fully-connected (FC) layers. The training loss function can be expressed as:

$$\ell = \ell_{CLASS} + \sum_{i=1}^{n} \lambda_i \ell_{DA} \tag{5.1}$$

where n is the number of DA regularization layers in the network and λ denotes the hyperparameter that trades off the adaptation with classification accuracy. Since our CNNs only had one FC layer, we only used one DA loss.

Specifically, we used the CORAL regularization loss Sun et al. [2016]; Sun and Saenko [2016]. One of its advantages is that only the hyperparameter λ requires to be set. In this context, the output features of the source and target layers are said to come from the source domain $\mathcal{D}_S = \{\mathbf{x}_i\}$, $\mathbf{x} \in \mathbb{R}^d$ and the target domain $\mathcal{D}_T = \{\mathbf{u}_i\}$, $\mathbf{u} \in \mathbb{R}^d$, respectively. Then the CORAL regularization loss can be defined as:

$$\ell_{CORAL} = \frac{1}{4d^2} \| C(\mathcal{D}_S) - C(\mathcal{D}_T) \|_F^2 \tag{5.2}$$

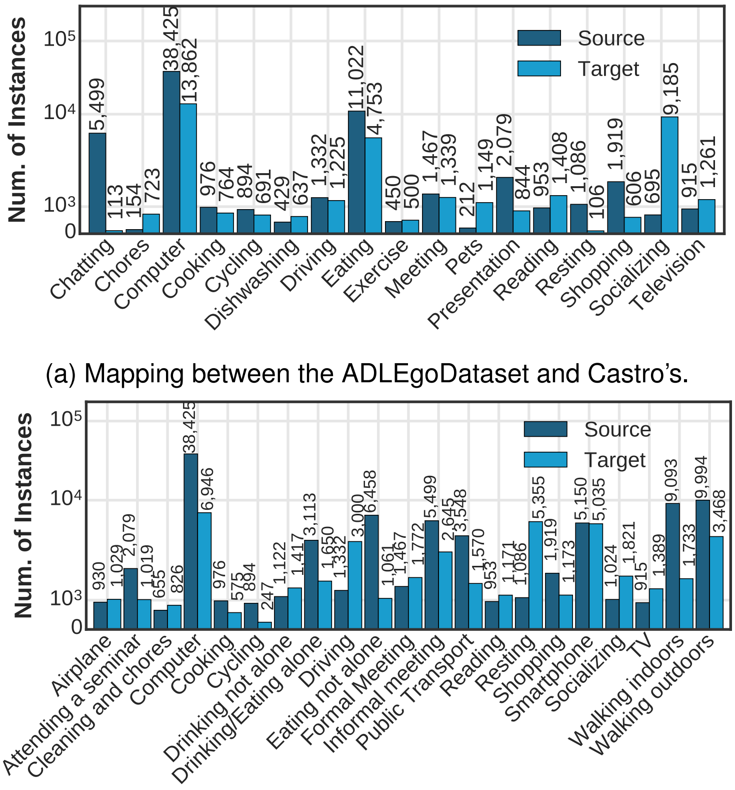
where $\| \cdot \|_F^2$ denotes the squared matrix Frobenius norm and C is the covariance of $\mathcal{D}$ given by:

$$C(\mathcal{D}) = \frac{1}{m}(\mathcal{D}^\top \mathcal{D} - \frac{1}{m}(1^\top \mathcal{D})^\top (1^\top \mathcal{D})) \tag{5.3}$$

where m is the number of data in the domain $\mathcal{D}$ and 1 is a column vector with all elements equal to 1. The CORAL loss penalizes the discrepancy between domain features, so that when the source and target images correspond to different classes the penalty is high.

5.2 Source and Target Datasets

In our experiments, we used the ADLEgoDataset as the source domain dataset, and the NTCIR-12 [Gurrin et al., 2016] and Castro et al. [2015] as target domain datasets. Both datasets were selected as target domains since they used different cameras and have more annotated categories and images than other lifelogging datasets, as can be appreciated in Table 4.1. Additionally, the domain visual difference with respect to our dataset can be appreciated in Fig. 5.1. We did not consider using the NTCIR-12 Gurrin et al. [2016] and Castro's datasets as source domains since they have fewer people, half of the images, and fewer activity categories. Since their labels correspond to a different set of activity categories than ours, we manually mapped the matching categories. More categories would have required an automatic matching between words. The resulting categories and data distributions are shown in Fig. 5.3. The corresponding number of images of the source and the target for the NTCIR-12 was 96,632 and



(a) Mapping between the ADLEgoDataset and Castro's.

(b) Mapping between the ADLEgoDataset and NTCIR-12.

Figure 5.3: Categories mapping between the source and target domains with their data distributions. The source domain corresponds to the ADLEgoDataset, whereas the target domains are the (a) Castro's dataset and the (b) NTCIR-12 dataset. Note that the vertical axis has a logarithm scale.

44,902, and for the Castro's dataset was 68,507 and 39,166. The specific data splits for each experiment are detailed below.

Training without Domain Adaptation The goal of this experiment was to measure the performance of adding images from two different domains only during training without using DA. Therefore, we combined the source dataset with each target dataset for the training and validation splits, but the testing split only considered images from the source domain. Explicitly, we used the same splits for the source images as described in Section 4.3.4. The images from the target domains were randomly stratified in a 90/10% proportion for the training/validation splits.

Domain Adaptation on the Target Datasets The objective of this experiment was to (i) use transfer learning in a practical setting and (ii) determine the required amount of labeled data from the target domain to obtain a good classification performance. The initial setting of the experiment considered that only the source domain data was labeled, but later different proportions of labeled data from the target domain were added.

First, we randomly stratified the source data into training and validation sets, and the target data into training and testing sets. Throughout the experiment, the proportion of training and validation data of the source images was fixed and set to 90/10%, whereas the proportion of training and testing data of the target images was initially set to 85/15%. Subsequently, different proportions $(10, 20, \ldots, 50\%)$ of images were randomly and incrementally removed from the target training split. These images were added to the training/validation splits of the source domain while maintaining their original 90/10% proportion.

5.3 Implementation

The following paragraphs describe the training settings for each experiment.

Training without Domain Adaptation We used a ResNet-50 [He et al., 2016] network as a CNN model and replaced its top layer with a FC layer of 28 outputs. In order to have comparative results with the classification baseline presented in the last chapter, we explicitly used the same network. It was trained using stochastic gradient descent (SGD) with its weights initialized on ImageNet [Russakovsky et al., 2015]. The last ResNet block and the FC layer were unfrozen during fine-tuning procedure. The training parameters were a learning rate $\alpha = 1 \times 10^{-2}$, a learning rate decay of 5×10^{-4}, and a momentum $\mu = 0.9$. Since we used two validation splits, the training was stopped when their epoch losses were not further improved. The number of epochs for the target datasets NTCIR-12 and Castro's one were 6 and 9, respectively.

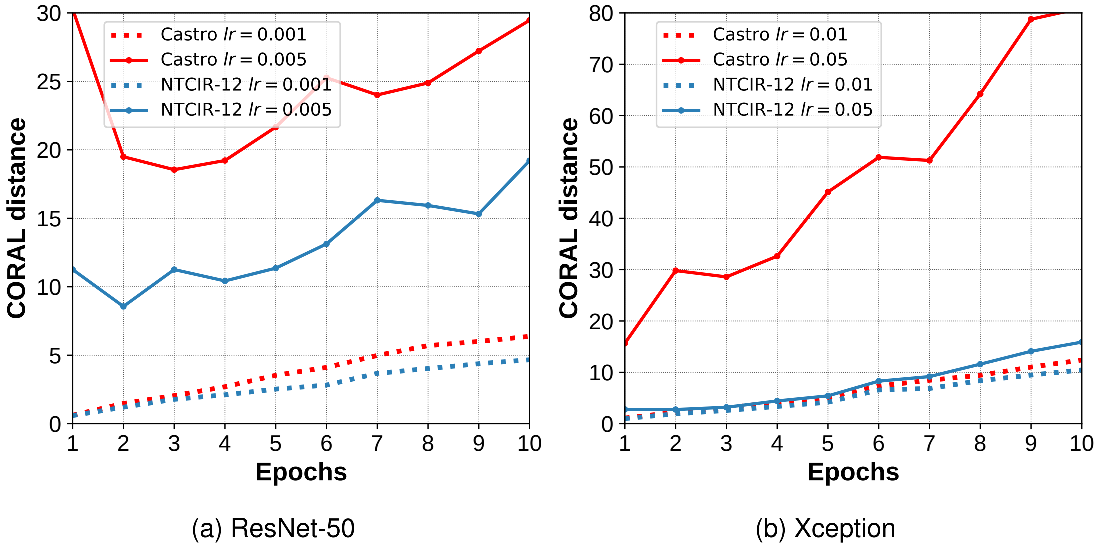
Domains discrepancy As a means to quantify the difference between the source dataset and the target datasets, we calculated the maximum mean discrepancy (MMD) [Fortet and Mourier, 1953] between them for each shared category. First, we sampled between 500 and 1,000 images per category that were both in the source and the target datasets. These sampled images took into account all users and all days. Then, for each sampled image, we extracted a feature vector from the last pooling layer of a ResNet-50 CNN pre-trained on ImageNet [Russakovsky et al., 2015]. Finally, we calculated the MMD between the sets of feature vectors of the source and target datasets using a Gaussian kernel with a $\sigma = 0.1$.

Domain Adaptation on the Target Datasets We initially used two CNN architectures, i.e. Xception [Chollet, 2017b] and ResNet-50, as they are more robust and have better performance than AlexNet [Krizhevsky et al., 2012], the original network used by Sun and Saenko [2016]; Long et al. [2015]; Zellinger et al. [2019].

Architecture setup In comparison with AlexNet, the Xception and ResNet-50 architectures have only one FC layer, making it the only layer suitable for the CORAL loss. The weights of this FC layer were initialized with $\mathcal{N}(0, 0.005)$ and its learning rate was set ten times bigger than the other layers, as stated by Sun and Saenko [2016]. The rest of the layers were initialized using pre-trained weights on ImageNet [Russakovsky et al., 2015]. We initially kept frozen all the layers except the classification layer, but it had a negative impact on the performance in the target domain. Hence, the layers from the last ResNet block of the ResNet-50 architecture and the exit flow block of the Xception architecture were unfrozen. We used SGD as an optimization method for both networks.

Learning rate α tuning We experimentally found that an *adequate* learning rate α had to be high enough to produce a significant CORAL distance between the source and the target domain, but not so high that it did not converge. In order to find it, we first varied the learning rates while maintaining the other parameters constant and setting $\lambda = 0$. In other words, the training was performed without penalizing the discrepancy between domains, but measuring their distance. For instance, Fig. 5.4 illustrates significantly different CORAL distances for two different learning rates on both target datasets. In both cases, the highest learning rates were used as their training converged. Additionally, in our experiments, the lower learning rate did not produce higher accuracy scores for the training split of the target domain.

The final training parameters for ResNet-50 were a learning rate of $\alpha = 5 \times 10^{-3}$, a batch size of 60, a momentum equal to 0.9, and a weight decay equal to $5 \times 10 - 4$. Additionally, the training parameters for the Xception network were a learning rate of $\alpha = 5 \times 10^{-2}$, a batch size of 40, a momentum equal to $\mu = 0.9$, and a weight decay equal to 5×10^{-4}.



(a) ResNet-50

(b) Xception

Figure 5.4: Sensitivity of the CORAL distance due to different learning rates using (a) ResNet-50 and (b) Xception network. These results were obtained by only measuring the CORAL distance and not penalizing it (i.e. by having a fixed $\lambda = 0$).

CORAL loss weight λ tuning After finding an adequate learning rate, we trained the ResNet-50 and Xception networks for $\lambda = 0, 0.1, \ldots, 1$. The best value of λ was obtained considering only the highest validation accuracy of the source domain, as the target data is supposed to be unknown. The best values of λ for ResNet-50 were 0.3 and 0.5 on the NTCIR-12 and Castro's datasets, respectively; whereas the best values of λ for Xception were 0.5 and 0.2 on the NTCIR-12 and Castro's datasets, correspondingly.

The validation accuracy plots for ResNet-50 and Xception networks on both datasets are shown in Fig. 5.5. Two observations can be made from these plots. First, the areas between the minimal and maximal values of the accuracy obtained using the different values of λ suggest that the training of Xception network is more unstable than the ResNet-50 network. Consequently, no further experiments were implemented using the Xception network. Second, the difference between the target accuracy of both datasets ($\approx 73.22\%$ for Castro and $\approx 47.92\%$ for NTCIR-12) shows that a good performance is not always achieved using the CORAL loss alone. Therefore, more data from the target domain is needed to be labeled during training.

Addition of target labeled data to the source domain After *fine-tuning the hyperparameters*, we separately trained the ResNet-50 network adding different percentages of random target labeled data to the source domain. The considered percentages of tar-

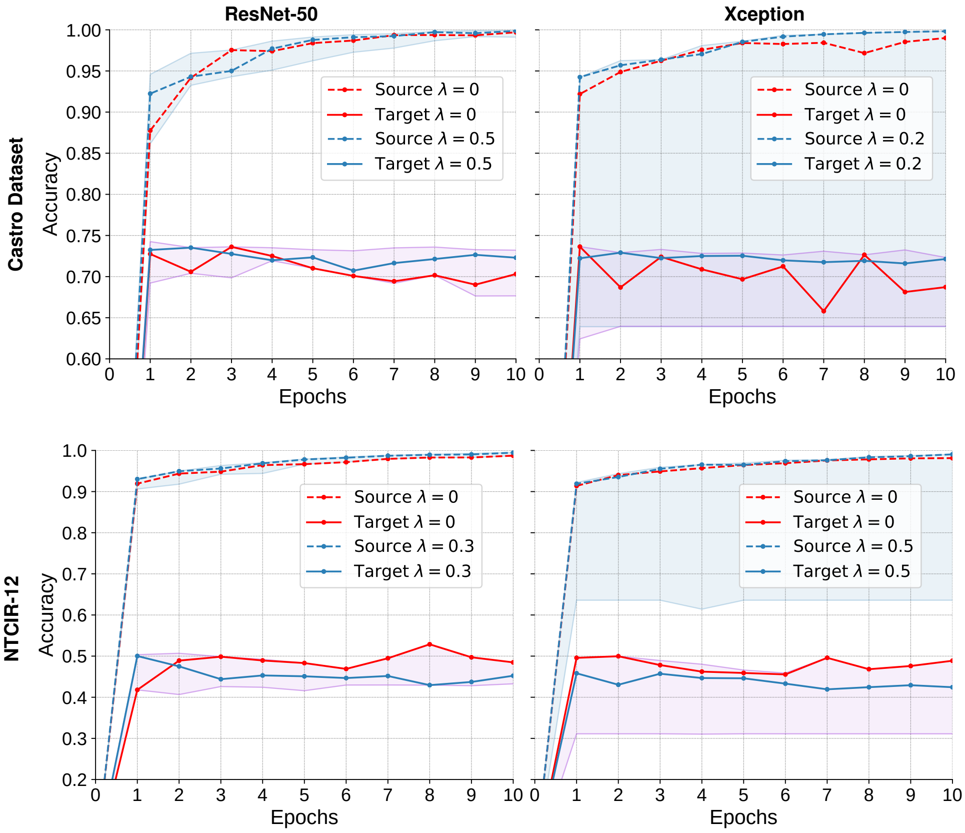


Figure 5.5: Validation accuracy for the ResNet-50 and Xception on the transfer learning to the Castro's dataset (top) and the NTCIR-12 dataset (bottom). Each plot shows the validation accuracy obtained with the best value of λ and without using domain adaptation ($\lambda = 0$). Additionally, the **blue** and **violet** areas represent the range between the minimal and maximal values of the accuracy for $\lambda = 0.1, 0.2, \ldots, 1.0$ on the source and target domains, correspondingly.

get data were $0, 10\%, \ldots, 50\%$ and were selected as described in Section 5.2.

5.4 Results

Training without Domain Adaptation The objective of this experiment was to (i) measure the activity classification performance when mixing the source and target datasets during training without DA method and (ii) estimate how different were the source and target domains. Given the class imbalance present in the dataset and for comparative purposes, we used the same performance metrics as the experiments presented in Section 4.5. The discrepancy between shared categories of the source and target domains was calculated using the MMD as described in Section 5.3.

	SEEN			UNSEEN			ALL		
Measure	ADLEgoDataset	ADLEgoDataset+Castro	ADLEgoDataset+NTCIR-12	ADLEgoDataset	ADLEgoDataset+Castro	ADLEgoDataset+NTCIR-12	ADLEgoDataset	ADLEgoDataset+Castro	ADLEgoDataset+NTCIR-12
Accuracy	**79.46**	67.87	67.44	**75.44**	58.95	55.02	**78.30**	65.31	63.87
mAP	**63.06**	43.08	43.03	**53.42**	36.99	37.59	**59.02**	40.17	40.08
Macro precision	**67.83**	47.93	54.31	**41.24**	31.50	32.21	**64.74**	45.69	49.99
Macro recall	**65.04**	52.34	45.71	**43.74**	30.61	33.79	**60.95**	47.40	41.97
Macro F1-score	**64.22**	48.71	45.80	**39.52**	28.59	27.89	**60.80**	44.67	41.09

Table 5.1: Activity classification performance results obtained by adding the Castro's [Castro et al., 2015] and NTCIR-12 [Gurrin et al., 2016] datasets for training without domain adaptation. The best result per measure is shown in bold. Note that not all categories appeared on the unseen users test set.

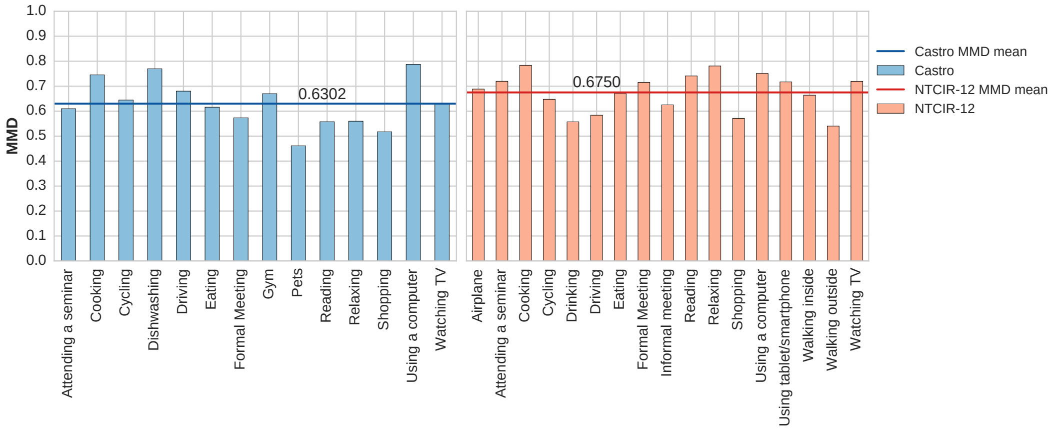


Figure 5.6: Maximum mean discrepancy (MMD) between the categories from the source and target datasets. The closer the value to zero the more similar the domains for that category.

The classification results of separately adding Castro's and NTCIR-12 datasets for training are presented in Table 5.1. It shows that the addition of labeled data from the target domains diminished all the evaluated performance metrics; in particular, the accuracy was lower by 13.71% on average. The overall classification performance of adding Castro's dataset was better than when adding the NTCIR-12 dataset. This is also reflected in their calculated discrepancy with respect to the source domain. Fig.

Domain shift	Percentage of random target data used during training											
	0%		10%		20%		30%		40%		50%	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Castro	58.34±0.2	72.47	46.70±0.4	75.58	78.75±0.0	79.04	79.50±0.0	79.73	**80.07±0.0**	81.18	65.83±0.3	**81.88**
NTCIR-12	45.21±0.0	45.14	77.85±0.0	78.46	82.60±0.0	81.81	84.91±0.0	84.90	87.34±0.0	87.40	**87.77±0.0**	**87.76**

Table 5.2: Action recognition accuracy for the domain shifts from the ADLEgoDataset dataset using ResNet-50.Best result per measure is shown in bold.

5.6 shows the MMD for each shared category between each target and source domains, and a horizontal line representing its mean. The MMD mean for the Castro's dataset is lower than for the NTCIR-12 dataset, thus meaning that it is more similar to the ADLEgoDataset. This difference in discrepancy also is reflected in the performance of domain adaptation as describe below.

Domain Adaptation on the Target Datasets The objective of this experiment was to use transfer learning in a practical setting and to determine the required amount of labeled data from the target domain to obtain a good classification performance. As in previous works [Sun and Saenko, 2016; Donahue et al., 2014; Ganin and Lempitsky, 2015; Gong et al., 2012; Long et al., 2015; Zellinger et al., 2019], we use the prediction accuracy as evaluation metric for five different training runs. Our results only consider ResNet-50 architecture, since the training of the Xception network was unstable as discussed above. The summarized results are shown in Table 5.2 and plot in Fig. 5.7.

The results in Table 5.2 show that ResNet-50 was also susceptible to instability during training, producing a high variance in some training runs. This instability only affected Castro's dataset and can be visually seen in the plot of Fig. 5.7. Therefore, the accuracy median was also considered to measure performance improvement.

The results confirm that performing domain adaptation without using labeled target data does not necessarily achieve a good performance on all target datasets. Specifically, the median accuracy of the NTCIR-12 was 45.14% whereas for Castro's dataset was 72.58%. This low performance was improved by adding a small subset of labeled target data to the training. The largest increment in median accuracy was obtained by adding 10-20% of labeled data, i.e. for the NTCIR-12 it improved by 33.32% when adding 10% and for Castro's dataset it improved by 6.57% after adding 20%. The

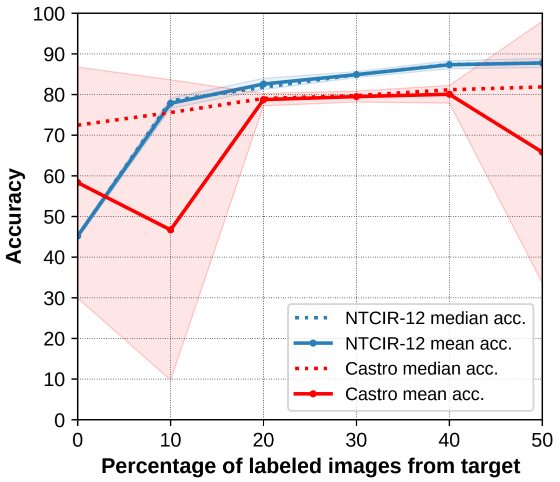


Figure 5.7: Mean test accuracy with respect to different percentages of added target labeled images to the training/validation source data using ResNet-50. The colored areas denotes the mean value ± standard deviation.

most benefited dataset was the NTCIR-12 since their initial discrepancy was higher as shown by the previous experiment. The mean and median accuracy curves from Fig. 5.7 show a decreasing increment that settles around 40%. Although a straight comparison with previous works cannot be made [Castro et al., 2015], the mean accuracy values at 40% of added data are competitive. Originally, the accuracy obtained for Castro's and NTCIR-12 datasets were 83.07% and 94.08%, correspondingly.

5.5 Conclusions

In this chapter, we evaluated the performance of trained methods in new data for real scenarios. Our aim was to mitigate the performance drop by applying a semi-supervised learning technique, namely domain adaptation (DA). Specifically, we presented experiments of generalization in different domains. We first showed that the evaluated source and target datasets have a large discrepancy that diminished the classification performance by 13.71% on average. Finally, we used the CORAL loss function as a DA technique and showed that a good performance is not always achieved on different target datasets. Specifically, we obtained a median accuracy value of 72.47% and 45.14% on Castro's and the NTCIR-12 datasets. We also showed that the performance can improve by incorporating a small percentage of labeled target data to the training. In the case of the NTCIR-12 dataset, the performance improved to 78.46% by randomly adding 10% of target data.

Part II

Egocentric Action Recognition From Videos

Chapter 6

Region-Based Action Recognition From Egocentric Videos

Contents

6.1 Proposed approach 83

6.2 Datasets 91

6.3 Implementation details 92

6.4 Evaluation 93

6.5 Results 95

6.6 Conclusions 99

In this chapter we propose a semantically driven approach to classify *actions* in egocentric videos that learns first-person object interactions at frame level and the dynamics of *actions* at temporal level. In the context of this chapter, we refer to an *action* as a short-term human-object interaction such as *close fridge*, consisting of an action verb (i.e. close) and an interacting object (i.e. fridge). Similarly, we say that an *activity* indicates a sequence of *actions* such as *take bread*, *take cheese*, *open ketchup*, etc. performed with a specific goal, i.e. *making a sandwich*.

Our approach aims at reasoning about different semantic entities in images across space and/or time since it is able to learn relevant image regions at frame-level and the dynamics of *actions* at temporal level. At frame-level, we model first-person object interactions as spatially structured regions in a deformable star configuration, consisting of primary and secondary regions corresponding to the user hands and to the objects being potentially interacted with, respectively (see Fig. 6.1).

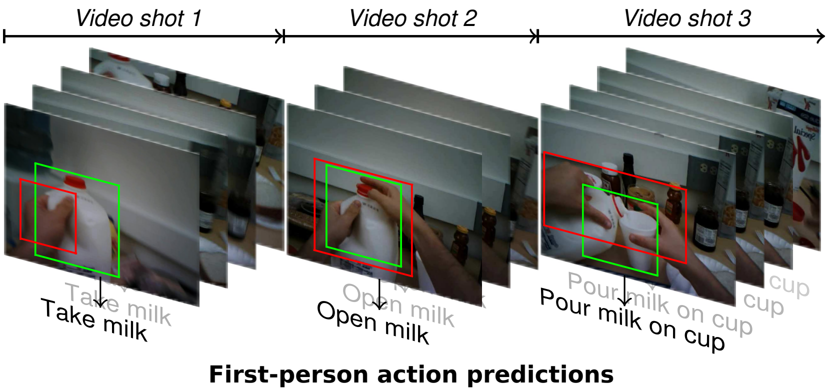


Figure 6.1: Our model takes as input for each frame a primary region corresponding to the user hands (red boxes) and a set of secondary regions corresponding to object proposals (green boxes). These regions are used to make action predictions at frame level, that are aggregated firstly at shot level and then across shots to capture the temporal structure of human-object interactions.

At temporal-level, we model first-person activities as sequences of *actions* and we aim at exploiting temporal structure to improve predictions. Our temporal model is based on the observation that when different people perform an *activity* with a particular goal, i.e. *make tea*, the temporal order matters. For instance the *action poor milk in cup* is typically preceded by *take milk*, *open milk*, etc.

As stated in chapter 2, several approaches to first-person object interactions have been presented. They can roughly be grouped according to their used egocentric features like hand location and pose [Bambach et al., 2015; Fathi et al., 2011a], head motion and gaze [Fathi et al., 2012], manipulated objects [McCandless and Grauman, 2013; Pirsiavash and Ramanan, 2012; Baradel et al., 2018], or as a combination of them [Li et al., 2015; Ma et al., 2016; Singh et al., 2016; Sudhakaran and Lanz, 2018]. However, only a few of them [Fathi et al., 2011a; Baradel et al., 2018; Sudhakaran and Lanz, 2018] are specifically aimed at reasoning about different semantic entities in images across space and/or time. Our approach follows the same direction by capturing object-interactions using regions and modeling their temporal structure. A comparison of all these approaches and ours is presented in Table 6.1.

We evaluated our approach on three public datasets: the GTEA [Fathi et al., 2011b], Gaze [Fathi et al., 2012], and the GAZE+ [Li et al., 2015] benchmarks. Quantitative comparisons to five state-of-the-art methods demonstrated both the effectiveness and the potential of our approach, which is able to outperform current methods even with-

Method	Hands	Objects	Motion	Gaze	Spatio-temporal feature modelling	Spatio-temporal reasoning	Spatial reasoning on hand-objects	Temporal reasoning on actions
Fathi et al. [2011a]	✓	✓	✓		✓			✓
Fathi et al. [2012]	✓	✓	✓	✓	✓			
Pirsiavash and Ramanan [2012]		✓	✓		✓			
McCandless and Grauman [2013]	✓	✓	✓		✓			
Bambach et al. [2015]	✓							
Li et al. [2015]	✓	✓	✓	✓				
Ma et al. [2016]	✓	✓	✓		✓			
Singh et al. [2016]	✓	✓	✓		✓			
Sudhakaran and Lanz [2018]		✓	✓			✓		
Baradel et al. [2018]		✓	✓			✓		
Ours	✓	✓					✓	✓

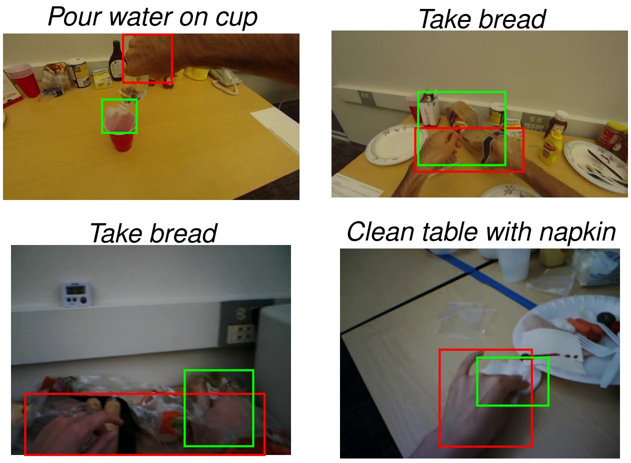
Table 6.1: State of the art egocentric object-interaction recognition methods and their characteristics

out relying on motion information. Unlike most available architectures, where motion is processed through a dedicate stream, in our approach, we explicitly model the dynamics of human-object interaction at temporal-level.

The reminder of this chapter is organized as follows. First, in section 6.1 we introduce our proposed approach and data augmentation strategies. Then in section 6.2 we detail the datasets we used, namely the GTEA, GAZE, and GAZE+. Next in sections 6.3 and 6.4 we describe the implementation details and evaluation settings of our model, respectively. In section 6.5 we throughly discuss our ablation studies and benchmark results. Lastly, in section 6.6, we draw our conclusions.

6.1 Proposed approach

The proposed approach aims to model: 1) the contextual cues that characterize person-to-object interactions, 2) the temporal evolution of contextual cues within a shot, and 3) the temporal order of actions. In section 6.1.1 and section 6.1.2 we detail how the proposed approach models and exploits contextual information of actions and the temporal order that characterizes activities, respectively.



Pour water on cup

Take bread

Take bread

Clean table with napkin

Figure 6.2: Examples of primary and secondary regions. On top of each image it is described its corresponding *action* . The primary and secondary regions are highlighted in red and green boxes, respectively.

6.1.1 Frame-Level Modeling

In order to capture contextual cues of *actions* , we adapted the R*CNN framework proposed by Gkioxari et al. [2015b] for action recognition in still images captured by a third-person camera. This approach takes as input a manually selected primary region containing the person whose action has to be predicted, and a set of secondary regions that are supposed to capture contextual information such as objects the person is interacting with, the pose of the person or what other people in the image are doing.

The primary region is manually selected, whereas the set of candidate secondary regions are provided by using object proposal methods such as selective search [Uijlings et al., 2013], constrained parametric min-cuts (CPMC) [Carreira and Sminchisescu, 2012], and multiscale combinatorial grouping (MCG) [Pont-Tuset et al., 2015]. The convolutional network automatically estimates the secondary region, among those provided automatically, that helps to describe an action the most.

In our egocentric setting, the model is defined by a previously detected primary region corresponding to hands plus a set of parts regions corresponding to manipulated objects with associated CNN features.

Primary region detection Since the person-to-object interactions are mainly done by manipulating the objects with the hands, we focus on the area covering the hand starting from the wrist to the tip of its middle finger. We define a primary region as

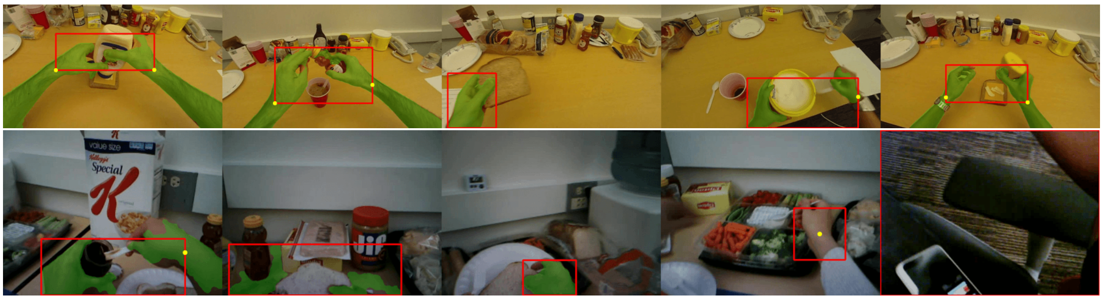


Figure 6.3: Examples of primary regions extraction taken from random frames from the GTEA and GAZE+ datasets. The green-colored regions show detected skin pixels, the yellow dots are the wrist center locations, and the red bounding box shows the located primary region.

the single area covered by one or both hands of the person. Fig. 6.2 depicts four examples where the primary regions detected with the proposed approach are shown in red bounding boxes.

Our method determines the primary region based on skin regions and wrists located points. First, we perform skin pixel detection using the algorithm proposed by Li and Kitani [2013]. The training examples for this method are images containing visible arms and their respective skin masks. This method trains a random forest for each training example using a cluster from a set of local appearance features, such as different colorspaces (RGB, HSV, LAB) or spatial descriptors (SIFT [Lowe, 2004], HOG [Dalal and Triggs, 2005], BRIEF [Calonder et al., 2010]). The skin classification is done by globally combining the output of each random forest using a probabilistic model. We restricted the skin pixel detector to trained using only colorspaces.

The location of the wrists is performed using the parts affinity fields (PAFs) model introduced by Cao et al. [2017]. This CNN model has two branches that separately predict confidence maps for body part detection and part affinity fields. The PAFs encode the orientation and location of limbs over the image. Both kinds of confidence maps are associated using a bipartite graph matching algorithm. The results are body poses for each person on the scene. Specifically, we only used the confidence maps generated for body part detection, since other body joints are not visible and their association cannot be made.

Once the skin regions and wrist location points were calculated, the primary region is determined as follows. If no skin regions and wrist location points are found, then the primary region is considered to be the full-frame (see the last column in the second row of Fig. 6.3). When no wrist points are located, then the complete skin region is determined to be the primary region. This case usually happens when a small part of the hand is seen (see the third column on the first row Fig. 6.3). In case, where no skin regions are found, then a fixed area near the wrists is considered as the primary region (see the fourth column of the second row on Fig. 6.3). Finally, when both wrists points are located and skin regions are found, then wrists points determine the right, left and lower sides of the primary region, while the skin determine its upper side (i.e. first two columns of the first row of Fig. 6.3).

Secondary regions prediction In contrast to Gkioxari et al. [2015b], where secondary regions are defined as those that overlap the primary region, our set of secondary regions coincides with the set of proposed regions. The reason is that often while performing an action as *take*, *poor water* etc., the hands are not in contact with the object during the full duration of the shot. Considering the performance results obtained by Hosang et al. [2014], we compute region proposals by using MCG [Pont-Tuset et al., 2015] instead of Selective Search [Uijlings et al., 2013] as originally presented by Gkioxari et al. [2015b].

Action prediction The score function implemented by the R*CNN architecture corresponds to those of a latent SVM formulation. In a latent SVM, each example x is scored by a function of the following form

$$f_w(x) = \max_{z \in Z(x)} w \cdot \phi(x,z),$$

where w is a vector of model parameters, z are latent values, and $\phi(x,z)$ is a feature vector.

In our particular case, given an image x and the primary region p in x containing the hands, the score for the action α is defined as

$$f(\alpha; x, p) = w_p^\alpha \cdot \phi(x,p) + \max_{z \in Z(p,x)} w_z^\alpha \cdot \phi(x,z) \tag{6.1}$$

where z is the latent variable representing a secondary region corresponding to an object, $Z(p,x)$ is the set of candidates for the secondary region (object proposals), w_p^α and w_z^α are the model parameters.

The probability that given the hand p on the image x, and the action being performed α, is computed using softmax as in Gkioxari et al. [2015b].

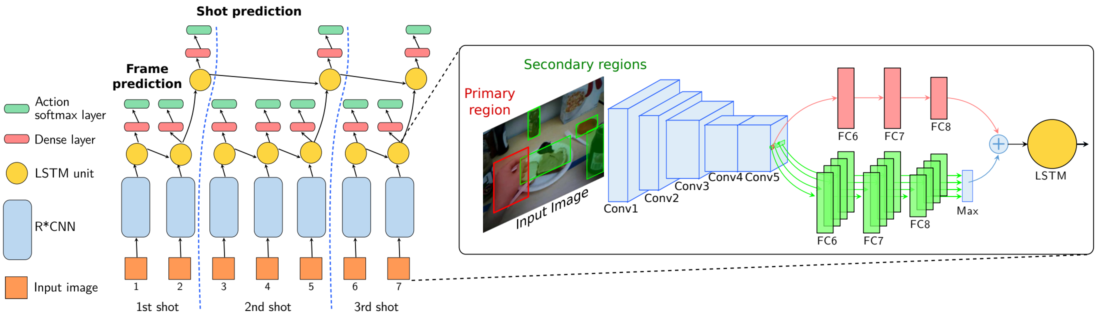


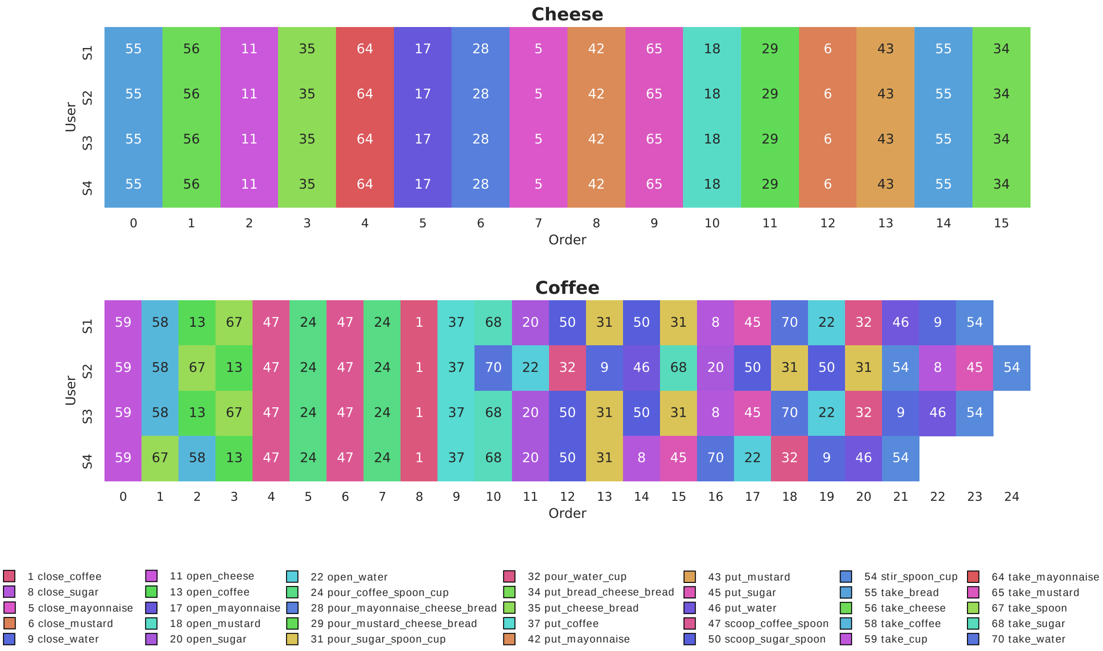
Figure 6.4: Overview of our proposed region-based architecture. The left part shows the hierarchical action recognition pipeline across different video shots. The output of the last LSTM unit of each shot in the first level serves as input for an LSTM unit of the second level of the hierarchy. The right part shows the frame level processing, the primary (red), and secondary regions (green) boxes are previously estimated and passed as input to the network. The R*CNN architecture is applied to each frame within the video shot. The *action* prediction is measured at frame-level, as the output of the softmax of the first layer of LSTM.

The feature vector $\phi(\cdot)$ and the vectors of model parameters, w_p^α and w_z^α, are learned jointly for all actions using a CNN in an end-to-end fashion (see Fig. 6.4).

6.1.2 Temporal modeling

Our architecture models sequences of human-object interactions using a hierarchical training structure. Variations between successive frames in a video encode useful information to make more accurate action predictions. A standard way to account for such variations is to feed the output of a CNN to an LSTM in an end-to-end fashion. Consequently, at the bottom level of our hierarchy, we use a many-to-many LSTM to explicitly capture the temporal coherence of action probabilities within a video shot. Every LSTM unit at time step i takes as inputs the action probabilities returned by R*CNN for the i-th frame and the hidden state from the previous LSTM unit. Additionally, each LSTM output is connected to a dense layer and a softmax to output action probabilities that account for these of previous frames (see Fig. 6.4). The top-level of our hierarchy captures a long time temporal dependencies across *actions* . Specifically, the LSTM output of the last frame of each shot is fed to another LSTM unit, which also takes as input the hidden state of the LSTM units of the previous shot.

The loss function of our complete hierarchical LSTM (HLSTM) architecture is as



Cheese

User	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	55	56	11	35	64	17	28	5	42	65	18	29	6	43	55	34
S2	55	56	11	35	64	17	28	5	42	65	18	29	6	43	55	34
S3	55	56	11	35	64	17	28	5	42	65	18	29	6	43	55	34
S4	55	56	11	35	64	17	28	5	42	65	18	29	6	43	55	34

Order

Coffee

User	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
S1	59	58	13	67	47	24	47	24	1	37	68	20	50	31	50	31	8	45	70	22	32	46	9	54	
S2	59	58	67	13	47	24	47	24	1	37	70	22	32	9	46	68	20	50	31	50	31	54	8	45	54
S3	59	58	13	67	47	24	47	24	1	37	68	20	50	31	50	31	8	45	70	22	32	9	46	54	
S4	59	67	58	13	47	24	47	24	1	37	68	20	50	31	8	45	70	22	32	9	46	54			

Order

Figure 6.5: Two different action sequences for all users from the GTEA dataset.

follows:

$$\mathcal{L} = (1-\beta)\mathcal{L}_N + \beta\mathcal{L}_M \tag{6.2}$$

where β $(0 \leq \beta \leq 1)$ is a hyper-parameter used to trade-off between two objectives and $\mathcal{L}_N$ and $\mathcal{M}_2$ are the cross-entropy loss defined at frame-level and shot-level, respectively. They are defined as follows:

$$\mathcal{L}_N = \sum_{i=1}^{N} L(y^i, \hat{y}^i), \ \mathcal{L}_M = \sum_{j=1}^{M} L(y^j, \hat{y}^j) \tag{6.3}$$

where N is the total number of frames, M is the total number of shots, y^i is the ground truth action at the $i-$th frame and $\hat{y}^i$ is the action prediction, accordingly. Similarly, y^j is the ground truth action at the $j-$th shot and $\hat{y}^j$ is the action prediction.

6.1.3 Data augmentation

We performed visual and temporal data augmentation in two separate ways. The visual data augmentation rotates the video frames while maintaining the temporal consistency between consecutive ones. The proposed algorithm tries to mimic the head movements of the camera user. The goal of the temporal data augmentation was to extend the sequences of actions in a logical way. The tasks for the camera users in all the evaluated datasets were cooking/preparing something in a kitchen or table. Since

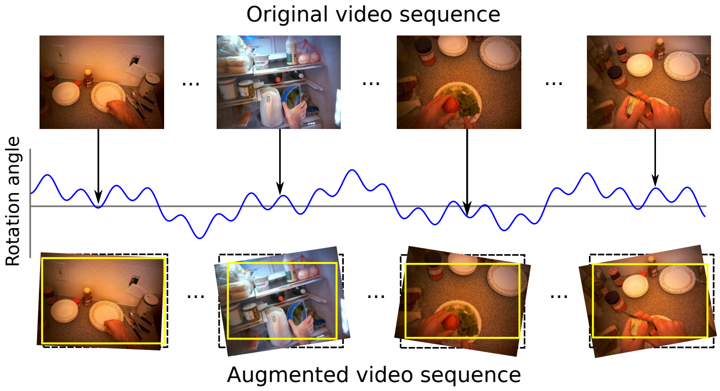


Figure 6.6: Visual data augmentation process.

a cooking recipe can be followed by doing its steps in different order and time, it can have slightly different action sequences as long as it preserves the logical order, as seen in the original sequences shown in Fig. 6.5. For instance, the original order of three actions for a sequence named *making a ham sandwich* could be *take ham*, *take bread*, and *put ham on bread*, but it could be augmented by changing the order of the first two actions. Both data augmentation techniques are detailed in the following paragraphs.

Visual data augmentation This data augmentation process consisted in rotating the frames of all videos in a continuous manner as illustrated in Fig. 6.6. Given a video from the dataset containing N frames and a uniformly sampled random number $r \in \left[0, \frac{1}{2}\right]$, the rotation angle $\theta(n)$ for the $n-th$ frame is calculated using the following pair of equations:

$$\rho(n) = 2\pi\left(\frac{C}{N}n + r\right) \tag{6.4}$$

$$\theta(n) = \Theta_{max}\sum_{i}^{M}\lambda_i \sin\gamma_i\rho(n) \tag{6.5}$$

where the constants C, γ_i, and $\sum_i^M \lambda_i = 1$ determine the sinusoidal behavior of θ within a period. The purpose of the function $\theta(n)$ is to generate continuous rotation angles for each sequence in the range $[-\Theta_{max}, \Theta_{max}]$.

After rotating a frame n by the angle θ, then its largest inscribed rectangle is cropped. This rectangle represents the augmented version of the $n-th$ frame. The width w_r and height h_r of this rectangle is calculated as follows. Let $s = \min(w,h)$ and $l = \max(w,h)$ denote the value of the shortest and longest side respectively.

Original sequence	Grouped actions	Swapped/moved	Skipped	Addition
1. Turn on burner	1. Turn on burner	1. Open fridge	1. Turn on burner	1. Take oil container
2. Take oil container	1. Take oil container	2. Take carrots	2. Take oil container	2. Open oil container
3. Open oil container	2. Open oil container	3. Close fridge	3. Open oil container	3. Turn on burner
4. Pour oil on pan	3. Pour oil on pan	4. Turn on burner	4. Pour oil on pan	4. Pour oil on pan
5. Close oil container	4. Close oil container	5. Take oil container	5. Close oil container	5. Close oil container
6. Open fridge	1. Open fridge	6. Open oil container		6. Open fridge
7. Take carrots	2. Take carrots	7. Pour oil on pan		7. Take carrots
8. Close fridge	3. Close fridge	8. Close oil container		8. Close fridge

Figure 6.7: Examples of sequence data augmentation. The first and second columns show the original sequence and its grouped action subsequences. The last three columns show three distinct combination operations.

If $\frac{s}{l} > 2 \cdot |\sin\theta||\cos\theta|$

$$w_r = \frac{w|\cos\theta| - h|\sin\theta|}{\cos 2\theta} \tag{6.6}$$

$$h_r = \frac{h|\cos\theta| - w|\sin(\theta)|}{\cos 2\theta} \tag{6.7}$$

else if $w > h$

$$w_r = \frac{s}{2|\sin\theta|}, \quad h_r = \frac{s}{2|\cos\theta|} \tag{6.8}$$

otherwise

$$w_r = \frac{s}{2|\cos\theta|}, \quad h_r = \frac{s}{2|\sin\theta|} \tag{6.9}$$

The primary region previously obtained is rotated with respect to its new center location. Afterward, a new bounding box is computed using the rotated corners. The secondary regions are calculated again for the augmented version of the $n-th$ frame by using the MCG algorithm. This process duplicated our training data.

Sequence data augmentation The temporal sequences of the datasets are augmented in a two-step process. First, a meta-sequence is manually created for each video sequence in the dataset. This meta-sequence defines all the possible logical combinations that the sequence can have. Second, an alternate sequence is randomly generated from its meta-sequence during training.

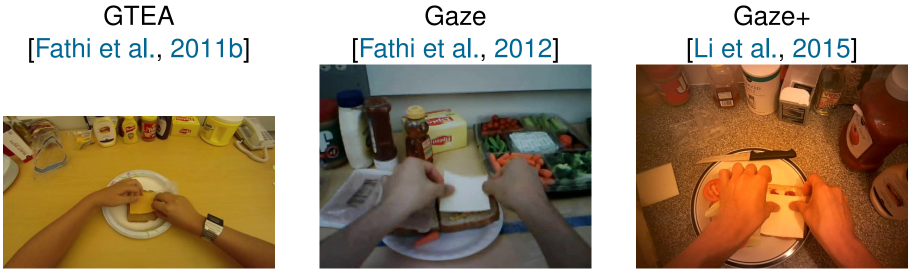
In order to create a meta sequence, groups of related actions are created along with their possible combinations. The groups can be combined using three different operations: they can be swapped/moved, skipped, or added to one another. For example, Fig. 6.7 shows the start of a cooking sequence with three groups of actions shown in different colors. Additionally, it also shows three combinations using each operation, but more combinations can be found by mixing them.

6.2 Datasets

Our experiments were done using the GTEA [Fathi et al., 2011b], GTEA Gaze [Fathi et al., 2012], and the GTEA GAZE+ [Li et al., 2015] datasets. In these datasets, all the subjects in the videos manipulated the same objects in the same place, thus preserving the same conditions through all videos. Their description is provided in the next paragraphs and they are illustrated in Fig. 6.8.

GTEA The GTEA dataset consists of 21 videos acquired by 4 different subjects while performing scripted activities such as making a *cheese sandwich*, an *instant coffee*, *sweet tea*, and a *peanut butter sandwich*. Each activity contains sequences of actions that are freely performed by the subject. For instance, the activity *making coffee* starts with the following action sequences: *take cup*, *take coffee*, *open coffee*, *take spoon*, and *scoop coffee with spoon*. The dataset includes two subsets of 71 and 61 actions.

Gaze The GTEA Gaze dataset consists of 17 videos from 14 different subjects following unscripted food recipes. The dataset has two subsets of 25 and 40 actions, such as *open milk* or *take carrot*. Both action subsets have the same fixed training and test data splits as presented by Li et al. [2015].


GTEA
[Fathi et al., 2011b]

Gaze
[Fathi et al., 2012]

Gaze+
[Li et al., 2015]

Figure 6.8: Examples of people performing the same *action* (*put cheese on bread*) from all the evaluated datasets.

Gaze+ The GAZE+ dataset has 37 videos of 7 activities recorded by 6 participants. The subjects were asked to perform some activity with the available objects. The total number of evaluated actions is 40. Some examples of the activities include *prepare a Greek salad* and *making cheese burger*, whereas some examples of actions are *put milk container* and *open fridge drawer*.

6.3 Implementation details

The following paragraphs describe the implementation details of our proposed approach.

Primary regions As stated in section 6.1.1, our method estimates the primary region by detecting skin regions and locating wrists points. The skin pixel detector presented by Li and Kitani [2013] was trained for each video of every dataset using skin masks. For datasets not including the skin mask, a different number of frames were uniformly sampled from each video and the skin contour was annotated using LabelMe [Russell et al., 2008]. In the case of the GTEA dataset, we used only the 814 skin masks provided. For the Gaze dataset, an average of 37 frames per video were annotated, giving a total of 625 skin masks. Lastly, 2,230 skin masks were annotated for the GAZE+ dataset, thus having an average of 60 annotated frames per video. The wrists points were detected by using the PAFs network [Cao et al., 2017]. This network was originally trained using the COCO 2016 keypoints challenge dataset [Lin et al., 2014], which contains over 100K person instances for training. Specifically, we only used the confidence maps generated for part detection and one scale for the wrist point detection for each dataset.

Secondary regions We used the MCG method for computing the object proposals for all datasets. This method was pre-trained on the PASCAL 2012 [Everingham et al., 2015] and the BSDS500 [Arbelaez et al., 2011] segmentation datasets. This method generated an average of 1,484 proposals per video frame.

Visual Data augmentation We visually augmented the training data for all our experiments following the description of section 6.1.3. The value of the angle of rotation was in the set $[-7^\circ, 7^\circ]$. This process roughly duplicated our training data.

Frame-level modeling We fine-tuned the R*CNN model on top of a pretrained VGG-16 network [Simonyan and Zisserman, 2014b] on ImageNet for each dataset. Moreover, all the fully-connected layers were fine-tuned. The optimization procedure used the stochastic gradient descent (SGD) with a step learning policy. The step decay learning rate was $\gamma = 1e^{-1}$ for every 30K iterations. Depending on the dataset, the learning rate α was between $1e^{-4}$ and $2e^{-4}$ with a momentum $\mu = 9e^{-1}$ and a batch size of 10. We randomly selected 10 regions from the set of proposal regions as described by Gkioxari et al. [2015b], but without considering overlapping thresholds with respect to the primary region. All the models were trained between 60K and 90K iterations.

Temporal modeling We trained our hierarchical model for all datasets by using the SGD optimization method and the resulting weights of the previous frame-level modeling stage as initialization. The output of the softmax for each frame is provided as input to the HLSTM model. We first trained the model with $\beta = 0$ and then we used the resulting weights as initialization for the training with $\beta > 0$. The model with $\beta = 0$ was trained by using a step decay learning rate $\gamma = 1e^{-1}$ for every 30K iterations. Its learning rate α was between $1e^{-4}$ and $2e^{-4}$, depending on the dataset, a momentum $\mu = 9e^{-1}$, and a batch size of 6. The optimization was performed between 30K and 60K iterations. For $\beta > 0$, we trained the model between 5 and 8 epochs depending on the dataset by using a step decay learning rate $\gamma \approx 9.5e^{-1}$, and starting with a learning rate $\alpha = 5e^{-5}$, a momentum $\mu = 9e^{-1}$, and a batch size of 6.

Since the second LSTM layer captures the sequence of actions across the videos, during test all the video frames are fed sequentially to the network, and the prediction at frame-level of the first layer of the LSTM is used to measure the performance. Thus, shot boundary information coming from the second LSTM layer is not needed for testing in our hierarchical model.

Sequence Data Augmentation We perform sequence data augmentation only for the GAZE+ dataset on a specific ablation study, and the comparison with other methods did not include it.

6.4 Evaluation

We evaluated the effectiveness of our trained model on the action recognition task. Since our model only uses the temporal structure of the entire video sequence during

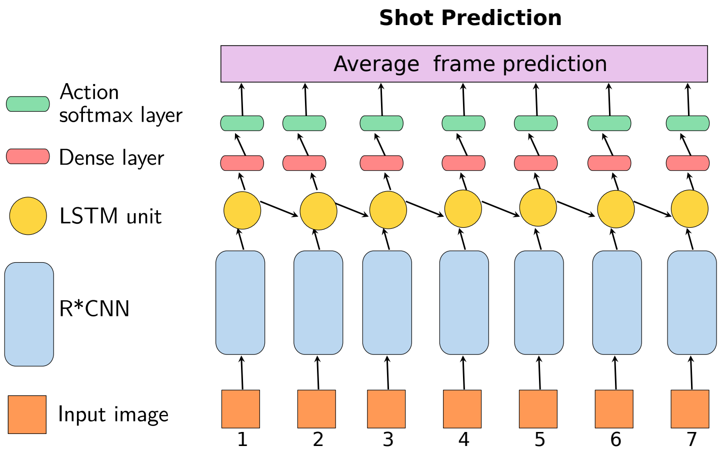


Figure 6.9: Diagram of the test architecture. We only used the first level LSTM to determine the action of every single video shot. The final prediction is the average prediction of all processed frames.

training, the evaluation was carried out considering individual video shots and only the first level LSTM. Thus we are using the same given information for testing as previous works on first-person action recognition [Simonyan and Zisserman, 2014a; Wang et al., 2016; Ma et al., 2016; Sudhakaran and Lanz, 2018]. Fig. 6.9 illustrates the configuration used during the testing of our model. The performance measurements were done using the frame average accuracy score per sequence. In addition, other methods that used the accuracy score at the frame level are indicated on the reported results.

Method	**GTEA 61***	**GTEA 71**	**GAZE 25***	**GAZE 40***	**GAZE+**
Fathi and Rehg [2013]†	39.7	-	-	-	-
Li et al. [2015]†	66.8	62.1*	60.9	39.60	60.50
Two-Streams Network Simonyan and Zisserman [2014a]	57.64	49.65	-	-	58.77
Temporal Segments Network Wang et al. [2016]	67.76	67.23	-	-	55.25
Ma et al. [2016]	75.8	73.24	62.40	43.42	**66.40**
Sudhakaran and Lanz [2018]	**77.59**	**77**	-	-	60.13
VGG-16 Baseline †	54.67	48.95	43.44	40.76	49.83
Ours (frame level)	68.97	64.74	56.94	47.25	52.75
Ours (1 level LSTM)	69.83	71.04	63.89	49.45	58.41
Ours (Hierarchical LSTM)	70.69	72.95	**65.28**	**52.75**	59.96

Table 6.2: Action recognition accuracy for different methods on the GTEA, Gaze, and Gaze+ datasets. Note: * fixed split, † accuracy measured at frame level.

	GTEA 61	GTEA 71	GAZE 25	GAZE 40	GAZE+
Avg. Levenshtein distance	2.36	2.57	15.12	19.40	50.40

Table 6.3: Similarity between sequences of actions in videos for each dataset. The closer the value to zero the more similar.

6.5 Results

Our main results are presented in Table 6.2. They are further commented along other ablation studies in section 6.5.1 and a comparison is draw in section 6.5.2.

6.5.1 Ablation Studies

Component Analysis On the last rows of Table 6.2 we report performances obtained by using different modules of the proposed architecture. The module that performs spatial reasoning alone determines a consistent improvement with respect to the VGG baseline (13.79 %). The first LSTM layer is responsible for an increase of 8.19 % of per frame accuracy on average on all datasets. The second LSTM layer is responsible for a further increase of 2.69 %. It is worth to stress that the hierarchical model has a much smaller training set compared to the single layer LSTM, which explains the relatively small improvement in performance. Moreover, the dissimilarity between the sequences of actions in the datasets explains the difference in performance improvement among them. We measure the similarity as the minimal cost of transforming one sequence into the other using the Levenshtein distance [Levenshtein, 1966], as shown in Table 6.3. The more similar the action sequences are, the higher the accuracy improvement. Since we work with action probabilities instead of image features, our model could benefit of the training of other datasets or action sequences augmentation techniques.

To better quantify the impact of our frame-level model, we compared its performance to those of the VGG-16 network [Simonyan and Zisserman, 2014b] pretrained on ImageNet, since it serves as the basis for the R*CNN architecture. We observed a considerable improvement of performance of 24.70% on average.

The effect of the hierarchical training over the first level LSTM can be seen using different values of β. First, we trained our model using a $\beta = 0$ (i.e. only the first LSTM level) until the optimization process converged. We then performed a grid search varying the learning rate α and the hyperparameter β on a single split of each dataset. In particular, we first found a suitable learning rate α, and then we looked for the best

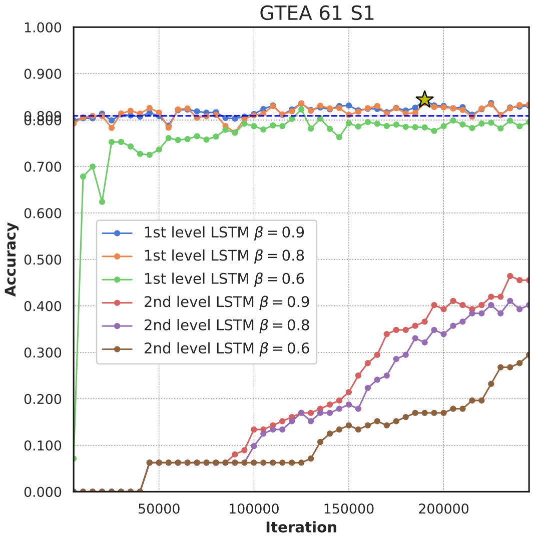


Figure 6.10: GTEA 61 test curves for different values of β. The horizontal dotted blue line indicates the test accuracy obtained for $\beta = 0$. See text for more details.

Method	Subject 1	Subject 2	Subject 3	Subject 4	Average
Ours (frame level)	62.6	68.42	58.46	69.47	64.74
Ours (1 level LSTM)	73.28	73.68	70.0	67.18	71.04
Ours (Hierarchical LSTM)	**74.05**	**73.68**	**73.08**	**70.99**	**72.95**

Table 6.4: Classification accuracy on each split of the GTEA 71 dataset.

value of β in the set $\{0.5, 0.6, \ldots, 0.9\}$. For example, the effect of different values for β can be seen in Fig. 6.10. Moreover, the plot shows the resulting test curves for the first split of the GTEA 61 dataset. The classification accuracy for the first and second levels are measured per frame and shot, respectively.

Primary Region Importance Furthermore, to gain understanding about the importance of the primary region in our model, we trained the R*CNN architecture with both manually provided regions and with fixed regions covering the low central part of each image. For a single split of the GTEA 71, we achieved 66.11% accuracy when trained with ground truth primary region, 63.80% when trained with our method, 65.05% when trained using the skin as primary region, and 61.93% when trained with a fixed region. Although limited to a single split, these experiments suggest the importance of accurately detecting the primary region. We also report the results of our approach on each evaluated split on Table 6.4 and Table 6.5.

Method	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Average
Ours (frame level)	49.8	58.05	46.11	60.66	51.02	50.83	52.75
Ours (1 level LSTM)	56.55	**65.77**	47.58	63.16	63.27	54.17	58.41
Ours (Hierarchical LSTM)	**58.53**	64.43	**51.37**	**64.54**	**64.63**	**56.25**	**59.96**

Table 6.5: Classification accuracy on each split of the GAZE+ dataset.

Shot Evaluation	Sequence	Splits						Average
		Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Avg. Prediction	Original	55.56	64.09	45.89	62.88	61.9	**56.67**	57.83
	Augmented	**56.55**	**65.77**	**47.58**	**63.16**	**63.27**	54.17	**58.41**
Weighted Avg. Prediction	Original	54.56	**63.76**	47.79	62.88	**62.59**	**57.08**	**58.11**
	Augmented	**56.94**	**63.76**	**49.47**	**63.16**	60.54	54.58	58.08

Table 6.6: Classification performance comparison using visual data augmentation on the GAZE+ dataset for 44 categories. See text for more details.

Visual Data Augmentation We measured the classification performance of the visual data augmentation by comparing it against the original sequence. We considered all the splits from the GAZE+ dataset using the first level LSTM model. The accuracy was obtained using the normal and weighted average prediction for each video shot. The results of this experiment are shown in Table 6.6.

Sequential Data Augmentation In order to understand the effect of temporally augmenting the data sequences, we measure the accuracy performance of our model on the GAZE+ dataset. We trained over all the splits of the dataset using the visually augmented frames. For each split, we trained using as initial weights the best resulting models from the first level LSTM. In addition to calculating the average prediction per shot, we also measured the linearly weighted average prediction. The results of this experiment is shown in Table 6.7. The results show that there is no significant improvement over the original sequences.

6.5.2 Comparison with the state of the art

We tested the proposed approach on five datasets from three well known benchmarks (GTEA, Gaze, Gaze+) and performed comparisons to five states of the art methods. On average, each competitive method has been tested on 3 datasets. We achieved a gain of 6.11%, 5%, on the GTEA Gaze, and GTEA Gaze+, respectively. On the GTEA dataset we achieved similar performance to Ma et al. [2016]. On this latter dataset, we

Shot Evaluation	Sequence	Splits						Average
		Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	
Avg. Prediction	Original	**58.53**	64.43	**51.37**	**64.54**	**64.63**	**56.25**	**59.96**
	Augmented	58.33	**65.44**	**51.37**	63.99	**64.63**	55	59.79
Weighted Avg. Prediction	Original	**58.13**	63.42	**52.21**	63.44	62.59	**57.08**	59.48
	Augmented	57.74	**64.09**	52	**65.10**	**63.26**	56.67	**59.81**

Table 6.7: Classification performance comparison using sequence data augmentation on the GAZE+ dataset for 44 categories. See text for more details.

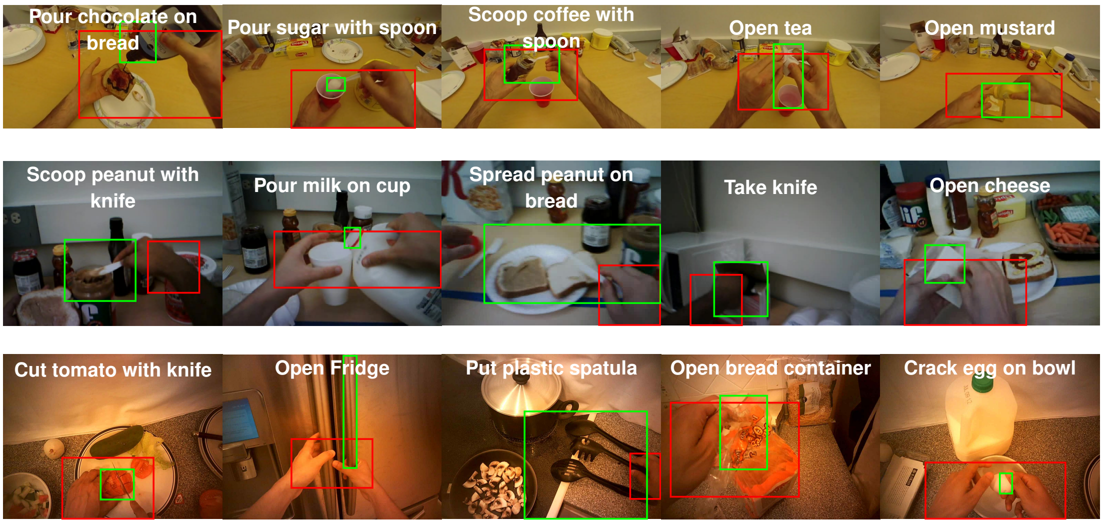


Figure 6.11: Examples of true predictions on all evaluated datasets using our proposed model. The primary and secondary regions are the bounding boxes highlighted in red and green colors, respectively.

observed that often the predictions at frame level within one shot were all incorrect, so that the HLSTM is not able to improve the results for such shots.

Note that with respect to the state of the art, our method does not rely on motion information and shot boundary information is required solely during training. These results demonstrate the effectiveness of reasoning about the spatial arrangement of relevant regions at frame level on the one side, and how important it is to exploit the temporal structure of videos on the other side. We also provide some examples of qualitative results in Fig. 6.11 and 6.12. In particular, in Fig. 6.11 we show examples of true predictions, whereas in Fig. 6.12, we show examples of failure. We observed that, in case of failure, very often either the action verb or the action object are correct. Specifically, a false verb prediction can often be attributed to the lack of motion modeling.

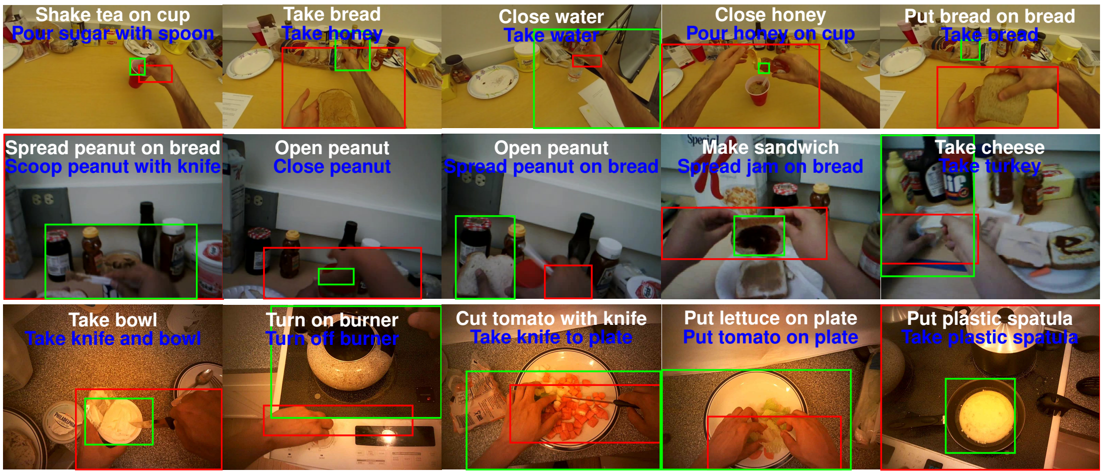


Figure 6.12: Examples of false predictions on all evaluated datasets using our proposed model. The true action category and its false prediction appears on top of the image on white and blue colors, respectively. The primary and secondary regions are the bounding boxes highlighted in red and green colors, accordingly.

6.6 Conclusions

This chapter introduced a novel approach for egocentric action recognition able to reason at spatial level about semantically meaningful regions of the image and at temporal level about the sequence of actions being performed. The proposed neural network architecture consists of a CNN module, followed by an HLSTM able to capture temporal dependencies within and across shots. Experiments demonstrate that the proposed architecture achieves significant improvement with respect to competitive methods modeling the temporal structure and/or object interactions. We have also performed a detailed ablation analysis of the various components of our network.

Chapter 7

Multimodal Action Recognition From Egocentric Videos

Contents

7.1 Proposed approach 102

7.2 EPIC Kitchens Dataset 105

7.3 Experimental Setup 107

7.4 Evaluation metrics 108

7.5 Results 109

7.6 Conclusions 115

Our interaction with the world is an inherently multimodal experience. We employ different senses to perceive new information both passively and actively when we explore our environment and interact with it. In particular, object manipulations almost always have an associate sound (e.g. open tap), and we naturally learn and exploit these associations to recognize object interactions by relying on available sensory information (audio, vision, or both). However, the understanding of human-to-object interactions has historically been addressed focusing on a single modality. In particular, a limited number of works have considered integrating the visual and audio modalities for this purpose.

In this chapter, we focus on the recognition of actions involving object manipulations. More specifically, our goal is to identify egocentric *actions* of the type *verb+noun*, i.e. *pouring+jam* performed in a kitchen environment. We propose a method that integrates visual and audio features. Fig. 7.1 shows how audio cues are

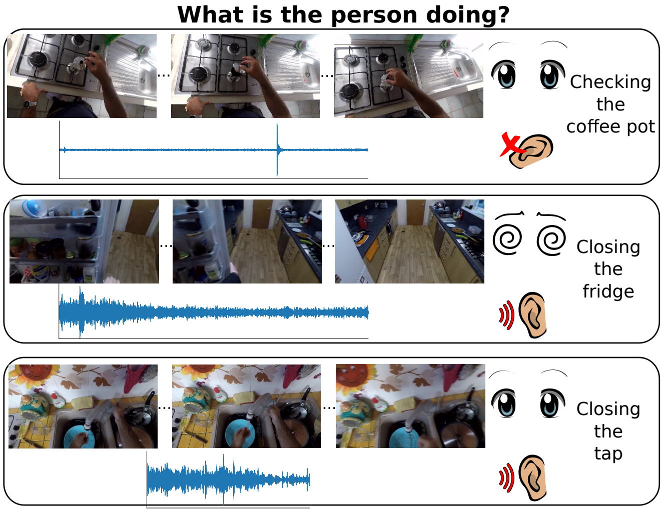


Figure 7.1: Audio and vision are complementary sources of information for recognizing egocentric object interactions. A limited number of interactions do not have an associated audio signal (top), but in most cases, auditory sources provide valuable information in situations such as the occlusion of the hands and objects (middle), and in some others they just strengthen the visual information (bottom).

crucial to identify the activity being performed especially when visual information is ambiguous from an egocentric perspective due to self-occlusions. Our proposal aims at exploiting the complementarity of visual and audio features to obtain robust multimodal representations.

Our approach combines a sparse temporal sampling strategy with a late fusion of audio, spatial, and temporal streams. Experimental results on the EPIC-Kitchens dataset show that multimodal integration leads to better performance than unimodal approaches. In particular, we achieved a 5.18% improvement over the state of the art on verb classification.

The rest of the chapter is organized as follows. First, section 7.1 introduces the proposed approach. Next, in section 7.2 the EPIC-Kitchens dataset is described. In sections 7.3 and 7.5 we detail the experimental setup and discuss the results, respectively. Finally, section 7.6 concludes by summarizing the main findings.

7.1 Proposed approach

In this section, we describe our multimodal approach for action recognition, which is summarized in Fig. 7.2. We first present our vision-based recognition approach that

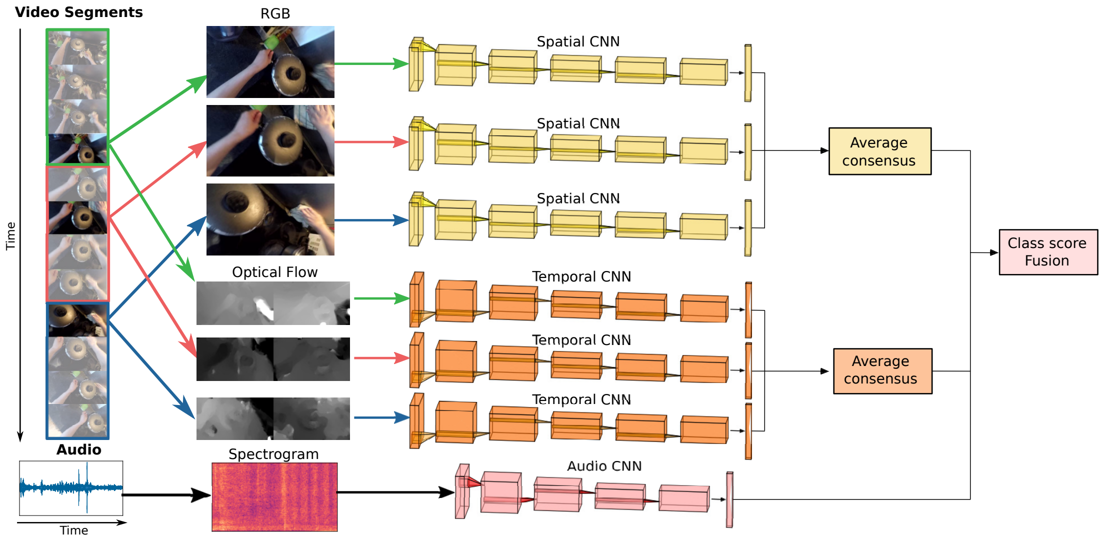


Figure 7.2: Pipeline of our multimodal proposed approach. A video is divided into $K = 3$ time segments shown in **green**, **red**, and **blue** colors. Then, RGB and optical flow frames are sparsely sampled from each time segment to be processed in their respective spatial and temporal streams. At the end of each stream, the average consensus of the softmax scores is computed. A spectrogram is calculated from the raw audio signal and processed in its audio stream. The class scores of each stream are joined together using late fusion.

uses Temporal Segments Network (TSN) [Wang et al., 2016] for the spatial (RGB) and temporal (optical flow) visual modalities. Then, we present our audio-based recognition approach that uses two different convolutional neural networks (CNNs): VGG-11 [Simonyan and Zisserman, 2014b] and a custom network based on the one presented by Sainath and Parada [2015]. Finally, we detail our fusion strategy to integrate the different modalities.

Vision The visual-spatial and temporal input modalities are RGB and optical flow frames calculated using the method proposed by Zach et al. [2007]. Each visual modality was trained as a TSN stream. On the TSN model, the frames of a video are grouped into K sequential segments of equal size. Similarly to Wang et al. [2016], we decided to set $K = 3$ as originally presented. Simultaneously from each segment, a frame is sparsely sampled and processed by a CNN. In our case, we used as backbone networks a ResNet-18 and ResNet-50 [He et al., 2016] in our experiments. Then, a consensus of the scores from each processed frame is done. We used as a consensus function the

Layer type	Output size	#Filters	Kernel size	Dilation
Conv2D	331×248	64	11×7	9×4
max-pool				
Conv2D	166×124	64	6×4	9×4
Conv2D	166×124	32	6×4	9×4
Conv2D	166×124	16	6×4	9×4
max-pool				
Dense	256	-	-	-
Dense	256	-	-	-

Table 7.1: Architecture of our proposed traditional dilated network for audio classification.

average of the softmax scores. This model is an extension of the two-stream model proposed by Simonyan and Zisserman [2014a], but it learns long-range temporal structure of the action in the video.

Audio The audio modality uses as input the spectrogram of the raw audio signal from the video. The spectrogram is calculated as follows. First, when the video has multiple audio channels, we join them by obtaining their mean. Then, we compute the short-time Fourier transform (STFT) from this signal and not the Mel-frequency cepstral coefficients (MFCCs), as we are interested in noises rather than in human voices. We use a sampling frequency of 16 KHz, as it covers most of the band audible to the average person [Heffner and Heffner, 2007].

The STFT uses a Hamming window of length equal to 30 ms with 50% time overlapping. The signal spectrogram is calculated as the logarithm value of the squared magnitude of its STFT. The final step consists in normalizing all the input spectrograms. The spectrogram has a resulting dimension size for the frequency of 331. We only consider the first four seconds of the audio spectrogram. When it has less than four seconds duration then a zero padding is applied. This constraint results in a time dimension size of 248 for the input spectrogram. Thus, the size of the input spectrogram image for CNN is 331×248.

A single spectrogram covers a larger time window than the visual input frames. Therefore, our model only needs one CNN to process the audio modality. Nonetheless, for longer video durations a long short-term memory (LSTM) could be added as proposed by Sainath et al. [2015]. Our backbone audio CNN models are a VGG-11 [Simonyan and Zisserman, 2014b] network and a proposed smaller CNN based on the one proposed by [Sainath and Parada, 2015]. We call the latter traditional dilated network and show its architecture in Table 7.1. This network was adapted to spectrograms with bigger sizes by using dilation convolutions introduced by Yu and Koltun [2016].

Audio-visual fusion In our experiments, we used two late fusion methods. The first method is the weighted sum of the class scores from each stream. The second method uses a network with two fully connected (FC) layers. Its input vector is calculated by concatenating the outputs of the penultimate FC layers from each stream. During training, the weights of each modality CNN stream are kept frozen.

7.2 EPIC Kitchens Dataset

We carried out our experiments on the EPIC Kitchens dataset introduced by Damen et al. [2018]. Each video segment in the dataset shows a participant doing one specific cooking-related *action* in a kitchen environment. Some examples of their labels are "cut potato" or "wash cup". The EPIC Kitchens dataset includes 432 videos recorded from a first-person perspective by 32 participants in their own kitchens while cooking/preparing something. Each video was divided into segments in which the person is doing one specific *action* (a *verb* plus a *noun*). The total number of verbs and nouns categories in the dataset is 125 and 352, correspondingly. Thus, it is naturally highly

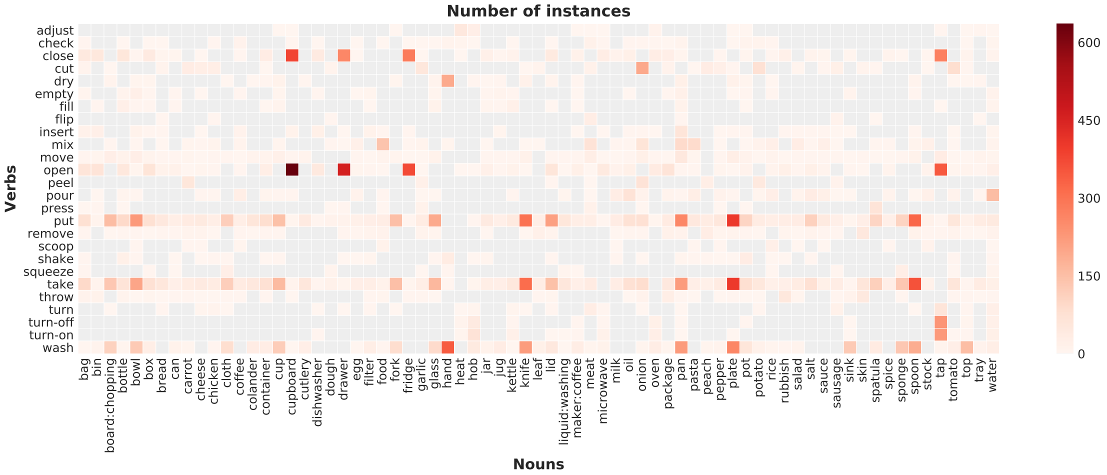


Figure 7.3: Heatmap of the number of *action* instances in the EPIC Kitchens dataset training split. This plot only considers 22,018 *action* segments of verbs and nouns having more than 100 instances.

Video segments time duration stats (secs)	
Min:	0.5
Mean:	3.38
Median:	1.78
Std. Dev.:	5.04
Mode:	1.0
Max:	145.16

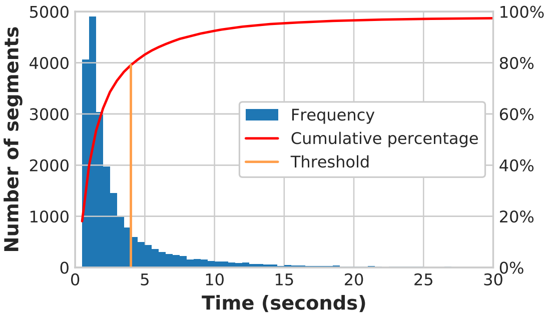


Figure 7.4: Statistics and histogram of the time duration of the video segments in the training partition of the EPIC Kitchens with a bin size of half a second.

unbalanced as seen in Fig. 7.3. Moreover, the duration statistics of the *action* segments are shown in Fig. 7.4. It shows that roughly 4 seconds covers 80.697% of all video segments.

For comparison purposes, we considered two data partitions derived from the labeled data of the EPIC Kitchen Challenge:

Home made partition In this partition, all the participants were considered for the training, validation, and test splits. The data proportions for the validation and test splits were 10% and 15%, accordingly. Since the resulting distribution of action classes is highly unbalanced, the data split was done as follows. At least one sample of each action category was put in the training split. If the category had at least two samples, one of them went to the test split. We also report the results obtained on the EPIC Kitchens Challenge board from the models trained on this partition.

Unseen verb partition The second data partition was the one proposed by Baradel et al. [2018]. This partition only used the *verb* classes from the labeled data. Moreover, the training and test splits are on the participants 01-25 and 26-31, respectively. Therefore, the test set is only composed of unseen kitchens. In order to train our methods, we created a randomly stratified validation split with 10% of data from the training split.

7.3 Experimental Setup

The main objective of our experiments was to measure the performance of our proposed multimodal approach on an egocentric object interaction recognition task from videos. More specifically, the task consists of predicting what a person is doing (*verb*) using a specific object (*noun*). Both classifications can be trained and evaluated separately or combined as a single *action* classification. Therefore, our secondary objective was to determine the contribution of audio and visual information on each type of classification (noun, verb, action). Contrary to previous works reported by Fabian Caba Heilbron and Niebles [2015]; Ghanem et al. [2018, 2017], we did not make any assumption on which classification type the audio source would perform better.

7.3.1 Implementation

We first trained all modality streams separately on each training split for *verb*, *noun*, and *action*. Subsequently, we trained their late fusion on different combinations of audio and vision streams. In our experiments, we searched for the best learning rates while performing early stopping using the validation split. The following paragraphs provide more training details for each part of the model.

Audio We only trained our audio network on the spectrogram of the first four seconds of each video segment. As seen in Fig. 7.4, setting a time threshold of 4 seconds allows to completely cover 80.697% of all video segments using a single time window. For all our experiments we used the stochastic gradient descent (SGD) optimization algorithm to train both networks from scratch. We used a momentum and a batch size equal to 0.9 and 6, correspondingly. The learning rates for VGG-11 [Simonyan and Zisserman, 2014b] on *verb*, *noun*, and *action* classification were 5×10^{-6}, 2.5×10^{-6}, and 1.75×10^{-6}, respectively. The learning rates for the traditional dilated network on *verb*, *noun*, and *action* classification were 4.5×10^{-4}, 7.5×10^{-5}, and 1×10^{-4}, accordingly. It was trained using a learning rate equal 1×10^{-5} and a batch size of 22 during 65 epochs. The difference between the different data splits was the number of training epochs. The training times for the VGG-11 and the Traditional Dilated network were around fifteen and eleven hours on an Nvidia GeForce GTX 980, respectively.

Vision We followed similar training specifications used by Damen et al. [2018], but considering the spatial and temporal CNNs as single networks rather than two joined streams. As previously stated, we used ResNet-18 and ResNet-50 as backbone CNNs. The former was used only in the data split presented by Baradel et al. [2018] for comparison purposes, while the latter was used in all other experiments. Each backbone CNN was used for both visual streams and was initialized using pre-trained weights on ImageNet Russakovsky et al. [2015]. Moreover, we trained both visual modalities between 40 and 80 epochs using the same learning rate of 1×10^{-3} and decreasing it by a factor of 10 after epochs 20 and 40. The tests were done using 25 samples with 1 spatial cropping. The training of each modality and category took approximately twelve hours on an Nvidia Titan X GPU.

Audio-visual fusion For the weighted sum of class scores, the weights were found using a grid search of values between 1 and 2. For the neural network method, depending on the backbone network used on each modality, the length of the input vector of the first FC layer was between 4,352 and 5,120. The second FC layer had and input vector length of 512. We used a momentum and a batch size equal to 0.9 and 6, correspondingly. The learning rates for the fusion of all modalities on *verb*, *noun*, and *action* classification were 1×10^{-4}, 1×10^{-3}, and 3×10^{-4}, respectively.

7.4 Evaluation metrics

Following the work of Damen et al. [2018], we measured the classification performance using aggregate and per-class metrics. As aggregate metrics for measuring the classification performance, we used the top-1 and top-5 accuracy, whereas as per-class metrics we used precision and recall. Moreover, the per-class classification improvement was measured by calculating the accuracy difference between the visual (RGB+Flow) and the audiovisual (RGG+Flow+Audio) sources. We also computed two baselines using the largest classes and random classifiers for each experiment. The latter baseline was approximated by sampling a multinomial distribution, but specifically, the random classification accuracy of N categories was calculated as

$$acc = \sum_{i}^{N} p_i^{train} \cdot p_i^{test} \tag{7.1}$$

where p_i^{train} and p_i^{test} are the occurrence probability for class i in the train and test splits, accordingly.

			Top-1 Accuracy			**Top-5 Accuracy**			**Avg Class Precision**			**Avg Class Recall**		
			VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION
		Chance/Random	11.38	01.58	00.43	47.58	07.74	02.12	01.00	00.34	00.06	01.00	00.34	00.07
		Largest class	20.19	04.11	02.10	66.93	18.38	07.54	00.21	00.01	00.00	01.05	00.36	00.07
Audio		VGG-11	34.48	09.51	03.56	74.50	26.63	12.17	05.26	01.32	00.28	04.04	01.32	01.72
		Traditional Dilated	34.82	15.44	06.26	74.72	36.96	17.83	04.53	05.77	01.12	03.88	04.95	01.39
Vision		TSN Flow	49.08	22.72	13.54	81.60	46.32	30.77	10.80	08.81	02.53	07.12	04.97	02.23
		TSN RGB	50.65	54.01	32.51	88.63	80.87	59.72	25.96	38.83	16.23	**19.36**	**34.43**	18.94
		RGB+Flow	55.47	52.82	32.76	88.48	78.01	58.13	28.94	39.82	13.53	14.25	27.81	14.22
Multimodal		Flow+Audio	50.06	26.26	15.13	81.02	51.45	33.49	11.72	11.32	02.99	06.61	06.42	02.42
		RGB+Audio	53.51	53.11	32.21	87.35	79.63	57.82	26.57	38.89	13.67	13.07	28.98	14.94
		RGB+Flow+Audio	56.27	51.09	32.27	87.24	77.15	55.96	25.06	37.17	11.81	11.34	24.28	12.05
	Weighted	Flow+Audio	51.40	26.39	15.62	81.57	51.54	34.24	11.86	12.57	03.37	06.98	06.48	02.72
		RGB+Audio	54.24	54.90	33.84	88.19	80.89	59.72	**31.21**	38.96	15.03	15.07	31.74	16.73
		RGB+Flow+Audio	56.65	53.90	33.86	87.70	79.47	58.37	25.75	**40.87**	13.36	12.58	28.17	14.02
	FC	Flow+Audio	52.00	33.02	20.22	82.24	58.50	39.37	09.72	18.58	05.80	08.11	16.63	06.58
		RGB+Audio	55.41	55.08	35.21	87.15	79.78	60.27	20.49	38.47	15.35	14.28	33.55	17.64
		RGB+Flow+Audio	**60.21**	**56.14**	**38.55**	**89.07**	**80.96**	**62.97**	27.05	39.89	**17.14**	19.09	33.85	**19.28**

Table 7.2: Classification performance for the equally stratified *action* data split for the *verb+noun* fusion. The scores in **gray** color were calculated based on the *verb+noun* classifiers.

7.5 Results

Noun The performance results for the *noun* classification on the home made partition are shown in Table 7.2. The best accuracy score was achieved by the weighted multimodal combination that improved the visual baseline by 1.24%. These results also indicate that the separated or combined unweighted fusion of the optical flow and audio decreases the top-1 and top-5 accuracy of the task. This effect can also be seen on the higher number of classes that decreased their accuracy on Fig. 7.5. The most misclassified pair of objects are *spoon-knife*, *spoon-fork*, *plate-bowl*, *tap-sponge*, and *knife-fork*.

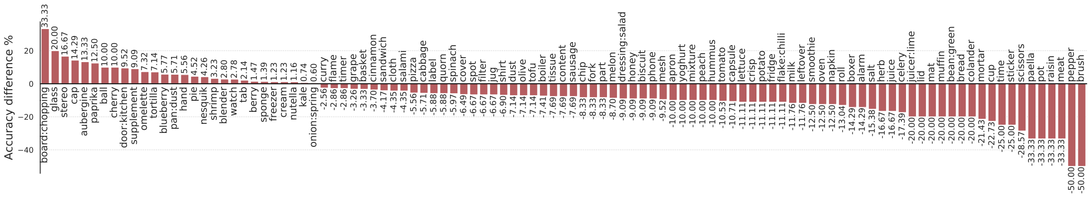


Figure 7.5: Accuracy difference of the *noun* split for the unweighted test predictions that changed with respect to RGB+Flow and RGB+Flow+Audio.

Verb The *verb* classification results on the home made partition are also presented in Table 7.2. Each visual method combination boosted their respective performance by adding the unweighted audio score. The multimodality combination is greater than the best visual method by 4.74%. According to Fig. 7.6, the multimodality helps to disambiguate *turn-on* and *turn-off* verbs, but fails on verbs that lack of sound like *scoop* or *adjust*. The most misclassified pair of verbs are *take-put*, *put-open*, *take-close*, *take-open*, and *put-close*.

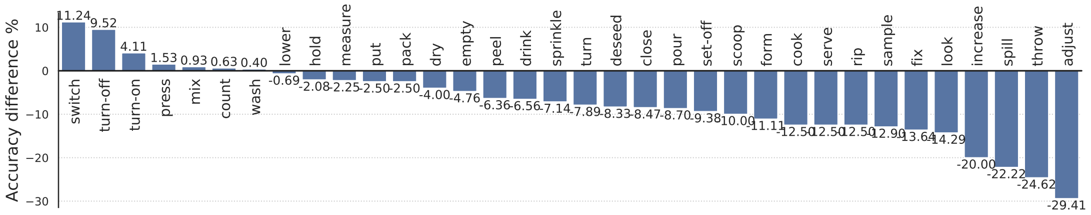


Figure 7.6: Accuracy difference of the *verb* split for the unweighted test predictions that changed with respect to RGB+Flow and RGB+Flow+Audio.

Action The results of our experiments on the homemade partition are presented in Table 7.3. They show that using multimodal information outperforms single audio or visual classification when training separately the *verb* and *noun* classifiers. Even though the best top-1 accuracy was achieved when considering the *verb* and *noun* labels as a single action classification problem, all other performance metrics were lower than when considering them separately. Table 7.2 shows that the accuracy of *noun* is diminished when adding audio scores to the *action* classification. The categories that changed their prediction on the multimodality setting for the action classifier are presented in Fig. 7.7.

Some qualitative results of true and false positive predictions are shown in Fig. 7.8. The action predictions made by the audio modality rely more on the *verb* than *noun* classification. For instance, it correctly predicts the verb *wash* in Fig. 7.8, but falsely predicts the noun *hand* instead of *cloth*. The visual modality obtained similar accuracy on *verb* and *noun* classification. Their predictions fail in cases where the actions occur in similar contexts, for example, the actions *take cup* and *wash cup* can occur in the sink, as illustrated in in Fig. 7.8. The combination of audio and visual modalities help to disambiguate actions where the objects are occluded such as *open drawer* and *close drawer*, as shown in first row of the fusion column in Fig. 7.8. Their complementarity also helps on the classification of actions with the same *verb*, but different *noun*, such

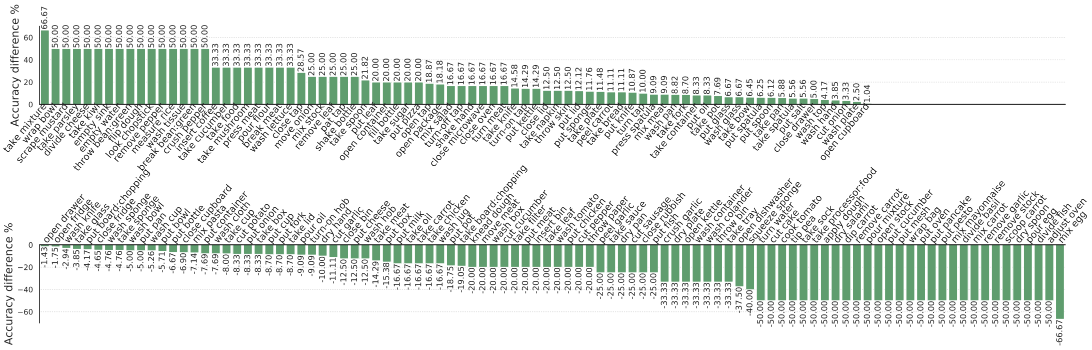


Figure 7.7: Accuracy difference of the *action* split for the unweighted test predictions that changed with respect to RGB+Flow and RGB+Flow+Audio.

as the case of *washing glass* and *bowl*, shown in the second row of the fusion column in Fig. 7.8.

Comparison with other methods Tables 7.4 and 7.5 show the results obtained on the EPIC Kitchen Challenge board using the models trained on the homemade partition. These results indicate that directly training over the *action* performs better than the combination of *verb+noun*. Our multimodal models obtained better scores than the challenge baseline and have similar results as previous works [Sudhakaran et al., 2019b]. Additionally, the results obtained on the unseen participants (S2) test split are

			Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
			VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION
		Chance/Random	11.38	01.58	00.43	47.58	07.74	02.12	01.00	00.34	00.06	01.00	00.34	00.07
		Largest class	20.19	04.11	02.10	66.93	18.38	07.54	00.21	00.01	00.00	01.05	00.36	00.07
Audio		VGG-11	29.44	05.99	05.84	70.49	16.90	17.41	00.92	00.51	00.20	01.86	00.79	00.45
		Traditional Dilated	32.16	06.86	10.77	72.22	22.34	25.86	01.64	00.84	03.04	02.31	01.90	04.58
Vision		TSN Flow	36.70	14.82	22.54	75.98	37.25	41.52	02.94	03.46	05.98	03.89	03.06	07.04
		TSN RGB	34.06	31.01	32.80	76.71	63.90	59.06	04.17	13.45	15.37	04.83	11.16	19.62
		RGB+Flow	37.12	30.15	36.19	79.56	62.02	59.92	04.96	11.94	15.14	04.19	09.21	17.84
Multimodal		Flow+Audio	38.04	13.03	25.35	77.42	35.81	45.06	03.89	04.52	09.25	03.59	03.21	09.32
		RGB+Audio	37.65	27.03	34.93	79.52	59.41	60.69	04.03	13.88	16.08	03.66	09.49	18.50
		RGB+Flow+Audio	39.95	27.45	36.78	80.47	59.17	60.38	03.30	11.03	15.51	04.02	08.26	16.96
	Weighted	Flow+Audio	39.66	14.44	26.03	77.53	39.15	46.16	03.60	04.78	09.30	03.97	03.39	09.29
		RGB+Audio	38.80	30.15	35.50	80.31	63.44	61.71	04.22	14.97	16.71	04.20	10.91	19.63
		RGB+Flow+Audio	40.06	29.68	36.92	**80.82**	61.69	61.14	03.16	12.47	15.56	04.12	09.47	17.76
	FC	Flow+Audio	41.58	21.26	27.34	79.63	48.33	47.84	05.81	07.16	12.63	05.08	05.69	14.01
		RGB+Audio	40.85	36.39	35.94	76.84	70.05	61.31	**09.03**	17.13	16.40	07.11	12.88	19.57
		RGB+Flow+Audio	**42.56**	**36.81**	**40.15**	77.06	**70.38**	**64.19**	08.48	**18.08**	**19.21**	**07.55**	**12.93**	**22.68**

Table 7.3: Classification performance for the equally stratified *action* for the *action* classifier. The scores in **gray** color were calculated based on the *action* classifier.

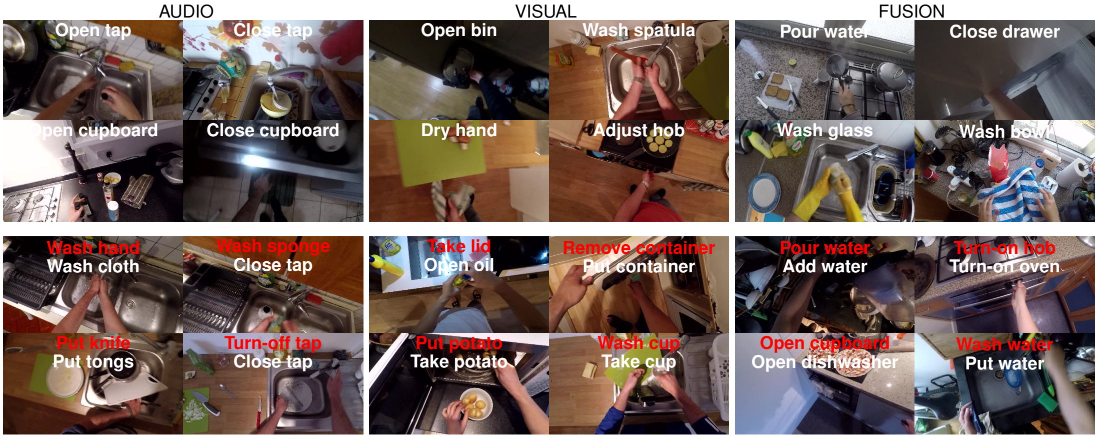


Figure 7.8: Qualitative results for the unweighted multimodal *action* classification experiment. The top and bottom rows shows true and false positive prediction, respectively. The columns indicate when the audio, vision, or their fusion scores were the final multimodal decision. The true and false labels are shown in white and red colors.

			Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
			VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION
		Chance/Random	12.62	01.73	00.22	43.39	08.12	03.68	03.67	01.15	00.08	03.67	01.15	00.05
		Largest Class	22.41	04.50	01.59	70.20	18.89	14.90	00.86	00.06	00.00	03.84	01.40	00.12
Audio		Traditional Dilated (Verb+Noun)	35.11	10.65	03.95	75.33	28.63	13.01	15.43	06.19	01.75	11.03	06.64	01.26
		Traditional Dilated (Action)	34.06	05.31	07.43	73.51	18.24	20.94	05.19	02.74	01.81	07.85	04.14	03.08
Visual		TSN BNInception (FUSION) Damen et al. [2018]	48.23	36.71	20.54	84.09	62.32	39.79	47.26	35.42	10.46	22.33	30.53	08.83
		TSN ResNet-50 (Verb+Noun) (FUSION)	55.08	38.59	24.38	86.36	64.16	45.37	43.69	38.59	14.91	28.63	32.10	12.12
		TSN ResNet-50 (Action) (FUSION)	38.62	25.84	27.95	79.51	54.18	49.12	10.50	24.63	14.13	14.30	21.12	14.61
		LSTA (two stream) Sudhakaran et al. [2019b]	59.55	38.35	30.33	85.77	61.49	49.97	42.72	36.19	14.46	38.12	36.19	17.76
		3rd Place Challenge Sudhakaran et al. [2019a]	63.34	44.75	35.54	89.01	69.88	57.18	63.21	42.26	19.76	37.77	41.28	21.19
		2nd Place Challenge Ghadiyaram et al. [2019]	64.14	47.65	35.75	87.64	70.66	54.65	43.64	40.52	18.95	38.31	45.29	21.13
		1st Place Challenge Wang et al. [2019]	**69.80**	**52.27**	**41.37**	**90.95**	**76.71**	**63.59**	**63.55**	**46.86**	**25.13**	**46.94**	**49.17**	**26.39**
Multimodal	Verb+Noun	Ours	56.37	37.69	24.00	85.47	63.45	44.66	48.15	38.02	13.49	25.54	30.31	10.50
		Ours (Weighted)	56.44	**39.42**	25.26	85.87	**65.27**	46.27	**51.39**	**38.36**	14.88	26.66	32.88	11.90
		Ours (FC)	**58.88**	39.13	27.35	**87.15**	64.83	47.68	46.36	37.92	**16.63**	**38.13**	**34.90**	15.00
	Action	Ours	40.80	22.29	28.83	81.04	50.59	49.68	11.43	23.00	15.89	13.40	17.90	14.18
		Ours (Weighted)	41.22	24.29	29.09	81.16	53.25	**50.57**	11.24	23.61	15.35	14.04	19.72	14.51
		Ours (FC)	44.64	30.64	**29.13**	76.41	59.39	49.71	19.90	32.28	16.51	21.99	25.28	**16.54**

Table 7.4: Performance comparison with EPIC Kitchens challenge baseline results for the seen test partition. The results highlighted in **bold blue** are the best obtained by our method.

			Top-1 Accuracy			Top-5 Accuracy			Avg Class Precision			Avg Class Recall		
			VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION	VERB	NOUN	ACTION
		Chance/Random	10.71	01.89	00.22	38.98	09.31	03.81	03.56	01.08	00.08	03.56	01.08	00.05
		Largest Class	22.26	04.80	00.10	63.76	19.44	17.17	00.85	00.06	00.00	03.84	01.40	00.12
Audio		Traditional Dilated (Verb+Noun)	30.73	07.20	02.53	67.26	21.65	09.90	13.92	04.52	01.90	09.79	04.68	01.20
		Traditional Dilated (Action)	31.96	03.89	03.96	64.73	13.96	13.45	05.05	01.66	00.97	07.91	04.04	01.94
Visual		TSN BNInception (FUSION) Damen et al. [2018]	39.40	22.70	10.89	74.29	45.72	25.26	22.54	15.33	05.60	13.06	17.52	05.81
		TSN ResNet-50 (Verb+Noun) (FUSION)	45.72	24.89	14.95	77.06	49.37	31.07	24.44	20.30	08.79	18.04	18.96	10.10
		TSN ResNet-50 (Action) (FUSION)	36.63	18.06	17.14	75.28	42.03	34.65	11.38	10.89	07.27	12.96	14.38	10.78
		LSTA (two stream) Sudhakaran et al. [2019b]	47.32	22.16	16.63	77.02	43.15	30.93	31.57	17.91	08.97	26.17	17.80	11.92
		3rd Place Challenge Sudhakaran et al. [2019a]	49.37	27.11	20.25	77.50	51.96	37.56	31.09	21.06	09.18	18.73	21.88	14.23
		2nd Place Challenge Ghadiyaram et al. [2019]	55.24	33.87	23.93	80.23	58.25	40.15	25.71	28.19	**15.72**	25.69	29.51	17.06
		1st Place Challenge Wang et al. [2019]	**59.68**	**34.14**	**25.06**	**82.69**	**62.38**	**45.95**	**37.20**	**29.14**	15.44	**29.81**	**30.48**	**18.67**
Multimodal	Verb+Noun	Ours	46.88	25.16	14.58	77.13	48.69	31.00	**28.72**	**16.63**	08.93	17.25	17.83	08.55
		Ours Weighted	47.46	25.95	15.74	**77.16**	**50.12**	31.85	28.71	16.47	09.26	17.85	19.21	09.94
		Ours FC	**47.49**	**26.36**	15.98	76.68	49.37	31.75	24.64	20.61	**09.80**	**20.59**	**21.35**	10.03
	Action	Ours	38.37	15.23	**18.40**	75.15	39.84	35.64	10.93	11.60	06.88	11.75	13.31	10.91
		Ours (Weighted)	38.10	16.76	18.23	75.38	42.23	**35.68**	11.58	12.83	07.72	11.79	14.16	11.26
		Ours (FC)	40.87	20.38	17.65	69.27	45.82	33.73	15.72	15.35	09.61	17.34	16.95	**12.20**

Table 7.5: Performance comparison with EPIC Kitchens challenge baseline results for the unseen test partition. The results highlighted in **bold blue** are the best obtained by our method.

in the top-ten ranking of the first challenge. We can also observe that the unweighted addition as a fusion method for *noun* diminishes the aggregate and per-class performance, not only for our method but also in the baseline results [Damen et al., 2018].

The results on the comparison data split originally presented by Baradel et al. [2018] are shown in Table 7.6. Our method obtained an improvement in accuracy of 5.18% with respect to the ORN method [Baradel et al., 2018]. This test on unseen kitchens showed that methods that only rely on RGB underperform optical flow methods. Likewise, they also showed that audio classification methods can have similar performance to visual methods. The addition of auditory sources increased the performance of the best visual method by 3.47%.

Comparison of audio CNN architectures The VGG-11 and the Traditional Dilated network have similar classification performance. The results in Tables 7.2 and 7.3 indicate that the Traditional Dilated network has better results on seen test users, but the results on Table 7.6 shows that the VGG-11 network outperforms the Traditional

		Method	Top-1 Accuracy	Top-5 Accuracy	Avg. Class Precision	Avg. Class Recall
		Chance/Random	11.75	48.87	00.99	00.98
		Largest class	21.27	69.44	00.31	01.41
Audio		Traditional Dilated	30.51	74.19	04.71	03.60
		VGG-11	33.27	74.13	05.72	04.08
Vision		ResNet-18 He et al. [2016]†	32.05	-	-	-
		I3D ResNet-18 Carreira and Zisserman [2017]†	34.20	-	-	-
		TSN ResNet-18 RGB	34.69	77.13	08.38	05.08
		ORN Baradel et al. [2018]†	40.89	-	-	-
		TSN ResNet-18 Flow	44.48	77.88	08.15	06.18
		RGB+Flow	43.36	78.77	09.98	05.58
Multimodal		RGB+Audio VGG-11	41.09	80.10	08.82	04.76
		Flow+Audio VGG-11	45.75	80.34	08.41	05.57
		RGB+Flow+Audio VGG-11	45.86	80.72	09.33	05.25
	Weighted	RGB+Audio VGG-11	40.83	79.42	09.45	04.85
		Flow+Audio VGG-11	43.74	79.80	09.68	05.31
		RGB+Flow+Audio VGG-11	**46.07**	**80.76**	09.34	05.30
	FC	RGB+Audio VGG-11	42.08	79.44	**12.51**	06.55
		Flow+Audio VGG-11	44.50	78.68	08.17	06.86
		RGB+Flow+Audio VGG-11	45.92	80.33	11.01	**07.31**

Table 7.6: Classification performance results on the comparison *verb* data split. The results marked with † were originally reported in Baradel et al. [2018].

Dilated network on unseen test users.

7.5.1 Discussions

The experimental results show that our model improves the top-1 accuracy by 3.61% in average for *verb*, *noun*, and *action* classification on our homemade data partition. Likewise, the results suggest that audiovisual multimodality benefits the classification of *verb*, and consequently *action*, more than for the classification of *noun*. Furthermore, although multimodality improves the aggregate performance metrics and avg. class precision for *noun* classification, the unweighted fusion decreases their value as observed in Table 7.2. This might be as a consequence of three main reasons. First, an interacting object can produce several sounds and not being described by one in particular. For instance, the sounds of a fridge being opened and closed are characteristically different. Second, rather than describing an object, sounds are better suited for describing the materials they are made of. For example, water and milk are liquids, and their emitting sound while being poured is indistinguishable. Third, objects lack a time dimension, but doing an action and making a sound involve time. Nonetheless, not all actions produce any sound, like checking the coffee pot. Our results show that the multimodality classification on verbs fails on categories that does not produce any sound and that are more visually abstract, like *checking* the heat. Additionally, harder audiovisual classes are *empty*, *flip*, and *squeeze*. The most discriminative input source for the *noun*, *verb*, and *action* comes from the RGB frames. However, their performance decreases when the test is performed on images from unseen persons and the optical flow achieves higher accuracy, as seen in Table 7.6.

7.6 Conclusions

We presented a multimodal approach for egocentric action classification and we validated it on the EPIC Kitchens dataset. Our approach combines audiovisual input sources. Specifically, its audio input is the spectrogram extracted from the raw audio of the video, while its visual inputs are RGB and optical flow frames. We tested and analyzed our approach for classifying each separate category (*verb* and *noun*) or merged (*action*). The obtained results show that our model improves the top-1 accuracy by 3.61% on average. Additionally, the results suggest that multimodal information is spacially beneficial for the *verb* recognition problem. Indeed, our multimodal approach outperformed the state of the art methods on *verb* classification by 5.18% accuracy.

Chapter 8

Conclusions

Contents

8.1 Summary of Contributions 117

8.2 Future Work 119

In this dissertation, we presented novel approaches for the egocentric action recognition from lifelogs and videos. The proposed models for lifelogs were defined according to different time scales. Whereas the models proposed for videos had a particular focus on object-interactions.

In this final chapter we present a summary of our contributions and findings in Section 8.1 and discuss about future research lines in Section 8.2.

8.1 Summary of Contributions

In chapter 2, we discussed the ambiguity of the terms *action* and *activity* in the context of computer vision and provided our definition throughout this thesis. Later on, we briefly presented a historically relevant context of action recognition that highlighted the low-level features approaches. Since our models are based on deep learning, we thoroughly described convolutional architectures. We started describing relevant works on third-person action recognition and finished detailing first-person methods for lifelogs and egocentric videos.

In chapter 3, we presented an ensemble method for still images from lifelog sequences, namely the CNN+RF model. This method combines the output of different fully connected layers from a backbone convolutional network using a random forest. This method can be seen as the generalization of the CNN LFE approach that uses the softmax prediction of a CNN with information such as date&time. In order to make a robust evaluation of our method, we extended the NTCIR-12 dataset consisting of three people by annotating it with 21 different *activities* . Moreover, we considered more metrics than accuracy alone and the tests were not only in a random split but also a temporal split. Our preliminary results in this chapter showed that our approach benefited early convolutional architectures.

In chapter 4 we argued that although lifelogs lack of motion features, they still show temporal coherence within neighboring frames that can be exploited. Under this premise, we first proposed two temporal models. The first is the extension of the CNN+RF by adding a long short-term memory (LSTM) unit. The second is the CNN+Piggyback LSTM that models the temporal relation between adjacent overlapping frames in subsequent input batches. Both models used as training strategy a *sliding window* approach that samples a lifelog sequence in overlapping frame segments of fixed length. The initial tests over the NTCIR-12 showed that both architectures improved the baseline performance of a CNN, thus showing that they were able to capture the temporal evolution of features over time. However, the CNN+Piggyback LSTM model did not present any improvement over the CNN+LSTM architecture.

Since a robust generalization test required data from multiple users, more lifelog sequences, and more *activities* , we introduced the ADLEgoDataset. Our dataset consists of 105,529 annotated images, captured by 15 different people performing 35 different *activities* . We propose a robust benchmark that considers unseen full-day sequences and unseen users during training. In this dataset, we tested the temporal methods described above using a more modern convolutional architecture and the bidirectional counterpart of LSTMs. Our results showed that, in comparison with full-day sequences, the best training mechanism for lifelogs was the *sliding window* indifferently of the LSTM or BLSTM temporal mechanism. We also showed that the best overall architecture was the CNN+LSTM, but the CNN+BLSTM had a better performance in the seen users testing partition. The last result confirmed that the CNN+RF model overfits when trained on a wider context having more users and images, and thus its temporal version (CNN+RF+LSTM model) overfits as well.

We also presented an event-based approach that takes into account the temporal boundaries from segmented events of a lifelog. These events were extracted using the SR-Clustering algorithm that extracts temporal segments sharing semantic and contextual information. This approach was also tested in the ADLEgoDataset benchmark but did not prove to be as effective as training with the full-day sequence. The smoothing effect that the LSTM has over neighbor frames in a sequence neglected the effect of the temporal boundaries, as the events showed to contain sparse *activities* .

In chapter 5 we evaluated the performance of trained convolutional networks in new data for real scenarios and proposed domain adaptation strategies to cope with them. First, we experimentally measured the discrepancy of two target datasets with respect to a source domain. The metric we used was the maximum mean discrepancy and then showed the diminishing of the classification performance. Then we added different amounts of labeled data from the target domains and showed that competitive results can be obtained with a little amount of extra data.

In chapter 6, we proposed a novel deep architecture that learns spatial and temporal egocentric object-interactions. The spatial object-interactions were modeled using a region-based approach. In this model, the region containing the hands of the person was considered to be the primary region. The interacting object was found using a set of region candidates. After an *action* was inferred from the first level of an LSTM, then an *activity* was inferred from a second level LSTM. We demonstrated that spatial egocentric relations of persons and interacting objects along their logical sequence into actions improved the classification.

In chapter 7, we presented a deep model that combines audiovisual modalities using a late fusion method. Visual information coming from RGB and optical flow frames was modeled using a temporal segments network. The audio information was learned using a convolutional network that takes as input its frequency spectrogram. The classification was independently modeled for *actions* and the combination of a *verb* plus a *noun*. Using different late fusion strategies, the results showed that audio consistently improved the classification performance for all of them, but especially for *verb* and *action*. Audio also proved to be a reliable modality when evaluating videos not seen during training.

8.2 Future Work

We outline the possible future research venues in the following points:

- Since a person can do one or more *activities* at the same time, we consider that the activity recognition from lifelogs is more naturally posed as a multi-label problem. For instance, a person might be exercising on a static bicycle while watching the TV or reading a book while commuting on a train. This kind of more precise description has implications in monitoring health patients of lifelogging applications.

- The temporal activity boundaries for lifelogs are still not well understood. In order to exploit the temporal boundaries, we believe that the sparse occurrence of *activities* should be taken into account and modeled.

- Classification uncertainty of activity recognition models for lifelogs is another open research line. It is important to quantify the confidence of egocentric classification methods in images from new unseen people, as they can completely have different lifestyles that do not reflect at all in the trained model.

- In this work, we modeled person-object interactions for a set of *actions* with a single purpose or *activity*. This can be extended by also modeling person-to-person interactions with a collaborative goal. For example, a team of basketball players perform several *actions* with a ball, but also among each other, and they collaboratively play to score points.

- We explored the addition of an audio modality for the classification of egocentric *actions* . On one hand, this exploration needs to be validated in uncontrolled environments to measure its performance. On the other hand, new devices are offering new sources of information. Their incorporation in new machine learning methods could not only predict what *action* is doing a person but also how well is he/she doing it.

Appendix A

Publications

This thesis has led to the publications summarized below.

Journals submissions

- Cartas A., Marín J., Radeva P., and Dimiccoli M. (2018). Batch-based activity recognition from egocentric photo-streams revisited. *Pattern Analysis and Applications*, 21(4):953-965.

- Cartas A., Radeva P., and Dimiccoli M., (2020) "Activities of Daily Living Monitoring via a Wearable Camera: Toward Real-World Applications," in *IEEE Access*, vol. 8, pp. 77344-77363.

Book chapters

- Dimiccoli M., Cartas A., and Radeva P. (2019). Chapter 6 - Activity recognition from visual lifelogs: State of the art and future challenges. In Alameda-Pineda, X., Ricci, E., and Sebe, N., editors, *Multimodal Behavior Analysis in the Wild*, Computer Vision and Pattern Recognition, pages 121-134. Academic Press

International Conferences

- Cartas A., Marin J., Radeva P., and Dimiccoli M. *Recognizing Activities of Daily Living from Egocentric Images*. In Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), Faro, Portugal, 2017.

- Cartas A., Dimiccoli M., and Radeva P. *Detecting Hands in Egocentric Videos: Towards Action Recognition.* In 16th International Conference on Computer Aided Systems Theory, Canarias, Spain, 2017.
- De Oliveira G., Cartas, A., Bolaños M., Dimiccoli M., Giró-i-Nieto X., and Radeva P. (2016). *LEMoRe: A Lifelog Engine for Moments Retrieval at the NTCIR-Lifelog LSAT Task.* In Lifelog Semantic Access Task, NII Testbeds and Community for Information access Research (NTCIR).

Workshops

- Cartas A., Luque J., Radeva P., Segura Perales C., and Dimiccoli M. *Seeing and hearing egocentric object interactions: How much can we learn?.* In International Conference on Computer Vision (Egocentric Perception, Interaction and Computing Workshop), Seoul, South Korea, 2019.
- Cartas A., Dimiccoli M., and Radeva P. *Batch-Based Activity Recognition from Egocentric Photo-Streams.* In International Conference on Computer Vision (Egocentric Perception, Interaction and Computing Workshop), Venice, Italy, 2017.

Other

- Cartas A., Luque J., Radeva, P., Segura Perales C., and Dimiccoli M. *How Much Does Audio Matter to Recognize Egocentric Object Interactions?.* In Computer Vision and Pattern Recognition Conference (Egocentric Perception, Interaction and Computing Workshop), Long Beach, USA, 2018.
- Cartas A., Talavera E., Radeva, P., and Dimiccoli M. *On the Role of Event Boundaries in Egocentric Activity Recognition from Photo Streams.* In European Conference on Computer Vision (Egocentric Perception, Interaction and Computing Workshop), Munich, Germany, 2018.
- Cartas A., Radeva P., and Dimiccoli M. *Contextually Driven First-person Action Recognition From Videos.* In International Conference on Computer Vision (Egocentric Perception, Interaction and Computing Workshop), Venice, Italy, 2017.

Bibliography

O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4277–4280, March 2012. Cited on page 17.

G. Abebe, A. Cavallaro, and X. Parra. Robust multi-dimensional motion features for first-person vision activity recognition. *Computer Vision and Image Understanding*, 149:229 – 248, 2016. Special issue on Assistive Computer Vision and Robotics - "Assistive Solutions for Mobility, Communication and HMI". Cited on pages 3 and 23.

G. Abebe, A. Catala, and A. Cavallaro. A first-person vision dataset of office activities. In F. Schwenker and S. Scherer, editors, *Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction*, pages 27–37. Springer International Publishing, 2019. Cited on page 23.

U.-V. Albrecht, U. von Jan, J. Kuebler, C. Zoeller, M. Lacher, O. J. Muensterer, M. Ettinger, M. Klintschar, and L. Hagemeier. Google glass for documentation of medical findings: Evaluation in forensic medicine. *J Med Internet Res*, 16(2):e53, Feb 2014. Cited on page 2.

M. A. Arabaci, F. Özkan, E. Surer, P. Jancovic, and A. Temizel. Multi-modal egocentric activity recognition using audio-visual features. abs/1807.00612, 2018. Cited on page 23.

P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. 33(5):898–916, 05 2011. Cited on page 92.

M. Arif, M. Bilal, A. Kattan, and S. I. Ahamed. Better physical activity classification using smartphone acceleration sensor. 38(9):95, Jul 2014. Cited on page 18.

R. B. Ribas Manero, A. Shafti, B. Michael, J. Grewal, J. Ll. Ribas Fernandez, K. Althoefer, and M. Howard. Wearable embroidered muscle activity sensing device for the human upper leg. volume 2016, 08 2016. Cited on page 2.

M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In A. A. Salah and B. Lepri, editors, *Human Behavior Understanding*, pages 29–39, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. Cited on pages 16 and 46.

S. Bambach. A survey on recent advances of computer vision algorithms for egocentric video. *CoRR*, abs/1501.02825, 2015. Cited on page 12.

S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1949–1957, 2015. Cited on pages 18, 20, 82, and 83.

L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, pages 1–17. Springer Berlin Heidelberg, 2004. Cited on page 18.

F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori. Object level visual reasoning in videos. In *European Conference on Computer Vision (ECCV)*, June 2018. Cited on pages 82, 83, 106, 108, 113, and 114.

A. Behera, D. C. Hogg, and A. G. Cohn. Egocentric activity monitoring and recovery. In *Asian Conference on Computer Vision*, pages 519–532. Springer, Springer Berlin Heidelberg, 2012. Cited on page 22.

M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl. Actiserv: Activity recognition service for mobile phones. In *International Symposium on Wearable Computers (ISWC) 2010*, pages 1–8, Oct 2010. Cited on page 18.

T. O. Binford. Visual perception by computer. In *Proceedings of the IEEE Conference on Systems and Control*, December 1971. Cited on page 12.

M. Blum, A. Pentland, and G. Troster. Insense: Interest-based life logging. *IEEE MultiMedia*, 13(4):40–48, 2006. Cited on page 3.

A. F. Bobick. Computers seeing action. In *Proceedings of the British Machine Vision Conference*, pages 4.1–4.10. BMVA Press, 1996. doi:10.5244/C.10.4. Cited on page 12.

A. F. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 352(1358):1257–1265, Aug 1997. Cited on page 12.

S. Z. Bokhari and K. M. Kitani. Long-term activity forecasting using first-person vision. In *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V*, pages 346–360, 2016. Cited on page 22.

M. Bolaños, M. Dimiccoli, and P. Radeva. Toward storytelling from visual lifelogging: An overview. 47(1):77–90, Feb 2017. Cited on pages 12 and 23.

M. Bolaños, Álvaro Peris, F. Casacuberta, S. Soler, and P. Radeva. Egocentric video description based on temporally-linked sequences. 50:205 – 216, 2018. Cited on page 19.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984. Cited on pages 35 and 57.

P. Budner, J. Eirich, and P. A. Gloor. "making you happy makes me happy" - measuring individual mood with smartwatches. abs/1711.06134, 2017. Cited on page 2.

A. Bulling, J. A. Ward, H. Gellersen, and G. Troster. Eye movement analysis for activity recognition using electrooculography. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(4):741–753, Apr. 2011. Cited on page 21.

V. Bush. As We May Think. *Atlantic Monthly*, 176(1):641–649, March 1945. Cited on pages 2 and 3.

D. Byrne, A. R. Doherty, C. G. Snoek, G. J. Jones, and A. F. Smeaton. Everyday concept detection in visual lifelogs: validation, relationships and trends. 49(1):119–144, 2010. Cited on pages 43 and 46.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 778–792. Springer Berlin Heidelberg, 2010. Cited on page 85.

J. Cao, Y. J. Wen, and P. Z. Zhang. Research on application of google glass in the field of journalism. In *Information Technology Applications in Industry III*, volume 631 of *Applied Mechanics and Materials*, pages 180–183. Trans Tech Publications Ltd, 11 2014. Cited on page 2.

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. Cited on pages 85 and 92.

J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. 34(7):1312–1328, 2012. Cited on page 84.

J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017. Cited on pages 16 and 114.

A. Cartas, M. Dimiccoli, and P. Radeva. Batch-based activity recognition from egocentric photo-streams. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 2347–2354. IEEE, 2017a. Cited on pages 8, 24, 50, and 52.

A. Cartas, J. Marín, P. Radeva, and M. Dimiccoli. Recognizing activities of daily living from egocentric images. In L. A. Alexandre, J. Salvador Sánchez, and J. M. F. Rodrigues, editors, *Pattern Recognition and Image Analysis*, pages 87–95, Cham, 2017b. Springer International Publishing. Cited on pages 7, 24, 50, 51, and 52.

A. Cartas, P. Radeva, and M. Dimiccoli. Contextually driven first-person action recognition from videos. Accepted for presentation at EPIC@ICCV2017 workshop, Oct. 2017c. Cited on page 8.

A. Cartas, M. Dimiccoli, and P. Radeva. Detecting hands in egocentric videos: Towards action recognition. In R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, editors, *Computer Aided Systems Theory – EUROCAST 2017*, pages 330–338, Cham, 2018a. Springer International Publishing. Cited on page 8.

A. Cartas, J. Marín, P. Radeva, and M. Dimiccoli. Batch-based activity recognition from egocentric photo-streams revisited. 21(4):953–965, Nov 2018b. Cited on pages 8, 50, and 52.

A. Cartas, J. Luque, P. Radeva, C. Segura, and M. Dimiccoli. How much does audio matter to recognize egocentric object interactions? Accepted for presentation at EPIC@CVPR2019 workshop, June 2019. Cited on page 9.

A. Cartas, J. Luque, P. Radeva, C. Segura, and M. Dimiccoli. Seeing and hearing egocentric actions: How much can we learn? In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4470–4480, 2019. Cited on page 9.

A. Cartas, E. Talavera, P. Radeva, and M. Dimiccoli. On the role of event boundaries in egocentric activity recognition from photo streams. Accepted for presentation at EPIC@ECCV2018 workshop, Sept. 2019. Cited on page 8.

A. Cartas, P. Radeva, and M. Dimiccoli. Activities of daily living monitoring via a wearable camera: Toward real-world applications. *IEEE Access*, 8:77344–77363, 2020. Cited on page 8.

D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, and I. Essa. Predicting daily activities from egocentric images using deep learning. In *Proceedings of the 2015 ACM International symposium on Wearable Computers*, pages 75–82. ACM, 2015. Cited on pages 18, 24, 30, 36, 40, 41, 50, 52, 57, 61, 68, 70, 76, and 78.

P. J. Chase. Algorithm 382: Combinations of m out of n objects [g6]. 13(6):368–, 06 1970. Cited on pages 33 and 53.

J. Cheng, O. Amft, and P. Lukowicz. Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In P. Floréen, A. Krüger, and M. Spasojevic, editors, *Pervasive Computing*, pages 319–336. Springer Berlin Heidelberg, 2010. Cited on page 18.

J. Cheng, O. Amft, G. Bahle, and P. Lukowicz. Designing sensitive wearable capacitive sensors for activity recognition. 13(10):3935–3947, Oct 2013. Cited on page 17.

K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. Cited on page 15.

F. Chollet. *Deep Learning with Python*, pages 219–221. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2017a. Cited on page 49.

F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807. IEEE Computer Society, 2017b. Cited on pages 15 and 73.

F. Chollet et al. Keras. `https://keras.io`, 2015. Cited on pages 34 and 55.

M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar. A survey on activity detection and classification using wearable sensors. *IEEE Sensors Journal*, 17(2):386–403, Jan 2017. Cited on page 11.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005. Cited on page 85.

D. Damen, T. Leelasawassuk, O. Haines, A. Calway, and W. Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. 2014. Cited on page 23.

D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. Cited on pages 23, 105, 107, 108, 112, and 113.

D.-T. Dang-Nguyen, L. Piras, M. Riegler, L. Zhou, M. Lux, and C. Gurrin. Overview of ImageCLEFlifelog 2018: Daily Living Understanding and Lifelog Moment Retrieval. In *CLEF2018 Working Notes*, CEUR Workshop Proceedings. CEUR-WS.org <http://ceur-ws.org>, September 10-14 2018. Cited on page 24.

H. Daumé, III and D. Marcu. Domain adaptation for statistical classifiers. 26(1): 101–126, 05 2006. Cited on page 24.

F. de la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. Technical report, April 2008. Cited on page 23.

J. Dezert. An introduction to the theory of plausible and paradoxical reasoning. In *Revised Papers from the 5th International Conference on Numerical Methods and Applications*, NMA '02, page 12–23, Berlin, Heidelberg, 2002. Springer-Verlag. Cited on page 19.

M. Dimiccoli, M. Bolaños, E. Talavera, M. Aghaei, S. G. Nikolov, and P. Radeva. Sr-clustering: Semantic regularized clustering for egocentric photo streams segmentation. 2017. Cited on pages 19, 45, 48, and 58.

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine*

Learning, volume 32 of *Proceedings of Machine Learning Research*, pages 647–655, Bejing, China, 22–24 Jun 2014. PMLR. Cited on page 77.

J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. Cited on pages 16 and 46.

J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, April 2017. Cited on pages 16, 46, and 47.

M. Ermes, J. Parkka, and L. Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4451–4454, Aug 2008. Cited on page 17.

M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. 111(1):98–136, 2015. Cited on page 92.

B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. Cited on pages 17 and 107.

C. Fan, Z. Zhang, and D. J. Crandall. Deepdiary: Lifelogging image captioning and summarization. 55:40 – 55, 2018. Cited on page 19.

C. Fanti. *Towards Automatic Discovery of Human Movemes*. PhD thesis, California Institute of Technology, Pasadena, CA, March 2008. Cited on page 12.

A. Fathi and J. M. Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586, 2013. Cited on pages 21 and 94.

A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 407–414. IEEE, 2011a. Cited on pages 21, 23, 82, and 83.

A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. pages 3281–3288, 2011b. Cited on pages 21, 82, and 91.

A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision (ECCV)*, pages 314–327. Springer, 2012. Cited on pages 21, 23, 82, 83, and 91.

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sep. 2010. Cited on page 20.

S. Fidler, G. Berginc, and A. Leonardis. Hierarchical statistical learning of generic parts of object structure. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 182–189, June 2006. Cited on page 14.

J. Foer. *Moonwalking with Einstein: The Art and Science of Remembering Everything*. Penguin Publishing Group, 2011. Cited on page 1.

F. Foerster, M. Smeja, and J. Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. 15(5):571 – 583, 1999. Cited on pages 4 and 18.

R. Fortet and E. Mourier. Convergence de la répartition empirique vers la répartition théorique. *Annales scientifiques de l'École Normale Supérieure*, 3e série, 70(3): 267–285, 1953. Cited on page 72.

K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119 – 130, 1988. Cited on page 14.

A. Furnari, G. M. Farinella, and S. Battiato. Temporal segmentation of egocentric videos to highlight personal locations of interest. pages 474–489, 2016. Cited on page 45.

A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, page 3201–3208, USA, 2011. IEEE Computer Society. Cited on page 12.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189. PMLR, 07–09 Jul 2015. Cited on pages 25 and 77.

A. Garcia del Molino, J.-H. Lim, and A.-H. Tan. Predicting visual context for unsupervised event segmentation in continuous photo-streams. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 10–17, New York, NY, USA, 2018. ACM. Cited on page 19.

G. García Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. Cited on page 20.

F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000. Cited on page 47.

D. Ghadiyaram, M. Feiszli, D. Tran, X. Yan, H. Wang, and D. Mahajan. Large-scale weakly-supervised pre-training for video action recognition. abs/1905.00561, 2019. Cited on pages 112 and 113.

B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, R. Krishna, V. Escorcia, K. Hata, and S. Buch. Activitynet challenge 2017 summary. abs/1710.08011, 2017. Cited on pages 17 and 107.

B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, V. Escorcia, R. Krishna, S. Buch, and C. D. Dao. The activitynet large-scale activity recognition challenge 2018 summary. abs/1808.03766, 2018. Cited on pages 17 and 107.

R. Girdhar, J. João Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2019. Cited on page 16.

R. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, ICCV '15, page 1440–1448, USA, 2015. IEEE Computer Society. Cited on page 15.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '14, page 580–587, USA, 2014. IEEE Computer Society. Cited on pages 15 and 20.

G. Gkioxari, R. Girshick, and J. Malik. Actions and attributes from wholes and parts. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 2470–2478. IEEE Computer Society, Dec 2015a. Cited on page 15.

G. Gkioxari, R. Girshick, and J. Malik. Contextual action recognition with R*CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1080–1088. IEEE Computer Society, 2015b. Cited on pages 15, 84, 86, and 93.

X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 513–520, USA, 2011. Omnipress. Cited on page 24.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, June 2012. Cited on page 77.

I. González Díaz, V. Buso, J. Benois-Pineau, G. Bourmaud, and R. Megret. Modeling instrumental activities of daily living in egocentric vision as sequences of active objects and context for alzheimer disease research. In *Proceedings of the 1st ACM International Workshop on Multimedia Indexing and Information Retrieval for Healthcare*, MIIRH '13, pages 11–14, New York, NY, USA, 2013. ACM. Cited on page 20.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in*

Neural Information Processing Systems 27, pages 2672–2680. Curran Associates, Inc., 2014. Cited on page 25.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602 – 610, 2005. IJCNN 2005. Cited on pages 15 and 48.

C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, and R. Albatal. Overview of ntcir-12 lifelog task. In N. Kando, K. Kishida, M. P. Kato, and S. Yamamoto, editors, *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, pages 354–360. National Institute of Informatics (NII), 2016. Cited on pages 5, 24, 29, 31, 44, 46, 50, 68, 70, and 76.

C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, R. Gupta, R. Albatal, N. Dang, and T. Duc. Overview of ntcir-13 lifelog-2 task. In *Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies*, pages 6–11. NTCIR, 2017. Cited on pages 24 and 50.

C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, V.-T. Ninh, T.-K. Le, R. Albatal, D.-T. Dang-Nguyen, and G. Healy. Advances in lifelog data organisation and retrieval at the ntcir-14 lifelog-3 task. In M. P. Kato, Y. Liu, N. Kando, and C. L. A. Clarke, editors, *NII Testbeds and Community for Information Access Research*, pages 16–28, Cham, 2019. Springer International Publishing. Cited on pages 24 and 50.

Hao Tian, Pang Lei, Li Xingjuan, and Xing Shusong. Wearable activity recognition for automatic microblog updates. In *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1720–1723, July 2009. Cited on page 17.

D. F. Harwath, A. Torralba, and J. R. Glass. Unsupervised learning of spoken language with visual context. In *NIPS*, 2016. Cited on page 17.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. Cited on pages 15, 33, 35, 55, 57, 72, 103, and 114.

H. Heffner and R. Heffner. Hearing ranges of laboratory animals. 46:20–2, 02 2007. Cited on page 104.

S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. 60:4 – 21, 2017. Regularization Techniques for High-Dimensional Data Analysis. Cited on pages 11 and 12.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012. Cited on page 16.

I. Hipiny and W. Mayol-Cuevas. Recognising egocentric activities from gaze regions with multiple-voting bag of words. (CSTR-12-003), 2012. Cited on page 21.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997. Cited on pages 15 and 46.

D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5 – 20, 1983. Cited on page 12.

B. Horowitz and A. Pentland. Recovery of non-rigid motion and structure. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 325–330, June 1991. Cited on page 12.

J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014. Cited on page 86.

J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. Cited on page 15.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, Jan 1962. 14449617[pmid]. Cited on page 13.

Y. Iwashita, A. Takamine, R. Kurazume, and M. S. Ryoo. First-person animal activity recognition from egocentric videos. In *2014 22nd International Conference on Pattern Recognition*, pages 4310–4315, Aug 2014. Cited on page 3.

S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. 35(1):221–231, Jan 2013. Cited on page 16.

T. Kao, C. Lin, and J. Wang. Development of a portable activity detector for daily activity recognition. In *2009 IEEE International Symposium on Industrial Electronics*, pages 115–120, July 2009. Cited on page 18.

S. Karaman, J. Benois-Pineau, R. Mégret, V. Dovgalecs, J. F. Dartigues, and Y. Gaëstel. Human daily activities indexing in videos from wearable cameras for monitoring of patients with dementia diseases. In *Proceedings - International Conference on Pattern Recognition*, pages 4113–4116, 2010. Cited on page 3.

S. Karaman, J. Benois-Pineau, V. Dovgalecs, R. Mégret, J. Pinquier, R. André-Obrecht, Y. Gaëstel, and J.-F. Dartigues. Hierarchical hidden markov model in detecting activities of daily living in wearable videos for studies of dementia. 69 (3):743–771, Apr 2014. Cited on page 3.

A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, June 2014. Cited on page 16.

E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5491–5500, 2019. Cited on page 23.

H.-J. Kim, J. S. Lee, and H.-S. Yang. Human action recognition using a modified convolutional neural network. In D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, editors, *Advances in Neural Networks – ISNN 2007*, pages 715–723, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. Cited on page 16.

G. King and L. Zeng. Logistic regression in rare events data. 9(2):137–163, 2001. Cited on pages 34 and 57.

K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR 2011*, pages 3241–3248, June 2011. Cited on page 21.

P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulò. Deep neural decision forests. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1467–1475, Dec 2015. Cited on page 30.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. Cited on pages 14, 34, and 73.

I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108(3):207 – 229, 2007. Special Issue on Spatiotemporal Coherence for Visual Motion Analysis. Cited on page 13.

O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. 15(3):1192–1209, Third 2013. Cited on pages 11, 17, 18, 31, 49, 50, 51, and 52.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. volume 2, pages 2169 – 2178, 02 2006. Cited on pages 20 and 22.

Y. LeCun and Y. Bengio. *Convolutional Networks for Images, Speech, and Time Series*, page 255–258. MIT Press, Cambridge, MA, USA, 1998. Cited on page 17.

Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann, 1990. Cited on page 14.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. Cited on page 14.

C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. Cited on page 25.

I. Lee, D. Kim, S. Kang, and S. Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1012–1020, Oct 2017. Cited on page 47.

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966. Cited on page 95.

C. Li and K. M. Kitani. Pixel-level hand detection in egocentric videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3570–3577. IEEE, 2013. Cited on pages 85 and 92.

Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295, 2015. Cited on pages 5, 22, 82, 83, 91, and 94.

Y. Li, M. Liu, and J. M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *The European Conference on Computer Vision (ECCV)*, September 2018. Cited on pages 21 and 23.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014. Cited on page 92.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 97–105. PMLR, 07–09 Jul 2015. Cited on pages 24, 69, 73, and 77.

M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 136–144, USA, 2016. Curran Associates Inc. Cited on page 24.

M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2208–2217. JMLR.org, 2017. Cited on page 25.

X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018a. Cited on page 17.

X. Long, C. Gan, G. Melo, X. Liu, Y. Li, F. Li, and S. Wen. Multimodal keyless attention fusion for video classification. 2018b. Cited on page 17.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. Cited on pages 21 and 85.

M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1903, June 2016. Cited on pages 22, 82, 83, 94, and 97.

S. Mann. An historical account of the "wearcomp" and "wearcam" inventions developed for applications in "personal imaging". In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, ISWC '97, page 66, USA, 1997. IEEE Computer Society. Cited on page 2.

K. Matsuo, K. Yamada, S. Ueno, and S. Naito. An attention-based activity recognition for egocentric video. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 565–570, June 2014. Cited on page 20.

U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, pages 4–116, April 2006. Cited on page 18.

W. W. Mayol and D. W. Murray. Wearable hand activity recognition for event summarization. In *Ninth IEEE International Symposium on Wearable Computers (ISWC'05)*, pages 122–129, Oct 2005. Cited on page 19.

W. Mayol-Cuevas, B. Tordoff, and D. Murray. Designing a miniature wearable visual robot. volume 4, pages 3725 – 3730 vol.4, 02 2002. Cited on page 2.

T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013. Cited on pages 20, 82, and 83.

D. Minnen, T. Westeyn, D. Ashbrook, P. Presti, and T. Starner. Recognizing soldier activities in the field. In S. Leonhardt, T. Falck, and P. Mähönen, editors, *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pages 236–241. Springer Berlin Heidelberg, 2007. Cited on page 18.

A. Muaremi, B. Arnrich, and G. Tröster. Towards measuring stress with smartphones and wearable devices during workday and sleep. 3:172–183, 2013. 89[PII]. Cited on page 2.

E. Munguia Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 37–40, Oct 2007. Cited on page 18.

K. Nakamura, S. Yeung, A. Alahi, and L. Fei-Fei. Jointly learning energy expenditures and activities using egocentric multimodal signals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-230255, 2017. Cited on page 23.

J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition*, pages 4694–4702, June 2015. Cited on pages 16, 46, and 47.

N. J. Nilsson. *The Quest for Artificial Intelligence*. Cambridge University Press, USA, 1st edition, 2009. Cited on page 16.

F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, Sep. 2005. Cited on page 16.

K. Ogaki, K. M. Kitani, Y. Sugano, and Y. Sato. Coupling eye-motion and ego-motion features for first-person activity recognition. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–7, June 2012. Cited on page 21.

K. Ohnishi, A. Kanehira, A. Kanezaki, and T. Harada. Recognizing activities of daily living with a wrist-mounted camera. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3103–3111, June 2016. Cited on pages 3 and 18.

D. Oneață. *Robust and Efficient Models for Action Recognition and Localization*. PhD thesis, Université Grenoble Alpes, France, 7 2015. Cited on page 12.

K. Ozcan, A. K. Mahabalagiri, M. Casares, and S. Velipasalar. Automatic fall detection and activity classification by a wearable embedded smart camera. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(2):125–136, June 2013. Cited on page 3.

D. Palmer. 22,000 london police are getting wearable cameras to video crime. *ZDNet*, Oct 2016. Cited on page 2.

J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen. Activity classification using realistic data from wearable sensors. 10(1):119–128, Jan 2006. Cited on page 17.

T. Pederson. *From Conceptual Links to Causal Relations — Physical-Virtual Artefacts in Mixed-Reality Space*. PhD thesis, Umeå University, Computing Science, 2003. Cited on page 20.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. Cited on page 35.

H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2847–2854. IEEE, 2012. Cited on pages 12, 20, 23, 82, and 83.

Y. Poleg, C. Arora, and S. Peleg. Temporal segmentation of egocentric videos. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2544, June 2014. Cited on pages 21 and 45.

Y. Poleg, A. Ephrat, S. Peleg, and C. Arora. Compact cnn for indexing egocentric videos. In *Applications of Computer Vision (WACV), IEEE Winter Conference on*, pages 1–9, 2016. Cited on page 23.

J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. In *arXiv:1503.00848*, March 2015. Cited on pages 84 and 86.

R. Poppe. A survey on vision-based human action recognition. *Image Vision Computing*, 28(6):976–990, June 2010. Cited on pages 11 and 12.

W. Price and D. Damen. An evaluation of action recognition models on epic-kitchens, 2019. Cited on page 18.

S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. 6(2):13:1–13:27, 03 2010. Cited on page 17.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015. Cited on pages 14, 34, 57, 72, 73, and 108.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. 77(1):157–173, May 2008. Cited on page 92.

M. Ryoo and L. Matthies. First-person activity recognition: Feature, temporal structure, and prediction. 119(3):307–328, 2016. Cited on pages 22 and 23.

M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2737, 2013. Cited on page 22.

T. Sainath and C. Parada. Convolutional neural networks for small-footprint keyword spotting. In *Interspeech*, 2015. Cited on pages 103 and 104.

T. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. pages 4580–4584, 04 2015. Cited on pages 20 and 104.

K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Adversarial dropout regularization. In *International Conference on Learning Representations*, 2018a. Cited on page 25.

K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b. Cited on page 25.

H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, pages 338–342, 2014. Cited on page 17.

C. Schmid. Recent progress in spatio-temporal action location. In the First NIPS Workshop on Large Scale Computer Vision Systems, 2016. Cited on page 12.

Seon-Woo Lee and K. Mase. Activity and location recognition using wearable sensors. 1(3):24–32, July 2002. Cited on pages 4 and 18.

T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 994–1000 vol. 2, June 2005. Cited on page 14.

Y. Shiga, T. Toyama, Y. Utsumi, K. Kise, and A. Dengel. Daily activity recognition combining gaze motion and visual features. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1103–1111. ACM, 2014. Cited on page 21.

G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. 2018. Cited on page 23.

P. Siirtola and J. Röning. Recognizing human activities user-independently on smartphones based on accelerometer data. 1(5):38–45, 06/2012 2012. Cited on page 18.

K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576. 2014a. Cited on pages 16, 94, and 104.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. abs/1409.1556, 2014b. Cited on pages 14, 15, 33, 34, 55, 93, 95, 103, 104, and 107.

S. Singh, C. Arora, and C. V. Jawahar. First person action recognition using deep learned descriptors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. Cited on pages 22, 82, and 83.

S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. S. Babu, P. P. San, and N.-M. Cheung. Multimodal multi-stream deep learning for egocentric activity recognition. In *Workshop on Egocentric (First-person) vision, in conjunction with the International Conference on Computer Vision and Pattern Recognition*. IEEE, 2016a. Cited on page 22.

S. Song, N.-M. Cheung, V. Chandrasekhar, B. Mandal, and J. Liri. Egocentric activity recognition with multimodal fisher vector. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2717–2721. IEEE, 2016b. Cited on page 22.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. Cited on pages 34 and 55.

A. Storkey. When training and test sets are different: Characterising learning transfer. In J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, editors, *Dataset Shift in Machine Learning*, chapter 1, pages 3–28. MIT Press, 2009. Cited on page 24.

S. Sudhakaran and O. Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *Proceedings of the British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 229, 2018. Cited on pages 20, 82, 83, and 94.

S. Sudhakaran, S. Escalera, and O. Lanz. FBK-HUPBA submission to the epic-kitchens 2019 action recognition challenge. abs/1906.08960, 2019a. Cited on pages 112 and 113.

S. Sudhakaran, S. Escalera, and O. Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019b. Cited on pages 20, 111, 112, and 113.

B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 443–450, Cham, 2016. Springer International Publishing. Cited on pages 25, 68, 69, 70, 73, and 77.

B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2058–2065. AAAI Press, 2016. Cited on pages 25 and 70.

D. Surie, T. Pederson, F. Lagriffoul, L.-E. Janlert, and D. Sjölie. Activity recognition using an egocentric perspective of everyday objects. In J. Indulska, J. Ma, L. T. Yang, T. Ungerer, and J. Cao, editors, *Ubiquitous Intelligence and Computing*, pages 246–257. Springer Berlin Heidelberg, 2007. Cited on page 20.

I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), page 757–764, New York, NY, USA, 1968. Association for Computing Machinery. Cited on page 2.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. Cited on page 15.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. Cited on pages 33 and 34.

E. Talavera, M. Dimiccoli, M. Bolaños, M. Aghaei, and P. Radeva. R-clustering for egocentric video segmentation. In *Pattern Recognition and Image Analysis*, pages 327–336. Springer, 2015. Cited on pages 19 and 45.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. Cited on page 16.

D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. Cited on page 16.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2962–2971, 2017. Cited on page 25.

J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. 104(2):154–171, 2013. Cited on pages 84 and 86.

K. Vandecasteele, T. De Cooman, Y. Gu, E. Cleeren, K. Claes, W. V. Paesschen, S. V. Huffel, and B. Hunyadi. Automated epileptic seizure detection based on wearable ecg and ppg in a hospital environment. 17(10):2338, Oct 2017. PMC5676949[pmcid]. Cited on page 2.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. 2017. Cited on page 16.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, March 1989. Cited on page 16.

H. Wang and C. Schmid. Action recognition with improved trajectories. In *2013 IEEE International Conference on Computer Vision*, pages 3551–3558, Dec 2013. Cited on page 22.

H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103 (1):60–79, 2013. Cited on page 22.

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36, 2016. Cited on pages 16, 94, and 103.

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11):2740–2755, Nov 2019. Cited on page 16.

X. Wang, Y. Wu, L. Zhu, and Y. Yang. Baidu-uts submission to the epic-kitchens action recognition challenge 2019. abs/1906.09383, 2019. Cited on pages 112 and 113.

Y. Watanabe, T. Hatanaka, T. Komuro, and M. Ishikawa. Human gait estimation using a wearable camera. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 276–281, Jan 2011. Cited on page 3.

D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224 – 241, 2011. Cited on page 11.

M. Weiser. The computer for the 21 st century. *Scientific American*, 265(3):94–105, 1991. Cited on pages 1 and 3.

M. Weiser. Ubiquitous computing. *Computer*, 26(10):71–72, Oct. 1993. Cited on page 1.

Z. L. Wenhao Wu, Wenbo Chen. Samsung & siat submission to activitynet challenge 2018. In the ActivityNet Large Scale Activity Recognition Challenge, at the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2018. Cited on page 17.

Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 791–800, New York, NY, USA, 2016. ACM. Cited on page 17.

L. Xia, I. Gori, J. K. Aggarwal, and M. S. Ryoo. Robot-centric activity recognition from first-person rgb-d videos. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 357–364, Jan 2015. Cited on page 3.

K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR. Cited on page 15.

K. Yamada, Y. Sugano, T. Okabe, Y. Sato, A. Sugimoto, and K. Hiraki. Attention prediction in egocentric video using motion and visual saliency. In Y.-S. Ho, editor, *Advances in Image and Video Technology*, pages 277–288, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. Cited on page 20.

J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, June 1992. Cited on pages 12 and 13.

H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. Cited on page 24.

T. Yao and X. Li. YH technologies at activitynet challenge 2018. *CoRR*, abs/1807.00686, 2018. Cited on page 17.

C. Yu and D. H. Ballard. Understanding human behaviors based on eye-head-hand coordination. In H. H. Bülthoff, C. Wallraven, S.-W. Lee, and T. A. Poggio, editors, *Biologically Motivated Computer Vision*, pages 611–619. Springer Berlin Heidelberg, 2002. Cited on page 22.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. Cited on page 105.

H. Yu, W. Jia, Z. Li, F. Gong, D. Yuan, H. Zhang, and M. Sun. A multisource fusion framework driven by user-defined knowledge for egocentric activity recognition. 2019(1):14, 2019a. Cited on page 19.

H. Yu, G. Pan, M. Pan, C. Li, W. Jia, L. Zhang, and M. Sun. A hierarchical deep fusion framework for egocentric activity recognition using a wearable hybrid sensor system. 19(3), 2019b. Cited on pages 18 and 19.

C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In F. A. Hamprecht, C. Schnörr, and B. Jähne, editors, *Pattern Recognition*, pages 214–223. Springer Berlin Heidelberg, 2007. Cited on page 103.

H. F. M. Zaki, F. Shafait, and A. Mian. Modeling sub-event dynamics in first-person action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1619–1628, July 2017. Cited on page 22.

E. Zarepour, M. Hosseini, S. S. Kanhere, A. Sowmya, and H. R. Rabiee. Applications and challenges of wearable visual lifeloggers. *Computer*, 50(3):60–69, 2017. Cited on page 2.

W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. Cited on page 25.

W. Zellinger, B. A. Moser, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz. Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, 483:174 – 191, 2019. Cited on pages 25, 69, 73, and 77.

K. Zhan, S. Faux, and F. Ramos. Multi-scale conditional random fields for first-person activity recognition on elders and disabled patients. 16:251 – 267, 2015. Selected Papers from the Twelfth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2014). Cited on page 3.

X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4498–4507, June 2016. Cited on page 20.

X. Zhang, Y. Bao, F. Zhang, K. Hu, Y. Wang, L. Zhu, Q. He, Y. Lin, J. Shao, and Y. Peng. Qiniu submission to activitynet challenge 2018. abs/1806.04391, 2018. Cited on page 17.

Y. C. Zhang and J. M. Rehg. Watching the tv watchers. 2(2):88:1–88:27, 07 2018. Cited on page 3.

Y. Zhao, B. Zhang, Z. Wu, S. Yang, L. Zhou, S. Yan, L. Wang, Y. Xiong, D. Lin, Y. Qiao, and X. Tang. CUHK & ETHZ & SIAT submission to activitynet challenge 2017. In the ActivityNet Large Scale Activity Recognition Challenge, at the International Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, 2017. Cited on page 17.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, June 2016. Cited on page 20.