Temporal Labelling for Action Recognition in Videos

By

DAVIDE MOLTISANTI

University of BRISTOL

Department of Computer Science

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

OCTOBER 2019

Word count: **53,757**

Time is a lie
to limit our mind

The Swan With Two Necks

Abstract

Action recognition in computer vision is the task of understanding what a subject is doing in an environment. When performing recognition in videos, labels are typically provided in the form of a category class, along with the temporal boundaries of the action. These bounds delimit the span of the action in the video, allowing algorithms to build a representation of the action from the visual content of the video segment.

Labelling action boundaries entails that an annotator decides when the action starts and ends. This is a subjective and arbitrary task, i.e. different people are likely to identify the start and the end of an action differently. As action boundaries vary, salient and irrelevant video frames are included or excluded, thus the ability of a classifier to learn and detect actions may be influenced. This Thesis offers an insight into how action boundaries are perceived and how they can affect classification in videos. An important finding of this study is that accurate temporal labelling is crucial to learn discriminative representations of the actions, using current state-of-the-art methods. Indeed, classification results fluctuate as temporal bounds are altered. This Thesis also proposes the Rubicon Boundaries, annotation guidelines inspired by work in cognitive psychology that aim to alleviate labelling ambiguity, in the attempt to foster more precise and consistent annotations.

Action boundaries are not only arbitrary, but also expensive to annotate. Given that video datasets are growing increasingly larger, there is an intrinsic need for scaling the labelling process up. This Thesis proposes a novel level of temporal supervision for the task of action recognition, i.e. single timestamps roughly aligned with actions in untrimmed videos. Using this type of supervision, together with the proposed training algorithm, it is possible to achieve performance comparable to results obtained with full temporal supervision. The proposed method can operate under varying dataset complexity, highlighting that single timestamps constitute a good compromise between labelling effort and performance. Additionally, single timestamps also alleviate ambiguity, since annotators do not have to decide when the action starts and ends, but only to mark one frame within or close to the action.

Dedication and acknowledgements

I would like to express my utmost gratitude to my supervisor, Dr. Dima Damen. Dima is not only a passionate and competent researcher, but most importantly a thoughtful, dedicated and generous mentor. During these four years, I learnt so much about research from working with her. A sincere *thank you* is due.

I dedicate this Thesis to my parents and my brother, Concetta, Pippo and Marco, who have always being loving and caring despite the long distance. Thank you also for bringing some sun to the UK when visiting me!

A special thanks goes to all my friends, who made my years in Bristol an amazing time of my life. Thanks to Nicolás, for our endless conversations, the ping-pong, and above all his genuine friendship. Thanks to Alfredo and Charlie, my fantastic bandmates, for the joy of playing together during these years. The Swan With Two Necks will always be a great band.

Thank you Ruru, for the fond affection and support, for being an adorable person, for the marvellous time we had and hopefully will have soon together.

Thanks to my flatmates, Laurie, Gabriela and Lorenzo, for making our house a lovely home, for lending an ear and a hand whenever needed. Thank you Laurie, for the engaging discussions about science and for the home-made cinema, and thank you Gabriela and Lorenzo for the delicious food and the nice parties.

Thanks to all the people in the VILab who made our working place a fun one: Mike, Laurie, Hazel, Will, Eduardo, Jonny, Vangelis, Young, Ramón, Abel, Perla, Xingrui, Yanan, Jian, Miguel, Janis, Erik, Sasha, Farnoosh, Faegheh, Toby, Sam, Vahid, Austin, Richard, Miltos and Zeynel. A particular thanks goes to Mike, for having being a very nice companion during our contemporary journeys, to Hazel, for having many good laughters and drinks together, and Will, for his patient and often spontaneous help on many technical questions.

Thanks also to Fabio, Matilde, Giulia, Nanni, Gabriele, Antonino, Valerio, simply for being friends, however far they are.

LIST OF PUBLICATIONS

Some parts of the work presented in this Thesis have been published in the following papers:

- Michael Wray*, Davide Moltisanti*, Walterio Mayol-Cuevas, Dima Damen. **SEMBED: Semantic Embedding of Egocentric Action Videos**. In *European Conference on Computer Vision Workshops (ECCVW) - 2016;*
- Davide Moltisanti, Michael Wray, Walterio Mayol-Cuevas, Dima Damen. **Trespassing the Boundaries: Labelling Temporal Bounds for Object Interactions in Egocentric Video**. In *International Conference on Computer Vision (ICCV) - 2017;*
- Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, William Price, Michael Wray. **Scaling Egocentric Vision: The EPIC-Kitchens Dataset**. In *European Conference on Computer Vision (ECCV) - 2018*;
- Davide Moltisanti, Sanja Fidler, Dima Damen. **Action Recognition from Single Timestamp Supervision in Untrimmed Videos**. In *Conference on Computer Vision and Pattern Recognition (CVPR) - 2019*;

*Equal contribution.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Table of Contents

LIST OF TABLES

List of Figures

Figure **Page**

Chapter 1

Introduction

Action recognition in computer vision is the task of understanding what a subject is doing in an environment. This can be done by analysing various types of data, such as videos, static images, motion capture, etc. Action recognition is a classification problem. Given some labelled training data, a classifier is trained to recognise actions from the data. When performing action recognition in videos, labels are typically provided in the form of a semantic description of the action (i.e. a category class), along with the temporal boundaries of the action. These bounds delimit the span of the action in the video, allowing algorithms to learn discriminative features from the visual content of the video segment and build a representation of the action.

Labelling action boundaries in videos entails that a person decides when the action starts and ends in the video. From a human perspective, this might appear to be a trivial task. However, delimiting the span of an action is subjective, i.e. different people are likely to identify the start and the end of an action differently. This matter has been investigated in the field of cognitive psychology [38]. Psychologists analyse and decompose the course of an action to understand the cognitive processes involved in the intention and goal that drive an individual to perform an action.

In computer vision, action temporal boundaries have long been taken for granted. In fact, authors typically provide little insight into how actions are annotated in the published videos. A few issues arise from this oversight. Intuitively, when temporal boundaries vary, the visual data capturing the action varies too, i.e. salient or irrelevant frames are included or excluded depending on how the action is delimited in the video. While this generally does not constitute a problem to humans, classifiers can be sensitive to even minor boundary variations. The first finding of this Thesis is that accurate temporal delimitation is crucial to learn discriminative representations of the action. This implies that performance can deteriorate when bounds are imprecise, and accordingly that higher accuracy can be achieved by improving temporal labelling.

Considering that temporal bounds are subjective and that action recognition frameworks are designed and compared on such arbitrary labelling, it is important to shed some light onto this overlooked matter. This Thesis offers a thorough insight into how action boundaries are perceived by people and how they can affect classification in videos. In Chapter 4, the subjectivity of marking actions' start and end times is inspected by analysing multi-annotator labelling. Boundaries inconsistency in existing datasets is also scrutinised, before assessing how temporal boundaries variations impact classification performance. The Chapter also proposes The Rubicon Boundaries, annotation guidelines inspired by work in cognitive psychology that aim to alleviate labelling ambiguity, in the attempt to foster more precise and consistent annotations.

Delimiting the temporal span of actions is not only subjective, but importantly requires laboursome and expensive work. Until recently, datasets contained a relative small number of videos and actions, thus it was possible, although tedious, for a single person or research laboratory to annotate the entirety of a dataset in a reasonable amount of time. With the advent of deep learning pushing towards the need for large datasets, and with the explosion of video sharing platforms witnessed in the last few years, datasets have been expanding at a rapid pace in recent times. Authors nowadays commonly resort to crowd-sourcing to label videos. This has proven to be a scalable solution, however it also involves a painstaking endeavour to ensure sufficient labelling quality, given that crowd-sourced annotations are typically noisy. Controlling the quality of video annotations becomes remarkably impractical when dealing with the sheer size of current datasets (Kinetics [11], the largest video dataset to date, contains more than 650,000 videos).

Recent research adopts weak temporal supervision for various video understanding tasks. A well established line of works has achieved performances comparable to traditional fully supervised methods, despite using weaker and less expensive annotations. Nevertheless, datasets have not simply grown larger in the number of videos, but have also become more complex. This means that datasets now contain a greater number of actions per video, with actions belonging to a larger pool of classes that involve complicated temporal dynamics. Videos are also captured in unconstrained settings (e.g. the Activity of Daily Living [79] and EPIC Kitchens [21] datasets collect unscripted videos recorded in native environments), which increases complexity as well. As we will see, successful weakly supervised approaches are challenged under such conditions.

Within this context, this Thesis proposes a novel level of temporal supervision for the task of action recognition, i.e. single timestamps roughly aligned with actions in untrimmed videos. Chapter 6 shows that using this labelling, together with the proposed training algorithm, it is possible to achieve performance comparable to results obtained with full temporal supervision. Importantly, the Chapter also shows that the proposed method can operate under varying dataset complexity, highlighting that single timestamps constitute a good compromise between labelling effort and performance.

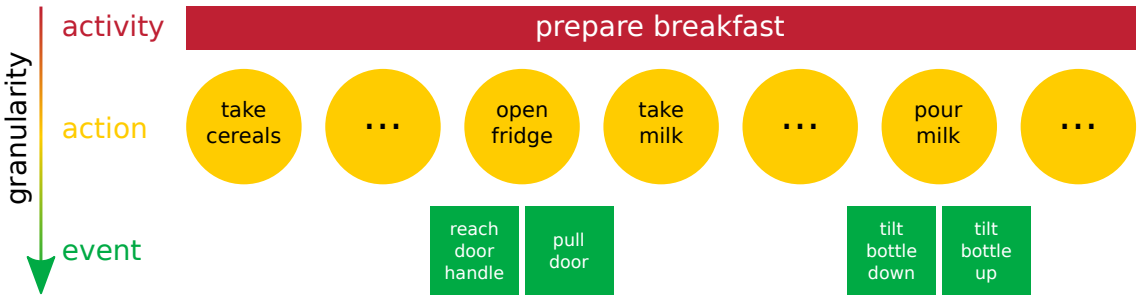

Figure 1.1: Defining activities, actions and events based on the temporal granularity of the interaction between a subject and the world.

The Thesis is organised as follows. Chapter 2 reviews works concerning temporal boundaries for action recognition in videos, as well as weakly supervised approaches for video understanding. Chapter 3 surveys important datasets and frameworks for action recognition. As data collection is an important part of research, Chapter 3 additionally discusses the collection and the annotation of two datasets (BEOID [20] and EPIC Kitchens) I was involved in during my PhD. Chapter 4 talks about temporal boundaries, how these are ambiguous to define and how this relates to classification. Chapter 4 also proposes labelling guidelines in the attempt to assist annotators in marking action bounds. Chapter 4 contains work published in [73]. Chapter 5 revisits the sensitivity of recent models to action bounds variations. Chapter 6 proposes single timestamp supervision for training classifier from untrimmed videos. This work was published in [74]. Finally, Chapter 7 summarises the findings of this Thesis and provides some insights into future research directions.

1.1 Definitions

This Section defines some terminology that will be used throughout the Thesis. Since this Thesis concerns action, videos and time, the temporal granularity of human interactions is first discussed.

1.1.1 Activities, Actions and Events

There are a few possible ways to decompose the temporal granularity of an interaction between a person and the external world. A possible decomposition of human-world interactions could devise three hierarchical categories: **activities**, which are composed of **actions**, which in turn are composed of **events**. Activities refer to a generic task (e.g. preparing tea, painting a wall, etc.) which is accomplished by means of several actions. Actions can be further broken down into events, i.e. an atomic motion a particular action requires in order to be carried out.

Figure 1.1 shows an example of such hierarchical definition. The Figure illustrates the task of preparing cereals with milk for breakfast. Such activity might be decomposed into the following actions: take cereals, then open the fridge and take some milk, then pouring the milk, etc. Each of these actions may be split into multiple events: for example, opening a fridge entails first reaching

the door handle and pulling then the door, while in order to pour milk a person needs to tilt the bottle twice, the first time to let the milk out of the bottle, and the second time to stop the milk flowing. Notice that other possible ways of decomposing human-world interactions can be devised. In fact, there is generally no agreement in the literature as to how to temporally decompose interactions, yet a coherent terminology is needed. This Thesis adopts the activity-action-event hierarchy.

Different granularities entail very distinct temporal extents. In fact, the length of an activity is usually in the order of minutes, whereas actions and events are significantly shorter, spanning from fractions of a second to few seconds. As a consequence, approaches designed to recognise activities are different from those designed for recognising actions, due to the visual content varying at intrinsically different scales. This Thesis concerns action recognition. As events and actions are close in the temporal scale, the distinction between actions and events is often subtle and based on semantic labelling. Events are thus within the scope of this Thesis as well.

1.1.2 Temporal Labels

For the task of action recognition, labels are usually provided in the form of action class and temporal bounds. The former is typically given as a combination of verbs and nouns that describe the action, while the latter specify the start and end times of the action in the video. Notice that datasets often contain trimmed videos enclosing one action only. This Thesis regards temporal labelling, thus focuses on cases where videos are untrimmed. Temporal boundaries will be referred to as full temporal supervision in this Thesis.

Weaker temporal labels are also used for action recognition. The weakest temporal cue is given by **video-level** labels, which only tag the presence of an action in the video. Following in the rank of temporal cue strength are **action sets**, which annotate all the actions contained in a video, with no ordering information. **Action transcripts** provide an ordered sequence of all the actions in the video, but still without start and end timestamps. Finally, **single timestamps**, proposed in Chapter 6, signal the rough temporal location of an action occurring in a certain proximity, with only one timestamp attached to each action.

1.1.3 Temporal Intersection Over Union

Intersection Over Union (IOU) is a metric commonly used to measure the spatial overlap between two rectangles in an image. In the video domain, IOU measures the temporal overlap of two segments extracted from a video:

$$\mathrm{IOU}\left(V_a^b, V_c^d\right) = \frac{\left|\{V_a, V_{a+1}, \ldots, V_b\} \cap \{V_c, V_{c+1}, \ldots, V_d\}\right|}{\left|\{V_a, V_{a+1}, \ldots, V_b\} \cup \{V_c, V_{c+1}, \ldots, V_d\}\right|} \tag{1.1}$$

where V_a^b and V_c^d denote two sets of contiguous frames indices relative to the same video V, with (a, c) and (b, d) respectively indicating the first and last frame indices of the two segments. IOU

Figure 1.2: Comparison between first-person (left) and third-person (right) videos.

ranges between 0 and 1, with higher values corresponding to larger overlaps. When IOU equals 1, V_a^b and V_c^d overlap completely (i.e. they contain the very same frames), whereas when IOU equals 0, they do not overlap at all.

1.1.4 First-person and Third-person Videos

First-person (or egocentric) videos are recorded by means of a wearable camera. Cameras are typically fastened to the head or the chest of the user, allowing a unique perspective of what the camera wearer is doing, who and what they are interacting with, where they are located. The egocentric viewpoint is especially advantageous for capturing object interactions, since it gives a close picture of how object are manipulated. At the same time, several difficulties arise from the camera-wearer bond. Because cameras are attached to people who usually move freely within an environment, videos tend to be unsteady and to exhibit abrupt appearance changes.

Third-person videos are recorded with a fixed camera or by a person not involved in the action. As the recorder is not participating in the actions, cameras are typically more stable, which mitigates the jitter often visible in egocentric videos. Third-person videos generally provide a wider view of the scene. Figure 1.2 illustrates two frames extracted from a first-person (left-hand side) and a third-person video (right-hand size).

1.1.5 Acronyms

- SVM: Support Vector Machine;
- HMM: Hidden Markov Model;
- RNN: Recurrent Neural Network;
- CNN: Convolutional Neural Network;
- LSTM: Long Short-Term Memory;
- IOU: Intersection Over Union;
- MAP: Mean Average Precision.
- PCA: Principal Component Analysis.

Chapter 2

Related Work

This Chapter reviews papers closely related to the work presented in this Thesis. Section 2.1 gives an overview of works that concern temporal boundaries for action recognition in videos. More precisely, Section 2.1 reviews works that discuss the subjectivity of annotating action boundaries and the issues related to variations in such bounds. This topic is revisited in detail in Chapter 4.

Section 2.2 reviews weakly supervised approaches for video understanding. Namely, Section 2.2 surveys works that use video-level, transcript and point supervision (in the form of a single pixel or frame) for the tasks of action recognition, localisation and segmentation. Chapter 6 returns to weak supervision, presenting an approach for action recognition in untrimmed videos that uses single timestamps annotations.

Chapter 3 reviews additional related work, scrutinising various methods for action representation and classification, both before and after the advent of deep learning. Chapter 3 also offers an insight into several video datasets for action recognition.

2.1 Temporal Boundaries for Object Interactions

Defining the beginning and the ending of an action is a highly subjective matter. As noted by previous work [2, 14, 32, 86, 93, 111], when different people are asked to identify the temporal extent of an action they will likely recognise its start and end differently. This is because the perception of when an action starts and ends varies between individuals. For example, consider the action of washing a cup showed in Figure 2.1. It is subjective to say whether the action starts at frame 1, 2 or 3, when the person is pouring some washing up liquid into the cup (1), turning the tap on (2) or lathering the cup (3). Likewise, it is subjective to say whether the action ends at frame 6 or 7, when the subject turns the tap off (6) or puts the cup onto the drying rack (7).

Figure 2.1: The perception of the start and the end of an action is a subjective matter. When does the action of washing the cup begin and end?

Ambiguity in temporal boundaries entails potential disagreement amongst annotators who delimit the scope of the actions contained in videos. As we will see in Chapter 4, annotators disagreement and inconsistent bounds relates to the robustness of a classifier, i.e. the susceptibility of a recognition algorithm to variations in the start and end times that enclose action frames. This Section reviews prior work that concerned the subjectivity of the temporal scope of an action or addressed the robustness of methods for action recognition in videos.

* * *

The problem of annotators agreement and its impact on the design, performance and evaluation of computer vision algorithms has received much more attention on still image tasks compared to video-based ones. Lampert et al. [58] observe that most semantic segmentation algorithms are tailored to the available ground truth, and highlight that in most cases such ground truth is provided by a single annotator. Noting that algorithms are thus evaluated on the decision of a single annotator, the authors extensively assess the impact of ground truth variability, i.e. annotator disagreement, for the task of semantic segmentation in images. Firstly, they correlate the level of disagreement in labelling pixels to several visual properties of the images (e.g. contrast and intensity), and find that a strong correlation appears in most cases. Secondly, the authors also highlight that the performance of detection algorithms increases as the agreement between annotators increases. Intuitively, this corresponds to the fact that a good segmentation algorithm is able to highlight discriminative regions of the image which, in turn, are also more likely for different annotators to agree on. The authors conclude questioning the fact that most approaches are evaluated using labels coming from a single annotator. While they do not propose a methodology to combine multiple annotations, nor provide guidelines on how to assist people to recognise relevant information in images, the proposed evaluation framework sheds some critic light on the way algorithms are often biased by arbitrary annotations.

Similarly and more recently, Tanno et al. [102] too observe that the performance of recognition algorithms are directly influenced by the annotators agreement. The authors note that the disagreement stems from the different annotators' skills and biases, and remark that treating noisy labels without precaution hinders the efficacy of learning algorithms. The authors thus propose a method to learn individual annotator models in a multiple annotators setting. By modelling each annotator, they also estimate the distribution of the "true" labels. The authors focus on image recognition, i.e. classifying a single image into a predefined set of classes. For this task, the truth of a label is arguably

less ambiguous compared to more complicated tasks such as semantic segmentation. This is because describing an object in a still image is less arbitrary than recognising the spatial extent of the object.

Satkin and Hebert [86] were amongst the first to note that determining the temporal extent of an action is subjective. Interestingly, they compare temporal boundaries to object boundaries, stating that while one can often unambiguously identify the boundary between an object and its surrounding background, the same does not always apply to the start and the end of an action. The authors observe that training segments are typically cropped in a qualitative manner based on the semantic definition of the action, and note that the annotation task is not only deemed a very difficult one by previous work [16, 60], but importantly it becomes laborious and impractical when dealing with large datasets.

The authors avoid any temporal labels altogether, and propose instead a method to automatically determine the extent of a single action from a training video, via iterative cropping, in order to optimise the performance of the classifier. More specifically, they define a crop to be of optimal length if a shorter segment would not enclose the most discriminative part of the action, while a longer segment would increment noise by including irrelevant frames. Given a training video, their approach starts by splitting the video considering all the possible start/end combinations for the action of interest. Using Histograms of Oriented Gradients [18], Histogram of Optical Flow [60] and Point-Trajectory Features [68], they exhaustively search for the crop that leads to the highest classification accuracy. Their results show that the performance of the recognition system varies considerably depending on how the training samples are trimmed.

Satkin and Hebert [86]'s work was motivated by the fact that the temporal scope of an action is ill-defined. Importantly, their approach aims to refine the boundaries of a single action in a video, and thus assumes that the videos do not contain multiple actions. Moreover, given the exhaustive search based on all start/end combinations, the videos were also expected to be relatively short. These two assumptions were satisfied in the evaluated datasets, the University of Rochester Activities of Daily Living dataset [69] and the Hollywood 2 dataset [67], as well as in most datasets available at the time (2010).

How many frames? Schindler and Van Gool [87] tackled the question of how many frames are needed for a machine to recognise a simple action, such as walking, running or jumping. They extract both shape and motion features from short video segments ranging from 1 to 10 frames, respectively using Gabor filters and optical flow. These features are given in input to an SVM to classify the actions. Their main finding is that very short snippets (1-7 frames) are sufficient to achieve high recognition accuracy, with rapidly diminishing returns as more frames are added. Nevertheless, the evaluation was conducted on the Weizmann [7] and the KTH [59] datasets. These datasets, although being the de-facto standard benchmarks at the time, contain few basic actions and are nowadays considered easy to optimise on.

More recently, a similar conclusion has been drawn by Yang and Tian [118], who proposed a method to recognise human actions using 3D skeleton joints. The authors show that the first 30-40%

Figure 2.2: Actom annotaion example for the actions “sit down” and “open door”. Figure from [32].

of the video (i.e. 15-20 frames) are sufficient to achieve results comparable to those obtained using the entire sequence. Notice that both [87] and [118] assessed *how long* a sequence should be in order to obtain good results, but neither of the two evaluated the impact of varying the beginning and the ending times separately, taking these for granted.

Alternative annotations and representations Gaidon et al. [32] deviate from the standard labelling pipeline which entails annotating the start and end times of an action. They instead propose to annotate a small set of frames called *actoms*, for each action instance. An actom is a frame in a video such that its neighbouring frames are visually representative of a portion of the action. Figure 2.2 shows an example of actom annotations for two actions. Annotators were asked to label a minimum amount of key moments (at least two) that unambiguously identify the action, without receiving any instructions to recognise these key moments. Actoms should be annotated so that they are semantically consistent between different videos, i.e. the i-th actom of a given action in a video should identify the same i-th key moment of the same action in a different video. The authors define the temporal span of an actom with an adaptive radius around its timestamp. Such radius is used to pool visual features around the actom, which are encoded using a bag of features approach. To localise actions in a test video, an SVM classifier is used with a sliding window.

The authors claim that actom labelling is more consistent than the standard start/end times approach. However, by not instructing annotators how to recognise the key moments of an actions, and hence the corresponding actoms, this labelling is still potentially as subjective as traditional temporal bounds. Furthermore, the authors also note that not all actions can be easily decomposed into actoms. This is especially the case for high speed actions, like clicking a button, and complex activities, like cooking a meal. For these cases the authors propose to label actoms at evenly spaced intervals between the annotated temporal boundaries. Again, given that no guidance to recognise the extent of an action is offered, the problem of subjective bounds remains unsolved.

Wang et al. [112] argue that the essence of an action lies in the change that an action applies to the interacted environment and objects, as previously noted also by [27]. Based on this, they represent actions as a transformation from a precondition state to an effect state. The authors use a Siamese network with two towers encoding the two states. Each tower is fed with frames extracted from the corresponding precondition and effect portions of the action. The inner boundary between

the two parts is however estimated with a brute force approach, and standard start/end timestamps are used to train the model.

Richard et al. [81] proposed a weakly supervised approach to recognise actions in videos using only the ordered sequence of actions for training (i.e. action transcripts). They notice that action labels typically describe a task that involves multiple movements of the subject. For example, they note that the action "take cup" might implicate approaching and opening the cupboard, picking up a cup and placing the cup somewhere. Again, this reflects how ill-posed the definition of what an action is and how arbitrary the granularity of a human-world interaction can be. In fact, if we define an action as a single object interaction, then we could argue that "take cup" would only entail picking the cup up, and that opening the cupboard and putting the cup down are separate, distinct actions.

The authors do not focus on the definition of an action. They instead propose to decompose actions into sub-actions, following their interpretation mentioned above. These sub-actions are treated as latent variables to be learnt. More precisely, an HMM is designed to represent the sequence of sub-actions, for each action, and sub-action probabilities are modelled using the class predictions produced by an RNN that receives frame-wise features in input.

Evaluating annotators disagrement Sigurdsson et al. [93][1] pose a few questions on how human actions in videos should be reasoned about. As similarly noted by [86], they observe that unlike objects, whose physical boundaries and semantic categories are overall well defined, object interactions are more difficult to detect and categorise unambiguously. In order to gauge how much different people disagree when identifying the start and the end of actions in videos, the authors asked multiple annotators to label a set of actions from the Charades [91] and the Multi-THUMOS [119] datasets, and compare the collected annotations to the ground-truth bounds. Their experiment showed that people disagree to a large extent. Indeed, the authors report an average IOU of 72.5% (Charades) and 58.7% (Multi-THUMOS) between the crowd-sourced and the published annotations. The authors highlight a correlation between the action length and the amount of disagreement amongst annotators, suggesting that longer actions and activities are easier to temporally localise. Additionally, they notice that annotators tend to be more consistent with each other in identifying the start of an action, especially for longer activities, compared to the end which most people struggle to coherently recognise.

Based on these findings, the authors question the way action localisation is evaluated, i.e. measuring the overlap between a single arbitrary annotation and the action extent predicted by a model. To investigate on this matter, they refine the temporal bounds of actions in testing videos using the crowd-sourced labels, and evaluate the refined test set with several baselines (Two Steam CNN [95], IDT [108], LSTM on top of VGG-16 [96], Action VLAD [37] and Asynchronous Temporal Fields [92]), observing an increase in localisation accuracy compared to the original annotations. The authors

[1] Concurrent to the work presented in Chapter 4, also published in ICCV 2017.

Figure 2.3: Asking multiple annotators to guess the action and its temporal extent, given a few frames. Figure from [2].

conclude their work evaluating what state-of-the-art recognition approaches learn, analysing how classes size, motion, temporal extent/context and actors presence impact performance.

Alwassel et al. [2] [2] developed a novel diagnostic tool to thoroughly evaluate action localisation performance, going beyond the standard single-scalar metric MAP that is typically used to evaluate the predicted location of actions in videos. Similarly to [93], the authors investigate whether humans can precisely locate an action in a video. Figure 2.3 shows a motivational example from [2]. Different people were asked to guess the action depicted in the Figure, as well as to decide where the action ends. Interestingly, while an unanimous response was collected for the first question, a certain disagreement was discovered for the second question, with 67% of the people choosing frame B from the right-hand side of Figure 2.3.

The authors re-label the actions contained in Activity Net [10] employing multiple annotators. Comparing both the newly collected boundaries and the original ones, Alwassel et al. [2] too observe a large disagreement when different people are asked to identify the temporal extent of an action, reporting an average IOU of 64.1%. The authors define several characteristics of an action instance in order to analyse the performance of a localisation framework: instance length, coverage (how much of the untrimmed video is covered by the action), context (how easy is for the annotators to guess the action from frames near its occurrence) and annotators agreement. The agreement of an action instance is measured calculating the median IOU between all its multiple annotations. The authors report that only few instances exhibit a very low agreement (2.1% actions have median IOU less or equal to 0.2), while most of the actions (83.8% of the dataset) show a larger agreement, with median IOU greater than 0.4.

The authors analyse the performance of several action localisation models using the aforementioned action characteristics. All the evaluated methods follow a similar trend in performance, and prove more sensitive to changes in instance length, coverage and context compared to agreement. The authors thus argue that annotators disagreement is not a major hurdle for action localisation, at least in comparison to the other action characteristics they define. Yet, they also acknowledge a direct correlation between bounds agreement and performance. In their experiments, actions where annotators most agree are better localised than those where annotators most disagree. This

[2] Published after the work presented in Chapter 4.

highlights the persistent ambiguity in defining the extent of an action, as well as the limited insight that we still have into what do models learn from videos to classify actions, and consequentially what makes an action hard or easy to be recognised.

Varying actions length Hussein et al. [43] [3] propose a novel architecture to capture complex dynamics in long activities. They observe that spatio-temporal 3D convolutions with a fixed kernels size struggle to cope with large variations in the actions duration, i.e. the kernel size may be too short for long activities or may be too large for short actions. To address this issue the authors use multi-scale temporal convolutions stacked in an Inception [101] fashion, designing a model that is able to reason about minute-long complex actions. To evaluate how well the proposed framework can deal with varying action lengths, the authors alter the temporal extents of test activities. Specifically, they first divide the video in segments of equal size, and change then the length of each segment individually by either dropping or duplicating frames. The authors show that multi-scale temporal convolutions are better able to cope with these alterations with respect to fixed-size kernel convolutions on the Charades [91] and Multi-THUMOS [48] datasets.

Liu et al. [65] propose a framework to produce action proposals in untrimmed videos following a coarse-to-fine approach. Action segments are first produced using features with positional encoding [36], a technique that embeds the temporal order of the frames into the corresponding visual features. In order to obtain more precise segments, the authors design a module that estimates the "action-ness" of a single frame. More precisely, the module predicts the probability of a frame being the start, middle and end point of an action. By extracting these additional scores, the authors refine the original action proposal by adding or removing frames to the segment. The authors do not define what the start, middle and end points correspond to in the temporal execution of an action, i.e. they do not explain how an annotator could interpret or locate either of these points, and take the action bounds for granted.

During training, the authors expand the temporal bounds of an action by a 10th of the action length, and divide the expanded segment into a start, middle and end region. The middle region corresponds to the unexpanded segment, while the shorter start and end regions are defined over the neighbourhood of frames centred on the annotated start/end times. These regions are used as strong temporal ground truth to learn the start/mid/end probability of a frame. By enlarging the annotated temporal bound the authors are potentially introducing background frames or frames that belong to a neighbouring distinct action. However, they gloss over this matter and do not discuss the potential noise that is introduced when using unlabelled frames with strong confidence.

Adversarial attacks With the increasing popularity of adversarial attacks [100], recent works have assessed the robustness of action recognition methods to perturbations in the input video[44, 47, 113, 115]. These approaches, whether target specific (white box) or class agnostic (black box), manipulate

[3] Published after the work presented in Chapter 4.

RGB or optical flow frames in order to fool the classifier. Although these works provide insightful analysis on the susceptibility of state-of-the-art recognition methods, here robustness is defined from a still image perspective. In fact the adversarial attacks, even when applied to optical flow, are crafted by adding noise to the frame pixels. Importantly, the temporal extent of an action is not questioned, but assumed to be given in the form of ground truth start and end times in all cases. This does not address the distinctive nature of videos, i.e. time. Arguably, attacking an action classifier should not only entail transforming pixel values, but especially should also alter the temporal extent of the action in order to expand our understanding of what video recognition models learn. Chapter 4 will explore another definition of robustness for recognition in videos.

Focusing on the start of an action Kwak et al. [57] argue that in many applications detecting the start of an action is more important than localising its end. The authors observe that this is especially the case for neuroscience, where researchers often use the beginning of a behaviour as a temporal reference point to inspect neural activity prior to the action. Pinpointing the start of an action also helps understanding the causal relationship of human-world interactions. For example, to highlight salient actions in a football or basketball game, one could detect the time when the audience starts cheering, and analyse the preceding moments to interpret the outstanding actions. The authors also note that identifying the beginning of an action is often less ambiguous than identifying its ending. This is particularly the case of actions that involve repeated movements, such as walking or stirring food in a pot.

The authors observe that training losses based on frame-wise errors typically penalise predictions that are off by a small amount of frames. This kind of objective function focuses less on false positive or negative predictions. Based on this, they propose a loss that weighs the errors relative to true positive offsets, as well as false positive and false negative predictions. Formulating the loss as a combination of different types of error favours more accurate predictions and especially reduces the number of false positive detections. The classifier is implemented with an RNN that receives HOG-HOF [18, 60] or I3D [11] features. The authors also collected a new dataset to evaluate their approach. This dataset contains videos of mice performing several tasks in a laboratory environment. The dataset was collected by neuroscientists who labelled only the start times of the actions in the video.

* * *

To summarise, temporal bounds of human actions have often been overlooked by datasets collectors. Thanks to the growth in the field of action recognition and localisation, there has been some recent attention on the arbitrariness of identifying the extent of an action. Nevertheless, despite the comprehensive study presented by some works [2, 93], the definition of when an action starts and ends remains ambiguous. Chapter 4 will revisit this important matter, as well as the sensitiveness of action recognition algorithms to changes in these temporal bounds.

2.2 Weakly Supervised Approaches for Video Recognition

Annotating the temporal bounds of an action is not only subjective, but importantly also laboursome and expensive. With the increasing need for larger datasets, manual extensive labelling hinders advance in the field of video recognition. For these reasons, weak supervision for video understanding has received much attention recently. The most popular types of weak supervisions are video-level labels, typically used for localisation, and action transcripts, which are generally used for action segmentation. Some works also use point supervision in the form of a single pixel annotated in an image. This is used for semantic segmentation in still images or for spatio-temporal action localisation. This Section reviews important works that use such levels of supervision.

2.2.1 Video-level Supervision for Action Localisation

A stream of works has been recently exploiting the weakest form of temporal supervision provided by video-levels in untrimmed videos. Some approaches [78, 111, 111] follow a two stages pipeline where action proposals are first generated to supervise a classifier to learn actions without temporal labels. Attention mechanisms are often employed to determine the most salient frames in order to learn a more discriminative representation [30, 75, 78, 90, 111].

Gan et al. [33] designed a model to identify one key frame for the single action contained in a trimmed video, as well as to spatially recognise the most salient area within such frame. The authors start by detecting shot boundaries based on the colour difference between adjacent frames. The middle frames of the detected shots are then used as a set of potential key frames. The objective of the framework is to rank such frames according to their relevance to the action, and correspondingly to detect the most salient pixels in each image. This assumes that each video, though not temporally annotated, contains only one action. To achieve this goal the authors map the final activations of a CNN to the input image (i.e. each key frame candidate), obtaining a saliency score for each class and for each pixel. Key frames are ranked based on the average saliency score. The top ranked frame is selected as the key frame for the action of interest. A graph-cut algorithm is then applied on the selected key frame's saliency map to spatially segment the most relevant regions for the action.

The framework is evaluated on the TRECVID MED 2014 dataset [77] for the tasks of action classification and recounting, i.e. the task of finding the evidences of the action in the video, which in this case correspond to the key frame with its salient regions. The authors show that the framework is able to outperform a baseline based on Improved Dense Trajectory [108] for the classification task. However, they do not compare their framework, which uses CNN features, to other CNN-based approaches.

Wang et al. [111] were the first to use video-level labels to train a network to recognise actions in untrimmed videos. The authors first generate clip proposals from a video. These clips are obtained either by splitting uniformly the video, based on the number of action instances occurring in the video, or by means of shot sampling. In the latter case, Histogram of Gradient (HOG) [18] features

are used to detect visual changes between adjacent frames. When the HOG difference between consecutive frames is above a certain threshold, a clip of fixed length is automatically trimmed. This approach is suitable mainly for third person videos exhibiting little camera motion, e.g. professionally shot sport videos like those contained in THUMOS 14 [48], one of the two evaluated datasets.

The generated clip proposals are fed to a CNN to extract features, which are then passed in input to a classification and a selection module. The classification module employs a standard softmax layer to produce class scores for a clip. The selection module is designed to select the clips that are most likely to contain an action. Two methods are proposed for the selection module. The hard selection method ranks the clips according to their classification scores, and averages the scores of the top k proposals to predict the action contained in the untrimmed video. The soft selection method learns an attention weight to estimate the relevance of a proposal clip to the labelled class. This weight is learnt using a simple network that receives the features extracted from a clip and outputs an attention weight for the single proposal. The attention weights obtained for each clip are used to compare the relevance of all proposals. The soft selection attention scores are used to weigh the classification scores of each clip, which are finally combined to predict the action in the untrimmed video,

The whole architecture is trained with a standard cross entropy loss. The trained network is then used to predict frame-level scores to classify and localise actions in a test video. When using the soft attention selection module, the attention weight produced by the network is also used to further refine the location of the action at test time. The authors show that the proposed method achieves results that are comparable to and sometimes even superior than those obtained with fully supervised baselines, on the THUMOS 14 [48] and Activity Net [10] datasets.

Nguyen et al. [75] proposed a similar approach to use video-level labels in untrimmed videos. The authors first split the video uniformly to obtain video segments, as in [111]. Features extracted from these segments are then fed to an attention module, a small *class agnostic* network that outputs an attention score for a given segment. Such score is used to weigh the features extracted from the segment, i.e. to gauge the "action-ness" of the visual content of the segment. The weighted features are then fed to a classification module. The authors argue that an action can be identified by recognising a sparse set of salient segments containing discriminative information about the action. They thus design a sparsity loss calculated over the attention weights vector to encourage the attention module to assign a high score to few segments. The model is trained combining the sparsity loss with a standard cross entropy loss.

To localise the actions the authors devise a *class specific* activation map called Temporal Class Activation Map (T-CAM). T-CAM is a one dimensional map in the temporal domain that assigns a class specific weight to a video segment, using the output of the classification module. The class specific T-CAM weight and the class agnostic score produced by the attention module are combined together to predict the action location in the untrimmed video. The authors show that the proposed approach outperforms [111] on the the same datasets.

Singh and Lee [97] observe that models for object and action localisation focus mainly on the most discriminative portions of images and videos, rather than on all relevant parts, in order to optimise classification accuracy. The authors argue that this can potentially cause suboptimal performance, and propose to randomly hide patches of the input data at training time, to force models to concentrate on all related parts. In the case of object localisation from images, the hidden patches are squared masks extracted from a uniform grid, while for action localisation the patches are frames removed from the untrimmed videos. The videos are uniformly split in segments of equal length, and each segment is hidden with a random probability during training. C3D features [103] extracted from the visible segments are then fed to a CNN. At test time the whole video is fed to the network, and class activation maps are used to locate the actions temporally, similarly to [75]. The authors compare their method to a baseline that receives the whole data during training. While this baseline performs worse than the proposed patch-hiding approach, the action localisation results obtained on THUMOS 14 [48] are largely inferior to other works that use video-level supervision [30, 63, 75, 78, 90, 111]. This suggests that the patch-hiding technique, albeit effective for object localisation in still images, may not be efficient to highlight the salient portion of an action in a video, especially in long videos like those contained in THUMOS 14.

Paul et al. [78] exploit the visual similarities in pairs of videos belonging to the same class to train a classifier. They start extracting features from the video frames using pre-trained Untrimmed Nets [111] or I3D [11]. Interestingly, unlike most related works, they do not split the untrimmed videos into multiple segments, arguing that the initial segments hardly align well with the action and that trimming the video without a priori knowledge likely introduces noise in the action representation. The authors instead attempt to feed the whole video to the network to extract the features. They do not fine-tune the feature extraction modules but only learn the classification parameters in their model. This allows entire large videos to be fed to the network. When an untrimmed videos does not fit into the GPU memory due to its length, they extract a single random segment of consecutive frames from the video.

The authors observe that frames where the same action is occurring in distinct videos should share a similar representation. Likewise, frames where an action is taking place should have a representation different from that encoded for frames where the action is not occurring. Based on this, they use a ranking hinge loss to train the model. Since no temporal annotations are used, the author use frame-wise softmax scores to estimate which regions of the video contain the action, in order to sample relevant and irrelevant portions of the video for the loss. Localisation is performed by applying a threshold on the classification and attention values of each frame, similarly to [111].

As seen above, the typical approach to leverage video-level labels is to first generate action proposals from the untrimmed video, and then learn to classify the actions based on such proposals. During testing, localisation is usually performed by sliding a window and obtaining classification scores for the frames contained in the window. A threshold is finally applied to these scores to locate the action, often in combination with an additional attention or “action-ness” score to refine the

action segment. Works that use this approach do not directly learn the action boundaries, but rather learn a classifier which is used to score individual frames or segments. Shou et al. [90] attempt instead to regress the boundaries of an action. While this strategy had been adopted by fully supervised localisation methods [34, 35, 62, 117, 122], Shou et al. [90] were the first to regress action bounds using video-level labels. To do so, they first split the untrimmed video in non overlapping snippets, and use Untrimmed Nets [111] to extract features, as well as to produce a classification and an attention score for each snippet. Features are fed to a localisation network that predicts the centre and the length of the action segment, in a class-agnostic way. The authors introduce a novel loss to refine the predicted action segments, using the combined classification and attention scores. The loss is designed to encourage high scores within a segment and low scores outside.

More recently, Liu et al. [63] highlighted two key issues related to the lack of temporal annotation in untrimmed videos: action completeness and context. As discussed above, most approaches cast the localisation task mainly as a classification problem followed by some heuristic to provide the temporal extent of the action during inference. When adopting this paradigm models can successfully recognise discriminative action portions, however they lack the knowledge of when an action is complete, which is necessary for accurate localisation. The authors follow prior works' observation that an action can be decomposed into multiple sub-actions [81]. Based on this, they design a classification architecture composed of multiple branches. The branches share the same input, i. e. features that are extract from segments of equal length, and attempt to focus on different parts of the action. This is enforced with a diversity loss that encourages each branch to output strong activations for distinct portions of the action.

The authors note that due to the unavailable temporal bounds, weakly supervised models tend to confuse the action with its enclosing context. For example, consider a short video segment preceding the action "cut tomato", where a cutting board, a tomato and a knife are visible and still, with no motion occurring in the snippet. Due to the visual cues associated with the objects, the classifier might incorrectly predict the action with high confidence in the snippet. The authors address this problem generating hard negative samples. In particular, they create new video snippets labelled with a new background class, extracting stationary frames from the training untrimmed video. Static frames are found by measuring the average optical flow magnitude of the pixels. This new class is in addition to the already existing background class of the evaluated datasets. While the standard background label is attached to irrelevant videos (unrelated context, easy sample), the proposed background class is assigned to visually highly correlated clips (hard sample), which pushes the classifier to discriminate between the action and its surrounding context. These two novel components help the model to achieve better performance compared to other weakly supervised approaches. Interestingly, the authors note that at higher IOU thresholds Shou et al. [90] achieve superior results, due to the fact that [90] attempts to directly regress the action boundaries.

Fernando et al. [30] propose a novel way to find relevant action frames by comparing local predictions (across frames in a video) and global predictions (across videos in a dataset). More

precisely, the local attention is estimated with a Gaussian modelled with the mean and standard deviation classification score of all frames *in a single video*, for each action class individually. By comparing the score of a single frame to the average score, the authors aim to gauge the relevance of a frame with respect to the entirety of the video it belongs to. The global attention works in a similar way: frame-wise class-specific scores are compared to the average score of the action *in the whole dataset*, using again a Gaussian model. The local and global attention scores are then fused together to weigh the classification score of each frame in a video, which are temporally aggregated to produce a single activation for each class. This approach relies on good quality features to be effective. The authors use VGG-16 [96] and ResNet34 [41] pre-trained on ImageNet [23], as well as I3D [11] pre-trained on Kinetics [50].

The local and global attention proposed in [30] differ from the commonly adopted attention method [78, 111] where a softmax function is applied to the classifications scores along the temporal dimension. While this can effectively point at discriminative temporal location in a video, this simple strategy does not attempt to model what is relevant for an action. Estimating the relevance of a frame considering local and global statistics seems instead to be a more robust approach. Furthermore, by using the average classification score and taking the variance of such score into account as well, it is possible to detect outliers in a more robust way. This is in contrast to the softmax function applied to the scores along the temporal axis. In fact, the softmax function squashes non maximum scores to separate the highest value without considering any statistical measure of the input data. This is potentially more sensitive to outliers, because highly scoring noisy frames could be highly rewarded causing the remaining frames to be considered irrelevant due to the potential extreme separation applied by the softmax.

To conclude, video-level labels provide the weakest temporal cue, signalling only the presence or absence of an action in an untrimmed video, discarding any temporal ordering. As shown in the works reviewed in this Section, when only a few different actions are present in an untrimmed video, this supervision is sufficient to learn the actions even from long videos. This is in fact the case for the two datasets on which all these works have been evaluated, i.e. THUMOS 14 [48] and Activity Net [10], which contain mainly one class per training video. The only exception to the above is [30], which was evaluated on Charades [91]. The Charades dataset contains six different actions per training video on average. Due to its complexity, the evaluation protocol commonly used for localisation is less strict. Specifically, 25 evenly spaced frames are selected, and MAP is calculated from the classification scores of such frames. Given that the evaluation is done on individual sparse frames rather than segments, MAP is not reported in association with temporal IOU thresholds, which raises the question of whether using the term “localisation” is appropriate for the standard evaluation protocol of this dataset.

When multiple different actions are present in a video, video-level labels no longer provide sufficient temporal supervision for recognition tasks. Chapter 6 will return to this point.

2.2.2 Transcript Supervision for Action Segmentation

Transcript supervision corresponds to an ordered list of action labels in untrimmed videos, without any temporal annotations [8, 9, 24, 42, 56, 81, 82]. Transcripts provides a much stronger temporal cue than video-level labels, given that in the former case action labels are ordered temporally, whilst in the latter case only the presence of an action is signalled. Most works [8, 24, 42, 56, 82] assume the transcript includes knowledge of "background", specifying whether the actions occur in succession or with gaps. When such background class is available, and when the actions (including the background) are of comparable length, transcript supervision can provide a good segmentation of the untrimmed video.

Bojanowski et al. [8] pose the task of action localisation as a temporal assignment with ordering constraints, using a discriminative clustering algorithm. The untrimmed videos are initially split into short segments of equal length (10 frames). The authors aim to classify each segment to temporally localise the actions in the video, respecting the order of the actions given by the transcript. This is achieved by training a classifier that takes in input the short segments and is optimised using a cost function that minimises the overall classification loss of the segments in the video. An extension of this work is proposed in [9], where the same authors address the task of aligning videos with natural language (free-form text), contrary to [8] where the videos were aligned with a sequence of pre-defined class labels.

Huang et al. [42] base their approach on [40], where an RNN was used to recognise the phonemes in an audio sequence. The authors observe that [40] does not exploit the fact that consecutive frames are potentially strongly correlated. The authors argue that this additional information helps excluding incorrect sequence labelling, and accordingly design a loss function that encourages visually similar consecutive frames to be assigned to the same label. Using this loss, the authors evaluate all the possible frame-label assignments, which are constrained by the order of the actions given at training time. Without temporal annotations, however, the frame-label assignments can easily drift. The authors shows that annotating less than 1% of the frames in the videos was sufficient to tackle incorrect labelling and achieve results comparable to fully supervised baselines.

Kuehne et al. [56] address the task of action segmentation, modelling each individual action class with a dedicated HMM. The action specific HMMs are concatenated using the ground truth order of the actions. The task is to assign each video frame a state belonging to one of the HMMs in the chain. The class associated with each state's HMM is thus used to segment the video. Starting by uniformly splitting the videos using the transcript, which include also a background class, the authors iteratively refine the action segments during training. This is done by updating the parameters of each action HMM, specifically the parameters of the model that represent the observation probabilities (in this work, either a multivariate Gaussian distribution or a CNN). The observation probabilities model is updated using the frame-wise features (IDT [108]) aligned with the action segment. During inference, when the order of the actions is unavailable, the authors evaluate all the valid action sequences seen in the training set, and use the sequence providing the highest confidence to segment the video.

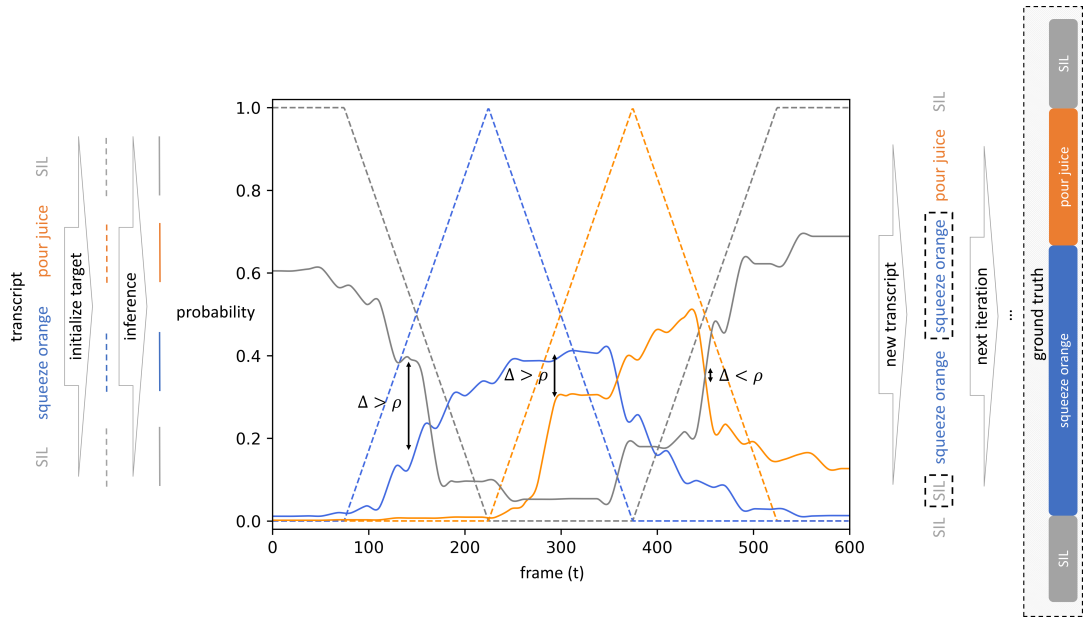

Figure 2.4: Starting from uniform segmentation obtained from the action transcripts, pseudo frame labels are iteratively refined comparing class probabilities at each boundary frame. Figure from [24].

The same authors extend their work in [81]. Noting that actions can be decomposed into shorter sub-actions, they model actions on both a coarse and fine grained scale. More precisely, each action is split into a fixed number of sub-actions which, as discussed in Section 2.1, are treated as latent variables, i.e. the authors do not attempt to classify the sub-actions which are not labelled. Rather, frame-wise IDT features aligned with the sub-actions are fed to an RNN in order to represent the actions. The RNN's output provides the observation probabilities used by class specific HMMs that are chained together to model the sequence of actions in a video, as in [56]. Training and inference are also similar to [56]. The initial alignment is provided by uniformly splitting the video using the transcript and further dividing each segment into sub-actions of equal length.

The authors observe that the initial uniform segmentation plays a crucial factor for the training convergence. In fact, if the action length varies considerably the uniform segments will likely be misaligned with the actions. In such conditions, it is difficult to obtain a discriminative initial representation of the actions. To overcome this issue, in [82] they propose another approach that does not require an initial segmentation. Specifically, they devise a network to estimate the observation probabilities, feeding IDT features to the network, similar to [56, 81]. They then design a model based on the Viterbi algorithm that produces a class label for each frame in a video, given the action transcript and the corresponding observation probability estimated by the network. The estimated frame-wise labels are used as pseudo ground truth. Namely, the pseudo labels are employed to apply a cross entropy loss that is used to train the model. Given that the transcripts are directly used during training, this approach does not rely on an initial uniform segmentation, alleviating the initialisation issues discussed above.

Ding and Xu [24] adopt an encoder-decoder architecture to produce softmax scores from frame-wise features. This architecture aims at incorporating high-level semantic information (provided by the decoder) into low-level visual features (provided by the encoder). This model is trained with a standard cross entropy loss, which requires a class label for each frame. These labels are provided, like in the approaches seen before, by a pseudo ground truth initialised with uniform sampling from the transcript. The authors propose a soft boundary strategy to iteratively refine the frame labels, accounting for the segmentation ambiguity resulting from the lack of the action bounds.

More precisely, the refinement is carried out by linearly interpolating the pseudo ground truth around the boundaries between adjacent actions. The main idea is that there is more uncertainty about frames close to the boundaries, given the weak temporal supervision. Hence, frames close to the bounds are assigned mixed probabilities of belonging to either of the neighbouring actions, depending on their proximity to the boundary. The pseudo ground truth is iteratively refined using the softmax scores produced by the encoder-decoder model. Specifically, boundaries are shifted according to the class predicted with highest confidence, between the neighbouring actions, at each boundary frame. Figure 2.4 illustrates the update method. This strategy does not allow gaps between action segments, and as a consequence a background label is necessary for the method to operate. The authors introduce a stop criterion to assess convergence. In particular, they obtain global classification scores for each ground truth action in the video by max-pooling the frame-level scores. The global scores are then used to calculate a video-level binary cross entropy loss, which is monitored to estimate convergence.

The works discussed this far share an underlying approach: they all produce a pseudo ground truth, chiefly starting with uniform sampling from the transcript, and use this pseudo labelling for training. As we already discussed, this approach is highly prone to suboptimal performance, especially when dealing with actions of very different lengths. The papers reviewed above, despite attempting different solutions to mitigate this issue, still treat the pseudo labels as strong labels, which potentially causes incorrect optimisation of the model.

To address this issue, Chang et al. [13] pose the problem of action segmentation as a Dynamic Time Warping problem. The main contribution of their work is the enforcement of an additional constraint to the objective function. This constraint is given by a margin loss based on positive and negative samples, i.e. the model is encouraged to maximise the discrimination between the correct and incorrect orders of actions. This is in contrast to previous works that only maximise the recognition of the actions given the pseudo labelled frames. While this kind of loss has been used for other tasks and other level of supervision, [13] is the first work to use it for action segmentation and alignment with transcript supervision.

2.2.3 Point-level Supervision

Point supervision refers to using a single pixel or a single frame as a form of supervision. This has been used mainly for semantic segmentation and action spatio-temporal localisation. One of the

pioneering works exploiting point supervision was [5], where a single pixel is used as annotation for semantic segmentation in still images. The authors show that such less expensive labelling can prove competitive with respect to full segmentation masks. The authors highlight that point supervision makes a more efficient use of the annotators time when the time budget is fixed.

Mettes et al. [70] address spatio-temporal action localisation in videos. For this task the common type of annotation is provided by action tubes, that is a set of consecutive frames labelled with a bounding box annotating the area where the action takes place. In [70] the authors use instead a sparse set of frames annotated with a single point placed around the area involving the action. The authors first generate a large collection of localisation proposals for an untrimmed video, using [105]. Each proposal is a sequence of frames estimating the spatial extent of the action with a bounding box. Based on the idea that ground truth action tubes can be substituted with action proposals, the authors find the proposal that best aligns with the actions in the video. In order to do so, they score each proposal calculating the distance between the proposal's bounding boxes and the annotated points.

The proposals scores are used in a Multiple Instance Learning (MIL) setting to train a model to discern good proposals from poor ones. Specifically, the authors use an SVM classifier incorporating the proposal scoring function in the classifier objective. Only the proposal with the highest score from each video is selected to train the classifier. The authors extract IDT features [108] from the proposals' frames. The features are then PCA-reduced and encoded with Fisher vectors [84] before being fed to the SVM classifier.

The authors compare results obtained with ground truth tubes to those obtained with the sparse point-annotated frames. The evaluated datasets, the UCF Sports [83], UCF 101 [98] and the Hollywood 2 [67] datasets, contain mainly one or two kinds of action per video, though in UCF 101 repeated occurrences might occur. The authors report that results obtained with the weaker supervision are competitive to those obtained with the more expensive ground truth tubes. The authors also investigate how temporal sparsity (i.e. how many frames are labelled) affects performance, and show that sufficient results could be achieved by annotating 10% of the video frames.

In a follow-up work Mettes et al. [71] revise the point-supervision approach and propose instead to mine localisation proposals using pseudo-annotations, i.e. visual cues such as human detection, object proposals, motion proposals and image centre bias. These cues are obtained using pre-trained models, without human annotations. The pseudo-annotations are used to select spatio-temporal action proposals for training, employing a MIL setting similar to [70]. However, contrary to [70] which uses point labels provided by human annotators, only video-level class labels are utilised for training.

Chéron et al. [14] evaluate several forms of supervision for the task of spatio-temporal action localisation, namely: video-level labels, single temporal points, single bounding boxes, temporal bounds without and with bounding boxes and full action tubes. Figure 2.5 compares these annotations. Single temporal points roughly locate the moment when the action is taking place, without identifying neither its temporal nor spatial extent, as in [70]. The point is randomly sampled within

Figure 2.5: Different levels of supervision are compared for the task of spatio-temporal action localisation. Figure from [14].

the ground truth temporal bounds, and thus it is assumed to always lie within the scope of the action. The authors use an off-the-shelf human detector to extract human tracks from the videos. These tracks provide a spatio-temporal region to pool I3D [11] features from the video frames. The pooled features are utilised in a unified framework based on discriminative clustering, where the various levels of supervision impose different constraints on the model. In the single temporal point case, such constraint is given by a fixed-length segment of consecutive frames centred on the given temporal point. More precisely, the authors enforce the human tracks extracted within the segment to be associated with the action.

2.3 Conclusion

This Chapter reviewed papers closely related to the work presented in this Thesis, which focussed on temporal boundaries and point supervision for action recognition in untrimmed videos. As previously discussed, little attention has been aimed at the subjectivity of the bounds that enclose an action. Importantly, no work linked boundaries ambiguity and the resulting start/end variations to the robustness of recognition algorithms. Furthermore, there has been virtually no attempt to precisely define the start and the end times of an action to foster consistent labelling. Chapter 4 will revisit both of these important aspects.

Weakly supervised approaches constitute an appealing line of research, given the increasing need for larger datasets and the corresponding cost involved in labelling these. Weak supervision often comes with a compromise between the required annotation effort and the performance that it is possible to achieve. Chapter 6 will discuss the opportunity that single timestamp supervision offers for action recognition in untrimmed videos, in the attempt to find a good balance between labelling labour and accuracy. Chapter 3 will instead review seminal works in action representation and classification, as well as works that contributed to the computer vision community with the release of video datasets for action recognition.

CHAPTER 3

DATASETS AND MODELS FOR ACTION RECOGNITION

This chapter provides an overview of important datasets and models for action recognition in videos. Details regarding the characteristics of each datasets are provided in Section 3.1, with a focus on the annotation process involved in the data collection. Particular attention is dedicated to the BEOID [20] and EPIC Kitchens [21] datasets, since I was directly involved in their annotation and collection.

Section 3.2 surveys models that contributed notably to significant advancement in the field. The review also includes recent interesting works that take an alternative approach to recognising actions in videos efficiently. The Chapter concludes with Section 3.3, which briefly presents a framework for action recognition leveraging semantic knowledge.

3.1 Datasets

This Section reviews video datasets for action recognition. Whenever available, details regarding the datasets' temporal labelling are discussed. As we shall see, authors typically do not mention how the start and end times of actions in videos were annotated. Chapter 4 will return to the importance of labelling actions in video, examining the arbitrariness and ambiguity often involved with this task.

The datasets are reviewed following a chronological order and are divided into egocentric and non egocentric datasets. A summary of the characteristics of the datasets is provided in Table 3.1. Noticeably, there has been a momentous expansion during the last decade in terms of size and complexity of the available datasets. This is particularly embodied by EPIC Kitchens [21] and Kinetics [11], the largest video datasets in the first and third-person domain respectively. However, as we will see in this Section, the number of videos and classes should not be the only metrics to compare and appraise datasets. In fact other factors such as the density and the unconstrained nature of the actions are very indicative of the complexity of a dataset and should be also evaluated when

	Dataset	Year	Untrimmed	Videos	Classes	Instances	Average action length	Average video length	Average instances per video	Average classes per video
Egocentric	CMU-MMAC	2008	Yes	13	31	450	8.7	466.1	34.6	26.9
	GTEA Gaze+	2012	Yes	27	42	1,130	2	604.7	41.8	16.6
	BEOID	2014	Yes	58	34	742	1.6	60.6	12.8	5.4
	EPIC Kitchens	2018	Yes	432	149[1]	39,448	3.7	451.8	91.3	42.7
Non egocentric	HMDB-51	2011	No	6,766	51	6,766	3.1	3.1	1	1
	UCF-101	2012	No	13,320	101	13,320	7.2	7.2	1	1
	THUMOS 14[2]	2014	Yes	2,584	101	6,310	4.2	204.8	15.4	1.1
	Activity Net 1.3[3]	2016	Yes	19,994	200	23,064	49.2	116.7	1.5	1
	Something-Something[4,5]	2018	No	220,847	174	220,847	4	4	1	1
	Kinetics[4]	2019	No	650,317	700	650,317	10	10	1	1

Table 3.1: Characteristics of seminal video datasets for action recognition. Average action and video length is in seconds. [1]Number of verb-noun classes with at least 50 instances (in total 3,033 verb-noun classes are present in EPIC Kitchens). [2]Relative to the validation and test sets (410 videos belonging to 20 classes were temporally annotated). [3]Relative to the published training an validation sets (test labels are withheld), with the exception of number of videos which counts all videos in all sets. [4]Relative to the latest version of the datasets. The first editions of Something-Something and Kinetics were both released in 2017. [5]Something-Something contains videos that might be considered egocentric since in some cases the camera is hand-held while performing the actions. However in most cases the camera is relatively static, i.e. does not move with the actor as it occurs in egocentric videos. For this reason Something-Something is put under the non egocentric category.

Figure 3.1: Example frames extracted from the reviewed datasets.

comparing datasets. Figure 3.1 depicts frames extracted from the reviewed datasets. Egocentric and non-egocentric datasets are showed respectively in the top and bottom row. Notice how the two fields of view offer different advantages. In first-person videos the interacted objects and the actor's hands typically dominate the frame, which can be helpful to distinguish the actions more clearly. Third-person videos are often shot with static or more stable cameras, and thus do not suffer from shaky motion usually present in egocentric videos. Third-person videos generally provide also a wider view of the scene.

First-Person Vision Datasets

3.1.1 CMU-MMAC

The Carnegie Mellon University Multimodal Activity dataset (CMU-MMAC) [22] is one of the first action recognition datasets captured using an Egocentric field of view. CMU-MMAC was collected in a kitchen built on purpose for the dataset. Five individuals were recorded while preparing one of five different meals, following a non scripted recipe: brownies, pizza, sandwich, salad and scrambled eggs. CMU-MMAC is multi-modal and includes video, audio, motion capture, accelerometer and gyroscope data.

The dataset contains actions related to the preparation of the aforementioned food, i.e. mainly interactions with kitchen items (e.g. jar, fridge, baking tray, etc.) and groceries (e.g. eggs, oil, etc.). The authors provide the actions annotations in [99], which presented a framework for action recognition and segmentation using both first-person videos and sensors. The authors observed that labelling actions is ambiguous, due to the several ways an action can be carried out and described. Nevertheless they do not discuss how they annotated the temporal bounds nor how they chose the granularity of the actions to be labelled. Action labels were provided only for one out of the five meals, i.e. for all videos showing people preparing brownies.

3.1.2 GTEA Gaze+

The Georgia Tech Egocentric Activities Gaze+ dataset (GTEA Gaze+) [28] records 10 individuals preparing a meal in a kitchen located in an instrumented house. The seven meals were prepared following a recipe, and involved the preparation of pizza, American breakfast, afternoon snack (e.g. sandwich), turkey sandwich, Greek salad, cheese burger and pasta salad. The preparation of each meal took on average between 10 and 15 minutes. The dataset contains video with audio, as well as the camera wearer's 2D gaze for each frame.

Each video displays around 42 action instances belonging to a total of 42 verb-noun classes. Like in most kitchen-oriented datasets, GTEA Gaze+'s actions consists of interactions with food and utensils. Compared to CMU-MMAC, which provides annotations only for one meal, actions in GTEA Gaze+ involve a large variety of objects.

3.1.3 BEOID

The Bristol Egocentric Object Interaction Dataset (BEOID) [20] collects videos showing subjects interacting with various objects in several locations. The interactions were captured using a head-mounted device, which recorded both the video and the gaze of the camera wearer. Seven individuals recorded the data in four locations in an office environment (kitchenette, workspace, printer desk, corridor), as well as in two locations within a gym (rowing machine, treadmill/bicycle). The subjects followed verbally communicated instructions to perform simple tasks. These include operating a

printer, preparing coffee, manipulating various tools on a worktop, running on a treadmill and using a rowing machine.

Actions were annotated in [114], a joint work I carried out during the first year of my PhD. Multiple annotators were asked to watch the dataset's untrimmed videos and to describe the observed actions with any verb and noun they saw fit. Annotators were not provided with a list of actions to be labelled, and were asked to annotate all the object interactions present in the videos. The annotators received instructions explaining the requested temporal granularity of the annotations. More precisely, they were told to split each activity (e.g. preparing a sandwich) into all the object interactions composing the activity (e.g. cut bread, spread butter, etc.). The guidelines also noted that actions can be interpreted as being composed of multiple sub-actions, illustrating the example of slicing bread, which requires a knife to be grabbed in advance. To obtain more consistent labelling, annotators were asked to label such cases as distinct object interactions.

The annotators were showed two video segments illustrating a good and a bad annotation example for the action "open fridge". The good example displayed only the act of pulling a fridge door open. The bad example showed the act of opening the fridge followed by the act of grabbing some vegetables from the appliance. The instructions explained that the bad example was not suitable because the two actions should be labelled separately with two distinct segments. Annotators received also directions for identifying the start and the end of an action. More precisely, annotators were told to identify the start of an object interaction as the first moment when they recognise that the subject in the video is initiating motion to interact with the object. Similarly, annotators should identify the end of an object interaction as the first moment when they recognise that the interaction is complete.

The multiple annotators labelling conducted on BEOID sparked a few questions regarding the perception and the consistency of the temporal span of an action. Chapter 4 will return to this point. In total 21 people annotated BEOID's 58 videos, with a variable number of annotators per video.

3.1.4 EPIC Kitchens

The egocentric domain has lacked large datasets for a long time since its onset. In fact, compared to third-person video datasets, egocentric datasets typically have significantly fewer videos. For example, GTEA Gaze+ has 1,130 action segments, compared to 13,320 in UCF-101 [98], which was released in the same year (see Table 3.1). Action and scene variability is also usually limited in first-person datasets compared to non egocentric counterparts. This is reflected by a small number of action classes and actors, as well as by less variability in the environment where actions take place. For example, in CMU-MMAC and GTEA Gaze+ all actions occur in one place, in both cases a kitchen that was equipped on purpose to record the dataset. Furthermore, actions in egocentric datasets are usually scripted, i.e. participants follow a script (or a recipe) when performing the tasks, which restrains the natural diversity in how activities are carried out by different individuals.

Some efforts attempted to fill this gap. The Activity of Daily Living (ADL) [79] dataset collected videos from 20 people who recorded themselves in their own homes with a chest-mounted camera

while performing daily life activities. ADL contains 436 action segments categorised in 32 classes, for a total of around 10 hours of footage (1 million frames). Although ADL is not fully scripted, participants were assigned a predetermined list of actions they needed to perform. More recently, the Charades-EGO [94] dataset crowd-sourced a large number of videos showing people performing a pre-defined task in their homes. Charades-EGO offers a large number of videos (2,751, for a total of 2.3 million frames) recorded in native environments, however its actions were scripted. Additionally, based on visual inspection it appears that in many videos the egocentric field of view is obtained with questionable means. For example, in some cases actors held the camera on their forehead with one hand and performed actions with the other hand. In other cases videos where recorded by a second person who closely followed the actor holding a camera around the actor's head. These improvised solutions affect the overall quality of the recordings in the dataset.

EPIC Kitchens [21] was collected specifically to address all the aforementioned issues, i.e. to provide a large dataset of unscripted activities recorded in native environments. By featuring 432 videos for a total of 11.5 million frames, EPIC Kitchens is the largest egocentric video to date. Videos were gathered from 32 participants who recorded themselves in their kitchens for a minimum of three days. The participants were told to simply wear a head-mounted camera and record any activity they performed in the kitchen at any time during the collection period, except when they were eating meals.

Given that activities were fully unscripted and recorded in real-life scenarios, videos are naturally heterogeneous in the type of actions and the way they are performed. Participants exhibited high levels of multi-tasking while operating in their kitchens, pausing tasks in order to complete other actions or performing multiple actions at the same time. The dataset contains 39,596 action segments and 149 action classes with at least 50 instances. Object bounding boxes were also annotated, for a total of 323 distinct object classes. Details regarding the annotation process of EPIC Kitchens are provided next.

Annotating actions in EPIC Kitchens The first step in the annotation pipeline involved a novel live commentary approach where the participants narrated the videos they had recorded. The participants narrated their actions as they watched the video. The participants recorded the narration using their personal phone, producing a separate audio file for each video. The participants were asked to provide narrations in a verb-noun form, using any verb and noun they wished to describe the actions and the involved objects. People could speak in English or in their own native language if preferred. As a results, narrations were collected in English, Spanish, Italian, Greek and Chinese.

The narrations formed the base for the subsequent crowd-sourced annotations. Narrations were first manually transcribed (and translated in English if in another language) using Amazon Mechanical Turk (AMT). As participants were free to use any word when narrating, the abundant collection of different verbs and nouns were clustered to form action and object classes. The audio tracks were parsed to extract the timestamps localising each spoken sentence. This gave a rough initial temporal location of each narrated action in the video.

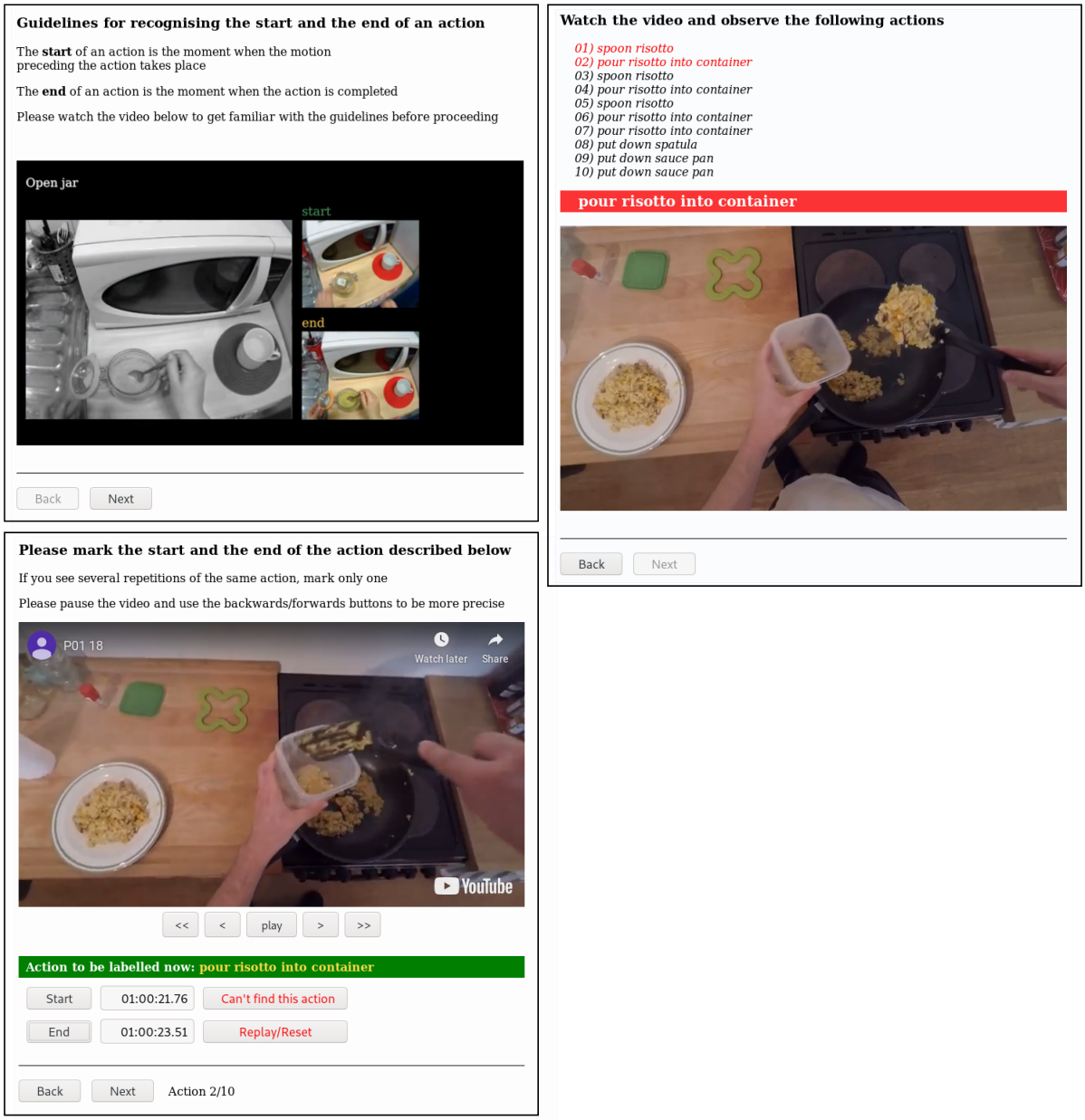

Figure 3.2: EPIC Kitchens: AMT interface to annotate action temporal bounds. Clockwise from top-left: first page showing guidelines to recognise the temporal span of action; list of actions to be annotated roughly aligned with the video; page to annotate a single action.

Starting from the narration timestamps, the start and end times of the actions were then labelled, using again AMT. In order to collect precise temporal annotations, annotators were first provided with textual guidelines instructing how to recognise the beginning and ending of an object interaction. According to the guidelines "the start of an action is the moment when the motion preceding the action takes place", while "the end of an action is the moment when the action is completed".

The annotators had to watch a video showing some annotations examples before proceeding

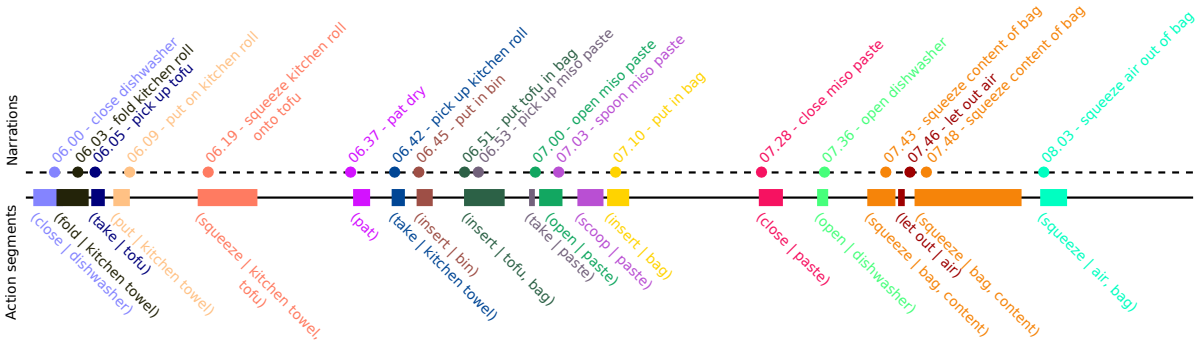

Figure 3.3: EPIC Kitchens: annotating action segments (bottom) starting from narration timestamps (top). Narrations correspond to the transcribed text as narrated by the participant. The corresponding action/object classes were obtained matching the narrated verb and noun to the actions/objects clusters.

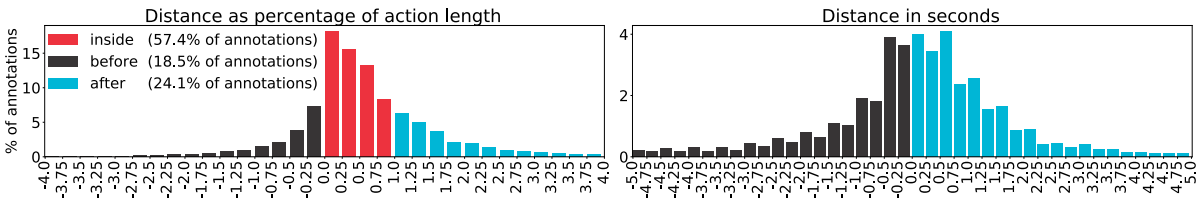

Figure 3.4: EPIC Kitchens: Distance between narration timestamp and ground truth action segments. Left: distance is normalised using the action segment length. Right: distance is reported in seconds. Red bars indicate narrations whose timestamp is contained in the corresponding action segments. Black and blue bars indicate narration timestamps that are respectively before and after the action's start and end.

forwards. Annotators were then showed the video and the actions to be annotated. The narration text was aligned with the video using the narration audio timestamps, in a subtitle fashion. This allowed the annotators to familiarise with the actions before annotating them. Figure 3.2 shows the web interface used by AMT workers to annotate the actions. Each action was annotated by four different people for robustness. The final action temporal boundaries were formed combining the start/end times provided by the four annotators.

Figure 3.3 illustrates narrations timestamps and the transcribed text (top) along with the corresponding action segments (bottom). Notice that despite being close, the narration timestamps are sometimes before or after the corresponding action segments. This is due to a few factors. Firstly, the narration timestamp is relative to a separate audio file which was not accurately synced with the video, and thus there is an intrinsic misalignment between videos and narrations. Secondly, as the videos were narrated live without pausing, narrations are in some cases belated. The density of the actions also amplified the misalignment in those video sections where many actions occur closely in a fast sequence.

This is better analysed in Figure 3.4, which illustrates the distance between the narration timestamp and the ground truth action segments. In the left-hand side plot the distance is normalised

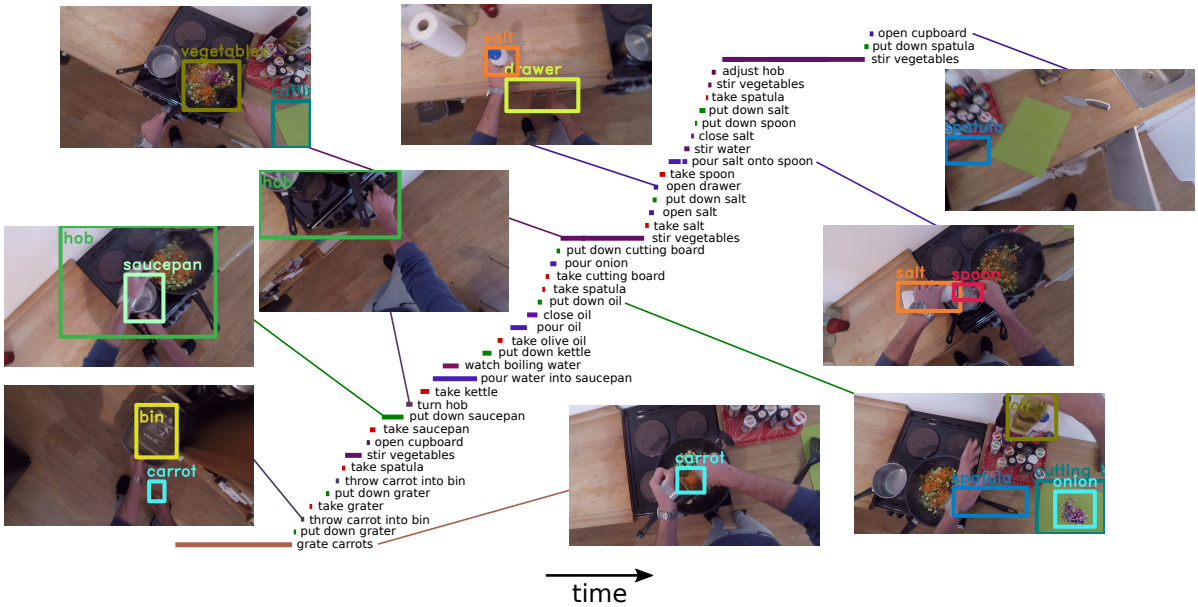

Figure 3.5: EPIC Kitchens: action and object annotations examples. A subset of consecutive action segments is showed together with the bounding boxes of the involved objects.

by the segment's length, with red bars depicting narrations whose timestamp is enclosed by the corresponding action segment (values between 0 and 1). Black and blue bars indicate narrations that are respectively before and after the action's start and end time. In the right-hand side plot the distance is instead reported in seconds. The Figure shows that most narrations are close to the action's segment (within 25% of its length), with more narrations being belated rather than anticipated. This is expected since narrators had to first recognise the action before narrating it, although in many cases the narrators were also able to predict their own actions. In total, 18.5% and 24.1% of narrations were respectively narrated before and after the action, while 57.4% of the narrations was contained in the action segments.

Overall, 30.6% of the narration timestamps were contained within a different action. This shows the challenges involved in annotating the actions starting from the narration timestamps, as well as the issues involved in using these timestamps as weak temporal supervision. Chapter 6 will return to this, proposing a method for action recognition using single timestamps. Specifically, the narration timestamps of EPIC Kitchens will be evaluated and compared to the ground truth action segments.

Finally, Figure 3.5 illustrates a subset of consecutive action segments labelled in a video, together with the bounding boxes annotated for the involved objects. Note how actions are very dense, overlap one with the other and vary remarkable in length and frequency.

Third-Person Vision Datasets

3.1.5 HMDB-51

The Human Motion DataBase (HMDB-51) [54] was introduced when existing video datasets were limited in terms of size (number of videos) and complexity (number of classes, simplicity of the scene and the actions). In fact, back then (2011) the most popular datasets for video recognition (e.g. KTH [59] and Weizmann [7]) contained 6-10 action classes.

Prior to HMDB-51, other datasets such as UCF Sports [83] and Hollywood 2 [67] offered a greater number of videos, however, their number of classes remained small (respectively 9 and 12). UCF-50 [80], released before HMDB-51, contains 50 classes and a minimum of 100 videos per class. However, UCF-50's actions are mostly unambiguously characterised by their appearance and can be easily recognised using static cues alone. e.g "playing guitar/piano" or "diving" which are strongly characterised by the presence of the musical instruments or the swimming pool. Interestingly, this issue still applies to contemporary datasets, where the appearance of a scene or an object is often sufficient for a classifier to recognise an action.

Within this historical context, Kuehne et al. [54] proposed HMDB-51 collecting actions that are more characterised by motion rather than static appearance. The dataset contains 51 classes with a minimum of 101 videos per class, for a total of 6,766 actions. The videos were collected from various internet sources and show daily-life actions such as eating, drinking and brushing hair, as well as less ordinary actions like riding a horse, shooting a bow and drawing a sword. Videos are trimmed and contain one single action.

HMDB-51 was one of the first large datasets collecting non professional videos from multiple online sources, an approach many other researchers have since adopted. Compared to other datasets, HMDB-51 proved to be more challenging, due to the broad variety of the actions and the more realistic videos. Together with UCF-101, HMDB-51 has been a standard benchmark for action recognition for nearly a decade.

3.1.6 UCF-101

The UCF-101 [98] dataset is an extension of UCF-50 [80]. By containing 13,320 actions belonging to 101 classes, UCF-101 has been the largest dataset for video recognition for a long time. UCF-101 also gathers videos from YouTube, and collects actions taking place in unconstrained environments. The videos display unsteady camera motion and various quality and lighting conditions, making UCF-101 a more realistic dataset, especially compared to its predecessor UCF-50. The nature of the actions ranges from music (playing an instrument), sport (both indoors and outdoors) and daily life tasks (e.g. brushing teeth). Videos are trimmed and contain only one action.

As mentioned above, UCF-101 and HMDB-51 constituted together a standard benchmark for action classification. Interestingly, despite having twice as many classes and videos as HMDB-51, all works in the literature report a significantly higher classification accuracy on UCF-101 compared to

Figure 3.6: Activity Net semantic hierarchy for action classes. Figure from [10].

HMDB-51. For example, I3D [11] (reviewed in Section 3.2.4) reports a top-1 accuracy of 93.4% and 66.4% on UCF-101 and HMDB-51 respectively.

3.1.7 THUMOS-14

THUMOS-14 [48] was the first action recognition challenge involving untrimmed videos. THUMOS-14 released a large dataset consisting of several components: a training set coinciding with the entirety of UCF-101, validation and test sets containing untrimmed videos and a background set. The untrimmed videos were temporally annotated for actions belonging to 20 classes. Additionally, spatio-temporal annotations were released for a subset of 24 actions.

The untrimmed videos in the validation and test sets display one or multiple action instances belonging to one class in most cases. The action classes correspond to those of UCF-101, i.e. range from daily-life activities to sport actions. The untrimmed videos were chiefly professionally shot and edited, however in some cases videos were filmed with consumer cameras by people recording some daily activity. The THUMOS-14 dataset is large in size, containing more than 254 hours of videos (over 25 million frames). Both action classification and localisation were evaluated on THUMOS-14.

3.1.8 Activity Net

Activity Net [10] is to date one of the largest and most complex datasets for video understanding. It comprises both trimmed and untrimmed videos, for a total of 200 classes and nearly 20,000 videos. Activity Net organises actions in a hierarchical manner. The authors define 7 main categories, namely: personal care, eating and drinking, household, caring and helping, working, socialising and leisure, sports and exercises. Each of these category is further divided into three sub-categories according to the involved social interactions and the place where the action occurs, forming thus a 4-level hierarchy for each action instance. Figure 3.6 depicts two hierarchy examples for “cleaning window” (top) and “brushing teeth” (bottom).

The authors compiled first a list of action classes to be collected. Given this list, videos were searched on various online sources such as YouTube. The authors then employed AMT workers to filter and annotate the videos. More precisely, given a video retrieved with the textual query, annotators needed first to verify that the video effectively contained the class associated with the query. After the filtering stage, the verified videos were temporally trimmed around each action instance. To ensure more precise labels, temporal bounds were obtained combining annotations from multiple AMT workers. The untrimmed videos are on average 5-10 minutes long, with a maximum length of 20 minutes. On average there are 1.5 action instances per untrimmed video. The number of instances per class is well balanced, displaying a nearly uniform distribution across the whole dataset. Compared to THUMOS-14 and many other datasets, Activity Net contains a greater number of action classes. Moreover, actions are more varied and include activities that are typically not present in other datasets, such as personal care, household, working and education activities.

Together with THUMOS-14, Activity Net is a standard benchmark for action recognition and localisation. More recently, both datasets have also been used to develop and test approaches that use video-level labels to learn actions from long untrimmed videos.

3.1.9 Kinetics

The video datasets reviewed so far are still relatively small in terms of classes when compared to still-image datasets. For example, Image Net [23] has 1000 classes, an order of magnitude more than any other existing video dataset. The Kinetics [12, 50] dataset was collected to narrow this gap. By containing 700 classes[1], with at least 600 videos for each class, Kinetics is the largest video dataset to date, both in terms of number of classes and videos, which amount to 650,317.

Kinetics was collected for classification only, and thus contains trimmed videos of an average length of 10 seconds. Kinetics was gathered to provide a dataset large enough to train deep networks for action classification. In fact, by becoming increasingly deeper and more complex, CNNs cannot be effectively optimised on small datasets. Kinetics can be seen as the video counterpart of Image Net: deep models pre-trained on Kinetics for the task of video classification can be adapted for several other tasks, in the same way image-based CNNs are typically pre-trained on Image Net for a wide variety of tasks.

The videos were all sourced from YouTube. The vast majority of the videos were not professionally shot and display substantial illumination variation, camera shake and clutter. The videos also show a great variety of performers (e.g. different clothing, age, body pose) who carry out actions in different ways and varying speeds. The dataset embraces a wide range of activities, such as single person actions (e.g. drinking, drawing), multiple people actions (e.g. hugging and shaking hands) and object interactions. As a result, some actions are more characterised by temporal dynamics, while other are more distinguished by the involved objects. Classes are organised in a two-level hierarchy, where each

[1] Kinetics contained 400 classes in its first release. This Section reviews the latest version available.

action class is grouped under a non-exclusive broad category (e.g. “playing the drums” is grouped under “music”).

Kinetics was compiled following an approach similar to that of Activity Net: a list of actions was first produced, and videos were then queried and filtered using AMT workers. To obtain a rough temporal location of the action, an image classifier was used to detect the object or scene related to the action label, for each frame of the video. A clip of 10 seconds was then trimmed around each frame where the classifier detected the object with high confidence. The video clips were then showed to AMT annotators, who had to indicate whether the video effectively contained the labelled action. In order to enforce a higher variability, only one clip was kept for each queried untrimmed video. This is in contrast to other datasets where multiple clips are typically trimmed from the same video.

3.1.10 Something-Something

The Something-Something dataset [39] contains videos showing people interacting with a large variety of objects in a number of ways. The distinctive trait of Something-Something is that its action labels generalise the involved objects. More precisely, action classes are textual descriptions of the object interaction where all objects are labelled as “something”, e.g. “putting *something* on a surface” or “pushing *something* from right to left”. The complexity of the actions ranges from simple interactions (e.g. “picking something up”) to more complex actions involving multiple objects (e.g. “attaching something to something”). Some actions are also purposely incomplete (e.g. “pretending to open something without actually opening it”) or unusual (e.g. “poking something so lightly that it doesn’t or almost doesn’t move”, “putting something that can’t roll onto a slanted surface, so it stays where it is”).

Instead of gathering existing videos, Something-Something was created by asking crowd-sourced workers (AMT) to record a short video given an action description like those mentioned above. Participants could choose any object to perform the action, which introduced a remarkable diversity in the dataset. In total 174 action classes and 22,0847 videos are available in the dataset[2]. Videos are trimmed and contain only one action, with an average length of 4 seconds.

The objects diversity, the complexity of the actions and perhaps above all the marginalisation of objects in action labels pushes recognition models to their limits. In fact, Something-Something is one of the most challenging datasets for action recognition to date. The dataset requires models to be able to reason about temporal relations, given that the object interactions are often complex and entail intricate temporal dynamics. Indeed, as showed in recent works [61, 123], performance on Something-Something is typically lower compared to other datasets where temporal reasoning is less needed (e.g. Kinetics, UCF-101). This was also showed in [123], which shuffled frames in Something-Something and UCF-101 and compared classification accuracy on the shuffled and normal videos. On UCF-101, where actions can be learnt via static appearance or short motion patterns, performance on the shuffled videos was virtually identical to the normal videos. On Something-Something instead,

[2]Numbers relative to the second and most recent version of the dataset.

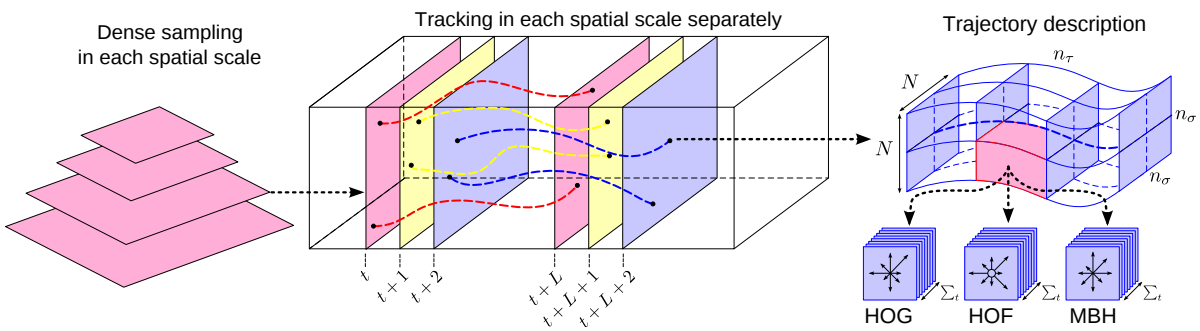

Figure 3.7: Dense trajectories overview. Points are first sampled at multiple spatial scales (left). Trajectories are produced tracking each point over subsequent frames in the corresponding spatial scale (middle). Trajectories are then described by their shape and the appearance and motion information extracted around them. Figure from [109].

results were considerably worse on the shuffled videos. This not only proves the difficulty of the dataset, but importantly shows that learning complicated temporal dynamics is the current step forward in advancing action recognition and video understanding.

3.2 Models

This Section analyses influential models for action recognition in videos. Starting from the last popular hand-crafted features before the advent of deep learning, this Section provides a brief overview of recent milestones in the field, such as two stream architectures and 3D convolutional models. The review is limited to temporally fully supervised approaches, i.e. methods that rely on the availability of trimmed action instances. The reader can refer to Section 2.2 for a review of works that use weak temporal supervision.

3.2.1 Dense Trajectories

Wang et al. [109] proposed one of the most successful and popular approaches to encode visual features before the spread of deep learning. Figure 3.7 illustrates the method. The authors start by densely sampling points at multiple spatial scales. Trajectories are then formed by tracking points in an optical flow field, at each scale separately. To prevent trajectories from drifting, points are tracked for 15 frames. Trajectories with sudden large displacements, which are likely to be noisy, are removed. Static trajectories are also removed.

Once the trajectories are obtained, several information is extracted to describe the motion and visual appearance around the tracked points. Firstly, the authors observe that the shape of a trajectory encodes local motion patterns, and thus measure the relative movements of the trajectory points in the x and y axes to represent its shape. Additionally, local spatio-temporal information is captured from a 3D volume around the trajectory. Specifically, HOG [18] and HOF [60] features are extracted

from the frames contained in the volume. The optical flow aligned with the volume is also described using Motion Boundary Histogram (MBH) [19].

HOG, HOF and MBH features are finally encoded with standard bag of words, which are then classified with an SVM. The authors evaluate four datasets with different complexity, namely KTH [59], YouTube [64], Hollywood 2 [67] and UCF Sports [83]. Experiments showed that the proposed dense trajectories attained a new state-of-the-art in action recognition.

Improved Dense Trajectories The same authors improved their approach in [108], where camera motion is estimated to enhance the quality of the trajectories. Assuming that the global motion between two consecutive frames is typically small, Wang and Schmid [108] find the homography between contiguous frames to estimate background motion, using RANSAC [31] on SURF [4] features points. The estimated homography is utilised to rectify the video frames, which reduces the camera motion. Trajectories are then extracted from the warped video as in the previous approach [109]. Suppressing camera motion is beneficial for the motion descriptors employed along the trajectories, i.e. HOF [60] and MBH [19], given that the optical flow for background movement non related to the actions is reduced. Trajectories that were likely generated by camera motion are also removed.

The authors note that people's bodies often occupy a large portion of video frames. Given that the method relies on matching feature points between frames to estimate the homography, a dominant human figure can be problematic since people generally do not move consistently with the camera. To address this issue the authors detect people in the video and remove features points lying within the persons' bounding boxes. This is done only to estimate the camera motion, i.e. trajectories that align with moving people are kept. The authors show that this approach further refines the camera motion estimation and consequently the quality of the trajectories.

Trajectories are encoded with bag of features and Fisher vector [84]. The encoded features are reduced with PCA and finally classified with an SVM. The authors show that the improved dense trajectories outperforms their previous approach, as well as other baselines, on the Hollywood 2 [67], HMDB 51 [54], Olympic Sports [76] and UCF 50 [80] datasets.

3.2.2 Two-Stream Convolutional Networks

Simonyan and Zisserman [95] proposed the first two-stream architecture for action recognition in videos. The model, which proved highly influential for momentous advance in the field, is based on two separate spatial and temporal streams that capture the static and dynamic information contained in a video. This approach combines complementary cues. The spatial stream essentially classifies actions from still images, focusing on scene appearance and objects. This is useful since many actions are strongly characterised by the interacted objects and the place where the action occurs. Certain actions are however better represented with motion, especially in those cases where an object can be manipulated in multiple ways. The temporal stream captures motion cues to help disambiguate such cases, as well as to provide supplementary information to the whole model.

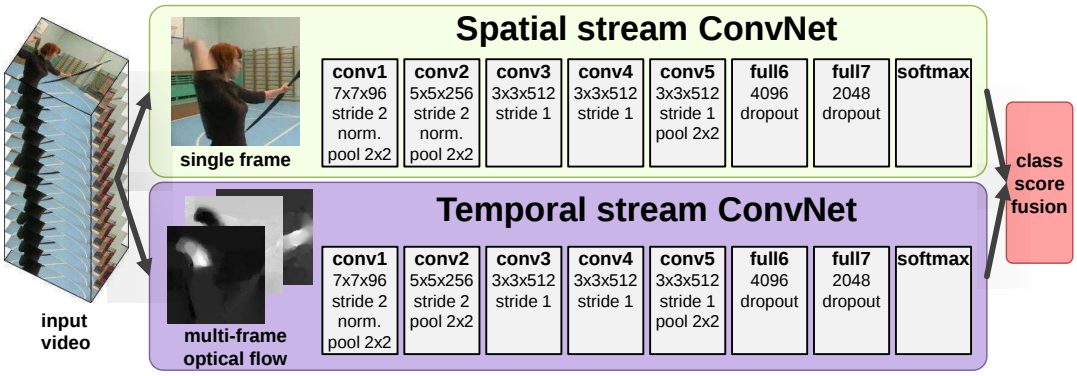

Figure 3.8: Two-Stream architecture for action recognition in video. Figure from [95].

Figure 3.8 depicts the proposed architecture. During training, the spatial stream receives a single frame randomly sampled from the video. The temporal stream instead receives a randomly sampled stack of $2n$ consecutive optical flow frames (both horizontal and vertical components). In the experiments, $n \in \{1, 5, 10\}$, with 10 yielding higher accuracy. At test time, 25 evenly spaced frames (or stacks) are sampled.

The two streams are trained independently. The softmax scores produced by the two CNNs are late-fused to predict the action class during testing. Two different fusion methods are evaluated: averaging the classification scores of the two streams and training a multi-class SVM using the softmax scores as features, which experimentally proved to be more effective.

The spatial network is pre-trained on Image Net [23]. To fine-tune the temporal CNN on the smaller UCF-101 and HMDB-51 datasets, the authors adopt a multi-task learning strategy and train the temporal stream using both datasets at the same time. This training technique allows using additional data to prevent overfitting.

Simonyan and Zisserman [95] proposed the two-stream architecture at a time (2014) when CNNs were still behind models based on hand-crafted features. The framework was one of the first CNN models able to reach comparable results to those obtained with Improved Dense Trajectories [108], which was the de-facto dominant action recognition framework. For these reasons, [95] can be considered a ground-breaking work in the field of action recognition.

Convolutional Two-Stream Network Fusion Late fusion is a simple way to combine streams. Given that the CNNs are trained independently, and because fusion takes places only at the classification level, two-stream architectures that use this approach cannot learn pixel-wise correspondences between spatial and temporal features. To address this issue, Feichtenhofer et al. [29] proposed a two-stream network based on [95], where the spatial and temporal streams are fused with a 3D convolutional layer. This allows the model to learn discriminative spatio-temporal features for the actions.

The authors evaluate two different approaches for fusing the streams, which are illustrated in Figure 3.9. The spatial and the temporal streams can be combined into a single CNN after a certain

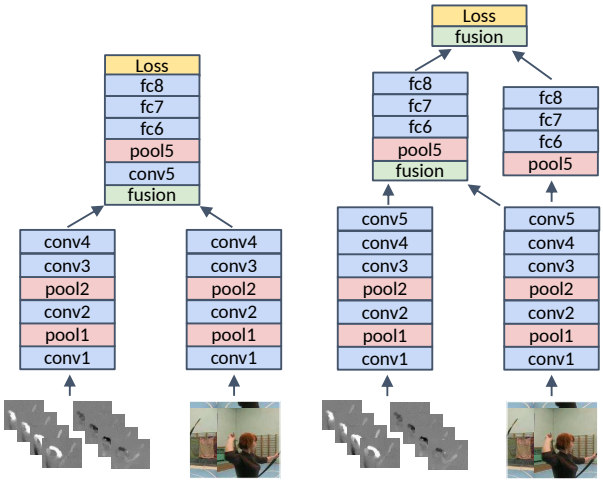

Figure 3.9: Fusing the spatial and temporal streams in two different ways. Left: the two streams are combined into one single CNN after the fourth convolutional layer. Right: the spatial stream is fused into the temporal stream after the fifth convolutional layer. The spatial CNN is kept and later on fused with the spatio-temporal hybrid CNN. Figure from [29].

convolutional layer, as depicted on the left-hand side of the Figure. This has the advantage of lowering the number of parameters. Alternatively, an architecture with two towers can be adopted, as showed on the right-hand side of the Figure. In this case the spatial stream is merged with the temporal stream and is kept as a separate CNN, which is finally fused with the spatio-temporal CNN after the fully connected layers.

The authors also consider two ways to aggregate features maps over time. The first method is 3D pooling, which is simply the extension of 2D max pooling to the temporal domain. The other aggregation method is 3D convolution followed by 3D pooling. In this case the temporally stacked feature maps are first convoluted with a 3D kernel, and then max-pooled as in the first method. This approach allows to learn spatio-temporal relationships between features, which can be more effective to model actions in time.

Based on experimental evaluation on UCF-101 and HMDB-51 the authors find the architecture that delivers the best performance. The proposed model fuses the two streams after the last convolutional layer using 3D convolution followed by 3D pooling. The temporal CNN is kept and 3D pooling is performed over the stacked temporal feature maps. The prediction scores of the two streams are averaged to produce the final classification score. During training and testing 5 frames are randomly sampled. These constitute the input to the spatial stream, whereas the temporal CNN receives stacks of 10 optical flow frames centred on each frame. The authors employ VGG-16 [96] pre-trained on Image Net as the backbone CNN for both streams.

Learning spatio-temporal correspondences between the two streams proves effective. In fact, on both UCF-101 and HMDB-51 the convolutional two-stream fusion architecture outperforms other CNN based approaches, particularly models based on other two-stream designs, as well as LSTM and

models entirely based on 3D convolutions. Indeed, Feichtenhofer et al. [29] combine the advantages of using 3D kernels, to learn spatio-temporal relationships between features, with the advantages of using two streams, to capture both spatial and temporal cues from the videos. The proposed model also outperformed Improved Dense Trajectories [108] by around 5% on the two evaluated datasets, marking the beginning of deep learning models taking over hand-crafted approaches.

3.2.3 Temporal Modelling with 2D CNNs

Temporal Segment Networks Wang et al. [110] note that most algorithms sample a small amount of frames for training, mainly due to memory and computational limitations. This corresponds to feeding the classifier with a temporally limited view of the video, which can be suboptimal for long actions. Some approaches [25, 106, 120] employ denser sampling, however these involve large computational costs and importantly are still unable to cover long actions effectively.

Wang et al. [110] thus proposed an efficient way to capture long-range dynamics from videos. Their framework, called Temporal Segment Networks (TSN) is based on the observation that consecutive frames are vastly similar, and thus dense sampling is often redundant. Instead, TSN sparsely samples frames across the entire length of a video. Specifically, TSN splits a video into n segments of uniform length, and then randomly draws one frame from each segment. This allows to capture complex long-term action dynamics while keeping computational requirements low. During training, each sampled frame is fed to a frame-based CNN to produce class predictions, for each of the n frames separately. The scores obtained from each frame are then aggregated with a consensus function to combine the n predictions for the whole video.

The authors employ Inception with Batch Normalisation [45] as the backbone of TSN, using a two-stream architecture with late fusion. For testing, the RGB and optical flow scores of 25 uniformly sampled frames and stacks are fused with a weighted average, where the spatial stream is assigned a higher score, based on empirical evaluation. Motion is represented with optical flow images as well as with two other modalities, namely RGB difference and warped optical flow. RGB difference corresponds to the pixel-wise RGB changes between two consecutive frames, which can be also used to encode motion. Warped optical flow aims to reduce the amount of camera motion to obtain a representation that is more focused on the action movements. This is done as in [107], where the camera motion is suppressed by first estimating the homography between contiguous frames, with optical flow then being computed on the warped frames.

TSN is evaluated on UCF-101 and HMDB-51. The authors analyse the different components of the framework, fixing the number of training frames to 3. On UCF-101, the best performance is achieved fusing all modalities except RGB difference. When using the same backbone CNN without TSN sampling, the authors report a 1.5% accuracy drop, proving that long-range dynamics are better captured when using the proposed sampling strategy. The authors achieve a new state-of-the-art on both datasets, outperforming [95] (the first late-fusion two-stream architecture) by 10% on HMDB-51 and 6% on UCF-101, showing the efficacy of the employed sparse sampling and training techniques.

Temporal Relation Network Temporal relation reasoning refers to the ability of linking the transformation of an entity (an object or a person) over the course of time. Zhou et al. [123] observed that actions in many popular datasets such as UCF-101 can be recognised without temporal relations reasoning, i.e. RGB or optical flow images without further temporal analysis are sufficient for state-of-the art methods to succeed well on such datasets. This is the case when actions are strongly characterised by the appearance of the involved objects and actors, or by the motion patterns. However, when the actions are characterised by temporal relationships and transformations between the involved entities, typical recognition approaches fail to successfully model the action.

Motivated by this, Zhou et al. [123] proposed Temporal Relation Network (TRN). TRN was inspired by [85], which proposed a module to learn the spatial relationship of objects in static images. TRN is effectively simple: a multi layer perceptron (MLP) θ is employed to model the relation between temporally ordered pairs of frame. More precisely, θ receives the frames' features produced by a given CNN. Another MLP ϕ operates on the output produced by θ on all the combinations of temporally ordered pairs of frames. The two MLPs are then extended to work on ordered tuples of n frames. This amounts to encode the relation between a sequence of frames at multiple temporal scales. The output of ϕ is used to predict the action. During training, n random ordered frames are sampled, while during testing frames are uniformly sampled throughout the video. The whole network with the TRN module is optimised with a standard cross-entropy loss.

The authors plug the TRN module to Inception with Batch Normalisation [45] to evaluate their method on the Something-Something [39], the 20BN Jester [1] and Charades [91] datasets. An all datasets, TRN outperforms other approaches, highlighting the importance of temporal relation reasoning. This is especially the case for Something-Something, which as seen in Section 3.1.10 contains actions that are heavily characterised by intricate temporal dynamics.

As mentioned earlier when reviewing Something-Something, the authors also present an interesting experiment. They shuffle the frames of UCF-101 and Something-Something and compare the performance obtained with TRN on the shuffled and normal videos. On UCF-101 classification accuracy remains high on the shuffled videos, while on Something-Something the authors report a wider gap (15% top-1 accuracy) between shuffled and normal videos. This confirms the aforementioned observation that UCF-101 contains actions that do not need temporal relation reasoning to be successfully recognised, unlike Something-Something which benefits from temporal relation modelling.

Temporal Shift Module Lin et al. [61] note that while 2D CNNs are usually computationally efficient, they struggle to capture salient spatio-temporal relationships between frames. On the other hand, 3D architectures can model temporal patterns more effectively and achieve higher accuracy, however this comes at a greater computational cost. The authors thus design a module called Temporal Shift Module (TSM) to obtain high performance with low computational overhead.

TSM is a module that can convert any image classification CNN to a pseudo-3D model. The module works by shifting spatial feature maps along the temporal dimension. The TSM module is

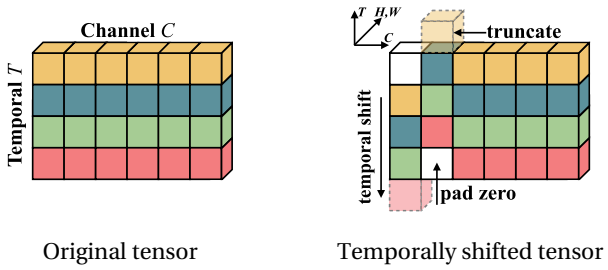

Figure 3.10: Temporal Shift Module. Spatial feature maps extracted for 4 frames are stacked. The values of the first channel are pushed one frame backwards, while the values of the second channel are pushed one frame forwards. The remaining channels are not shifted. Figure from [61].

illustrated in Figure 3.10. Given a random set of contiguous frames and their respective feature maps, a subset of channels is shifted one frame ahead, while another subset of channels is shifted one frame behind. The remaining channels are not shifted.

The temporal shift operation interleaves spatial information from temporally close frames, which endows the 2D CNN with the capability of modelling spatio-temporal relationships between frames. Both RGB and optical flow images are used in a late-fused two-stream fashion. During training, 8 or 16 consecutive frames are randomly sampled. For testing, the same number of frames are uniformly sampled and classification scores are averaged to predict the action.

TSM is not only cost-effective, given that the shift operation has a little overhead, but also delivers high accuracy on a number of datasets: Kinetics [50], Something-Something [39], 20BN Jester [1], UCF-101 [98] and HMDB-51 [54]. The authors first compare their model to a 2D baseline, i.e. TSN [110] equipped with the same backbone CNN. While TSM outperforms TSN on all datasets, the performance gap is more remarkable for those datasets that focus on temporal modelling. In fact, on Something-Something and 20BN Jester the authors report an improvement of respectively 29% and 12% with respect to TSN. On Something-Something, TSM also achieves higher accuracy than TRN [123] (7% and 8% higher on versions v1 and v2), which is designed to learn temporal relations between frames, and I3D [11], which by employing 3D convolutions is computationally expensive.

The results demonstrates that TSM, while still adopting a 2D architecture, is highly capable of modelling temporal relations. The authors compare also performance vs FLOP, showing that TSM, being entirely based on inexpensive temporal shifting, keeps computational requirements low.

3.2.4 Two-Stream Inflated 3D CNN

Carreira and Zisserman [11] proposed a 3D architecture where 2D filters of image classification CNNs are inflated to obtain a spatio-temporal model. This has the great advantage that successful architectures for image based tasks, and importantly their pre-trained weights, can be used for the task of video action classification. Carreira and Zisserman [11]'s work was motivated by the fact that CNNs for various tasks such as pose estimation and object segmentation have gained remarkable performance boost when using Image Net pre-training.

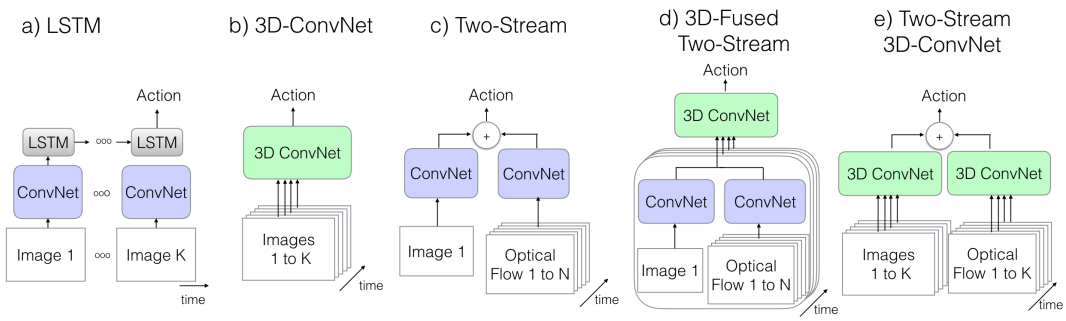

Figure 3.11: Comparison between different architectures for action recognition. Figure from [11].

The authors discuss several state-of-the-art architectures, which are illustrated in Figure 3.11. The first evaluated design is a CNN+LSTM model, where visual features produced by an image classification CNN are fed to an LSTM. This is a standard approach to overcome the lack of temporal modelling of CNNs that operate on single images. LSTMs are able to capture long-term high level dynamics, however they may struggle to model salient fine-grained brief motion when receiving solely spatial features.

Secondly, the authors evaluate a 3D CNN (C3D [103]). While 3D CNNs constitute a natural way to model time, their number of parameters makes them difficult to train. Furthermore, standard 3D CNN like C3D cannot benefit from 2D models pre-training due to their design, which limits their performance since they have to be trained from the scratch. This major drawback will be in fact the key factor that inspired the design of the inflated 3D model. The authors then evaluate two-stream architectures, namely the original late fusion approach proposed in [95] and the subsequent framework which employs convolutional fusion [29], both reviewed earlier in this Section.

Finally, the authors introduce their new model, Two-Stream Inflated 3D CNN (I3D). The key idea is that, rather than attempting to design another 3D model, state-of-the-art 2D CNNs can be converted to 3D models. This is done by inflating both convolutional and pooling kernels, i.e. by adding a third (temporal) dimension to the existing filters. The weights of pre-trained models are inflated too. Specifically, the parameters of 2D convolutional filters are replicated n times in order to form a cube. The weights of the repeated 2D kernels are then averaged along the temporal dimension. This is as a matter of fact the winning ingredient of I3D, especially in comparison to other 3D architectures.

The authors experimentally found that although 3D models can intrinsically learn temporal information from RGB images, using an additional stream that operates on optical flow images further improves performance. The authors thus train a spatial and a temporal CNN independently, and average their predictions during testing (late fusion). For training, a random stack of 64 consecutive frames is sampled, whereas for testing the whole video clips is fed to the network.

The I3D model is compared to the aforementioned recognition architectures on UCF-101 and

HMDB-51, as well as on the then-recently introduced Kinetics[3] dataset [50]. On all the three datasets, I3D outperforms all the other approaches by 5% top-1 accuracy on average. Late-fused and 3D fused two-stream models perform comparably to each other and follow I3D. The CNN+LSTM model, which operates only on RGB images, scores closely lower than the two-stream models. C3D achieves the lowest performance, which is likely due the lack of pre-training. In fact, all models were pre-trained on Image Net, with the exception of C3D which was trained from the scratch, given the previously discussed unavailability of pre-trained 2D weights for this type of architecture.

To highlight the importance of pre-training, the authors also use a subset of Kinetics to pre-train all the compared models before fine-tuning them on UCF-101 and HMDB-51. All architectures benefit from the Kinetics pre-training, especially C3D which gains a 30% and 25% accuracy boost on UCF-101 and HMDB-51. The CNN+LSTM and the two-stream models instead, which are trained using a much sparser input, do not experience a large improvement (reported boost was around 1% on average) when pre-trained on the Kinetics subset. This is probably because when models receive a sparse input, videos do not appear as different from still images as they do to 3D models.

3.2.5 Factorised 3D Convolutions

R(2+1)D [104] and S3D [116] factorise 3D convolutions into separate spatial and temporal convolutions. In both cases, 3D filters of size $t \times d \times d$, are replaced by one 2D filter of size $d \times d$ applied on t frames. A temporal convolution is then applied on the t spatial maps with a kernel of size $t \times 1$.

This approach has two advantages. Firstly, the models are easier to optimise given that there are no 3D kernels to be tuned. Secondly, decomposing 3D convolutions, which are more prone to overfitting, can lead to higher classification performance. This is supported by experiments on Kinetics [50], Something-Something [39] Sports 1M [49], UCF-101 [98] and HMDB-51 [54]. On these datasets, factorised 3D models were able to achieve comparable or superior performance to state-of-the-art accuracy, including I3D [11] which is a fully 3D model. More precisely, [104] scores less than 1% worse than I3D, while [116] outperforms I3D by 3% on Kinetics' validation set.

* * *

This Section provided an overview of recent milestones in the field of action recognition. Starting from hand-crafted dense trajectories, moving then from two-stream architecture and models employing 3D convolutions, we saw how video understanding has progressed over the last few years. Given the availability of increasingly more complex and realistic datasets (e.g. Something-Something), modelling complicated temporal dynamics (e.g. TRN and TSM) appears to be the current step forward in accelerating progress in the field. Increased computational resources also play an important role, making deep 3D CNNs widely used nowadays, a scenario that perhaps seemed unrealistic only a few years ago.

[3] Reporting results on the first edition of Kinetics with 400 classes.

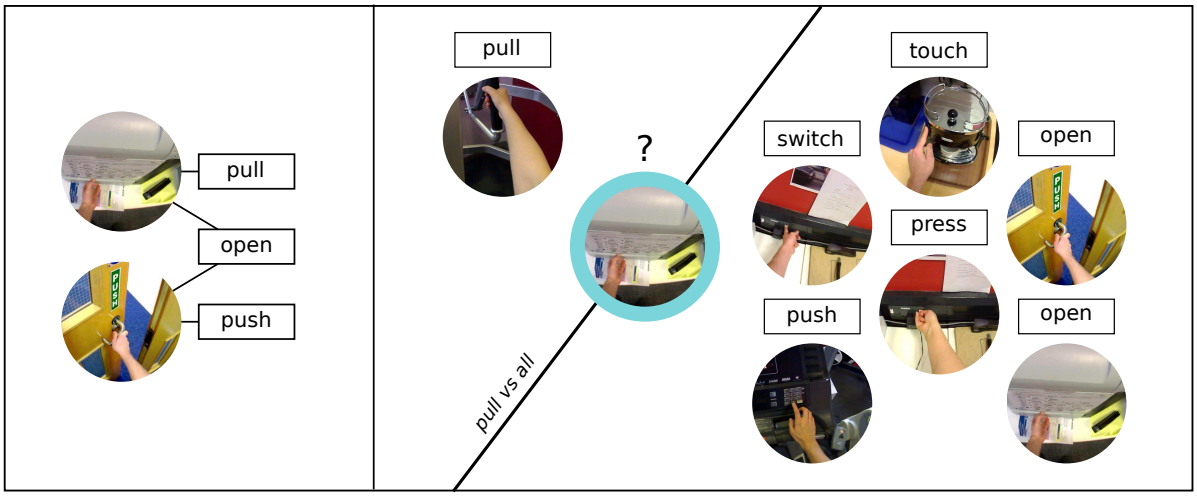

Figure 3.12: Open vocabulary labelling challenges standard classification approaches. Left-hand side: when different people label the same object interactions, a generic valid verb like "open" may be attached to visually distinct but semantically identical actions, such as opening a drawer or a door. Similarly, one object interaction can be described by multiple labels, e.g. the act of opening a door can be annotated with the verbs "push" and "open". Standard classification approaches (right-hand side) typically operate on classes that do not overlap semantically, and thus struggle to classify actions in open vocabulary settings.

3.3 Features Encoding for Action Recognition with Semantic Embedding

This Section provides an overview of a joint work carried out during the first year of my PhD [114]. The proposed framework exploits semantic knowledge and visual cues to classify action segments in an open vocabulary setting, i.e. a scenario where class labels may semantically overlap. The model was nicknamed SEMBED, standing for Semantic Embedding of Egocentric Action Videos. SEMBED helped disambiguate cases where identical actions were assigned distinct valid classes, a situation where standard classification approaches would struggle.

After SEMBED my research took other directions distant from semantics analysis. For this reason, a thorough review of the framework's semantic reasoning would be out of the scope of this Thesis. This Section instead mainly focuses on experiments conducted for SEMBED on the representation of action segments using different visual features and encodings.

Overview of SEMBED Figure 3.12 illustrates the scenario in which SEMBED operates. Given a dataset containing fine-grained object interactions that were annotated using open vocabulary (left-hand side), generic verbs like "open" can be attached to visually distinct object interactions, such as pull (drawer)" or "push (door)". Likewise, multiple verbs can describe the same object interaction. Typical classification approaches normally assume that classes do not overlap semantically (right-hand side of the Figure). When this assumption does not hold true, standard classifiers struggle to

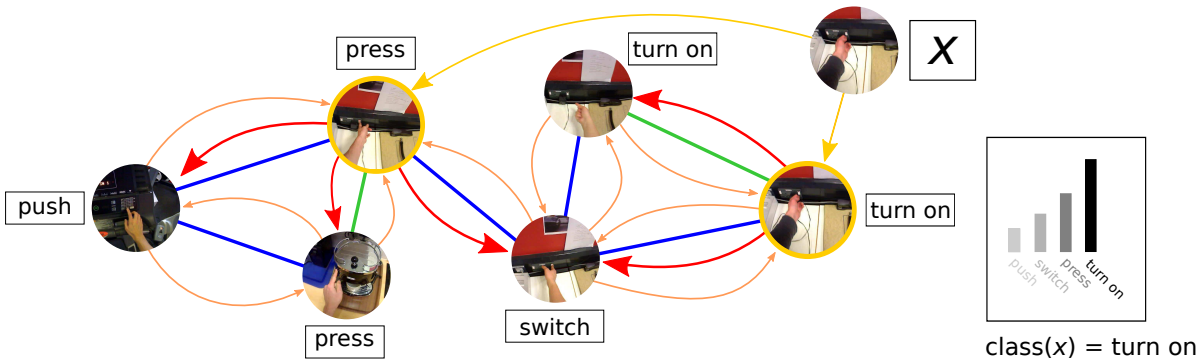

Figure 3.13: Semantic embedding for action recognition in open vocabulary setting. A graph is first built to link action segments (nodes) according to their semantic and visual relationship (green and blue edges respectively). Given an unknown action segment (node x), the most visually similar videos in the graph are first found (yellow nodes). Starting from these nodes a Markov walk of two steps is performed to estimate the class of x. Red and orange arrows correspond to the first and second step of the Markov walk.

build a discriminative representation of the actions.

SEMBED employs verb-only labels, discarding any noun annotations. This was motivated by the fact that clustering actions according to the involved objects separates identical actions into different classes. For example, the act of pouring some liquid into a container is the same regardless of the liquid (e.g. water or oil) or container (e.g. cup or bowl). While object super-classes could be used to group similar objects, ultimately very distinct objects can be interacted in the same way, like the drawer and the door seen in Figure 3.12, which can be both opened or closed.

Figure 3.13 illustrates the approach proposed in SEMBED. A graph is first built to link action segments (the nodes) based on visual similarity (blue edges) and semantic relationship (green edges). Note how the action of turning a socket switch on is represented in the graph with multiple nodes labelled with different valid verbs ("press", "turn on" and "switch"). Edges are assigned a weight according to the similarity between video frames and verbs. Semantic similarity is estimated using Word Net [72], whereas visual similarity is evaluated based on the distance between vectors in a visual feature space.

Given an unknown video (node x in the Figure), the most visually similar videos are first found in the graph (yellow nodes, "press" and "turn on"). From such nodes, a Markov walk of 2 steps (red and orange arrows) is then performed to estimate the class probability for x, based on the classes of the visited nodes.

3.3.1 Features and Encodings

Two different kinds of visual features were evaluated in SEMBED. These were IDT (reviewed in Section 3.2.1) and OverFeat [89], a CNN designed for the tasks of object recognition and localisation in still images. IDT was chosen as it was the leading approach in action recognition at the time of the

study (late 2015), whereas OverFeat was chosen for its efficiency and high performance on the Image Net [23] classification challenge. IDT and OverFeat features were encoded with Fisher Vector (FV) [84] and Bag of Words (BOW) [17].

Fisher Vector encoding is based on the Fisher Kernel principle of [46], which derives a kernel from a generative model of the data. More precisely, the Fisher Kernel works by characterising a given sample by its deviation from the generative model, producing a representation which is the Fisher Vector. In this case, the samples correspond to either the IDT trajectories or the OverFeat features. The generative model used in the FV encoding is the Gaussian Mixture Model (GMM). Consequently, the number of Gaussians that compose the mixture model is an important hyper-parameter of Fisher Vectors.

Bag of Words assigns a given sample to the closest entry in a "visual vocabulary", also called codebook. The codebook is learnt by clustering a set of samples with k-means. BOW works by producing a histogram (or bag) of "visual words", i.e. by identifying and counting the occurrence of each codebook entry in a given sample. The number of entries in the BOW codebook is an important hyper-parameter of the model. This parameter can be seen as the equivalent of the number of Gaussians in the GMM for the Fisher Vector encoding, since both change the dimension of the models' feature space. As we will see next, these parameters can affect the representation of a video (and therefore its classification) in a considerable manner.

3.3.2 Experiments on Feature Encodings

The CMU-MMAC, GTEA Gaze+ and BEOID datasets (reviewed in Section 3.1) were evaluated for this study. Classes were formed using the labelled verbs, discarding the object nouns. CMU-MMAC and GTEA Gaze+ provide annotations with semantically non overlapping verb-noun labels, with a total of 31 and 42 verb-noun classes for the two datasets respectively. As a result the number of verbs annotated in the two datasets is small, being respectively 12 for CMU-MMAC and 25 for GTEA Gaze+. BEOID offers instead annotations collected from 20 native English speakers who labelled the actions using any verb they saw fit. This resulted in a higher semantic variability, with a total of 75 verbs.

IDT and OverFeat features were extracted from each action video clip. To handle the abundant number of IDT features extracted from the videos (which were in the order of hundreds of millions for a single dataset), FV and BOW encodings of IDT were calculated using a random 25% of the feature vectors. Given that OverFeat operates on single images, features obtained from every 5th frame were concatenated in order to produce a descriptor for a video. OverFeat, which was pre-trained on Image Net, was not fine-tuned and was used exclusively to extract visual features from the RGB video frames.

In order to compare IDT to OverFeat and FV to BOW, the encoded action segments were classified using KNN (with $k = 5$) and a linear SVM. As mentioned above, the number of Gaussians and the number of words are two important hyper-parameters of FV and BOW. To assess how these parameters affect classification, results were obtained with the number of Gaussians (γ_{fv}) ranging in $\{5, 10, 50, 100, 256\}$ and number of words (γ_{bow}) ranging in $\{5, 10, 50, 100, 256, 512\}$.

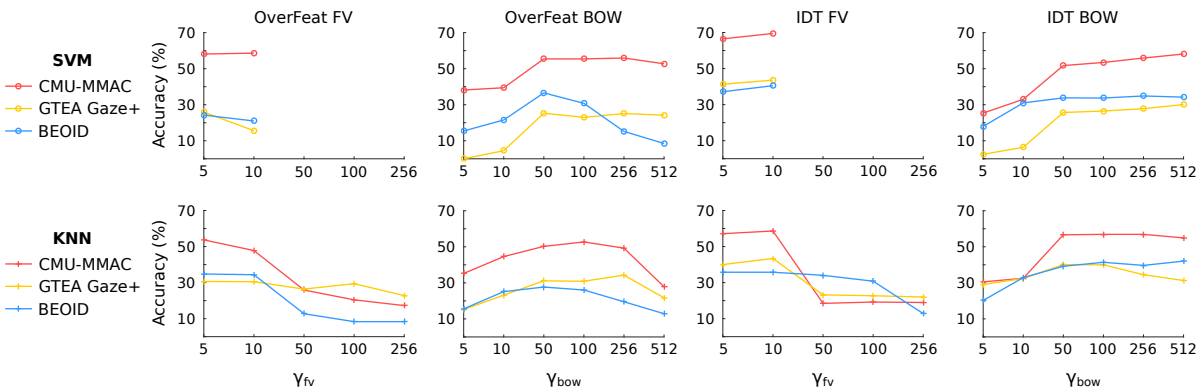

Figure 3.14: Top-1 accuracy obtained with SVM and KNN on CMU-MMAC, GTEA Gaze+ and BEOID, with different encoding parameters for FV and BOW. Number of classes in the three datasets is 12 (CMU-MMAC), 25 (GTEA Gaze+) and 75 (BEOID). SVM results using FV encoding are provided only with $\gamma_{fv} \in \{5, 10\}$ due to computational efficiency.

Figure 3.14 reports classification accuracy obtained on the three datasets. IDT outperforms OverFeat regardless of the classifier and encoding on all datasets. This was expected as IDT is a descriptor specifically designed for action recognition in videos based on both motion and static information. OverFeat on the other hand is a CNN designed for object recognition which extracts appearance features from static images. In spite of the deep learning architecture and the pre-training on Image Net, the lack of motion information impacts the effectiveness of OverFeat for action recognition in videos.

Classification accuracy varies according to γ_{fv} and γ_{bow} on all datasets with both features. When using FV encoding the highest accuracy is achieved with a small number of Gaussians (5 and 10). This is probably due to the fact that as the number of Gaussians increases, the encoded feature vectors become sparser. Sparsity negatively affects the dot product, which is used by both SVM and KNN to calculate the distance between feature vectors, and thus lower accuracy is observed with a larger number of Gaussians. Conversely, a greater number of visual words leads to higher performances when BOW is used. This is because the dimension of BOW-encoded feature vectors corresponds directly to the number of words of the codebook (γ_{bow}), i.e. vectors of small dimension (e.g. 5 or 10) are unlikely to provide a sufficient representation of the actions.

The accuracy trend of the four features-encoding combinations are by and large the same for the three datasets and the two classifiers. Due to its small number of classes (12), CMU-MMAC is the dataset where the highest accuracy is achieved. Despite the difference of classes between GTEA Gaze+ (25) and BEOID (75), the performance on these two datasets is comparable, with BEOID scoring higher than GTEA Gaze+ in some cases. This is perhaps due to the more varied environment in which actions take place in BEOID. In fact, actions in BEOID take place in 6 different locations, with many actions occurring exclusively in one place. This might help the classifiers which can associate actions with the discriminative appearance of the locations. In GTEA Gaze+ all actions take place in one single environment, and thus classifiers cannot rely on salient locations to disambiguate the actions.

SVM results with FV encoding were obtained only with $\gamma_{fv} \in \{5, 10\}$ due to the impractical size of the encoded vectors obtained with $\gamma_{fv} \geq 50$. The dimension of the FV-encoded feature vector (for a single action segment) is $s(v) = \gamma_{fv} \times d$, with $d_{cnn} = 12,288$ and $d_{idt} = 426$ being the dimension of the OverFeat and IDT descriptors respectively. Using $\gamma_{fv} \in \{5, 10, 50, 100, 256\}$ the resulting FV size would range between 61,440 and 3,145,728 using OverFeat features, while with IDT features the FV size would range between 2,130 and 109,056. Given that the tested datasets contain more than a thousand of actions, the computational cost of running a CPU implementation of SVM on such high dimensional vectors is highly burdensome both in terms of memory and time. Dimensionality reduction by means of PCA was tested to overcome this issue. Although PCA helped to reduce the computational time, no improvement in accuracy was observed when using $\gamma_{fv} > 10$ with the PCA-reduced features.

3.4 Conclusion

This Chapter scrutinised seminal datasets and models, providing an analysis of the main advances in the field during recent years. Section 3.1 started discussing egocentric datasets and their typical shortcomings (limited actions and variability), which pushed towards the collection of EPIC Kitchens, the largest first-person video dataset to date where actions were recorded in native environments in a fully unscripted manner.

The Chapter continued analysing the evolution of non egocentric datasets as well. From the early days of HMDB-51 and UCF-101, we observed a rapid expansion of video datasets over the last few years, thanks chiefly to the explosion of online video platforms which are now used to collect data in a more scalable way. This growth is perhaps best represented by Kinetics and Something-Something. The former provides 700 classes and more than half-million videos, narrowing the gap with image-based datasets, which however remains still wide. Something-something also constitutes an interesting and challenging benchmark, where marginalised object labels and complex temporal dynamics push current models to their limits.

Temporal modelling is indeed the main area of investigation in the field of action recognition. Starting from approaches modelling temporal dynamics with hand-crafted trajectories or stack of optical flow frames, the community has witnessed the development of more sophisticated methods for temporal reasoning. As discussed during the Chapter, temporal modelling is a crucial aspect of action recognition which is increasingly gaining more attention. This is especially the case now, with the availability of more complex and temporally dynamic datasets, which require models to reason more about time.

CHAPTER 4

TEMPORAL BOUNDARIES FOR ACTION RECOGNITION

Temporal boundaries refer to the start and end times delimiting the extent of an action in a video. These, together with a class label, form the base annotations for training current action recognition models. In fact, typical supervised approaches sample frames within the labelled bounds to provide a view of the action to a classifier.

Marking the beginning and the ending of an action requires deciding when it starts and ends. This is often a subjective task. For instance, consider the act of pouring oil onto a pan showed in Figure 4.1. When does the action start? Does it start at frame 1, when the user grasps the pan, or does it start at frame 3, when the subject tilts the bottle? Or perhaps the action starts at frame 5, when we can see some oil on the pan? What about the action's end? Does it end at frame 7, when oil stops flowing, or at frame 9 or 10, when the person puts down the oil bottle and the pan? Also, you may have noticed that the Figure shows the subject pouring some oil first, then pausing, finally pouring more oil before putting down the bottle and the pan. Does this mean there are two instances of the same action or is it just one action altogether?

As you can see, the definition of the temporal extent of an action can be ambiguous and, from a human perspective, there are generally no right or wrong answers to the questions raised before. Yet, most works in action recognition gloss over this important matter, and simply assume the temporal boundaries are given by some oracle.

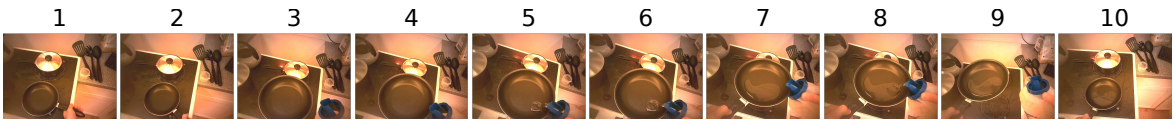

Figure 4.1: Defining the temporal extent of an action can be ambiguous. When does the action of pouring oil start and end?

This Chapter offers a deeper insight into the temporal labels that teach algorithms to learn actions from videos. First, a study on how different people disagree when annotating actions in a video is provided, and accordingly inconsistencies in three datasets for action recognition are inspected. As you might already expect, the study revealed that different annotators perceive the temporal span of an action in different ways.

Boundaries ambiguity and inconsistency relates to robustness. When temporal bounds vary, the visual content of the video segment enclosing the action changes too. This Chapter thus addresses the question whether small variations in temporal bounds can confuse recognition models. As we will see shortly, the performance of the evaluated algorithms dropped even when boundaries were modified negligibly.

The effect of temporal boundaries variations depends on the granularity of the considered human-world interaction. Intuitively, if we consider a short action spanning one or two seconds, shifting its start or end by one second will likely have a considerable impact. Conversely, if we examine a long activity extending over one minute, changing its bounds by one second is less likely to alter the visual content to an extent that changes the interpretation of the activity. Additionally, the disagreement between annotators is limited to relatively short amounts of time, i.e. annotators disagree in the order of seconds, not minutes, when marking either the beginning or the ending of an action.

For the reasons above, given that this study focuses on temporal bounds subjectivity, this work studies only minor boundaries variations and, accordingly, only short fine-grained actions. Specifically, videos showing humans interacting with objects for a short amount of time are evaluated, and therefore the terms “actions” and “object interactions” are used interchangeably. Interactions between people and their surrounding environment (such as walking or jumping) are not considered.

Finally, this Chapter proposes the Rubicon Boundaries, an annotation protocol that formally describes the temporal span of an action, in the attempt to alleviate ambiguity and assist annotators in identifying the temporal bounds of actions in videos. Experiments will show that annotators recognise the action intervals more consistently when instructed with specific guidelines. We will also see that classification algorithms benefit from more consistent temporal labels. In fact, when using new temporal annotations collected following the Rubicon Boundaries protocol, a performance boost on the biggest egocentric dataset available at the time was observed.

Before moving forward, it should be noted that this work was carried out between 2016 and 2017. As a consequence, due to the evaluated recognition models and datasets, some results and conclusions may be deemed outdated at the time of writing this Thesis. However, Chapter 5 will revisit some of the questions raised in this study, and will provide new analysis on more recent algorithms and datasets.

Figure 4.2: Video frames showing the actions labelled by multiple annotators. Actions depicted in a) were annotated locally, while actions in b) were labelled using AMT. Frames extracted from a video of the BEIOD [20] dataset.

4.1 Evaluating Agreement amongst Multiple Annotators

The annotations collected from multiple people in BEOID were inspected to assess how different annotators agree on labelling the temporal bounds of the same action. For this study the actions occurring in one video were selected. Figure 4.2a illustrates the actions. As detailed in Section 3.1.3, BEOID's annotations were collected by asking multiple people to label all the object interactions observed in an untrimmed video. The number of annotators per action varies in BEOID. The actions analysed here were labelled by five people. Annotators received instructions defining the requested temporal granularity of the labels, as well as guidelines on how to recognise the start and the end of an object interaction, as reported below:

> The **start** of an object interaction is the moment when you **first** recognise that the user is initiating **motion** to interact with the object. The **end** of an object interaction is the moment when you **first** recognise that the interaction is complete.
>
> Every activity (e.g. preparing a sandwich) should be split into its constituent actions (e.g. cut bread, spread butter, etc.). An action may also be interpreted as being composed of other sub-actions. In these cases, you should label each sub-action separately. For example, slicing bread requires a knife to be grabbed in advance: if the action of picking up the knife is visible, you should label "grab knife" and "slide bread" individually.

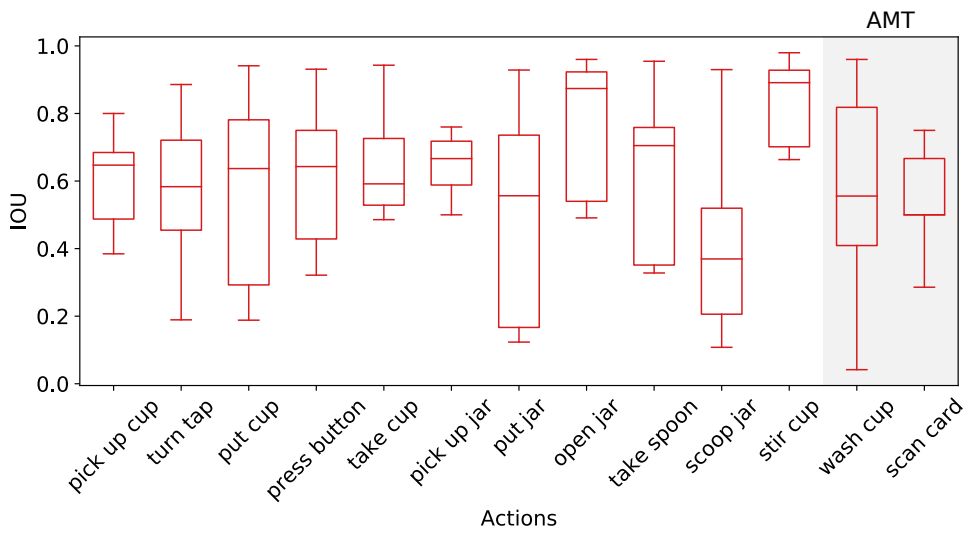

Figure 4.3: Temporal IOU box plots for several object interactions labelled by multiple annotators. AMT stands for Amazon Mechanical Turk, the crowd-sourcing platform employed to assess annotators agreement at a larger scale.

Annotations	**Average IOU**	**Average IOU SD**	**Average start SD**	**Average end SD**
Local (5)	0.62	0.19	0.62	0.25
AMT (100)	0.57	0.20	3.8	4.0

Table 4.1: Statistical comparison between local and AMT annotations. Numbers between parenthesis indicate the number of annotators. SD indicates Standard Deviation, which is expressed in seconds for the start and end times.

Annotators were also shown examples of good and bad annotations. The annotators were either Master or PhD students from the University of Bristol. Albeit knowledgeable in computer vision, they were not familiar with the arbitrariness of action temporal bounds, and most of them were not familiar with action recognition from videos altogether. This ensured a less biased response.

Figure 4.3 shows box plots for the temporal IOU calculated between all pairs of annotations, for each action. Annotators identified the extent of the actions in different ways. Indeed, for most actions the overlap between annotations varies considerably. As reported in Table 4.1, the average IOU is equal to 0.62, while its average standard deviation is equal to 0.19 seconds. These further indicate the low agreement amongst annotators. Marking the beginning of an action appears to be slightly more ambiguous than annotating its ending. This is reflected by a higher average standard deviation for the start time compared to the end, respectively equal to 0.62 and 0.25 seconds. These measures, also reported in Table 4.1, were obtained averaging the start/end standard deviation calculated for each action separately. Although the scale of the start/end standard deviation might seem insignificant, even half a second can involve substantial visual changes in egocentric videos, considering the typical level of density of the actions in the videos. The next Section will revisit this important aspect.

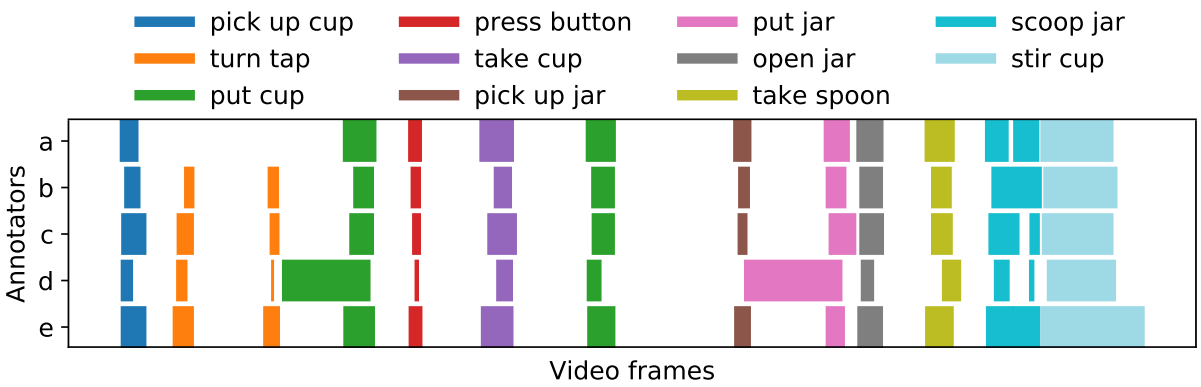

Figure 4.4: Action segments annotated by multiple people. The illustrated actions correspond to those depicted in Figure 4.3.

Certain object interactions exhibit more disagreement than others. For example, the IOU for the actions "put cup/jar" and "scoop jar" ranges within 0.15 and 0.95, with the latter action displaying the lowest median IOU. Conversely, actions like "pick up cup/jar", "open jar" and "stir cup" show less variability or a higher median IOU. This suggests that ambiguity might correlate with the nature of the action: temporal bounds for object interactions that are shorter (i.e. picking up an object) or that modify the state of an object (like opening a jar) may be more likely to be more consistently identified compared to actions that involve carrying an object (i. e. "put jar/cup"), where the start/end boundaries are more vague.

Figure 4.4 illustrates the action segments labelled by the five annotators. As noted above, actions entailing the transportation of an object ("put cup" and "put jar") can be validly labelled with a longer segment including the whole act of moving the object from a place to another, as annotator *d* did. Action granularity is interpreted in different ways as well. For the action "scoop jar" the video shows a person scooping sugar from the jar two times consecutively. Some people (annotators *a*, *c* and *d*) recognised the action as being split in two instances and labelled two segments accordingly. Annotators *b* and *e* instead viewed the two scooping movements as being part of one single action. Action granularity is another interesting issue which is however left for future work. Annotators were asked to label all the actions in the video, but were not provided with a list of actions to be labelled. As a result, some people missed one action (annotator *a* did not label "turn tap", annotator *d* did not label "pick up jar").

The annotators involved in this experiment were both native and non native English speakers. Language is another factor that may affect the perception of the temporal scope of an action. In fact, semantic ambiguity, stemming either from language proficiency or translation issues (i.e. no direct translation from English happens to exist for a certain class label), can affect the interpretation of the action itself. This is especially the case for more generic verbs like "take", which can be used both to describe the act of transporting something from a place to another, or to express just the act of picking something up. Semantics can be leveraged for the task of action recognition (as briefly seen in Section 3.3), however a deeper investigation on this matter is out of the scope of this Thesis.

In order to assess how consistency changes as more annotations are collected, 100 annotators were employed using Amazon Mechanical Turk (AMT) to label two more object interactions from BEOID, namely the actions of "wash cup" and "scan card". Figure 4.2b depict these two actions. AMT annotators received similar instructions to those provided to the local annotators. AMT workers marked the beginning and the ending of the object interaction in a loosely trimmed video containing the action to be labelled.

Box plots for the IOU calculated from the AMT labels are displayed at the right end of Figure 4.3. The average IOU and its standard deviation, reported in Table 4.1 and respectively equal to 0.57 and 0.20 seconds, are similar to the average IOU and standard deviation reported for the locally collected labels. The crowd-sourced annotations, however, exhibit a larger deviation for the start and the end times, respectively equal to 3.8 and 4 seconds. The difference between the two standard deviations in this case does not suggest that annotators disagreed more on neither of the two endpoints.

Ultimately, temporal bounds ambiguity does not seem to be correlated with the number of annotations, i.e. the IOU for the AMT and local annotations varies similarly. In order to gauge temporal bounds variability at a larger scale and in existing scenarios, the next Section inspects annotation inconsistency in three egocentric datasets. The ambiguity in annotating the start and end times of an action will then be revisited in Section 4.4.2, which introduces the Rubicon Boundaries labelling guidelines. As we will see, such annotation protocol helps to decrease ambiguity and assist people in marking action temporal bounds more consistently.

4.2 Inspecting Temporal Bounds Inconsistency in Egocentric Datasets

First person videos offer a unique point of view of the actions being performed by the camera wearer. For this reason, the egocentric field of view is often chosen to record daily life activities, usually in domestic environments. These videos are typically very dense, i.e. they capture a number of object interactions occurring one after the other in a quick succession. Action density is an important characteristic when considering temporal bounds. When a video contains many actions taking place in a tight sequence, imprecise or ambiguous boundaries can play an important role, given that an action may transition into another one within very few frames. This is especially the case for egocentric videos, where abrupt visual changes are common due to the movements of the user and the mounting of the camera. For these reasons, this work focuses on egocentric datasets containing task-oriented daily life activities.

The ground truth labels of the BEOID [20], GTEA Gaze+[28] and CMU-MMAC [22] datasets were inspected. The three datasets, reviewed in Section 3.1, contain untrimmed videos showing a number of object interactions occurring in a quick succession one after the other. Note that the authors of the datasets provide little or no information regarding the annotation process. CMU-MMAC's authors acknowledge that action labelling is ambiguous, however they do not explain how actions were temporally annotated. BEOID's authors provided brief instructions to their annotators, as reported in Section 4.1. GTEA Gaze+'s authors do not mention any details altogether.

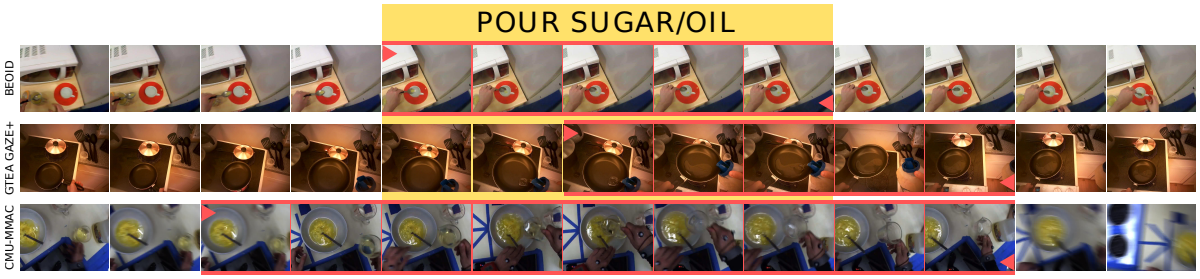

Figure 4.5: Annotations for the action "pour sugar/oil" from BEOID, GTEA Gaze+ and CMU-MMAC. Frames marked in red correspond to the labelled segments, as published with the datasets. The yellow rectangle encloses the motion strictly involved with the pouring motion.

Based on visual inspection of the published labels, in many cases the start and end of an action are identified respectively as the first and last frame when the hands are visible. Other annotations delimit an action more strictly, including only the most relevant physical object interaction within the bounds. Figure 4.5 compares three different temporal segmentations for the action of pouring oil or sugar across the three datasets. Frames marked in red are enclosed within the labelled start and end times provided by the datasets. The yellow rectangle highlights the motion strictly involved with the pouring action.

The annotated temporal bounds in this example vary remarkably. BEOID's bounds are the tightest, i.e. the segment's start corresponds to the moment when the subject tilts the teaspoon in order to pour sugar into the cup, while its end coincides with the instant the sugar sinks into the cup. The start of GTEA Gaze+'s segment is slightly belated, with the first frame in the annotated segment showing some oil already on the pan. Moreover, the segment includes two pouring motions, i.e. the individual pours some oil first, then pauses for a while before finally pouring some more oil. CMU-MMAC's boundaries include two extra object interactions besides the pouring action. In fact, the segment displays the user picking up the oil container and putting it down before and after pouring oil into the bowl. These observations extend to other actions in the three datasets.

Annotations are also inconsistent within the same dataset, as illustrated in Figure 4.6. For each dataset, two instances of the same action extracted from different videos are compared. For the action "open door" in BEOID, one segment includes the hand reaching the door, while the other starts with the hand already holding the door's handle. For the action "cut pepper" in GTEA Gaze+, in the first segment the user is already holding the knife and cuts a single slice of the vegetable. The second segment instead includes the action of picking up the knife, and shows the subject slicing the whole pepper through several cuts. The length difference between these two clips is considerable, with the segments being respectively 3 and 80 seconds long. This indicates that the granularity of an action can be ambiguous too. Finally, for the action "crack egg" in CMU-MMAC, only the top segment shows the user tapping the egg against the bowl, while the bottom segment starts just after the person taps the egg.

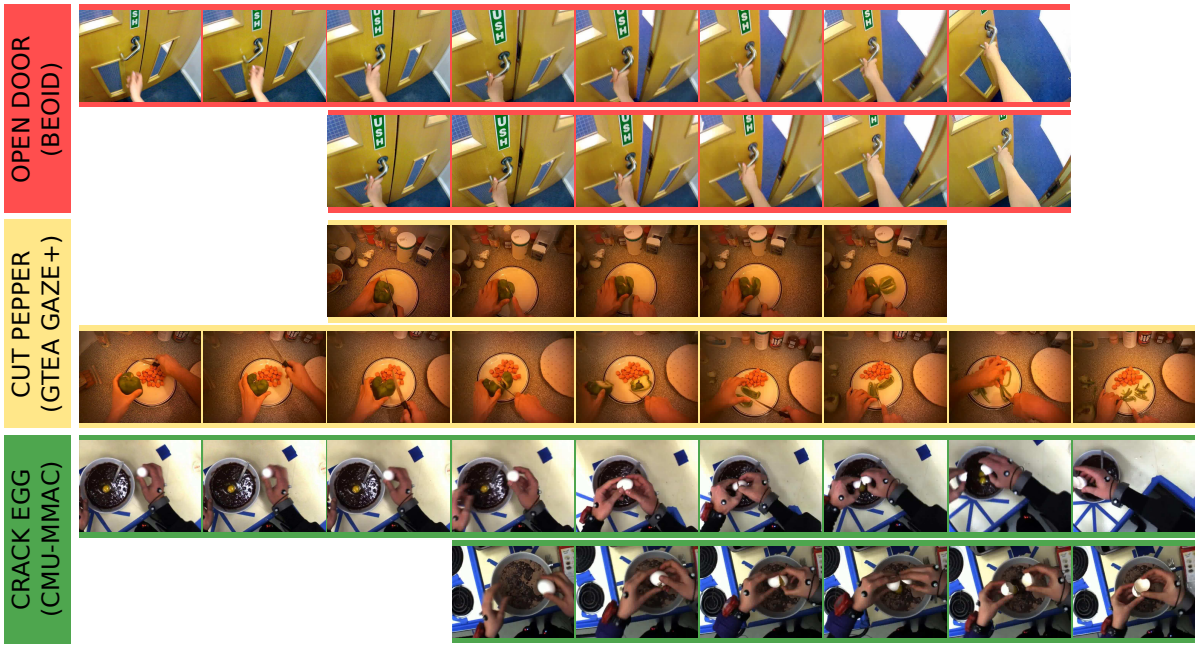

Figure 4.6: Inconsistency of temporal bounds within datasets. For each dataset two instances of the same action are shown, with visible differences in start and end times.

While Figure 4.6 shows only three examples, such inconsistencies have been discovered throughout the three datasets. GTEA Gaze+ generally exhibits more inconsistent boundaries, which could be due to its size, as it is the largest amongst the evaluated datasets.

When an object is partially occluded or when an action is only partially observed, humans are nevertheless typically able to recognise the object or the action. Does the same hold true for algorithms? What happens when a classifier sees a video segment that looks slightly different from what it is used to see? The next Section will provide some answers to these questions.

4.3 Assessing Robustness

To assess the effect of temporal bounds variation on action recognition, the start and end times of the annotated segments of BEOID, GTEA Gaze+ and CMU-MMAC were systematically varied. New test segments were generated from the annotated start/end times, deliberately allowing the new segments to discard relevant frames or include irrelevant ones. For training only the original ground truth bounds were considered, while during testing both the original and the altered boundaries were evaluated. Results were obtained using 5-fold cross validation.

The experiments were carried out following both a traditional approach using hand-crafted features, as well as employing a more recent deep learning model. The corresponding state-of-the-art classifiers available at the time of the study were thus chosen[1]. These were Improved Dense

[1]This work was finalised in 2016, when hand-crafted models were still competitive with CNNs.

Trajectories [107] encoded with Fisher Vector [84] (IDT FV) and Convolutional Two-Stream Network Fusion for Video Action Recognition (2SCNN) [29], reviewed respectively in Sections 3.2.1 and 3.2.2. IDT FV features were classified with a linear SVM. Experiments on 2SCNN were carried out using the provided code and the proposed VGG-16 [96] architecture pre-trained on Image Net and tuned on UCF-101 [98]. The spatial, temporal and fusion networks were fine-tuned on each fold's training set.

Theoretically, the two action recognition approaches are likely to respond differently to variations in start and end times. For testing, 2SCNN averages the classification responses of the fusion network obtained on n frames uniformly extracted from a test video v of length $|v|$ (in this work, $n = \min(10, |v|)$. IDT densely samples feature points in the first frame of the video, whereas in the following frames only new feature points are sampled to replace the missing ones. Importantly, IDT features are extracted from each frame. This entails that IDT FV should be more sensitive to start (specifically) and end time variations, especially for shorter videos. Moreover, 2SCNN's features are learnt and fine-tuned on each dataset, while IDT features are hand-crafted. This fundamental difference makes both approaches interesting to be studied for robustness.

Comparison to previous work Although some previous works concerned temporal boundaries ambiguity to different extents, this kind of study was not offered before. Satkin and Hebert [86], noting that defining the temporal span of an action is subjective, proposed a method to automatically crop *one* action from a *trimmed* training video, in order to optimise the performance of the classifier without temporal annotations. This is different from varying both the actions' start and end times in untrimmed videos containing multiple object interactions.

More recently, and after the study presented in this Chapter was published, Hussein et al. [43] also altered the temporal bounds of test action segments. However, the end goal of this work and [43] are different. This study concerns temporal bounds subjectivity for fine-grained actions and the corresponding impact that realistic variations may have on recognition models. Hussein et al. [43] proposed an architecture to model minute-long complex activities, and vary temporal bounds to assess robustness to variations in actions speed. More precisely, in [43] frames were duplicated or dropped to enlarge or shrink the segments length. This simulates actions occurring at different speeds, and is essentially different from the *trespassing* approach taken in this work, where both start and end bounds are shifted independently to intentionally include unrelated frames and remove relevant ones.

4.3.1 Generating Segments

Let $v_{gt} = [s, e]$ be a ground truth action segment obtained by clipping an untrimmed video v from s to e, which denote the annotated start and end times. Both s and e are varied in order to generate new action segments with different temporal bounds. More precisely, a neighbourhood of frames around the ground truth start and end times is considered, and candidate new start and end times are generated by sampling frames uniformly in such neighbourhoods.

Dataset	Number of ground truth segments	Number of generated segments	Number of classes
BEOID	742	16,691	34
GTEA Gaze+	1,141	22,221	42
CMU-MMAC	450	26,160	31

Table 4.2: Number of ground truth/generated segments and classes for BEOID, GTEA Gaze+ and CMU-MMAC.

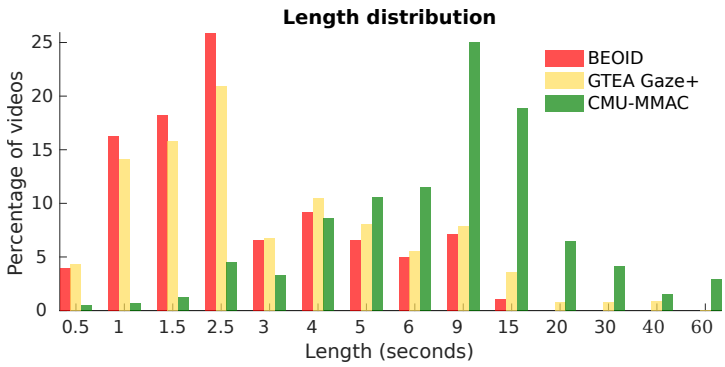

Figure 4.7: Length distribution of the actions contained in BEOID, GTEA Gaze+ and CMU-MMAC.

Formally, let $s_L = s - \Delta$ and let $s_R = s + \Delta$. The set containing candidate start times is defined as:

$$\mathbf{S} = \left\{ s_L,\ s_L + \delta,\ s_L + 2\delta,\ \ldots,\ s_L + (n-1)\delta,\ s_R \right\} \tag{4.1}$$

Where $n = (2\Delta)/\delta$. Analogously, let $e_L = e - \Delta$ and let $e_R = e + \Delta$. The set containing candidate end times is defined as:

$$\mathbf{E} = \left\{ e_L,\ e_L + \delta,\ e_L + 2\delta,\ \ldots,\ e_L + (n-1)\delta,\ e_R \right\} \tag{4.2}$$

Finally, all the possible combinations $s_{gen} \in \mathbf{S}$ and $e_{gen} \in \mathbf{E}$ are taken to generate new test segment from v_{gt}. Only the combinations such that the IOU between $[s,\ e]$ and $[s_{gen},\ e_{gen}] \geq 0.5$ are kept:

$$\mathbf{G} = \left\{ [s_{gen},\ e_{gen}] \quad \forall s_{gen} \in \mathbf{S}, \forall e_{gen} \in \mathbf{E} : \mathrm{IOU}\big([s,\ e], [s_{gen},\ e_{gen}]\big) \geq 0.5 \right\} \tag{4.3}$$

In the experiments Δ was set to 2 seconds while δ was set to 0.5 seconds.

4.3.2 Comparative Evaluation

Table 4.2 reports the number of ground truth and generated segments for BEOID, GTEA Gaze+ and CMU-MMAC, as well as the number of classes for the three datasets. BEOID is the second largest dataset in terms of ground truth segments, however it presents the smallest number of generated segments. Conversely, CMU-MMAC is the smallest dataset, but has the largest number of generated segments. This is due to the average action length of the datasets: when candidate start end times are generated from long segments, more start/end combinations are likely to pass the IOU threshold.

Dataset	IDT FV$_{gt}$	IDT FV$_{gen}$	2SCNN$_{gt}$	2SCNN$_{gen}$
BEOID	85.3	75.4	93.5	83.8
CMU-MMAC	54.9	52.8	76.0	71.7
GTEA Gaze+	45.4	43.3	61.2	59.6

Table 4.3: Top-1 accuracy obtained with IDT FV and 2SCNN, with ground truth and generated segments, on BEOID, CMU-MMAC and GTEA Gaze+.

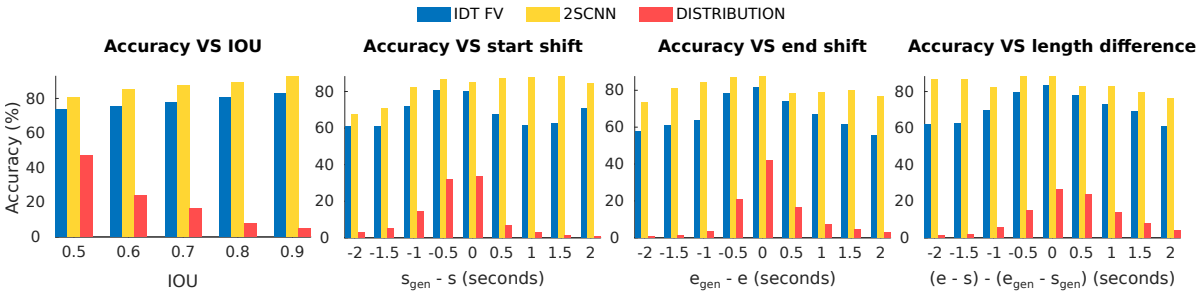

Figure 4.8: BEOID: classification accuracy vs IOU, start/end shifts and length difference between ground truth and generated segments.

Figure 4.7 illustrates the segments' length distribution for the three datasets, showing considerable differences: BEOID and GTEA Gaze+ contain mostly short segments (1-2.5 seconds), although the latter includes also videos up to 40 seconds long. CMU-MMAC has longer segments, with the majority ranging from 5 to 15 seconds. As discussed at the beginning of this Chapter, and as we will see shortly, action length is a factor to be taken into account when evaluating temporal bounds variation. For each dataset, accuracy vs overlap (IOU), start/end shifts and length difference between the ground truth and the generated segments are next reported.

BEOID [20] is the evaluated dataset with the tightest and most consistent temporal bounds. On this dataset both IDT FV and 2SCNN achieve the highest accuracy on ground truth segments amongst the three datasets - respectively 85.3% and 93.5% - as shown in Table 4.3. The high performance of both classifiers suggests that BEOID is a simple dataset. At the same time, BEOID exhibits the greatest decrease in accuracy: when classifying the generated segments, top-1 accuracy drops by 9.9% and 9.7% respectively for IDT FV and 2SCNN.

Figure 4.8 shows detailed results, where accuracy is reported vs IOU, start/end shifts and length difference between ground truth and generated segments. For all plots the illustrated bars are normalised using the corresponding distribution. For example, if 5 out of 10 segments that had start shift equal to 2 seconds were correctly classified, the reported accuracy would be 0.5.

A negative start shift implies that a generated segment begins before the corresponding ground truth start (new start is outside the bounds), while a positive start shift means that the segment begins after the ground truth start (new start is inside the bounds). Likewise, a negative end shift involves that a generated segment finishes before the corresponding ground truth end (new end is inside the

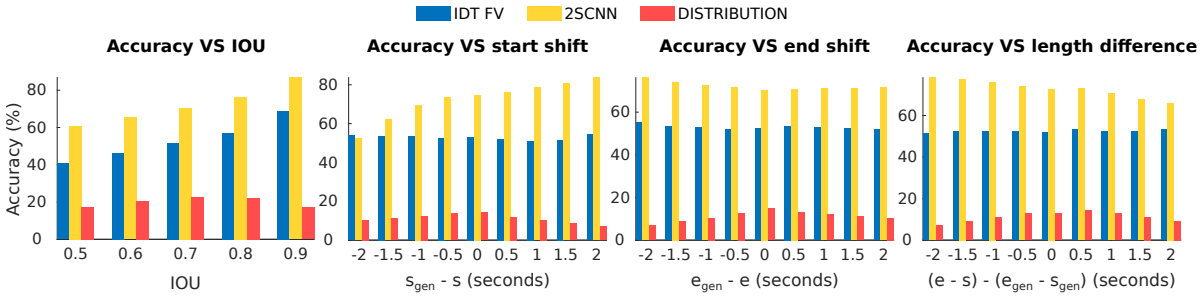

Figure 4.9: CMU-MMAC: classification accuracy vs IOU, start/end shifts and length difference between ground truth and generated segments.

bounds), while a positive end shift entails that the segment ends after the ground truth end (new end is outside the bounds). These terms are used consistently throughout this Section.

Figure 4.8 shows a direct correlation between IOU and performance drop, with the accuracy deteriorating consistently for both IDT FV and 2SCNN as the IOU decreases. This is the simplest signal showing lack of robustness to temporal bounds alterations for both approaches. Classification accuracy for IDT FV exhibits lower accuracy with both negative and positive start/end shifts. This is expected as BEOID segments are tight and short, i.e. by expanding a short segment a relatively large amount of irrelevant frames are potentially included. This applies particularly to IDT FV whose feature extraction strategy relies on the segment's first frame.

Similarly, IDT FV also exhibits lower accuracy with negative and positive length differences. This is due to the fact that IDT extracts trajectories uniformly from all frames in a video, which means that the number of features is directly proportional to the length of the video. As a consequence, IDT FV is more sensitive to length variations compared to 2SCNN, which is more robust given that it samples a fixed number of frames regardless of the length. Interestingly, 2SCNN is slightly more resilient to negative length differences in comparison to positive ones. This indicates that 2SCNN is more able to recognise the actions when frames are dropped compared to when frames are added. This hints that while 2SCNN can cope with missing information, it tends to struggle when noise is added. This hypothesis is reinforced by the fact that 2SCNN is also more robust to positive start shifts and, though to a lesser extent, to negative end shifts, when the generated start/end times are within the ground truth boundaries.

CMU-MMAC [22] is the dataset with the longest ground truth segments. As reported in Table 4.3, for this dataset IDT FV's accuracy drops by 2.1% for the generated segments, whereas 2SCNN's drops by 4.3%. As depicted in Figure 4.9, classification performance declines steadily with lower IOU, for both IDT FV and 2SCNN. Particularly, accuracy drops by 20% for both classifiers between IOU 0.5 and 0.9. This is the largest accuracy drop with respect to IOU observed for the three datasets. However, due to the long average length of segments in CMU-MMAC (see Figure 4.7), variations in start, end and length do not show particular patterns for IDT FV. This is probably due to the Fisher Vector encoding of the features. When videos are very long, i.e. when a large number of trajectories are encoded in a

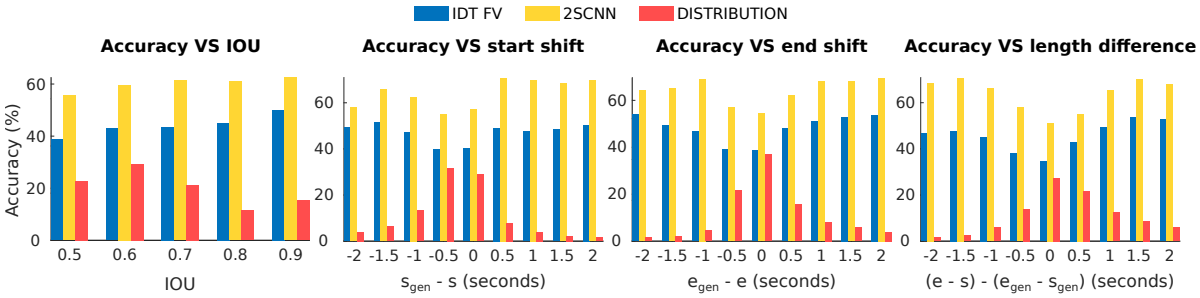

Figure 4.10: GTEA Gaze+: classification accuracy vs IOU, start/end shifts and length difference between ground truth and generated segments.

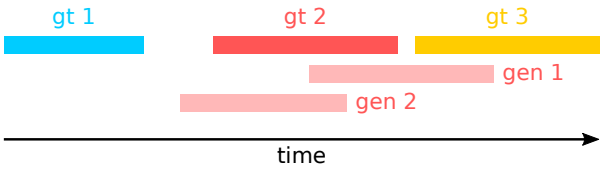

Figure 4.11: Illustration of overlapping and non overlapping neighbours. *gt 3* is an overlapping neighbour of *gen 1*, as well as a non overlapping neighbour of *gt 2* and *gen 2*. *gt 1* is a non overlapping neighbour of *gt 2*, *gen 1* and *gen 2*. *gt 2* is a non overlapping neighbour of *gt 1* and *gt 3*.

feature vector of fixed length, adding or removing a small percentage of frames/trajectories is unlikely to affect the encoding in a specific manner.

2SCNN displays an interesting behaviour. Accuracy consistently improves with positive start shifts, negative end shifts and negative length difference. This suggests that CMU-MMAC's ground truth bounds are somewhat loose and that tighter segments are likely to contain more discriminative frames for the labelled actions. This is in agreement with the annotations inspection presented in Section 4.2, where Figure 4.5 showed that the segment for the action "pour oil" included the act of picking the jar up and putting it down before and after the pouring motion.

GTEA Gaze+ [28] is the dataset with the most inconsistent bounds. Table 4.3 shows that accuracy for IDT FV drops by 2.1%, while accuracy for 2SCNN drops marginally (1.6%). This should not be mistaken for robustness, and that is evident when studying the results in Figure 4.10. For all variations the generated segments achieve higher accuracy, for both IDT FV and 2SCNN. When labels are inconsistent, shifting temporal bounds does not systematically alter the visual representation of the tested segments. Conversely, many generated segments include (or exclude) frames that increase the similarity between the testing and the training segments, which in turn increases their chance to be correctly classified.

Confusion The confusion with overlapping and non overlapping neighbours is now inspected. Confusion with a neighbour entails that a segment was incorrectly classified and was assigned the class of one of its neighbours. In this context, a neighbour of a given segment x is a ground truth

Dataset	**Segments with an overlapping neighbour**		**Average IOU with neighbour**		**Confused with overlapping neighbour**				**Confused with non overlapping neighbour**			
					IDT FV		**2SCNN**		**IDT FV**		**2SCNN**	
	gt	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*
BEOID	8.1	49.9	0.15	0.17	10.1	25.7	12.5	49.1	11.9	8.3	16.7	7.2
CMU-MMAC	0	39.4	/	0.08	/	8.4	/	28.7	12.3	6.4	23.1	10.6
GTEA Gaze+	3.9	22.3	0.12	0.13	0.3	5.5	0.9	12.6	9.1	9.4	8.7	14.3

Table 4.4: Confusion with overlapping and non overlapping neighbours. Neighbours are ground truth segments that are temporally close to or overlapping with another segment. All columns except "Average IOU" indicate percentages.

segment that is temporally close to or overlapping with x. If x is a generated segment, the ground truth parent segment (i.e. the segment from which x was generated) is excluded from x's neighbours. If multiple neighbours overlap with x, only the neighbour with the largest overlap was evaluated. For non overlapping neighbours, the closest segments both before and after x were considered, up to a maximum distance of 3 seconds. Figure 4.11 illustrates an example of overlapping and non overlapping neighbours for a few segments.

Table 4.4 shows the percentage of generated segments that have been confused with a neighbour. The reported confusion percentages are relative to the number of incorrectly classified segments. The Table also reports the percentage of ground truth and generated segments that overlap with a neighbour, along with the average IOU of the overlap. These were calculated considering all tested segments, i.e. both correctly and incorrectly classified segments.

As expected, a large portion of generated segments overlaps with a neighbour, however the average IOU between the generated segments and their neighbours is low in all datasets. Importantly, the average IOU with a neighbour is comparable between ground truth and generated segments, which reflects the realism of the generated annotations.

Confusion with an overlapping neighbour is the most straightforward reason to explain the accuracy drop observed when classifying the generated segments. When videos are dense of actions, altering temporal bounds of close segments can result in including frames belonging to a different action which can confuse the classifier. By comparing ground truth and generated segments, it seems indeed that confusion with overlapping neighbours is the main cause of performance drop when evaluating the generated segments. In fact, while a relatively small percentage of ground truth segments was confused with an overlapping neighbour, a larger portion of generated segments were incorrectly classified as their overlapping neighbour. However, two considerations should be made. Firstly, the generated segments substantially overlap with their parent ground truth segments (minimum IOU = 0.5). Secondly, the average IOU between generated segments and overlapping neighbours is low. This means that only a small portion of frames belonging to another action were included in the generated segments, and that the overall accuracy drop observed in this study is not simply due to the generated segments enclosing frames belonging to distinct actions.

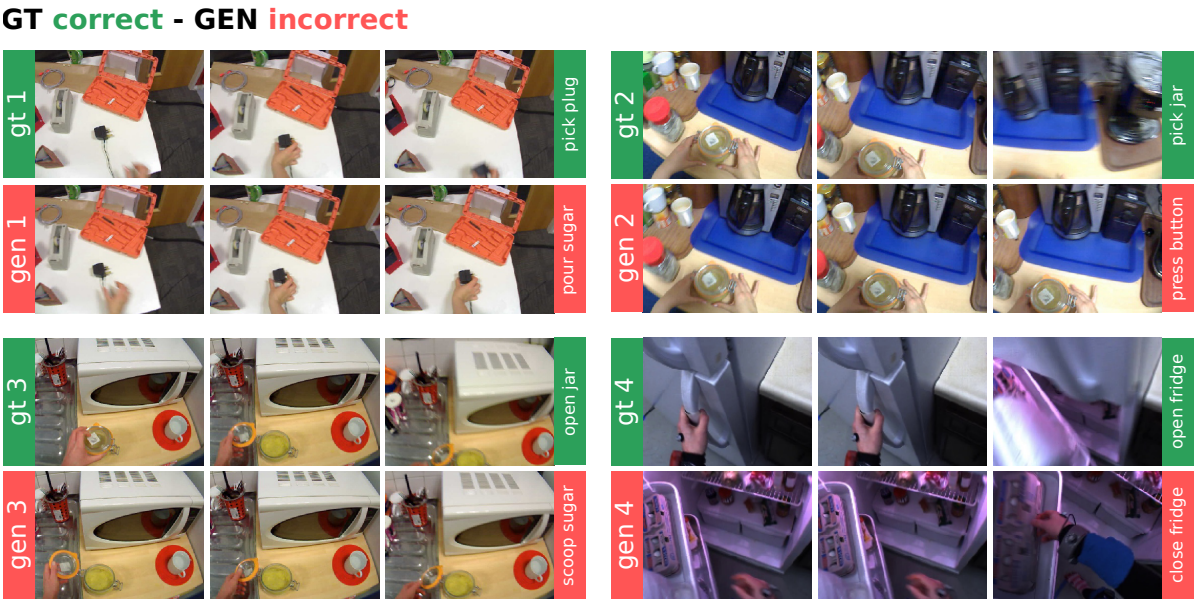

Figure 4.12: Qualitative results for correctly predicted ground truth (green) and incorrectly classified generated segments (red). Class labels correspond to the predicted class. Results shown with 2SCNN on BEOID (segments 1, 2, 3) and CMU-MMAC (segment 4).

Non overlapping neighbours are additionally evaluated to take bounds ambiguity into account. Overall, confusion with non overlapping neighbours for the generated segments is low. This further indicates that the accuracy drop observed with the generated segments is not merely due to an increased visual similarity between neighbouring actions, and that perhaps background frames are a more important culprit than expected.

Qualitative results Figure 4.12 shows classifications results obtained with 2SCNN for a few segments. The Figure illustrates cases where the ground truth segment was correctly classified (green) while the generated segment was not (red). Class labels indicate the predicted class. In some cases (gen 1 and 2) the incorrect prediction is wholly unrelated to the ground truth class, despite the significant overlap between ground truth and generated segments. In other cases (gen 3 and 4) the predicted class for the generated segment has a strong correlation with its visual content. For example, segment gen 3 was incorrectly classified as “scoop spoon” instead of “open jar”. In this segment the act of opening the jar, although still visible, was partially truncated. The shortened motion and the visual cues correlated with the sugar in the jar likely induced the classifier to predict the “scoop sugar” action. A similar reasoning can be done for gen 4, which was classified as “close fridge” as opposed to “open fridge”.

Figure 4.13 shows cases where the ground truth segment was incorrectly classified while the generated segment was successfully recognised. Ground truth segments 5 and 6 are loosely trimmed, including background frames at the end which probably confused the classifier. The generated segments happened to be more precise, enclosing more relevant frames which facilitated recognition.

Figure 4.13: Qualitative results for incorrectly classified ground truth (red) and correctly predicted generated segments (green). Class labels correspond to the predicted class. Results shown with 2SCNN on and GTEA Gaze+.

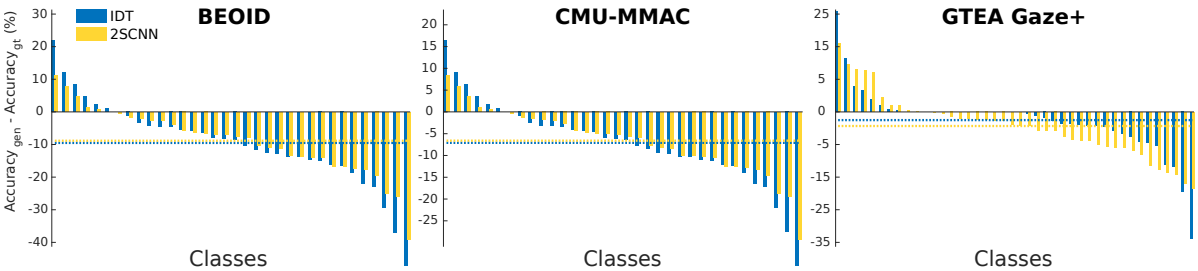

Figure 4.14: Per-class accuracy drop for BEOID, CMU-MMAC and GTEA Gaze+, for IDT FV and 2SCNN.

Dataset	Generated better than ground truth	
	IDT FV	2SCNN
BEOID	2.6	1.2
CMU-MMAC	4.1	3.1
GTEA Gaze+	17.8	16.9

Table 4.5: Percentage of generated segments that were "better" than the ground truth counterparts. This corresponds to cases where a generated segment x was correctly recognised while the ground truth segment that generated x was incorrectly classified.

Analysing classes Figure 4.14 illustrates the accuracy drop for each class separately, for the three datasets. Positive values entail that the accuracy for the given class was higher when testing the generated segments, and vice versa. Horizontal lines indicate the average accuracy difference. In total, 63% of classes in all three datasets exhibit a drop in accuracy when using IDT FV compared to 80% when using 2SCNN. This shows that robustness is not an issue related to only a few classes, and importantly proves that the accuracy drop observed in the experiments is not biased by class imbalance. In GTEA Gaze+ more classes (10) exhibit a higher accuracy with the generated segments compared to 6 for both BEOID and CMU-MMAC. This is in line with the previous observation regarding imprecise annotations, i.e. altered temporal bounds might cover the action better when the original boundaries are inaccurate. This is also showed in Table 4.5, which reports the percentage

Dataset	$2SCNN_{gt}$	$2SCNN_{gen}$	$2SCNN_{gt}^{aug}$	$2SCNN_{gen}^{aug}$
BEOID	**93.5**	83.8	92.3	86.6
GTEA Gaze+	**61.2**	59.6	57.9	58.1

Table 4.6: Top-1 accuracy obtained with 2SCNN on BEOID and GTEA Gaze+ with and without data augmentation, for both ground truth and generated segments.

of generated segments that were correctly classified while the ground truth segments that generated them were not. Table 4.4 shows a neat separation for this measure between GTEA Gaze+ (17%) and CMU-MMAC (4%) and BEOID (2%), which further proves GTEA Gaze+ to be the most inconsistently labelled dataset.

Data augmentation For completeness, data augmentation for 2SCNN on BEOID and GTEA Gaze+ is also evaluated. CMU-MMAC was not considered in this case due to its small size (with 450 action instances and 31 classes, it is the smallest datasets amongst the evaluated ones). The training sets were doubled in size by including segments generated for a random subset of training samples, whereas the test sets were the same as in the experiments presented above. Training segments were generated as explained in Section 4.3.1.

Results are reported in Table 4.6, where a boost in robustness can be observed, especially for BEOID which is the smallest between the two datasets. Nevertheless, the accuracy drop for BEOID remains high when using the augmented training set (5.7%). Additionally, when augmenting the training data, accuracy on ground truth segments dropped in both datasets, respectively by 1% and 4%.

* * *

In conclusion, both IDT FV and 2SCNN are sensitive to changes in temporal bounds for both consistent and inconsistent annotations. More sophisticated approaches using temporal augmentation could be attempted to improve robustness, however a broader look at how the methods could be inherently more robust is still needed, particularly for CNN architectures. Nevertheless, the temporal span of actions still remains an arbitrary matter.

The next Section presents the Rubicon Boundaries, a labelling protocol assisting annotators in identifying the temporal extent of actions in videos. The annotation guidelines facilitate more consistent temporal labelling, which is beneficial to obtain more precise annotations and, as a consequence, higher classification performance.

4.4 Labelling Proposal: The Rubicon Boundaries

The problem of defining consistent temporal bounds of an action is most akin to the problem of defining consistent bounding boxes of an object. Attempts to define guidelines for annotating objects bounding boxes started nearly a decade ago. Amongst others, the VOC Challenge 2007 [26] proposed what has become the standard for defining the bounding box of an object in images. These consistent labels have been used to train state-of-the-art object detection and classification methods. With this same spirit, this Section proposes an approach to consistently identify the temporal scope of an object interaction.

4.4.1 The Rubicon Model of Action Phases

The problem of defining the phases of a human action constitutes a broad area of research in the fields of cognitive neuroscience and social psychology. Given that temporal bounds constitute the base labels for recognition algorithms, computer vision researchers could take some inspiration from these fields. For example, Gollwitzer [38] defines the course of an action as "a temporal, horizontal path starting with a person's desire and ending with the evaluation of the achieved action outcome".

Specifically, in [38] an action is decomposed into four phases, namely the pre-decisional, post-decisional, actional and post-actional phase. The pre-decisional and post-decisional phases correspond to intention, i.e. the moments when an individual wishes to achieve some results by means of some actions. Depending on the desired outcomes, the subject will need to choose which results are to be prioritised, and accordingly decide which actions they want to carry out to fulfil their goal. Once this decision has been made, the subject enters the pre-actional phase, where intention is transformed to planning. The subject is now preparing the necessary steps to be performed to achieve their goal, focusing on the questions of where and when to start, as well as how long and how to undertake the action. After planning, the person begins to operate towards their goal, embarking thus on the actional phase, which ends when the desired goal has been achieved (assuming that the outcome is a discrete act, e.g. turn off a water tap). Finally, the individual reaches the post-actional phase, where they evaluate the results obtained with their action.

According to the model presented in [38], named "The Rubicon Model of Action Phases", the four actional steps are separated by three clear boundaries delimiting intention, initiation and conclusion of the actions associated with the goal. The model is named after the historical fact of Caesar crossing the Rubicon river, which became a metaphor for deliberately proceeding past a point of no return. In this case, the point of no return coincides with the transition point that signals the beginning of an action. The Rubicon Boundaries annotation guidelines, presented next, take inspiration from this model, specifically from the described transitions points.

4.4.2 The Rubicon Boundaries Guidelines

The Rubicon Boundaries guidelines focus on the pre-actional and actional phases presented in [38], and accordingly define two namesake stages for an action. The guidelines are specific to fine-grained object interactions, where an individual interacts with an object for a short amount of time in order to achieve some goal. The protocol is thus not designed for long activities (e.g. preparing a meal) or interactions that do not involve objects (e.g. walking or talking). The pre-actional and actional phases of an object interaction are defined as follows:

Pre-actional phase This sub-segment contains the preliminary motion that directly precedes the main action. When multiple preparatory acts can be identified, the pre-actional phase should contain only the last one.

Actional phase This is the main sub-segment containing the motion strictly related to the goal the user wants to achieve. The goal is semantically identified by the class label describing the action. The actional phase starts immediately after the pre-actional phase, i.e. the two sub-segments are temporally contiguous.

The beginning of the pre-actional phase and the end of the actional phase should be labelled to mark the span of an action in a video. In those cases where identifying the transition between the preparatory motion and the execution of the action is important (e.g. action anticipation), the beginning of the actional phase should be labelled as well. For the traditional case where the inner transition point is not of interest, it might be questioned whether the actional phase alone would be enough and whether the pre-actional phase would be necessary at all. Experiments will show that even in this case the pre-actional phase is helpful. In fact, including preliminary frames provides context that can be beneficial to a classifier. Furthermore, the pre-actional phase serves as a sort of landmark that facilitates annotating the beginning of the main action more precisely.

Figure 4.15 depicts three object interactions labelled according to the Rubicon Boundaries. The top sequence illustrates the action of cutting a pepper. The frames show the subject fetching the knife before cutting the pepper and taking it off the plate. Based on the aforementioned definitions, two preliminary acts preceding the cutting action can be identified: grabbing the knife and moving the knife over the centre of the vegetable. The pre-actional phase should contain only the last preparatory motion, and in this case is thus limited to the motion of moving the knife towards the pepper in order to slice it. This is directly followed by the actional phase where the user cuts the pepper. The actional phase ends when the goal is completed. In this case the goal is cutting the pepper, and accordingly the segment ends as soon as the vegetable is sliced.

The middle sequence shows the action of opening a fridge, displaying a person approaching the appliance, reaching towards the handle before pulling the fridge's door open. In this case, the

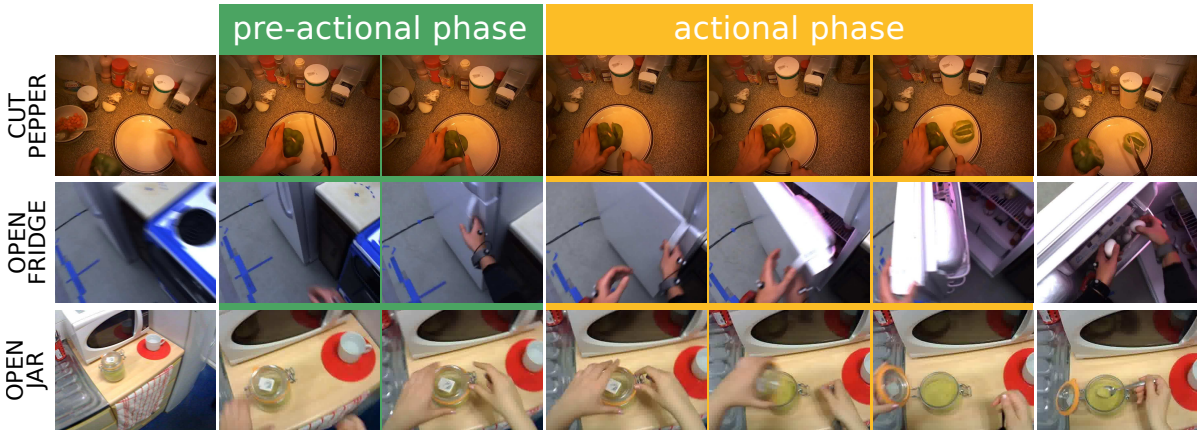

Figure 4.15: Three object interactions labelled according to the Rubicon Boundaries. Actions extracted from the GTEA Gaze+ (top), CMU-MMAC (middle) and BEOID (bottom) datasets.

pre-actional phase would enclose the reaching motion, while the actional phase would coincide with the pulling movement. Finally, the bottom sequence depicts a person opening a jar. The pre-actional phase in this case aligns with the motion of reaching the jar's lid, whereas the actional phase consists of the act of lifting the lid in order to open the jar. The actional phase, like in the case of "open fridge", ends at the moment when the object is opened (goal achieved).

The following Sections provide a thorough evaluation of the Rubicon Boundaries. Different aspects are inspected, namely consistency, intuitiveness as well as accuracy and robustness. In the remainder of this Chapter, "RB" refers to the Rubicon Boundaries.

4.4.3 Evaluating Consistency

The multi-annotator labels presented in Section 4.1 are here compared to a new set of annotations where multiple people were instructed using the Rubicon Boundaries guidelines. Ten people were asked to label three frames for a sequence of object interactions contained in a video: the beginning of the pre-actional phase, and the beginning and ending of the actional phase. The sequence of actions and the video were the same as those seen in Section 4.1.

The annotations analysed in Section 4.1 are from here on referred to as conventional annotations. As described in Section 3.1.3, these were labelled using open vocabulary. In order to isolate ambiguity stemming from different class labels, the RB annotators were given a list of class labels for the actions to be segmented. RB annotators were provided with the Rubicon Boundaries guidelines, i.e. a text document with the definitions of the pre-actional and actional phases as reported in Section 4.4.2. Like for the conventional annotations, annotators were showed some annotations examples before proceeding to label the video. Four out of the ten people who participated in the RB labelling were also involved in the first multi-annotator experiment.

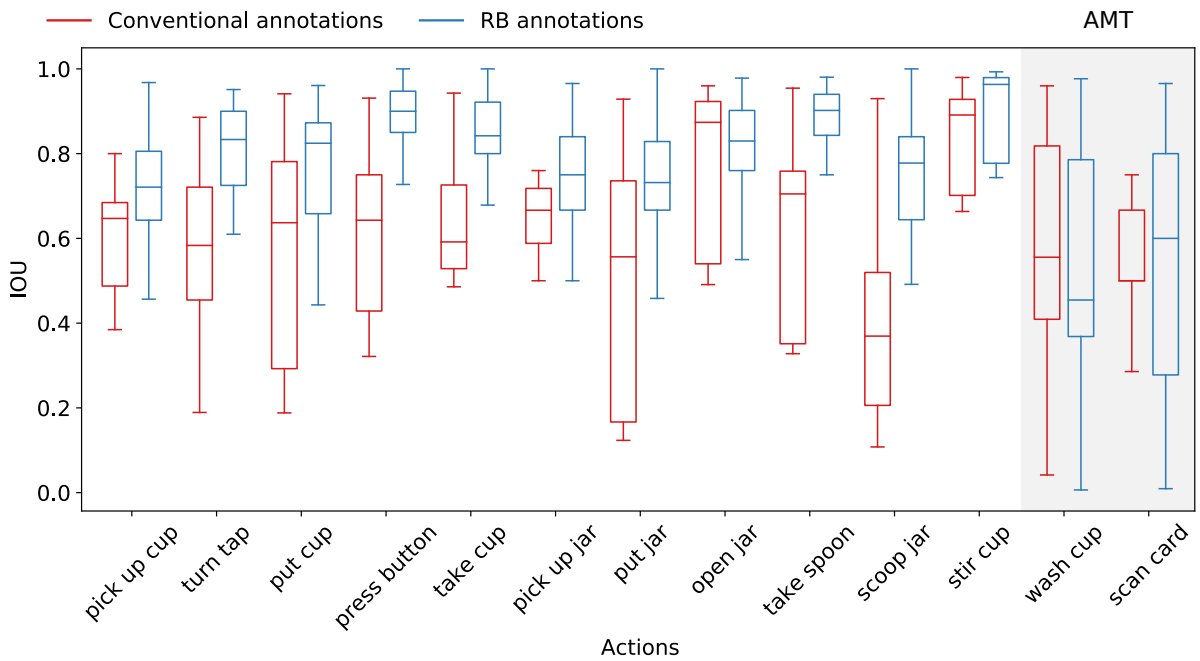

Figure 4.16: IOU comparison between conventional (red) and RB (blue) annotations for several object interactions. AMT stands for Amazon Mechanical Turk.

Annotations	Guidelines	Average IOU	Average IOU SD	Average start SD	Average end SD
Local (5, 10)	Conventional	0.62	0.19	0.62	0.25
	RB	0.81	0.11	0.14	0.16
AMT (100)	Conventional	0.57	0.20	3.8	4.0
	RB	0.54	0.24	6.1	5.0

Table 4.7: Statistical comparison between local and AMT labels, for conventional and RB annotations. Numbers between parenthesis indicate the number of annotators. SD indicates Standard Deviation, which is expressed in seconds for the start and end times.

Figure 4.16 compares the IOU box plots for the conventional and RB annotations (respectively red and blue boxes). For the RB labels the start of the pre-actional phase and the end of the actional phase were used to form a single segment. A distinctive separation between the two annotations sets is visible, with the RB labels consistently exhibiting an higher median IOU for all actions, with the exception of "open jar", where the RB annotations display a slightly lower median although showing also less IOU variation.

Table 4.7 compares the average IOU and its average standard deviation for the two annotations sets (local annotations, top two rows). The higher average IOU (0.81 vs 0.62) and lower average standard deviation (0.11 vs 0.19) further indicate that annotators were more consistent one with the other when following the Rubicon Boundaries. The RB guidelines specifically helped disambiguating the beginning of an object interaction. This is reflected by a lower average standard deviation for the start time, which was 0.14 seconds with the RB labels compared to 0.62 seconds for the conventional

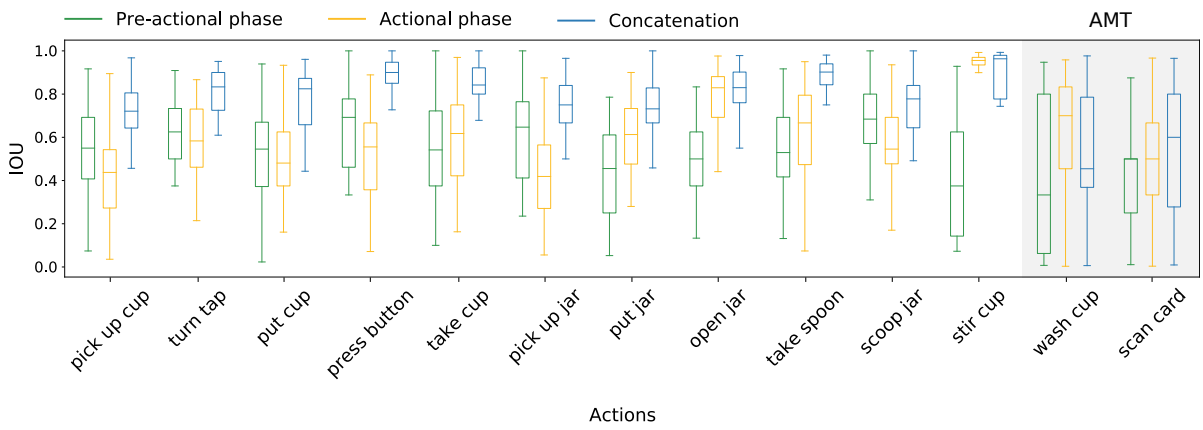

Figure 4.17: IOU comparison between the pre-actional (green), the actional (yellow) and the full (blue) segments, for several object interactions. AMT stands for Amazon Mechanical Turk.

labels. This is thanks to the decomposition of an action into its pre-actional and actional phases. By identifying the preliminary motion necessary to perform the action, it is possible to recognise the beginning of the action in a more unanimous way. The average standard deviation for the start and the end times of the RB labels are virtually identical, with the latter being equal to 0.16 seconds.

RB labels were also collected using AMT for the actions “wash cup” and “scan card”, which were previously annotated by AMT annotators in Section 4.1. Like in the local experiment, 100 AMT annotators were first presented with the Rubicon Boundaries guidelines, and then asked to mark the start of the pre-actional phase, as well as the start and the end of the actional phase. Box plots for the IOU calculated for the AMT RB labels are illustrated at the right-hand side of Figure 4.16. The graph shows that for the crowd-sourced annotations the RB guidelines did not prove as effective as for the local annotations. For “wash cup” a lower median IOU and a comparable variation was observed when evaluating the RB annotations, while for “scan card”, despite displaying a marginally higher median IOU, RB labels also show a larger agreement variability with respect to the conventional annotations. The average IOU and its average standard deviation, as well as the average start and end standard deviation are included in Table 4.7. All the four metrics are slightly worse and closely comparable to those reported for the conventional AMT annotations, which confirms that the crowd-sourced annotators did not benefit from the Rubicon Boundaries guidelines. The deployment of the Rubicon Boundaries to crowd-sourcing labelling will be discussed more in detail in the next Section.

Finally, Figure 4.17 compares the agreement on the two individual phases to the agreement on their concatenation. For 10 out the 13 actions the concatenation of the two phases shows a higher consistency compared to the two phases alone. By and large the pre-actional and the actional phases present a comparable agreement, with the exceptions of “open jar”, “stir cup” and “wash cup” (AMT labels), where the actional phase exhibits a significantly greater median IOU with respect to the pre-actional phase (“open jar”, “stir cup”) or the concatenation of the two sub-segments (“wash cup”). This suggests that for some kind of actions identifying the actional phase is less ambiguous than determining the preliminary motion required by the main action.

Figure 4.18: Frames extracted from the videos labelled by AMT annotators following the Rubicon Boundaries guidelines. Video sequences from BEOID.

4.4.4 Evaluating Intuitiveness

Although the Rubicon Boundaries annotations showed higher temporal consistency in the local experiment, any new labelling approach requires a shift in practice. This Section analyses how intuitive the RB protocol is when used in a larger scale annotation scenario, in order to assess the deployment of the RB guidelines to crowd-sourcing settings.

As previously mentioned, AMT annotators were first presented with the RB guidelines defining the pre-actional and actional phases for an object interaction. For each of the two actions, the annotators were then showed a video loosely trimmed around the action. After watching the video and before proceeding to mark the phases' bounds, the annotators were asked to answer two multiple-choice questions asking to identify the pre-actional and the actional phase, based on a textual description. The annotators could select only one answer per question.

Figure 4.18 illustrates the videos displayed to the annotators. Based on the RB definition, for the "wash cup" action the pre-actional phase is the act of moving the cup towards the water, while the actional phase corresponds with the act of rinsing the cup. For the "scan card" action the pre-actional phase coincides with moving the card towards the card reader, whereas holding the card by the reader constitutes the actional phase.

Figure 4.19 reports the answers provided by the AMT annotators. At a glance, it is evident that the majority of annotators were able to correctly identify the pre-actional phase of both actions, with 57.6% and 67.4% answering correctly for "wash cup" and "scan card" respectively. Conversely, it appears that the actional phase was harder to recognise. In fact, only 36.4% and 37.7% of annotators responded correctly for the two actions. In both cases, most annotators chose the answer including the largest number of movements (44% selected 'e' for "wash cup" and 40% chose 'f' for "scan card"). In total, 25.5% and 28% of the annotators responded correctly to both answers for the two actions.

Few factors can be ascribed to such results. Firstly, considering that AMT is commonly used by researchers to annotate objects and actions, annotators might have been highly accustomed to the conventional labelling method, and thus were confused when asked to follow the RB guidelines. Secondly and more generally, the RB definitions may have been difficult to understand, especially in a crowd-sourcing scenario like AMT where annotators typically do not spend sufficient time to

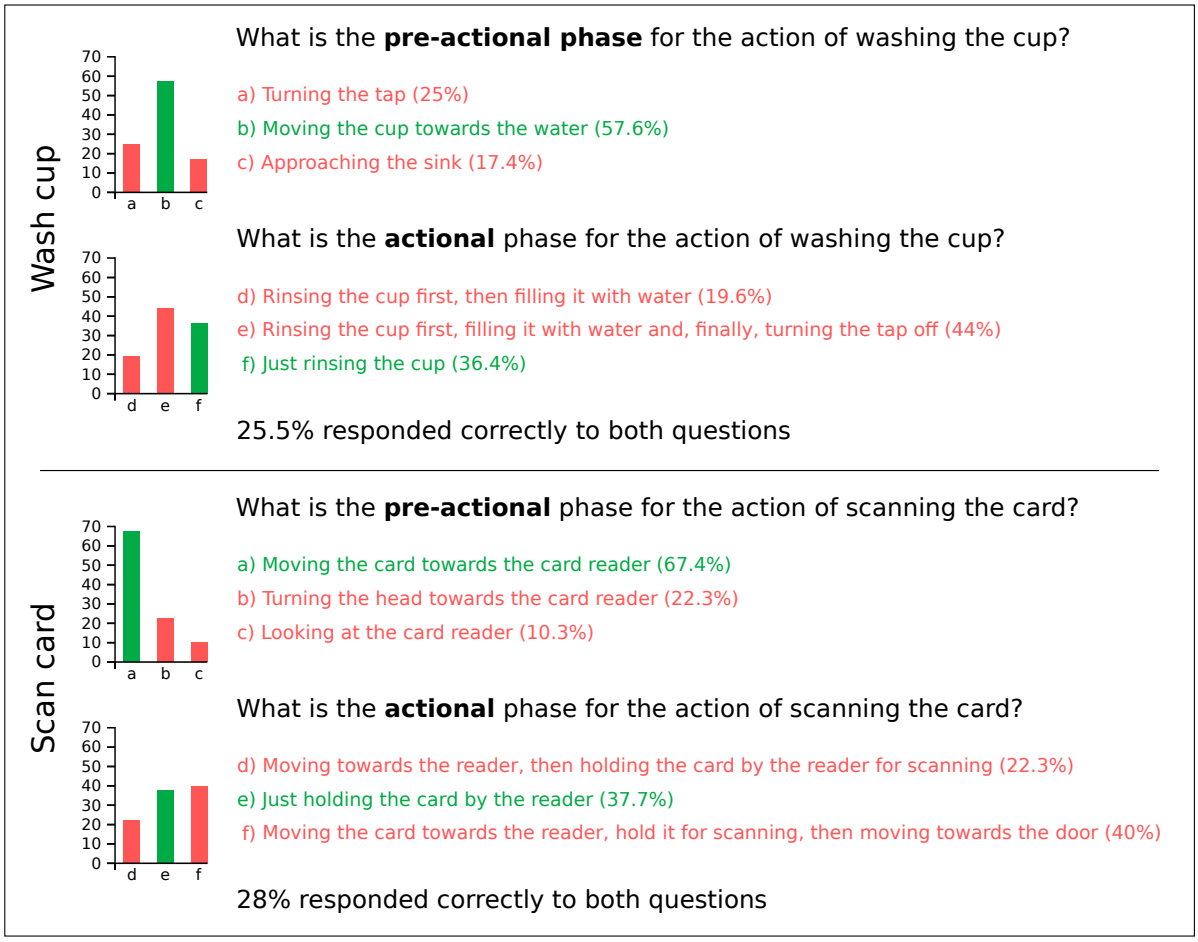

Figure 4.19: Evaluating RB guidelines intuitiveness for crowd-sourced annotations. Green and red indicate correct and wrong answers.

carefully read the instructions of the assigned task. While some work could be attempted to improve the readability of the RB definitions for a broader audience, other premium crowd-sourcing platforms offering high quality annotations could also be used to gauge the protocol usability in crowd-sourced settings.

4.4.5 Evaluating Accuracy and Robustness

In order to assess whether more precise temporal boundaries lead to a boost in accuracy and robustness, the GTEA Gaze+ dataset was locally re-annotated adopting the Rubicon Boundaries. GTEA Gaze+ was chosen since it is the largest dataset amongst the evaluated datasets. Importantly, as noted in Section 4.2, it is also the dataset with the most inconsistent temporal annotations. Before discussing results obtained using the RB segments, let us observe Figure 4.20, which compares the original and the RB annotations for a number of actions. RB labels are highlighted in blue, while original labels are coloured in red. To facilitate comparison, the illustrated RB segments correspond to the actional phase alone.

Figure 4.20: Qualitative comparison between RB and original annotations on GTEA Gaze+. Rubicon Boundaries annotations (blue segments) are compared to the corresponding original annotations (red segments) for a number of actions. The illustrated RB segments correspond to the actional phase. Each segment shows the labelled start and end frames (first and last images), as well as the frame sampled at the middle of the segment (centre image).

Original_{gt}	Original_{gen}	$\text{Original}_{gt}^{aug}$	$\text{Original}_{gen}^{aug}$	RB_{gt}^{act}	RB_{gen}^{act}	RB_{gt}	RB_{gen}
61.2	59.6	57.9	58.1	64.9	63.2	**65.6**	61.7

Table 4.8: Top-1 accuracy obtained with original annotations (with and without augmentation) and RB labels (actional phase alone and full segment). Accuracy is reported for both ground truth and generated segments. Results obtained with 2SCNN on GTEA Gaze+.

The Figure shows cases where the RB and original annotations differ significantly. Indeed, many original action segments are moderately loose and include frames belonging to the following action. For example, the original annotation for "open microwave" includes the action of putting the cup into the microwave, while "open freezer" includes the subsequent action of closing the freezer. Similarly, the original segment for "close fridge" shows part of the next action where the person opens the other fridge's door. In other cases, e.g. "put down bowl" the original segment does not enclose part of another action but includes background frames. The Rubicon Boundaries annotations are more precise and consistent than the original counterparts.

The newly collected RB annotations were used to evaluate 2SCNN on GTEA Gaze+. 2SCNN was trained and tested with the same settings used for the experiments presented in Section 4.3. Table 4.8 compares previously reported top-1 accuracy obtained with the dataset's original annotations, including performance obtained with data augmentation, to results obtained with the Rubicon Boundaries labels. Both the actional phase alone (RB_{act}) and the concatenation of the two phases (RB) are evaluated. For each annotation set, accuracy obtained on the ground truth and the generated test segments is reported.

Let us first focus on the performance of 2SCNN on the ground truth test set. The concatenated Rubicon Boundaries segments (RB_{gt}) proved the most accurate, leading to an increase of more than 4% in accuracy compared to the original ground truth segments (Original_{gt}). The RB segments delivered an accuracy 7.7% higher than that obtained with temporal augmentation ($\text{Original}_{gt}^{aug}$), showing that consistent labelling cannot be substituted with simple data augmentation. Interestingly, the actional phase alone (RB_{gt}^{act}) performs comparably to the full RB segment and still 3.7% higher than the original annotations.

Figure 4.21 shows per-class accuracy difference between RB annotations (both the full segment and the actional phase alone) and the original labels. When using the actional phase alone 21 out of 42 classes improved (top plot, yellow bars), whereas 11 classes were better classified with the original annotations (top plot, red bars). When evaluating the full RB segment, 23 classes received a boost in accuracy (bottom plot, blue bars), while the original annotations proved better for 10 classes (bottom plot, red bars). Certain classes achieved higher accuracy only with the actional phase, while some others did only with the full RB segment. Such cases are highlighted in bold in Figure 4.21. In total, 10 and 9 classes remain unchanged when comparing the RB annotations (actional phase and full) to the original annotations. No evident correlation between the kind of action and its variation in accuracy

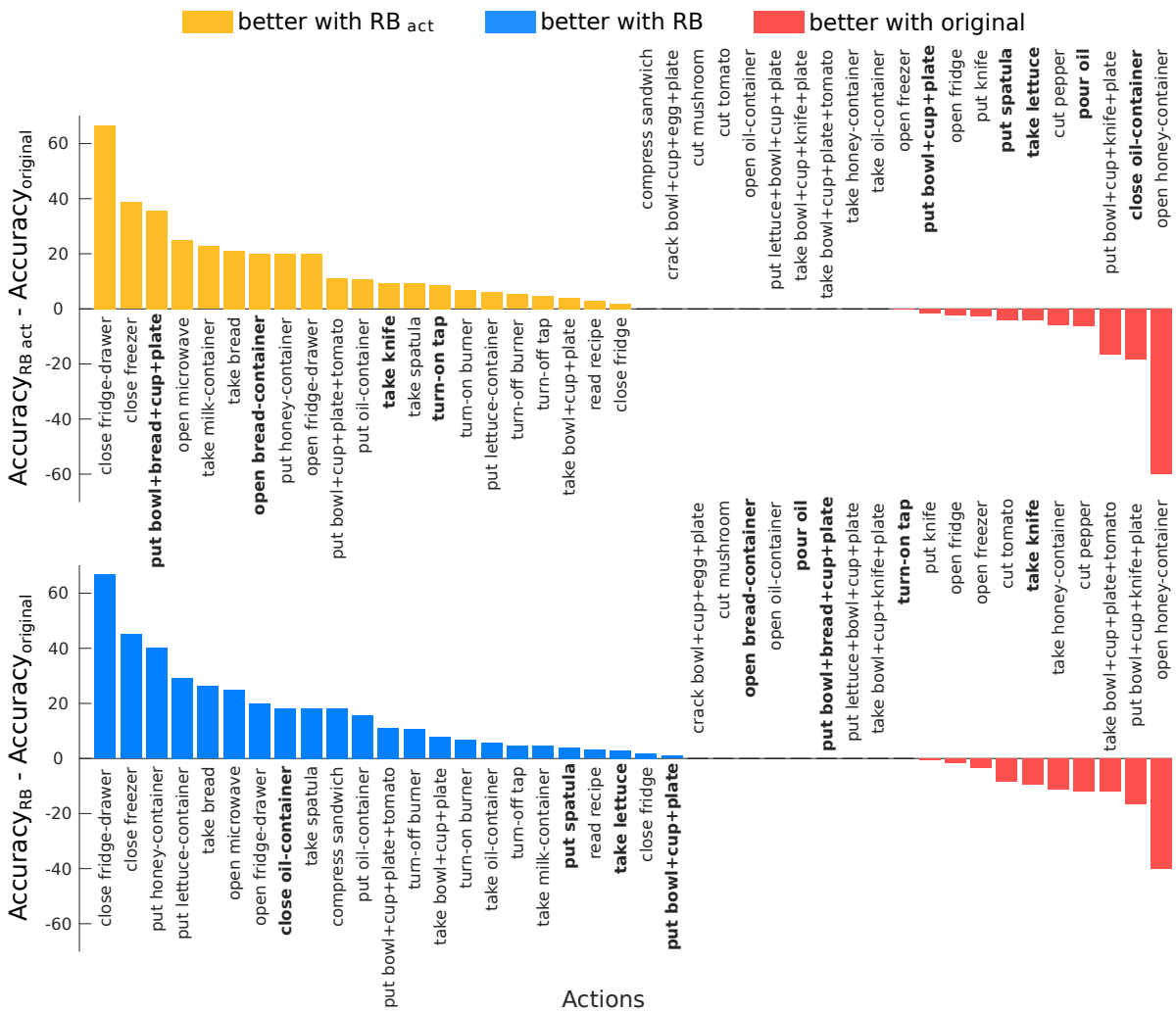

Figure 4.21: GTEA Gaze+: class accuracy difference between original and RB annotations. Some classes achieved higher accuracy only with RB_{act}, while other did only with the full RB segment. Bold highlights such cases.

appears when using the Rubicon Boundaries labels, which suggests that all types of actions benefit from consistent and precise temporal labelling.

Given that the experimental setup was identical to that used for the original annotations, the boost in accuracy can be ascribed solely to the new action boundaries. Indeed, the RB approach helped the annotators to more consistently segment the object interactions contained in GTEA Gaze+, which was one of the most challenging datasets for egocentric action recognition at the time of this work. This in turn produced more precise temporal boundaries, i.e. enhanced action samples that helped the classifier to better learn the actions. A 4% accuracy boost obtained by only relabelling the action temporal bounds should hopefully convince that good labels matter and can make a difference.

To assess robustness, new test segments were generated from the RB annotations, using the same method explained in Section 4.3.1. Table 4.8 reports results obtained on these generated segments. Although RB_{gen} exhibits higher accuracy than $Original_{gen}$ (61.7% vs 59.6%), a clear drop in performance between ground truth and generated segments is still observable, with a difference in accuracy of 3.9%. Interestingly, improved robustness can be noted when using the actional phase alone, where performance dropped by 1.7%. Recall that 2SCNN was trained again using the RB labels, and therefore both the training and testing aspects should be considered when analysing robustness in this case. From a training perspective, the actional phase alone, excluding motion that is not strictly related to the action, might force the classifier to focus on more specific action cues. This might build a representation that is more resilient to noise, although the slightly higher accuracy achieved with the full segment indicates that the context provided by the pre-actional phase is valuable.

From a testing viewpoint, given that the actional segment starts immediately after the pre-actional phase, the trespassing act is less severe. This is because the generated segments are likely to include frames belonging to the pre-actional phase, i.e. frames that were labelled to be relevant to the action.

To summarise, in this study high performance and robustness seem to be correlated, hinting that the evaluated classifiers possibly tend to overfit. This does not come to a surprise considering the small size of the analysed datasets. Although using the actional phase alone is a good compromise between accuracy and robustness, more insightful analysis of what recognition models learn is still needed. The spirit of this work was to draw attention to the long neglected problem of annotating actions in videos, and accordingly to propose labelling guidelines to foster more precise and consistent temporal annotations. The next Section provides some reflections on future directions.

4.5 Conclusion

Annotating temporal bounds for object interactions is the base for supervised action recognition algorithms. This work uncovered inconsistencies in temporal bound annotations, both within and across three egocentric datasets. The robustness of both hand-crafted features and fine-tuned end-to-end recognition methods was assessed. Results demonstrated that both approaches are susceptible to variations in start and end times. An approach to consistently label temporal bounds for object interactions was finally proposed and thoroughly evaluated. A few potential future directions can be taken from here.

Other CNN architectures Other architectures for action recognition could be tested for robustness. Particularly, it would be interesting to evaluate architectures that model temporal progression using recurrent networks (e.g. LSTM), those that model time by using three-dimensional convolutions (e.g. I3D [11]), and popular approaches that model long-term temporal dynamics with 2D CNNs (e.g. TSN [110]). Chapter 5 will revisit some of the questions presented here, evaluating more recent models and a larger egocentric dataset.

How can robustness be achieved? Although precise temporal labelling is beneficial, the volume and rapid pace at which increasingly larger datasets are being harvested inherently challenges the annotation quality. Approaches that are accurate, robust to bounds variations, and importantly can learn from noisy temporal labels are therefore highly desired. Chapter 5 will return to this question, suggesting some possible directions.

Which temporal granularity? The Rubicon Boundaries protocol addresses consistent labelling of temporal bounds, but they do not concern the granularity of the action. For example, is the act of cutting a whole tomato composed of several short cuts or is it one long action? The Rubicon Boundaries model is designed for actions relative to a goal a person wishes to accomplish, and accordingly depends on the way this goal is described. The granularity of an object interaction is another matter, and annotating the level of granularity consistently has not been addressed yet. Expanding the Rubicon Boundaries to enable annotating the granularity is another possible future direction.

Do models need accurate temporal bounds? This Chapter is based on the assumption that models need a start and end time to select frames during training. This is indeed the case for the vast majority of fully supervised approaches for action recognition. This labelling is expensive to collect and, as we saw here, importantly arbitrary. Chapter 6 will revisit this kind of supervision, proposing a novel type of annotation, i.e. one single timestamp roughly located around the occurrence of an action. Using a method designed to train a classifier with this temporally weaker but less expensive and arbitrary supervision, we will see that it is possible to obtain comparable or equivalent performance to that obtained with standard start/end times.

CHAPTER 5

REVISITING ROBUSTNESS

As seen in the previous Chapter, classifiers are sensitive to temporal boundaries variations. In fact, Chapter 4 showed that classification performance decreases when the start and end times of ground truth segments are altered during testing. The models assessed in the previous Chapter were state-of-the-art methods contemporary to the time of that study (late 2015). For completion, this Chapter evaluates more recent models in order to assess whether robustness is still an issue in current successful frameworks for action recognition.

The experimental setup and the analysis of the results presented here adheres to the same paradigm followed in the previous Chapter. Specifically, the same datasets inspected earlier are evaluated in these new experiments. Another recent dataset is additionally examined to determine whether robustness is also an issue in datasets labelled via crowd-sourcing, i.e. by means of a large number of annotators as opposed to a single person or laboratory, like in the older datasets evaluated before. Results will show that current popular approaches are sensitive to minor temporal boundaries variations in both scenarios.

5.1 Evaluating Common Approaches for Action Recognition

The models evaluated in this study are Temporal Segment Networks [110] (TSN, reviewed in Section 3.2.3), Two-Stream Inflated 3D CNN [11] (I3D, reviewed in Section 3.2.4) and a standard uni-directional LSTM. These models were chosen to compare different popular approaches for action recognition, i.e. 2D CNNs with basic temporal modelling (TSN), 3D CNNs (I3D) and recurrent networks (LSTM).

Varying the start and the end times of action segments has a theoretically different impact on the three models. Feed-forward architectures such as TSN and I3D operate on single images or stacks of frames. In this setting, video snippets are typically classified by uniformly averaging the predictions

of each frame/stack sampled from the segment. This way of temporally aggregating scores is prone to robustness issues. Because scores are averaged uniformly, predictions obtained from noisy frames have the same weight as the predictions obtained from relevant ones.

The density of the input plays also an important role in principle. Intuitively, large stacks of frames are more sensitive to noise compared to sparser samples (e.g. single images or small stacks), i.e. the amount of perturbation increases as a larger number of potentially irrelevant frames are aggregated to form inputs. I3D operates on dense stacks of 32 or 64 contiguous frames. As boundaries are varied, a single stack might thus include both relevant and irrelevant frames, which makes predicting I3D's robustness difficult. TSN receives single RGB frames or small stacks of 5 optical flow images. When using single RGB images, samples will be either outside or inside the bounds. When employing optical flow images, it will be less likely for TSN's small stacks to include both relevant and irrelevant frames, compared to I3D. This entails that for TSN the effect of altering action boundaries should be directly related to how many samples are outside the bounds.

LSTMs are able to forget irrelevant information passed through the various steps of a sequence. However, classification with recurrent networks is typically carried out by taking the predictions of the last step. This entails that the LSTM model should be more robust to variations to the start of an action segment in comparison to its end.

After this brief theoretical discussion on how the models can be sensitive in different ways, this Chapter moves to the experimental evaluation of the three frameworks. Training and testing details are provided next, together with information on the evaluated datasets.

5.2 Experimental Setup

5.2.1 Models

TSN and I3D were employed using Inception with Batch Normalisation [88] as the backbone CNN. All models were trained and tested using a late-fused two-stream architecture using RGB and optical flow inputs. The LSTM architecture was designed following prior common approaches [11, 15]. Specifically, the LSTM was composed of 128 hidden units and received RGB or optical flow features extracted with TSN's backbone. The features were extracted using the same TSN models fine-tuned on each dataset, i.e. TSN and LSTM use the same features and are thus directly comparable except for the classification layer. The sequence length for the LSTM was set on a per-dataset basis to optimise classification performance.

Training For TSN, 5 random frames or optical flow stacks per segment were selected using TSN's sampling strategy, for all datasets. Optical flow stacks contained 5 frames, as in [110]. For I3D, 1 stack of RGB or optical flow frames was randomly sampled, as in [11]. The size of the I3D stack was set based on preliminary testing in order to optimise classification accuracy. For BEOID and GTEA Gaze+ the stack was composed of 32 frames, while for CMU-MMAC and EPIC Kitchens the length of the

Dataset	Number of ground truth segments	Number of generated segments	Number of classes
BEOID	742	16,691	34
CMU-MMAC	450	26,160	31
GTEA Gaze+	1,141	22,221	42
EPIC Kitchens	9,009	40,009	274

Table 5.1: Number of ground truth/generated segments and classes for BEOID, CMU-MMAC, GTEA Gaze+ and EPIC Kitchens.

stack was 64 frames. On CMU-MMAC and EPIC Kitchens the higher accuracy observed with the larger stack is probably due to the longer action segments (CMU-MMAC) and the higher fps (EPIC Kitchens) of the datasets.

The input sequence length for LSTM was set to 50 time steps for CMU-MMAC and GTEA Gaze+, while for BEOID and EPIC Kitchens it was equal to 20 steps. The spatial and temporal networks of each model were trained independently. TSN and I3D models were pre-trained on Kinetics [50]. The models were trained using only the ground truth segments.

Testing TSN and I3D were evaluated by uniformly sampling n frames or stacks for each action segment, with $n \in \{1,2,5,10,25\}$. For these models, video segments were obtained by uniformly averaging the predictions obtained on each sample. As discussed before, averaging scores uniformly is prone to robustness issues. Aware of this, uniform aggregation was nonetheless adopted since it is the most commonly used way to combine frame predictions in strongly supervised approaches. Alternative score aggregation methods such as SCSampler [53] might mitigate robustness issues, however this is left as a direction for future research.

Given that LSTMs usually require a longer input to achieve good performance, the LSTM model was tested with a sequence of length $n \in \{5,10,25,50,100\}$. Features samples were uniformly drawn as for TSN and I3D. The output of the last time step was used to classify the sequence, following common practice. As mentioned earlier, this approach is susceptible to robustness. Nevertheless, in the same spirit as above, this testing approach was adopted since it is the most common one. Classifying temporal sequences in LSTMs taking uncertainty into account is accordingly left for future investigation. For all models, classification scores obtained with RGB and optical flow modalities were averaged to obtained the final prediction.

5.2.2 Datasets

The BEOID, CMU-MMAC and GTEA Gaze+ datasets were evaluated in order to compare more recent models with those analysed in Section 4.3. Additionally, the EPIC Kitchens dataset was also evaluated to assess whether robustness is an issue in datasets labelled via crowd-sourcing. EPIC Kitchens was chosen particularly for its high density of actions which, as discussed in Chapter 4, is an important characteristic to look for when temporal boundaries are concerned. All datasets are reviewed in Section 3.1. Experiments on EPIC Kitchens were conducted on a 25% subset of the dataset, which

Dataset	**TSN**$_{gt}$	**TSN**$_{gen}$	**I3D**$_{gt}$	**I3D**$_{gen}$	**LSTM**$_{gt}$	**LSTM**$_{gen}$	**IDT FV**$_{gt}$	**IDT FV**$_{gen}$	**2SCNN**$_{gt}$	**2SCNN**$_{gen}$
BEOID	93.5	86.7	**94.1**	89.0	93.1	85.3	85.3	75.4	93.5	83.8
CMU-MMAC	72.4	74.4	80.9	**83.1**	74.7	76.0	54.9	52.8	76.0	71.7
GTEA Gaze+	60.4	57.6	**63.6**	60.1	59.6	56.7	45.4	43.3	61.2	59.6
EPIC Kitchens	38.0	38.8	**43.6**	40.1	37.0	35.7	-	-	-	-

Table 5.2: Top-1 accuracy obtained with TSN, I3D and LSTM on BEOID, CMU-MMAC, GTEA Gaze+ and EPIC Kitchens, using ground truth and generated segments. Results obtained with fusion, testing 10 frames/stacks for TSN and I3D and 25 samples for LSTM. Previously reported results (see Table 4.3) obtained with IDT FV and 2SCNN on BEOID, CMU-MMAC and GTEA Gaze+ are included as well. Highest accuracy for each dataset is highlighted in bold.

still amounts to 13.5 hours footage containing 9,009 action segments belonging to 274 classes, as detailed in Table 5.1. The same subset of EPIC Kitchens will also be used in Chapter 6. Experiments on BEOID, CMU-MMAC and GTEA Gaze+ were conducted using the same generated segments and 5-fold cross validation used in the previous experiments. For EPIC Kitchens segments were generated as explained in Section 4.3.1, using single fold evaluation.

5.3 Results

Table 5.2 reports results obtained with TSN, I3D and LSTM on the four datasets, using ground truth and generated segments. The showed results were obtained with fusion, testing 10 frames/stacks for TSN and I3D and 25 samples for LSTM. The Table also includes results obtained with IDT FV and 2SCNN on BEOID, CMU-MMAC and GTEA Gaze+, previously reported in Table 4.3.

Let us first compare the new results with the old ones. At a glance, recent models still appear sensitive to temporal boundary variations. For instance on BEOID, the datasets where all models exhibit the largest accuracy drop, performance lowers by 7% with TSN and LSTM. Performance on BEOID's generated segments improves with I3D, where accuracy drops by 5%. I3D is also the model achieving the best performance on all datasets, which shows the advantage of employing 3D convolutions with CNNs pre-trained on large datasets. On GTEA Gaze+ accuracy on the generated segments lowers by around 3% with all models, whereas IDT FV and 2SCNN displayed a lower drop.

It is interesting to notice that 2SCNN, despite being older than the models evaluated here and although utilising an even older backbone CNN (VGG-16 [95] versus Inception with Batch Normalisation [88]), attains equal or superior performance compared to TSN and LSTM, and similar results compared to I3D. This is possibly due to the fact that fusion in 2SCNN is operated with a 3D kernel that learns the correspondences between RGB and optical flow features. The learnt fusion approach is more sophisticated than the simple late fusion approach, where streams are combined only at test time, with the spatial and temporal CNNs trained independently.

CMU-MMAC is the only dataset where all models achieve the best performance with the generated segments. As reported in Section 4.3.2, CMU-MMAC's segments are somewhat loose, and

Dataset	TSN			I3D			LSTM		
	F	S	T	F	S	T	F	S	T
BEOID	93.5	88.7	89.5	94.1	92.4	93.7	93.1	90.3	90.0
CMU-MMAC	72.4	69.5	62.9	80.9	78.9	77.3	74.7	74.7	66.2
GTEA Gaze+	60.4	54.5	56.1	63.6	61.6	60.2	59.6	55.7	57.1
EPIC Kitchens	38.0	34.4	31.6	43.6	40.9	35.7	37.0	31.8	32.8

Table 5.3: Comparing image modalities on the ground truth segments. Top-1 accuracy obtained with TSN, I3D and LSTM on BEOID, CMU-MMAC, GTEA Gaze+ and EPIC Kitchens. Results obtained with 10 testing frames/stacks for TSN and I3D and 25 samples for LSTM. F, S and T indicate results obtained respectively with fusion, spatial and temporal streams.

as a consequence generated segments that happened to be tighter were overall recognised more successfully than the original ground truth segments.

On EPIC Kitchens, the largest amongst the evaluated datasets, TSN and LSTM show little performance variation, with TSN slightly improving accuracy when testing the generated segments with fusion. However I3D, which outperforms TSN and LSTM by more than 5% on the ground truth segments, displays a drop in accuracy of 3.5% when evaluating the generated segments. This might be due to the dense 3D stacks utilised by I3D, which, as discussed in Section 5.1, while being effective to learn complex temporal dynamics might also lead to an increased sensitivity to temporal bounds variations.

The ground truth performance reported with fusion is now compared to the accuracy obtained on the ground truth segments with the individual spatial and temporal streams. As shown in Table 5.3, all models benefit from using a two-stream architecture, achieving their highest accuracy when fusing the spatial and the temporal streams. TSN and LSTM use the same visual features and differ only at the classification layer. This is reflected by an overall similar accuracy. On BEOID, CMU-MMAC and GTEA Gaze+, LSTM performs better than TSN on the individual spatial and temporal streams. This is possibly due to the intrinsic temporal modelling of recurrent networks, compared to the more limited temporal modelling employed by TSN with sample consensus.

Performance versus number of samples

Figure 5.1 plots accuracy versus number of tested samples, for all models, datasets and modalities. Green, red and blue correspond respectively to fusion, RGB and optical flow. Solid and dash lines depict accuracy obtained respectively with ground truth and generated segments. Coloured areas illustrate accuracy drop.

Testing more samples improves accuracy and reduces the gap between ground truth and generated segments, with diminishing returns as the number of samples grows. In some cases (CMU-MMAC with all models and EPIC Kitchens with TSN fusion and RGB) accuracy on the generated segments surpasses ground truth accuracy with a greater number of samples, which suggests that the ground truth temporal boundaries might be a little loose.

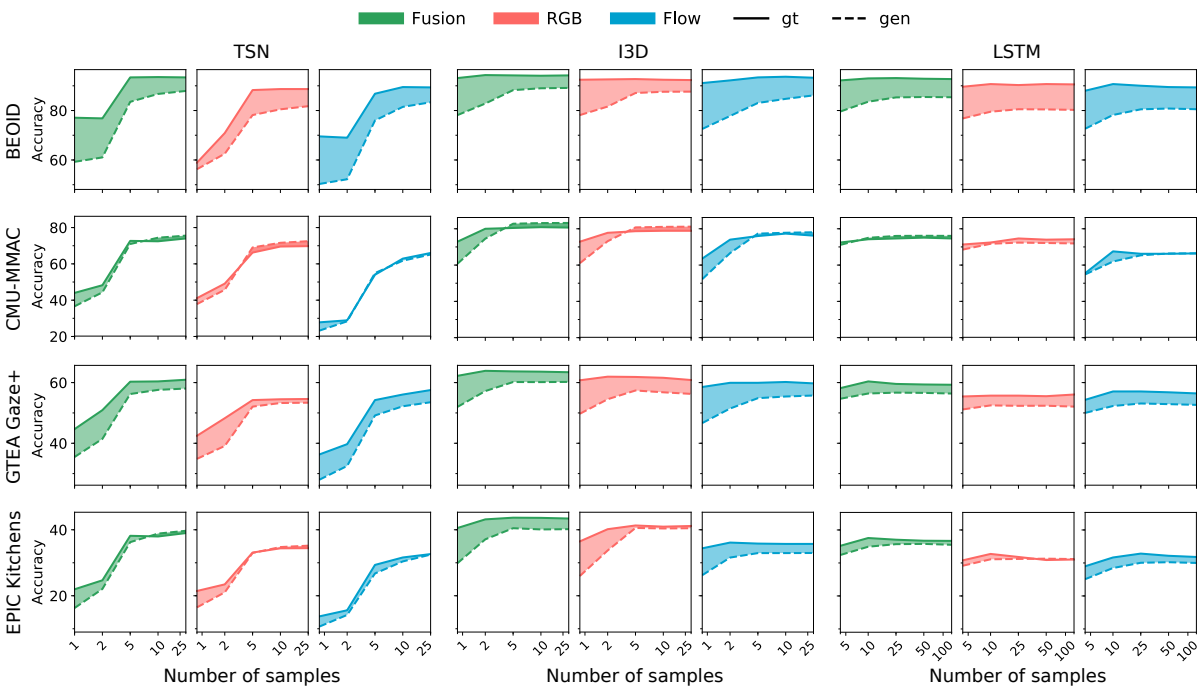

Figure 5.1: Accuracy versus number of testing samples. Solid and dash lines illustrate accuracy obtained with ground truth and generated segments. Coloured areas depict accuracy drop.

TSN is the model benefiting the most from observing a larger portion of the action, which is expected as it is based on sparse sampling. Conversely I3D, being based on dense sampling (each sample is a stack of 32 or 64 frames) gains a smaller boost when testing more samples, especially when evaluating ground truth segments. LSTM performance also varies little as the number of samples increases. This is probably due to the fact that LSTM was tested with at least 5 samples (compared to 1 for TSN and I3D), which appear to be sufficient for the model to achieve good performance.

By and large, the three modalities display a similar accuracy-drop trend on each dataset-model combination. In conclusion, while testing a greater number of samples helps narrowing the gap between ground truth and generated segments, denser sampling does not seem to be a solution to the sensitivity of recent models to bounds variation.

Performance versus IOU, start/end shift and length difference

BEOID Figure 5.2 shows accuracy versus IOU, start/end shift and length difference for BEOID. The plotted results were obtained with fusion, 10 frames/stacks for TSN and I3D and 25 samples for LSTM. This type of plots was presented in Section 4.3.2. To recall how to read these: i) the illustrated bars are normalised using the corresponding distribution; ii) a negative/positive start shift implies that a generated segment begins before/after the corresponding ground truth start (new start is inside/outside the bounds); iii) a negative/positive end shift involves that a generated segment finishes before/after the corresponding ground truth end (new end is inside/outside the bounds); iv) a negative/positive length difference entails that the generated segment is shorter/longer.

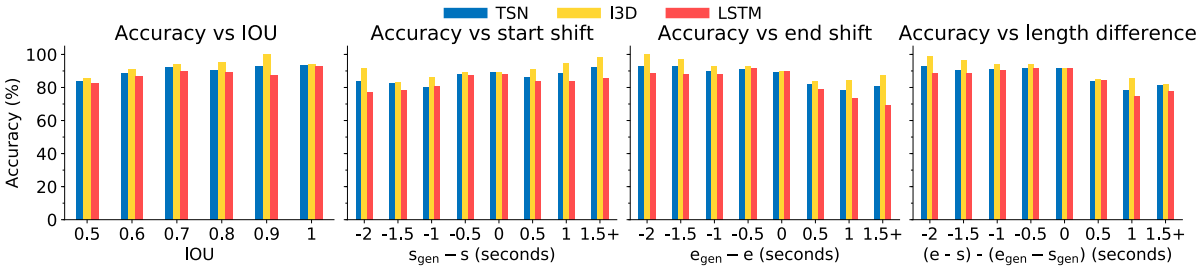

Figure 5.2: BEOID: classification accuracy vs IOU, start/end shifts and length difference, for TSN, I3D and LSTM. Plotted results obtained with fusion, 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

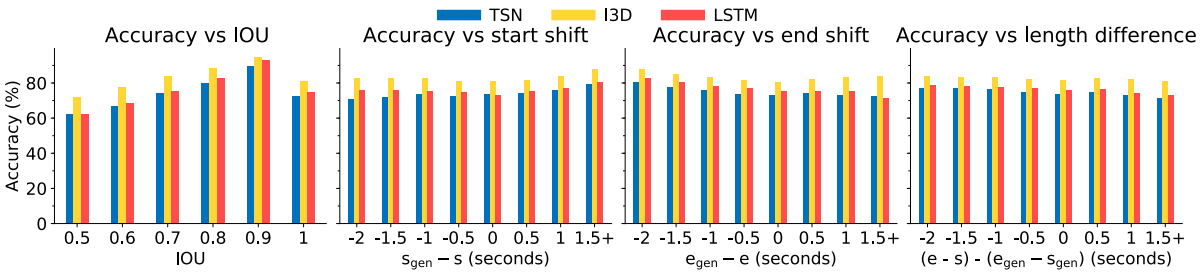

Figure 5.3: CMU-MMAC: classification accuracy vs IOU, start/end shifts and length difference, for TSN, I3D and LSTM. Plotted results obtained with fusion, 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

BEOID is the dataset where all models exhibit the highest accuracy drop. For TSN and LSTM performances decreases as the IOU between ground truth and generated segments lowers. I3D instead achieves a higher accuracy with generated segments at IOU 0.9. This suggests that I3D might benefit from shorter segments even when bounds are accurate like in BEOID. This hypothesis is reinforced when looking at the higher accuracy achieved with I3D with positive start shift, negative end shift and negative length difference.

Overall all models appear to be more sensitive when frames are added rather than removed, a trend that will appear for the other datasets as well. In fact accuracy varies little with negative length difference, while a larger drop is observable when generated segments are longer. Similarly, performance varies more when the shifted bounds are outside the ground truth boundaries, i.e. when the start and end shifts are respectively negative and positive. This suggests that the models are able to recognise the actions with fewer relevant frames, but are also sensitive to background frames or frames belonging to parts of neighbouring actions. This is because fully supervised approaches typically assume the action is fully contained in the given boundaries, and thus assign equal importance to all frames within the segment (recall that frame predictions are uniformly aggregated at test time).

CMU-MMAC The best performance on CMU-MMAC is obtained with the generated segments. This is observable with accuracy being higher for all models when the generated segments had an overlap of 0.8-0.9 with the ground truth segments, as depicted in Figure 5.3. As seen before, CMU-MMAC

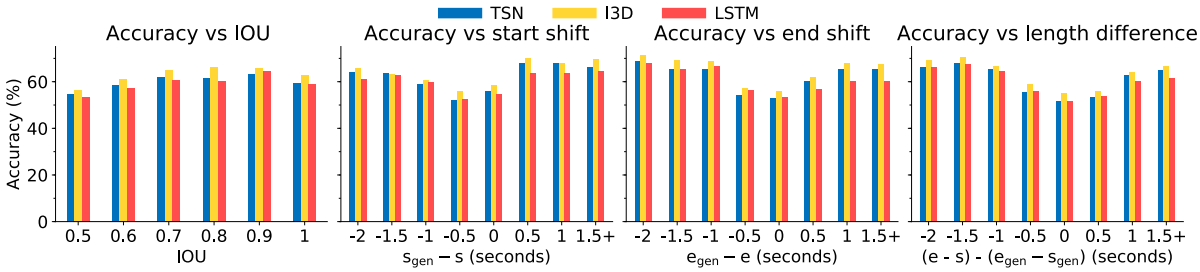

Figure 5.4: GTEA Gaze+: classification accuracy vs IOU, start/end shifts and length difference, for TSN, I3D and LSTM. Plotted results obtained with fusion, 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

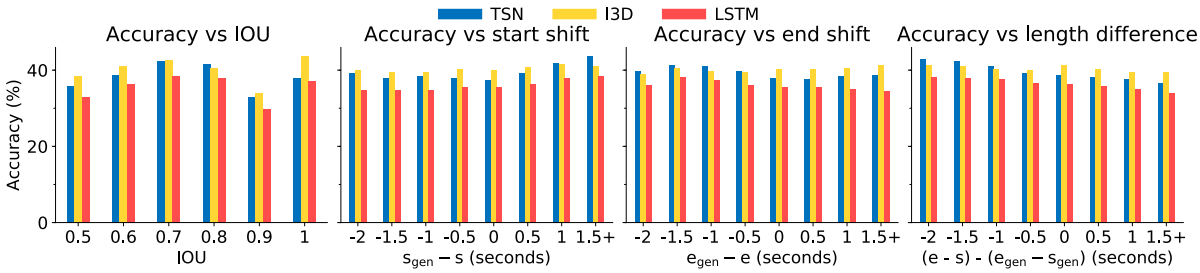

Figure 5.5: EPIC Kitchens: classification accuracy vs IOU, start/end shifts and length difference, for TSN, I3D and LSTM. Plotted results obtained with fusion, 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

annotations tend to be somewhat loose, including multiple object interactions in a single segment. Generated segments excluding unrelated frames were thus more discriminative and classified more successfully. Accordingly, like in the previous experiments discussed in Section 4.3.2, all models gain a slight boost in accuracy with positive start shift, negative end shift and negative length difference, i.e. when the generated segment is shorter. While this is also observable in BEOID in these experiments, this trend appears more prominent for all models in CMU-MMAC.

GTEA Gaze+ The annotated segments of GTEA Gaze+ are moderately inconsistent, as inspected in Section 4.2. Although all models display an accuracy drop of around 3% when evaluating the generated segments, Figure 5.4 shows that accuracy on the generated segments is consistently higher in almost all cases. This behaviour was also observed in Section 4.3.2. Because action boundaries are inconsistent in GTEA Gaze+, accuracy is not altered in a systematic way like in the other datasets, and thus it is difficult to draw any conclusions from the trends of the normalised and binned accuracies.

EPIC Kitchens Figure 5.5 shows results obtained on EPIC Kitchens. I3D, whose accuracy on the generated segments drops by 3.5%, displays worse performances with lower IOU, while LSTM seems to vary less with IOU. On the other hand TSN, which is the only model where accuracy was higher on EPIC Kitchens' generated segments, gains a slight accuracy boost with IOU 0.7-0.8. Accordingly,

Dataset	Segments with overlapping neighbour		Average IOU with neighbour		Confused with overlapping neighbour						Confused with non overlapping neighbour					
					TSN		I3D		LSTM		TSN		I3D		LSTM	
	gt	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*	*gt*	*gen*
BEOID	8.1	49.9	0.15	0.17	18.7	62.6	15.9	51.0	23.5	60.0	20.8	7.3	11.3	4.6	19.6	6.8
CMU-MMAC	0	39.4	/	0.08	/	25.5	/	14.3	/	23.2	26.6	10.6	19.8	10.5	20.2	8.5
GTEA Gaze+	3.9	22.3	0.12	0.13	6.0	20.0	4.8	16.5	5.4	19.1	20.2	14.5	17.9	12.7	18.6	11.9
EPIC Kitchens	42.9	61.2	0.15	0.17	6.6	11.6	5.4	9.8	5.1	10.5	4.7	1.5	4.5	1.2	5.1	1.6

Table 5.4: Confusion with overlapping and non overlapping neighbours. Neighbours are ground truth segments that are temporally close to or overlapping with another segment (either a ground truth or a generated one). Number of segments with an overlapping neighbour and average IOU are calculated considering all tested segments. The percentage of segments confused with a neighbour are calculated with respect to incorrectly classified segments only. All columns except "Average IOU" indicate percentages. Results obtained with fusion and 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

TSN exhibits larger accuracy improvement compared to I3D and LSTM when segments are shorter (positive start shift, negative end shift and negative length difference).

All models achieve their lowest performance when IOU equals 0.9, which is difficult to interpret. Firstly, it should be recalled that actions in EPIC Kitchens were labelled by a large number of crowd-sourced annotators. In spite of the effort put forth in collecting consistent and precise labelling, a larger variation in temporal consistency is expected due to the number of annotators and the scale of the dataset, which also makes a thorough inspection of the annotated bounds highly impracticable. Nevertheless, accuracy does not fluctuates in the same way when evaluating start/end shifts and length difference, which indicates that temporal annotations are overall sufficiently precise and consistent. Ultimately, this unexpected behaviour in accuracy versus IOU in EPIC Kitchens confirms that robustness is an issue in large datasets as well.

Inspecting robustness

Confusion Table 5.4 reports confusion with overlapping and non overlapping neighbours. This type of analysis was presented in Section 4.3.2 as well, where Figure 4.11 illustrated an example of overlapping and non overlapping neighbours. Let us recall here that a neighbour of a given segment x is a ground truth segment that is temporally close to or overlapping with x. If x is a generated segment, the ground truth parent segment (i.e. the segment from which x was generated) is excluded from x's neighbours. For non overlapping neighbours, the closest segments both before and after x were considered, up to a maximum distance of 3 seconds. Confusion with a neighbour entails that a segment was incorrectly assigned one of its neighbours' class.

Like in the experiments seen in the previous Chapter, confusion with an overlapping neighbour appears to be the main cause of error when testing the generated segments. However, it should be noted again that the generated segments had a large overlap with their parent ground truth segments

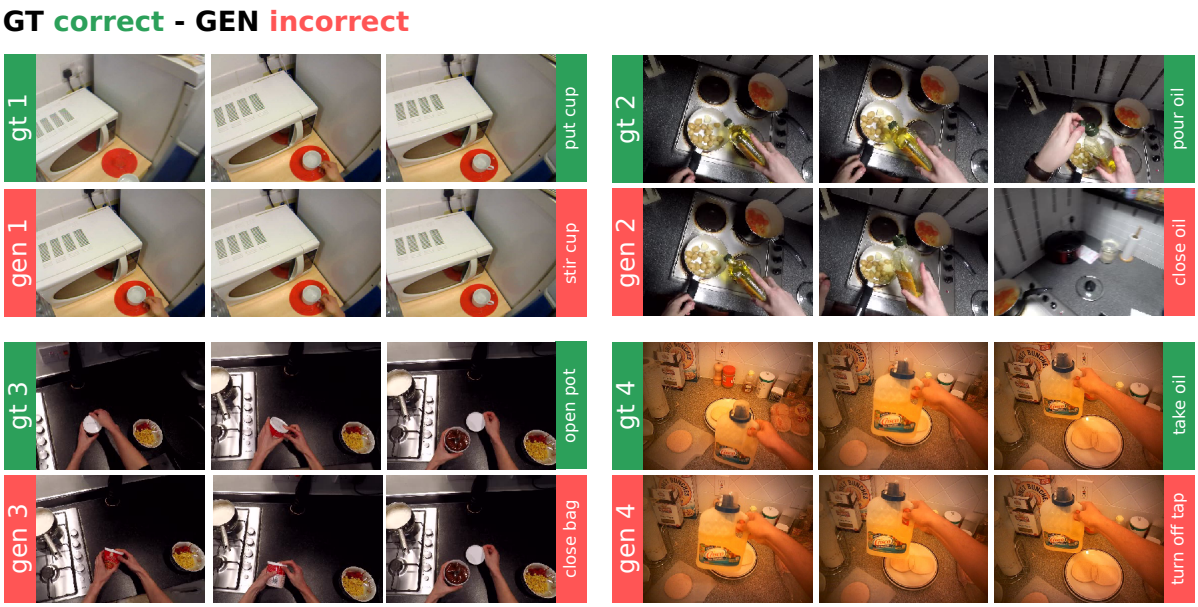

Figure 5.6: Qualitative results for correctly predicted ground truth (green) and incorrectly classified generated segments (red). Class labels correspond to the predicted class. Dataset and model: 1) BEOID with LSTM; 2-3) EPIC Kitchens with I3D; 4) GTEA Gaze+ with I3D. Showed results obtained with fusion, 10 stacks for I3D and 25 samples for LSTM.

(minimum IOU of 0.5), and that the average IOU with a neighbour is low on all datasets (maximum average IOU with a neighbour is 0.17). Notice how a large portion of ground truth segments (42.9%) in EPIC Kitchens overlaps with a neighbour. This is due to the realistic multi-tasking nature of the dataset, where multiple actions often occur concurrently. This is a distinctive trait of the dataset which challenges models to learn distinct actions from very similar visual cues.

Confusion with non overlapping segments is relatively low for the generated segments. In conclusion, confusion with neighbouring actions, although certainly an important factor, is not the only cause of accuracy drop. This suggests that models are still sensitive to frames not necessarily visually correlated to temporally close actions.

Qualitative results Figure 5.6 illustrates a few qualitative results, showing cases where the ground truth was correctly classified while the generated segment was not. Green and red depict correct and incorrect classification respectively. Class labels indicate the predicted class. In some cases the incorrectly predicted class is related to the visual content of the frames. This is the case for gen 1, where the action is truncated (but still visible) and gen 2, where a few background frames are added at the end of the segment. In both cases the models predicted a class that albeit incorrect involved the interacted object. Segments 3 and 4 depict cases where the predicted class is wholly unrelated to both the motion and the objects present in the video snippets. Note the high similarity between the ground truth and the generated segments, which shows how sometimes it is difficult to understand what models learn and how they internally represent actions.

Figure 5.7: Qualitative results for incorrectly classified ground truth (red) and correctly predicted generated segments (green). Class labels correspond to the predicted class. Dataset and model: 5) CMU-MMAC with LSTM; 6) EPIC Kitchens with TSN. Showed results obtained with fusion and 10 frames/stacks for TSN and 25 samples for LSTM.

Dataset	**Average class drop**			**Classes worsening with gen segments**			**Classes improving with gen segments**		
	TSN	**I3D**	**LSTM**	**TSN**	**I3D**	**LSTM**	**TSN**	**I3D**	**LSTM**
BEOID	4.9	5.9	7.1	67.6	73.5	76.5	20.6	11.8	23.5
CMU-MMAC	2.4	3.3	3.5	71.0	54.8	67.7	22.6	32.3	29.0
GTEA Gaze+	3.7	4.5	3.1	71.4	69.0	64.3	26.2	28.6	30.9
EPIC Kitchens	2.1	4.6	2.2	32.2	35.0	29.9	25.2	16.8	24.1

Table 5.5: Class accuracy drop. All reported numbers indicate percentages. Accuracy on generated segments did not change for some classes (i.e. percentage of classes worsening and improving for a given dataset-model combination does not sum to 1). Results obtained with fusion and 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

Figure 5.7 illustrates cases where the generated segment was correctly classified while its parent ground truth was not. Gt 5's end is slightly belated, showing the person's hand reaching out to open the cupboard before putting the oil bottle in it. Although the action of putting the oil into the cupboard is not visible, the classifier was likely confused by the motion present at the end of the segment and predicted the class "put oil". Gen 5 instead excludes the irrelevant motion and was correctly classified as "twist cap". Gt 6, a segment for the class "close oven", shows a person taking a cup from a microwave before closing the oven's door. The initial motion involved in grabbing the cup probably misled the classifier, which predicted an unrelated class ("take towel"). Gen 6, on the other hand, starts after the action of picking the cup is complete, including solely frames related to the act of closing the oven, which was correctly predicted.

Analysing classes Table 5.5 reports accuracy drop on a per-class basis. Performance on the generated segments deteriorates for the majority of classes for all datasets and models, showing that robustness is not an issue for only a few unfortunate actions. This is also reflected by the fact that the overall accuracy drop (see Table 5.2) and the average class drop are closely comparable for each dataset and model. Note that for some classes accuracy remained unchanged, thus the percentage of classes worsening and improving for a given dataset-model combination does not sum to 1.

Dataset	Number of generated segments	Generated better than ground truth		
		TSN	**I3D**	**LSTM**
BEOID	16,691	0.61	0.74	1.04
CMU-MMAC	26,160	2.96	1.40	4.05
GTEA Gaze+	22,221	2.30	3.48	3.17
EPIC Kitchens	40,009	3.81	2.72	4.22

Table 5.6: Percentage of generated segments that were "better" than the ground truth counterparts. This corresponds to such cases where a generated segment x was correctly classified while the ground truth segment that generated x was incorrectly classified. Results obtained with fusion and 10 frames/stacks for TSN and I3D and 25 samples for LSTM.

Table 5.6 shows the percentage of generated segments that were correctly classified while their parent ground truth segments were not. BEOID is the dataset with the most precise bounds amongst the evaluated datasets, and thus shows the lowest percentage of "better" generated segments. The remaining datasets, as discussed above, have imprecise boundaries to different extents, and therefore a greater (but still low) number of generated segments was better than their ground truth parents. Notice that the number of generated segments is very large in all datasets, hence the numbers reported in Table 5.6, although small, are still compatible with the percentages of classes improving with generated segments reported in Table 5.5.

5.4 Conclusion

Fully supervised approaches typically assume the action to be fully contained in the given boundaries, and thus assign equal importance to all frames within the segment. As seen previously in Chapter 4 and as revisited in this Chapter, when this assumption no longer holds true classification performance is likely to vary. Experiments in this Chapter showed that recent popular approaches employing different temporal modelling strategies are equally sensitive to bounds variation, both in small and large datasets. While the observed accuracy drop was often understandable, this study meant to communicate that precise temporal labelling is important and that temporal boundaries should not be taken for granted. This was perhaps more importantly highlighted by the fact that performance can even improve by simply adjusting the temporal boundaries, when these are not accurate. Alternative methods to aggregate frame predictions taking temporal uncertainty into account might make models more robust to imprecise or altered bounds. This constitutes an interesting direction for future research.

Video datasets are expanding at a very fast pace, which renders high quality temporal labelling an intrinsic challenge. Current research is exploring the opportunities offered by weak temporal supervision, however full temporal labelling is still often required to achieve high classification performance. Chapter 6 will propose a novel type of temporal supervision, i.e. single timestamps roughly aligned with actions in untrimmed videos, in the attempt to find a good compromise between

annotation effort and classification accuracy. As we will see shortly, the model was designed to achieve high performance under challenging conditions, but was not conceived nor evaluated for robustness to temporal bounds alteration during testing. Nevertheless, Chapter 6 will discuss some possible future directions that could alleviate robustness issues, borrowing ideas from the proposed training algorithm.

CHAPTER 6

ACTION RECOGNITION FROM SINGLE TIMESTAMP SUPERVISION IN UNTRIMMED VIDEOS

Typical approaches for action recognition rely on full temporal supervision, i.e. on the availability of the action start and end times for training. When the action boundaries are available, all (or most of) the frames enclosed by the temporal bounds can be considered relevant to the action, and frames can therefore be randomly or uniformly selected from the temporal segment to represent the action and train a classifier.

As we saw this far, delimiting the temporal span of an action is an often arbitrary matter. Moreover and importantly, collecting these boundaries is also notoriously laboursome and expensive. Video datasets are increasingly growing larger, thus it is crucial to speed up the annotation process to foster more rapid advancement in video understanding. Recent research employs weak temporal supervision to train action classifiers. A successful example of weak supervision is that of video-level labels, i.e. annotations that only signal the presence of an action in an untrimmed video. Albeit cheap to collect, video-level labels do not suffice when videos contain a large number of distinct actions, as we will see later on.

This Chapter proposes a novel type of temporal supervision, i.e. single timestamps roughly aligned with actions in untrimmed videos, together with a method to train a classifier using this supervision. This kind of annotation attempts to find a good balance between labelling effort and classification performance. In fact, single timestamps are quicker to collect and alleviate boundaries arbitrariness, given that annotators do not have to decide when the action starts or ends, but only label one timestamp within or close to the action. Single timestamps can alternatively be collected automatically from audio narrations or video subtitles, as seen for EPIC Kitchens [21] in Section 3.1.4. Single timestamps can offer a good temporal cue. Using the proposed algorithm, results will show that it is possible to achieve results comparable to those obtained with full temporal supervision.

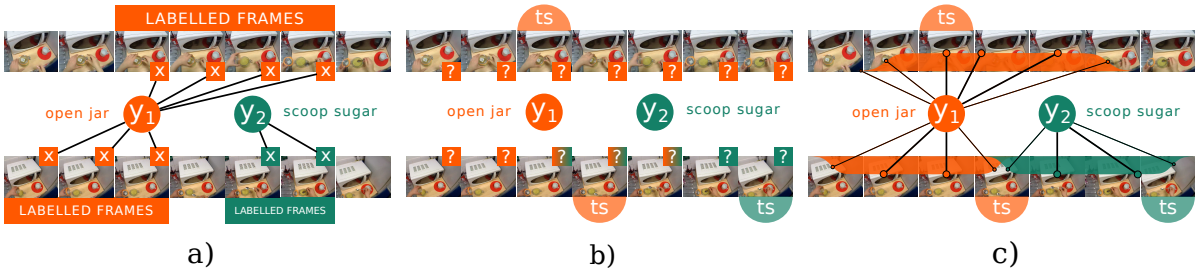

Figure 6.1: Comparison between temporal boundaries and single timestamps. When start/end times are available (a), all frames within labelled boundaries can be assigned to the class label. When action bounds are not available and only single timestamps are given (b), mapping frames to labels becomes a difficult task. In this work action bounds are replaced by sampling distributions (c), which iteratively update the mapping between frames and class labels. Top and bottom plots depict different videos.

The remainder of the Chapter is organised as follows. Section 6.1 provides an overview of the method, discussing the challenges involved in utilising roughly aligned single timestamps in untrimmed video dense of actions. The training algorithm is detailed in Sections 6.2 and 6.3. Results are discussed in Section 6.5. Finally, Section 6.7 concludes this study and provides some insight into possible future research directions.

6.1 Replacing Action Bounds with Sampling Distributions

This work considers the case where a set of untrimmed videos are provided for the task of fine-grained action recognition. That is the task of training a classifier $f(x) = y$ that takes one frame x in input to predict a class y from the visual content of x. For simplicity, from here on it is assumed that the classifier receives a single frame. This assumption will be relaxed later. The videos are expected to be dense, i.e. to contain multiple distinct actions, with actions likely occurring in a close succession.

The typical annotation for this task is given by the actions' start and end times, which delimit the temporal scope of each action in the untrimmed video, along with the accompanying class labels. As showed in Figure 6.1a, when using this supervision all frames enclosed by the labelled bounds are typically assumed to be relevant, and thus it is possible to train a classifier using any frame between the corresponding start/end timestamps.

Let us suppose now that one single timestamp is given instead of the action boundaries, together with a class label. Importantly, let us assume that the single timestamp, although being close to the action, is roughly aligned with the action. This means that there is no guarantee that the timestamp is located at a pertinent frame, and that the timestamp can be placed over background frames or even frames belonging to a distinct action. Figure 6.1b depicts such scenario, where three actions are labelled with single timestamps in two videos (top and bottom rows). Due to the approximate position of the timestamp (indicated with "ts" in the Figure), and because multiple temporally close actions are expected to be present in the video, using single timestamps to train a classifier is not straightforward.

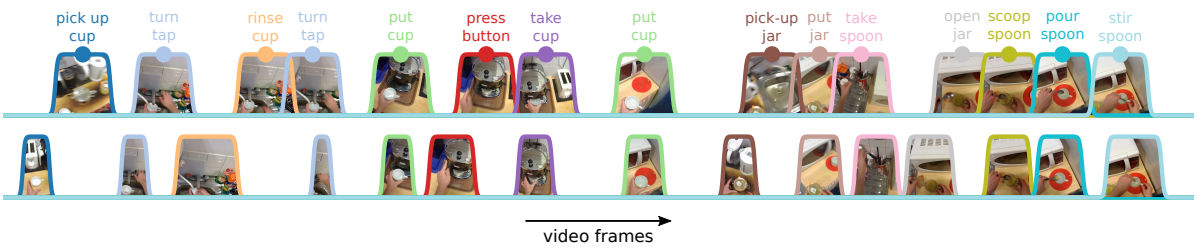

Figure 6.2: Replacing action boundaries with sampling distributions in an untrimmed video, given single timestamps (coloured dots). The initial distributions (top) may overlap (e.g. "pick-up jar", "put jar" and "take spoon") and contain background frames. The distributions are thus iteratively refined (bottom) using the classifier response during training, in order to sample more relevant frames and reinforce the classifier.

In this work action boundaries are replaced with sampling distributions, as illustrated in Figure 6.1c. The sampling distributions, initialised from the annotated timestamps, model the likelihood of a given frame of containing a certain action. Such distributions are used to sample training frames based on the action likelihood, which is iteratively refined based on the classifier response during training.

Figure 6.2 shows an untrimmed video with several actions annotated with single timestamps (coloured dots). A sampling distribution is initially centred on each labelled timestamps (top row). Due to the rough location of the timestamps, the action density and the different length of each action, the initial sampling distributions may overlap, encompass irrelevant frames or may not enclose enough frames to represent the action, which is not optimal.

The initial sampling distributions are iteratively refined during training, using the classifier's response (bottom row of Figure 6.2). This aims to sample more relevant frames and reinforce the classifier as training progresses. Details on the sampling distribution and the iterative update procedure are provided next.

Comparison to other works Mettes et al. [70] and Chéron et al. [14] also use a single temporal point per action for fine-grained recognition in videos. However, these works assume the given annotations to be correct at all times, and thus do not modify the temporal supervision used during training. In this work the temporal scope of the given supervision is instead actively refined, under the assumption that the annotated timestamps may not align well with the actions and thus lead to incorrect supervision.

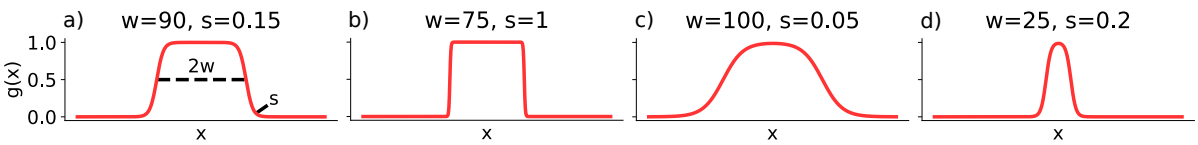

Figure 6.3: The plateau function showed in Equation 6.1, with different w and s parameters. The parameter c corresponds to the centre of the plateau, while w and s model its width and slope, respectively. The function ranges in $[0, 1]$.

6.1.1 Sampling Distribution: The Plateau Function

In this work, a sampling distribution is used to select frames to train a classifier, based on the likelihood that a given frame contains a certain action. The distribution should therefore resemble the output of a confident classifier, i.e. a smooth plateau of high classification scores for consecutive frames containing the action, with low response elsewhere. Another desirable property of this function is differentiability, so that it can be learnt in deep learning frameworks if needed. The Gaussian probability density function is commonly used to model likelihoods, however it does not exhibit a plateau response, peaking instead around the mean and steadily dropping from the peak. The gate function by definition exhibits a sharp plateau, however it is not differentiable. Based on the above observations, the probability density of the sampling distributions is here modelled with the following function:

$$g(x \mid c, w, s) = \frac{1}{\left(e^{s(x-c-w)}+1\right)\left(e^{s(-x+c-w)}+1\right)} \tag{6.1}$$

The parameter c models the centre of the plateau, while w and s model respectively its width (equal to $2w$) and the steepness of its side slopes. The range of the function is $[0, 1]$, thus it can be directly used to model likelihoods. The function is differentiable and is defined over the frames indices of an untrimmed video. Figure 6.3 illustrates the function with different w and s parameters. Note how the function can approximate the gate function (plot *b*) or the Gaussian distribution (plot *d*). The remainder of the Chapter will refer to g as the plateau function.

6.2 Initialising the Model

The sampling distributions are initialised from the single timestamp annotations. Let a_i^v be the i-*th* single timestamp in an untrimmed video v and let y_i^v be its corresponding class label, with $i \in \{1..N_v\}$ and $v \in \{1..M\}$. For each a_i^v, a sampling distribution centred on the timestamp is initialised with default parameters w and s. The parameters of the i-th distribution initialised for video v are denoted with $\beta_i^v = (c_i^v, w_i^v, s_i^v)$, where $c_i^v = a_i^v$, and accordingly $G(\beta_i^v)$ denotes the corresponding sampling distribution. $G(\beta_i^v)$ will be used to sample training frames from video v for the class indicated by y_i^v.

As noted before, due to the close proximity of some timestamps, the initialised plateaus may overlap considerably. Such overlap could be decreased by shrinking the plateaus, however, given that

the temporal extent of the actions is unknown, this may result in missing important frames. Instead, w and s are set to default values, allowing thus the overlap and giving all actions equal chance to be learnt from the same number of frames.

Frames sampled from these distributions might also be background frames or could be associated with incorrect action labels. To decrease noise, frames sampled from all untrimmed videos are ranked based on the classifier's response, after a few training iterations. Following a Curriculum Learning paradigm [6], only the frames where the classifier is most confident are selected for training. Let $P(k|x)$ denote the softmax score of a frame x for a class k. Let:

$$\begin{aligned} \mathscr{F}^k = \Big(x \leftarrow G(\beta_i^v) \ : \ y_i^v = k, \ \forall i \in \{1..N_v\}, \ \forall v \in \{1..M\}\Big) \\ s.t. \ P\big(k|\mathscr{F}_{t-1}^k\big) \geq P\big(k|\mathscr{F}_t^k\big) \end{aligned} \tag{6.2}$$

be the sequence of frames sampled from all the distributions for class k, ordered according their softmax scores. The top T frames in $\mathscr{F}^k$ are selected for training the model for class k:

$$\big(\mathscr{F}^k\big)_1^T \ : \ T = h|\mathscr{F}^k|, \ h \in [0,1] \tag{6.3}$$

This approach selects the frames where the classifier is most confident, which amounts to selecting the most relevant frames for each class within the plateaus. While this strategy feeds the classifier with fewer noisy samples, relevant frames outside the plateaus are still being missed. After training the model for a few iterations, the sampling distributions are thus updated, in order to sample more relevant frames and reinforce the classifier, as detailed next.

6.3 Updating the Distribution Parameters

The update procedure relies on the fact that multiple action instances are labelled for each class, and importantly, assumes that the initial plateaus are overall reasonably aligned with the actions. Under such assumptions, the parameters of the sampling distributions are iteratively updated, reshaping and moving the initial plateaus over more relevant frames, in order to reinforce the classifier.

The softmax scores for each frame of each video are first obtained from the classifier, at the beginning of a training iteration. Update proposals are then estimated from the softmax scores. The proposals are ranked to select the parameters that provide the most confident updates. The distributions are updated until convergence. For simplicity, the method is described here for one training iteration and one sampling distribution.

6.3.1 Producing Update Proposals

A series of consecutive classification scores can localise the temporal extent of an action in a video. In order to refine sampling distributions based on the classifier response, the plateau function can thus be fitted to the softmax scores. However, classification scores are typically noisy. For example, observe

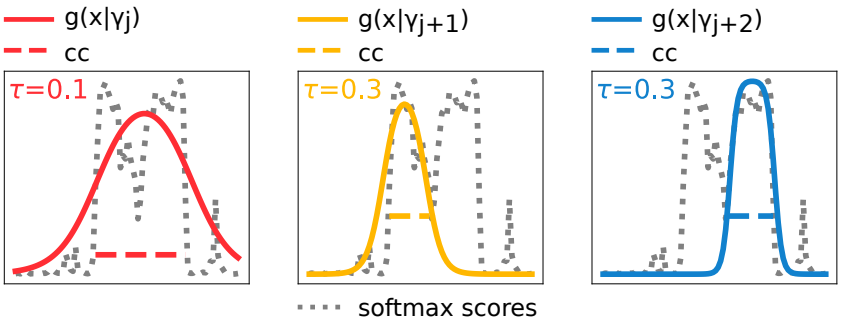

Figure 6.4: Producing multiple update proposals (red, yellow and blue lines) given softmax scores (grey dots) for a given class. "cc" denotes the connected components used to fit the softmax scores, obtained using different τ thresholds.

Figure 6.4, which illustrates softmax scores for a certain class (grey dots) obtained for a few frames. In this example, it is unclear whether the action is occurring throughout the two peaks (left-hand side plot), or whether the action is occurring over the frames corresponding to one of the peaks (centre and right-hand side plots).

To take this uncertainty into account, the plateau function is fitted to the softmax scores at multiple positions and temporal scales, as illustrated in Figure 6.4. The fitted plateaus represent different possible locations and durations of the action, which can be used to update the sampling distribution. More in detail, the fitting is done by first setting a threshold $\tau \in [0,1]$ over the softmax scores, and by finding all the connected components of consecutive frames with softmax score above τ. The plateau function is then fitted to each connected component. The parameters of each fitted plateau constitute an update proposal, denoted with $\gamma_j^v = (c_j^v, w_j^v, s_j^v)$. Given a sampling distribution $\beta_i^v = (c_i^v, w_i^v, s_i^v)$ for class y_i^v to be updated, the set of update proposals is thus:

$$\mathcal{Q}_i^v = \left\{\gamma_j^v \,:\, c_{i-1}^v < c_j^v < c_{i+1}^v\right\} \tag{6.4}$$

where j ranges over the number of plateaus fitted to the softmax scores of class y_i^v in video v. Note that the constraint $c_{i-1}^v < c_j^v < c_{i+1}^v$ enforces the order of the actions in v to be respected.

6.3.2 Selecting the Update Proposals

Each update proposal models a plateau covering a different segment of the video. In order to gauge which proposal is most likely located over more relevant frames, proposals are ranked and selected following a Curriculum Learning approach. Let us first define the score ρ for a given plateau function $g(x|\beta_i^v)$ by averaging the softmax scores enclosed by the plateau. Let $\mathcal{X}$ be the set of frames such that $\mathcal{X} = \{x \,:\, g(x|\beta_i^v) > 0.5\}$. The score ρ is then defined as follows:

$$\rho(\beta_i^v) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} P\left(y_i^v | x\right) \tag{6.5}$$

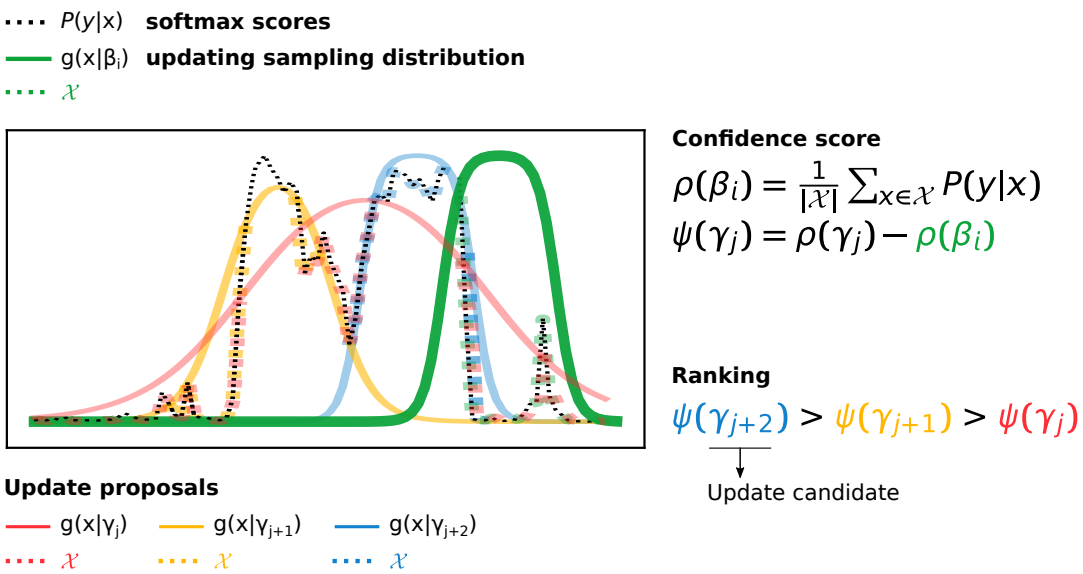

Figure 6.5: Selecting update candidates. The score ρ (Equation 6.5) of each plateau is first calculated, averaging the softmax scores x enclosed by each plateau (coloured dots, with colours indicating the corresponding plateau). The confidence score ψ (Equation 6.6) of each proposal is then calculated and used to select the update candidate. In this example γ_{j+2} (blue plateau) has the highest confidence score, and is thus selected as the updated candidate for β_i (green plateau).

The confidence ψ of each proposal $\gamma_j^v \in \mathcal{Q}_i^v$ is thus defined as:

$$\psi(\gamma_j^v) = \rho(\gamma_j^v) - \rho(\beta_i^v) \tag{6.6}$$

The underlying idea is to reward proposals whose plateaus contain frames that, on average, are scoring higher than those contained within the plateau to be updated, and thus are likely to be more relevant to the action. Accordingly, update proposals with non-positive confidence are discarded. For each β_i^v, the proposal $\widehat{\gamma_i^v}$ with highest confidence is selected:

$$\widehat{\gamma_i^v} = \underset{\gamma_j^v}{\arg\max}\, \psi(\gamma_j^v) \;:\; \gamma_j^v \in \mathcal{Q}_i^v \tag{6.7}$$

$\widehat{\gamma_i^v}$ is referred to as the update *candidate* for β_i^v. Figure 6.5 illustrates the candidate selection. The Figure depicts the same softmax scores and proposals showed earlier in Figure 6.4 (red, yellow and blue plateaus), as well as the sampling distribution to be updated (green plateau). The score ρ of each plateau is first calculated using Equation 6.5. The confidence score ψ (Equation 6.6) is then calculated for each proposal to choose the update candidate. In this example, proposal γ_{j+2} has the highest confidence score and is thus selected as the update candidate for the sampling distribution.

6.3.3 Ranking Update Candidates

The distribution update is guided by the classifier response. Given the weak temporal supervision, some update candidates might be erroneous. The ordering constraint imposed in Equation 6.4 alleviates this issue, ensuring that the order of the actions in each video is preserved. Additionally, to minimise noise during learning, the updates candidates selected for all sampling distributions in all videos are ranked according to their confidence score. Let:

$$\begin{aligned} \Gamma = \left(\widehat{\gamma_i^v}\ ,\ \forall i \in \{1..N_v\},\ \forall v \in \{1..M\}\right) \\ s.t.\ \psi(\Gamma_{t-1}) \geq \psi(\Gamma_t) \end{aligned} \tag{6.8}$$

be the sequence of all update candidates ordered according to their confidence score. The top R candidates in Γ are selected to update the corresponding sampling distributions:

$$\Gamma^R = \left(\Gamma\right)_{t=1}^{R}\ :\ R = z|\Gamma|,\ z \in [0,1] \tag{6.9}$$

This allows only the most confident updates to be applied, which ascribes some robustness to the update procedure.

6.3.4 Applying Updates and Measuring Convergence

Given the parameters of a distribution $\beta_i^v = (c_i^v, w_i^v, s_i^v)$ and the corresponding updated candidate $\widehat{\gamma_i^v} = \left(\widehat{c_i^v}, \widehat{w_i^v}, \widehat{s_i^v}\right) \in \Gamma^R$, the sampling distribution parameters are updated as follows:

$$\begin{aligned} c_i^v &= c_i^v - \lambda_c\left(c_i^v - \widehat{c_i^v}\right) \\ w_i^v &= w_i^v - \lambda_w\left(w_i^v - \widehat{w_i^v}\right) \\ s_i^v &= s_i^v - \lambda_s\left(s_i^v - \widehat{s_i^v}\right) \end{aligned} \tag{6.10}$$

where $\lambda_c, \lambda_w, \lambda_s$ are hyperparameters controlling the velocity of the update. The sampling distributions are updated iteratively until convergence, which is assessed by observing the average confidence of the update candidates. The confidence score measures the difference between the classifier response over the sampling distribution and the update candidate. An average confidence score approaching 0 thus entails that the update proposals are mostly aligned with the sampling distributions, i.e. the classifier is predicting the actions with steady confidence over the same video frames through different training iterations, which signals convergence.

Figure 6.6 illustrates a sampling distribution for the class “open fridge” being updated through different training iterations. The labelled timestamp and the corresponding initial sampling distribution (blue circle and dashed blue line, bottom plot) are not well aligned with the action, and are instead positioned before its occurrence. The initial predictions (dotted blue line, top plot) are naturally noisy. After a few iterations the classifier begins to learn the action, predicting it with more confidence

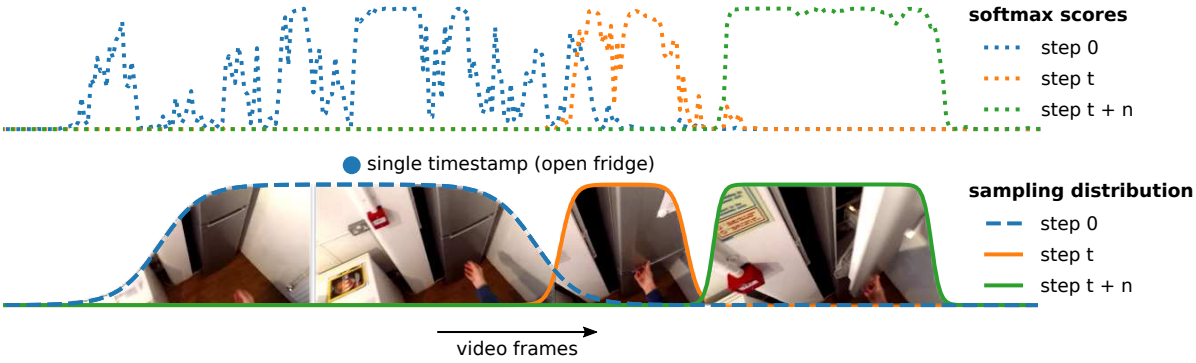

Figure 6.6: Updating the sampling distribution using the classifier response. Example from action "open fridge" in EPIC Kitchens [21]. Different colours indicate different training iterations.

over frames located after the initial plateau (dotted orange line, top plot). The updated distribution (solid orange line, bottom plot) encloses more relevant frames and reinforces the classifier. The final sampling distribution (solid green line, bottom plot) successfully aligns with the frames of the subject opening the fridge.

6.4 Experimental Setup

This Section provides details regarding the datasets and their annotations evaluated in this work. Implementation details are included as well, before discussing results in Section 6.5.

6.4.1 Datasets

The number of different actions (or classes) per video plays a crucial role when learning from untrimmed videos with weak temporal supervision. Intuitively, it becomes increasingly harder to learn discriminative features from videos displaying higher visual variability, when not knowing the temporal location and extent of the actions.

Figure 6.7 compares various common datasets [10, 20, 21, 48, 55, 66] for action recognition and localisation, based on the number of different actions per video in both the train (left-hand side) and test (right-hand side) sets. The Figure shows how the number of unique actions per video in these datasets ranges from an average of one action (Activity Net, THUMOS 14) to a maximum average of 34 actions (EPIC Kitchens).

Three datasets with increasing number of classes per video are evaluated in this work, namely THUMOS 14 [48], BEOID [20] and EPIC Kitchens [21]. These datasets were chosen to gauge the temporal cue offered by single timestamps and to assess the proposed training method under varying action density. EPIC Kitchens was selected also for its narration annotations, as detailed next. For THUMOS 14, experiments were conducted on the temporally labelled videos, which contain a total of 20 classes. For BEOID, the untrimmed videos were split in a random 80-20% proportion for training and testing. A subset of EPIC Kitchens was used for this work (the same subset was evaluated in

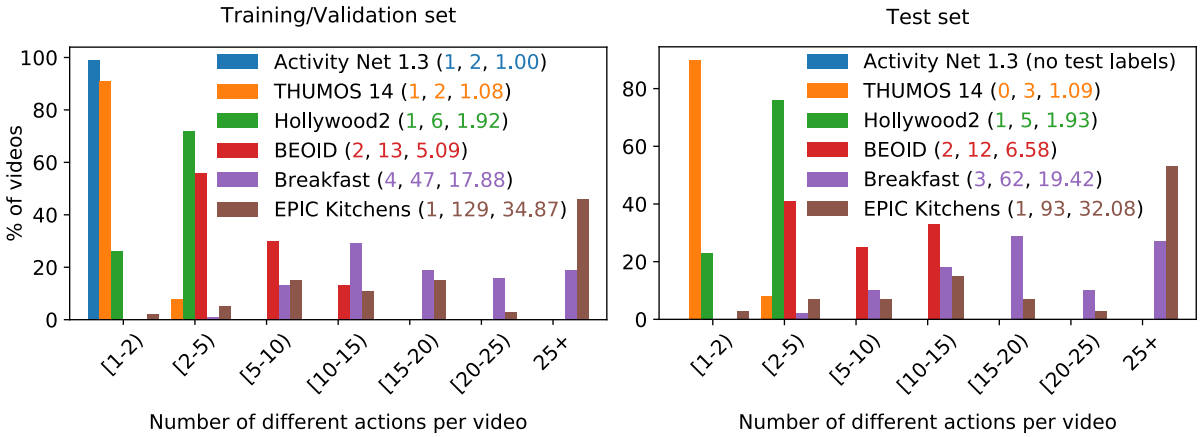

Figure 6.7: Different actions per video for various datasets. Numbers between parenthesis indicate (minimum, maximum, average) unique actions per video. For Activity Net [10] both train and validation sets were considered, while for THUMOS 14 [48] only the validation set was evaluated. The "s1" split and fine segmentation labels were used for Breakfast [55]. For EPIC Kitchens [21] only the videos in the used subset were considered.

Set	Dataset	Number of classes	Number of videos	Number of actions	Average video length	Average classes per video	Average instances per video
Train	THUMOS 14	20	200	3,003	208.90	1.08	15.01
	BEOID	34	46	594	61.31	5.09	12.91
	EPIC Kitchens	274	79	7,060	477.37	34.87	89.36
Test	THUMOS 14	20	210	3307	217.16	1.09	15.74
	BEOID	34	12	148	57.78	6.58	12.33
	EPIC Kitchens	274	26	1,949	399.62	32.08	74.96

Table 6.1: Information about the evaluated datasets. Average video length is in seconds.

Chapter 5 as well). With a total of 13.5 hours footage this subset amounts to 25% of full the dataset. Table 6.1 summarises various statistics of the chosen datasets. Despite considering a subset of the full dataset, EPIC Kitchens is by far the most challenging amongst the three datasets, given its very long videos containing many different actions.

6.4.2 Annotations

As discussed in Section 3.1.4, the EPIC Kitchens dataset was annotated using a two stages approach. Videos were firstly narrated to produce a rough temporal location of the actions, from which action boundaries were then refined using crowd sourcing. In this work the narration timestamps are used as the single timestamps for training. These timestamps are relative to the narration audio track and exhibit a challenging offset with respect to the actual occurrence of the actions in the videos. In fact, 55.8% of the narration timestamps are not contained in the corresponding labelled boundaries. For the timestamps outside the bounds, the maximum, average and standard deviation distance to the labelled boundaries are respectively 11.2, 1.4 and 1.6 seconds. The reader may refer to Figure 3.4 for a

visualisation of the distance between narrations timestamps and action segments for the full dataset. In the subset considered for these experiments, 26% of the narration timestamps were enclosed by a segment labelled for a distinct action. These numbers show the challenge involved in using EPIC Kitchens' narration timestamps as temporal supervision. This work offers the first attempt to train for fine-grained action recognition on EPIC Kitchens using only the narration timestamps.

THUMOS 14 and BEOID do not have single timestamp annotations. Rough single timestamps were thus simulated from the available temporal bounds, drawing each timestamp from the uniform distribution $[\sigma_i - 1\text{sec}, \epsilon_i + 1\text{sec}]$, where σ_i and ϵ_i denote the labelled start and end times of the i-th action in a video. The set of annotations presented so far are referred to with **TS**.

Another set of single timestamps for all the three datasets was additionally generated, where each timestamp is sampled using a normal distribution with mean equal to $\frac{\sigma_i+\epsilon_i}{2}$ and standard deviation of 1 second. This simulates the case where annotators are asked to precisely locate the actions providing one timestamp per action, assuming that they are likely to select a point close to the middle of the action. **TS in GT** denotes this second set of points.

6.4.3 Implementation Details

The training method abstracts the underlying classifier and image modality, and can be applied to any architecture that produces classification scores for single frames. In this work, experiments were conducted with the Inception architecture with Batch Normalisation (BN-Inception) [88] pre-trained on Kinetics [11], using TV-L1 optical flow stacks [121] of 5 images. For training, 5 stacks were sampled from each sampling distribution. Classification scores were aggregated using average consensus for back-propagation, as proposed in TSN [110], reviewed in Section 3.2.3. The model was trained end-to-end with Adam Optimiser [52], using a standard cross entropy loss. For testing, 10 stacks were uniformly sampled from the labelled bounds, taking the images' centre crop and averaging the frames scores for the final prediction.

The sampling distributions were initialised with $w = 45$ frames (1.5 seconds at 30 fps) and $s = 0.75$ for all datasets. To ensure adequate initialisation, the model was trained for 500 epochs before starting to update the sampling distributions. After the update started, the model was trained for 500 additional epochs. The initial 500 epochs were largely sufficient for the test error to converge in all experiments before the update started.

Training frames were ranked and selected following the Curriculum Learning paradigm described in Section 6.2. During the initial 500 epochs, before updating the sampling distributions, a fixed h (see Equation 6.3) was used. Once the update started, h was gradually increased until reaching $h = 1$, which corresponds to using all the sampled frames. Different h values were evaluated in the experiments to assess the impact of noisy training frames.

Update candidates were ranked using a fixed $z = 0.25$ (see Equation 6.9). To produce update proposals at multiple temporal scales, τ ranged in $\{0.1, 0.2, \ldots, 1\}$. Connected components shorter than

Dataset	CL h	Before update	After update
THUMOS 14	0.25	26.10	28.88
	0.50	32.69	55.15
	0.75	33.59	56.42
	1.00	63.41	63.53
BEOID	0.25	47.97	52.70
	0.50	71.62	83.11
	0.75	74.32	83.11
	1.00	64.86	70.27
EPIC Kitchens	0.25	20.47	22.83
	0.50	21.39	25.35
	0.75	20.73	23.86
	1.00	23.55	24.17

Table 6.2: Top-1 accuracy obtained with single timestamp supervision on the **TS** point set. CL h indicates the h parameter used for training the base model (see Equation 6.3).

15 frames were discarded. The update parameters $\lambda_c, \lambda_w, \lambda_s$ (see Equation 6.10) were respectively set to $0.5, 0.25, 0.25$, for all datasets. The sampling distributions were updated every 20 epochs.

6.5 Results

This Section analyses results obtained with single timestamps, comparing performance before and after updating the sampling distributions. The discussion begins with Table 6.2, which reports results obtained with the **TS** timestamps using Curriculum Learning (CL), with $h \in \{0.25, 0.50, 0.75\}$, as well as results obtained without CL, i.e. using all the sampled frames for training ($h = 1$). The evaluation metric used for all experiments is top-1 accuracy.

Results obtained after the update consistently outperform those obtained before the update, for all datasets and for all h values, showing the efficacy of the distributions update. For BEOID and EPIC, the CL strategy reduces the amount of noisy frames. On both datasets the best performance is in fact achieved with CL $h = 0.50$. This shows that learning a less noisy initial representation of the actions can lead to a more precise refinement of the sampling distributions, and in turn a higher accuracy can be achieved.

On THUMOS 14 the CL approach is less effective, with the best performance obtained when all frames are used in training. This is further analysed in Figure 6.8, which illustrates the percentage of selected and discarded frames that were enclosed by the labelled action boundaries (used only for plotting), before update. A higher percentage for the selected frames implies that the selected frames were overall more relevant than the discarded ones. For BEOID and EPIC Kitchens, a neat separation between the selected and discarded frames is noticeable. This shows that the CL strategy was effectively picking the most relevant frames within the plateaus during training.

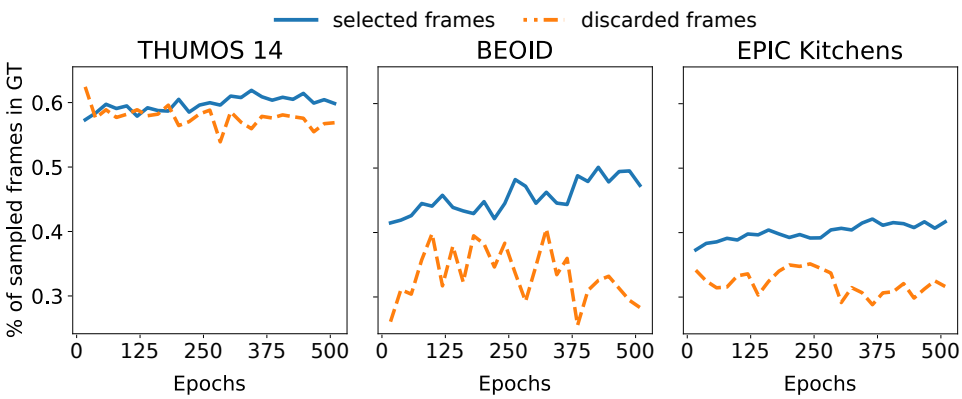

Figure 6.8: Percentage of sampled framed contained within labelled bounds, over training epochs. Plotted results obtained with CL $h = 0.50$ before update.

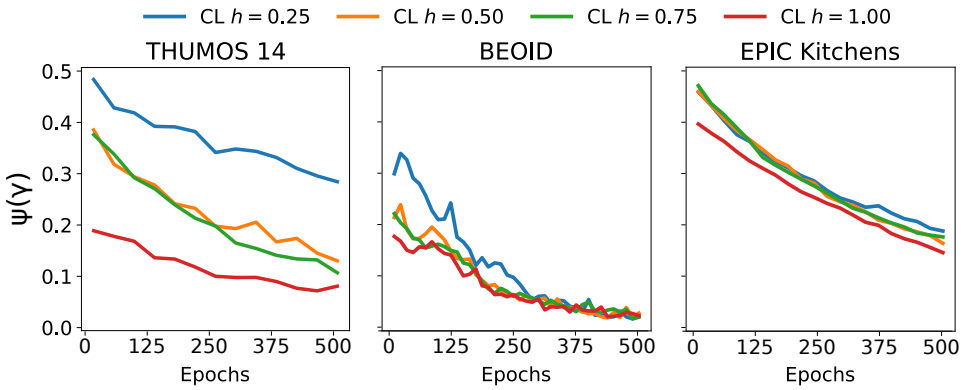

Figure 6.9: Average confidence of selected update candidates, as calculated in Equation 6.6, over training epochs.

In THUMOS 14 such distinct trend is instead not visible. This is possibly due the nature of the actions of the dataset. The 20 sports action classes present in THUMOS 14 are strongly characterised by their scene (e.g. football pitch, basketball court, etc.) or by short motion patterns, and require thus less temporal reasoning. In such scenarios, most frames sampled within the plateaus are likely to be equally relevant, thus ranking frames based on their prediction scores is less effective. This hypothesis is also confirmed by the marginal improvement obtained with the update with CL $h = 1$. As reported in Table 6.2, in this case accuracy before update is only 0.12% lower, which suggests that the initial sampling distributions, albeit roughly located, were sufficient to achieve high accuracy, given that accurate temporal labelling seems less important on this dataset. Note that a similar argument was also observed for UCF-101 [98] (whose classes form a superset of those of THUMOS 14) by Zhou et al. [123], as discussed in Section 3.2.3.

Figure 6.9 illustrates the average confidence of the selected update candidates during training, as calculated in Equation 6.6, for the three datasets and all CL h values. The average confidence decreases steadily in all cases, indicating the classifier's convergence.

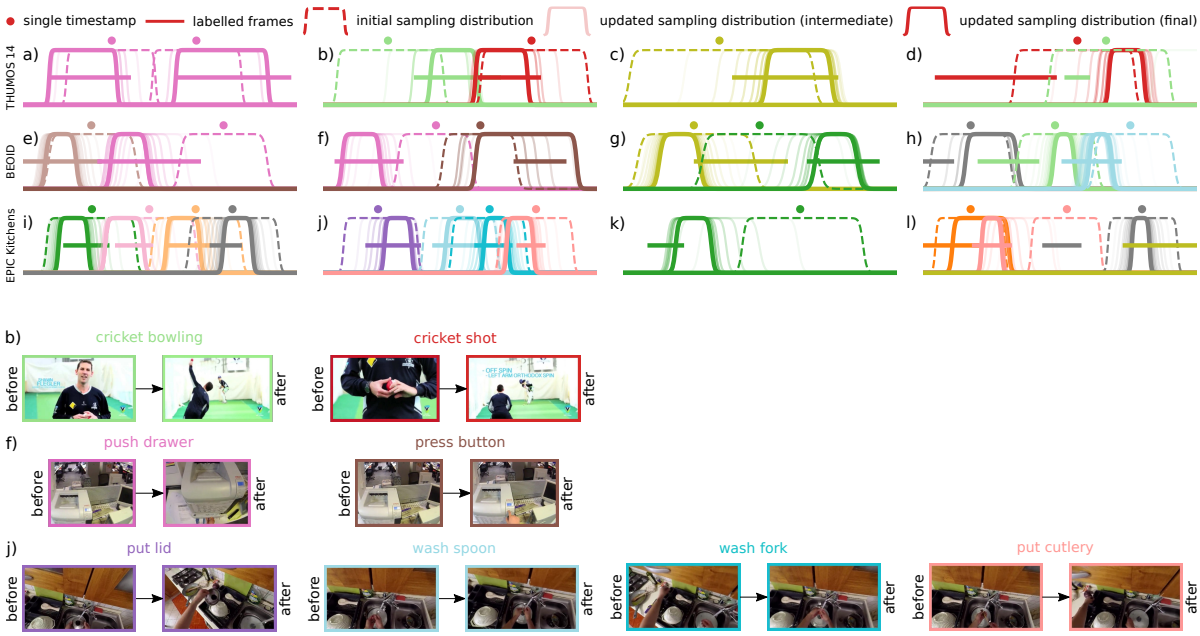

Figure 6.10: Qualitative results on THUMOS 14, BEOID and EPIC Kitchens, plotted from results obtained with CL $h = 0.50$. Top: each plot depicts a portion of an untrimmed video. Initial sampling distributions (dashed lines) are updated (washed-out lines) until convergence (solid lines). Different colours indicate different classes on a per-dataset basis. Labelled frames (horizontal lines) used only for plotting. Bottom: key-frames showed before and after update for plots *b, f* and *j*.

Qualitative results Figure 6.10 shows a few updates from each dataset. Each plot at the top of the Figure depicts a portion of an untrimmed video, illustrating the iterative update of the sampling distributions. The bottom part of the Figure displays key-frames before and after update for plots *b, f* and *j*. The examples are plotted from results obtained with CL $h = 0.50$ on the **TS** point set. The update method is able to successfully refine the sampling distributions even when the initial plateaus are considerably overlapping with other unrelated actions (plots *e, g, i, j*) or when the initial plateaus contain much background (plots *b, c, e, f, k*). Note how videos are dense of actions in BEOID and EPIC Kitchens which, together with the rough location of the single timestamps, illustrates the difficult conditions in which the method operates.

The Figure shows a few failure cases as well. In plots *g* (light green plateau) and *h* (grey plateau), the initial plateaus are pushed outside the relevant frames, due to poor initialisation. Indeed, in both cases the number of training actions was small (8 and 5 instances), with the single timestamps located outside the action in the majority of the cases. In plot *l*, the pink and grey initial plateaus were shifted with respect to the corresponding actions, reflecting the challenge EPIC Kitchens poses when using narration timestamps. While the update method managed to recover the correct location for the pink plateau, the grey plateau did not converge to the relevant frames.

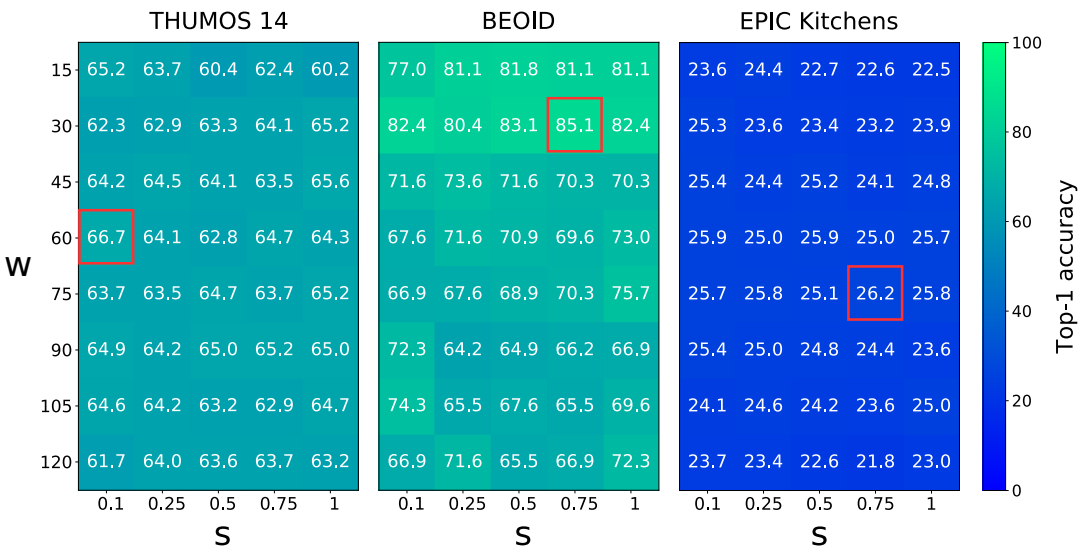

Figure 6.11: Top-1 accuracy obtained after update with different initial w and s parameters, with CL $h = 1$. Red boxes highlight best results.

Parameters initialisation The impact of the initial parameters w and s for the sampling distributions is assessed via a grid search. Figure 6.11 compares top-1 accuracy obtained after update with different (w, s) combinations, using CL $h = 1.00$. The method is robust to the initialisation of both w and s for the two large datasets (THUMOS 14 and EPIC Kitchens), i.e. similar performance is obtained for all parameters combinations. In BEOID accuracy fluctuates more, which is potentially due to the small size of the dataset (BEOID contains 594 actions in training, as reported in Table 6.1).

Note that the best results obtained with the grid search (highlighted with red boxes in the Figure) are slightly superior to those previously reported in Table 6.2. This is because when plateaus are optimally initialised they are better aligned with the actions, which leads to higher performance.

6.5.1 Comparing Levels of Supervision

As discussed in Section 2.2.1, a recent stream of works has successfully achieved results comparable to those obtained with full supervision, using only video-level labels. This type of temporal supervision, although being less expensive to collect, is not sufficient when dealing with videos containing multiple different actions. This is shown in the experiments presented in this Section, where different levels of temporal supervision are compared. More precisely, video-level labels are compared to single timestamps (both **TS** and **TS in GT** point sets) and full temporal boundaries.

The video-level baseline was obtained with Untrimmed Net [111] (reviewed in Section 2.2.1), which was trained using the same BN-Inception architecture and Kinetics pre-training used for the other baselines. For Untrimmed Net results are reported using RGB images, as these performed better than flow images in all experiments. For the fully supervised baseline, training stacks were sampled following TSN's approach, i.e. randomly within equally sized snippets.

Dataset	**ACV**	**Video-level**	**TS**	**TS in GT**	**Full**
THUMOS 14	1.08	64.92	66.68	64.53	67.10
BEOID	5.09	28.37	85.14	88.51	87.83
EPIC Kitchens	34.87	2.20	26.22	32.53	35.97

Table 6.3: Comparison between different levels of temporal supervision. All numbers expect ACV indicate top-1 accuracy. ACV reports the average number of classes (unique actions) per training video. Video-level results obtained with Untrimmed Net [111]. TS results refer to the accuracy obtained with the best initialisation (see Figure 6.11). Timestamp results are reported after update, with CL $h = 1.00$. Accuracy before update for TS was 64.74, 73.65 and 25.19 for THUMOS 14, BEOID and EPIC Kitchens. For TS in GT, accuracy before update was 64.74, 85.81 and 31.66.

Table 6.3 compares results obtained with the three levels of temporal supervision. When only one class of action is contained in the videos, as in THUMOS 14, Untrimmed Net's performance is notably close to the accuracy obtained with the fully supervised baseline. However, as the average number of different actions per video increases, it becomes increasingly harder for Untrimmed Net to achieve sufficient accuracy. In Untrimmed Net, as well as most other video-level approaches, when a video contains multiple different actions the label vector is L^1-normalised, thus all the present classes contribute equally to the cross entropy loss. As a consequence, without any temporal labels, it is very hard to train the model when a large number of classes are contained in a video, given that the standard cross entropy loss is not designed for multi-class instances.

Results obtained with single timestamps remain comparable to those obtained with full supervision, though requiring significantly less labelling effort. For THUMOS 14 and BEOID there is little difference between the **TS** and **TS in GT** point sets. In EPIC Kitchens a larger gap in performance is instead observed when comparing the narration timestamps **TS** to the fully supervised baseline. As mentioned before, the narration timestamps are relative to the live-audio track and are not well aligned with the actions. In fact, 56% timestamps are not contained in the corresponding annotated boundaries (with an average distance of 1.4 second to the labelled bounds), and 26% of the timestamps were even enclosed by a segment labelled for a distinct action. The particularly rough location of the timestamps, combined with the action density of the dataset lead to a challenging initialisation for the method, which affects the classifier's performance. Nevertheless, when simulating the initial timestamps from the labelled bounds (**TS in GT**), the method achieves a higher accuracy comparable to that obtained with the fully supervised baseline.

* * *

To summarise, the presented results show that single timestamps supervision constitutes a good compromise between accuracy and annotation effort. The method is robust to parameters initialisation and is able to attain high performance in settings entailing high action densities. The update method operates on roughly located timestamps and is able to correct misplaced annotations, converging

Baseline	**mAP@0.1**	**mAP@0.2**	**mAP@0.3**	**mAP@0.4**	**mAP@0.5**
This work (TS)	24.3	19.9	15.9	12.5	9.0
This work (Full)	26.7	22.5	18.5	14.3	11.1
Untrimmed Net [111]	44.4	37.7	28.2	21.1	13.7

Table 6.4: Localisation results on THUMOS 14 at different IOUs.

towards relevant action frames. Given that the update is based on the classifier response, an overall reasonable location of the timestamps is required to achieve high accuracy.

6.6 Future Direction: Localisation with Single Timestamps

This work focused on action classification using single timestamp supervision. Action classification and localisation are two often intertwined tasks, thus it would be natural to wonder if the proposed method could be used to temporally locate actions. Localising actions using a model designed for classification is however a sub-optimal approach. This is well known in the literature. For example, Liu et al. [63] argue that "identifying one fragment of an action is sufficient for classification but not for segment-level localisation".

This Section reports localisation results obtained on THUMOS 14 with BN-Inception (trained with single timestamp and full supervision), as well as Untrimmed Net, which is optimised for localisation. Results are presented in Table 6.4, which reports mean average precision (mAP) at different IOUs for the three baselines. The action temporal segments were produced following Untrimmed Net's testing pipeline. Classification scores were first obtained for each frame in the untrimmed test videos, using RGB and optical flow images. The scores obtained with the two modalities were then late-fused to produce a single class prediction. The action segments were finally obtained by applying a threshold and a smoothing-dilation operation on the classification scores.

As expected, the localisation performance obtained with the TS and full baselines are poor. While TS performs comparably to full supervision, even the fully supervised model is inferior to Untrimmed Net, proving that localising actions with a model optimised for classification is insufficient. Indeed, Untrimmed Net is able to outperform the other baselines despite using video-level labels. This is because the framework employs an attention module to refine the classification scores, which improves localisation accuracy.

This work could be extended to localisation by supervising a framework optimised for such task. For example, the plateau function could also be used to model the background between actions in untrimmed videos, either in a self-supervised manner (e.g. by automatically placing background plateaus between actions) or by means of single timestamps labelling idle sections. Learning the background can improve localisation accuracy, especially if used in combination with self-attention modules, as done in Untrimmed Nets.

6.7 Conclusion

This work proposed single timestamp supervision for training a classifier from untrimmed videos. The Chapter showed that this novel type of supervision appears to be a good compromise between labelling effort and performance. Indeed, results showed that with single timestamps and the proposed training algorithm it is possible to achieve high accuracy. Importantly, results are comparable to those obtained with full supervision (with an accuracy gap between 1% and 3%), despite using weaker and less expensive annotations, even in complex scenarios involving a high density of actions.

The proposed approach does not rely on a precise location of the timestamps, and is able to refine the temporal supervision used during training. This makes single timestamps an appealing labelling approach, especially considering the rapid growth of video datasets. Results also showed that the sampling distributions converge to discriminative action locations. The method is robust to the initialisation of the sampling distributions and is able to correct misplaced annotations, provided that the timestamps are overall reasonably aligned with the actions. In fact, a sensible location of the timestamps is required to attain high performances. This was observed in EPIC Kitchens, where results obtained using the challenging narration timestamps proved to be further from the fully supervised baseline.

A few future directions could be taken from this work. Firstly, as discussed above, single timestamps could be utilised for other tasks such as temporal localisation. Secondly, the optimal distribution parameters could be learnt in an end-to-end fashion, which could alleviate issues related to the coarseness of the annotated timestamps.

Lastly, it would be interesting to assess whether the plateau function introduced in this Chapter could be used to mitigate the robustness issues seen in Chapters 4 and 5. As discussed there, frame predictions are typically averaged uniformly to estimate the action contained in a segment. This entails that all the frames within the bounds are assigned equal importance, which is potentially problematic when boundaries include irrelevant frames. Classification scores could be aggregated using the plateau function. More precisely, frames predictions could be combined using a weighted average, where the weight of each frame is given by a plateau function fitted to the scores. This could be a more robust way of accumulating predictions in scenarios where temporal boundaries are uncertain.

Chapter 7

Conclusion

This Thesis concerned temporal labelling for action recognition in videos. The first step was to assess how action boundaries are perceived by different people, and how this affects classification algorithms. Almost surprisingly, this matter has received little attention within the action recognition community. In fact, dataset creators typically do not discuss how actions are segmented in the published datasets, and little work in the literature has focussed on how temporal labelling affects the performance of a classifier.

The first contribution of this Thesis was to explore this less trodden territory. As we saw in Chapter 4, annotators are likely to disagree when identifying the temporal scope of an action. As action boundaries vary, relevant and irrelevant frames are included or excluded, thus the ability of a classifier of learning and detecting actions may be influenced. The foremost finding of this study was that accurate temporal labelling matters: performance worsens when boundaries are imprecise, and accordingly improves when bounds are made consistent. Considering that temporal labelling is arbitrary and subjective, it was important to dedicate some attention to how actions are annotated, given that recognition frameworks are crafted and compared using such labels. The Rubicon Boundaries concluded that study, with an effort to precisely describe the temporal span of an action in order to alleviate labelling ambiguity.

Robustness to temporal bounds variations remains an open problem. In fact, Chapter 5 showed that recent state-of-the-art models (TSN [110] and I3D [11]) suffer from noisy frames introduced into action segments. As discussed during the course of this Thesis, this is mainly because fully supervised approaches are usually not designed taking bounds uncertainty into account, which is perhaps yet another sign that temporal labelling is an overlooked issue. Models aiming to address boundaries ambiguity constitute a direction for future research.

Action boundaries are not only arbitrary, but also expensive to annotate. Video datasets are expanding rapidly, thus there is an intrinsic need for scaling the labelling process up. This Thesis proposed a novel labelling approach, i.e. to annotate single timestamps roughly aligned with actions in untrimmed videos. This is associated with a learning paradigm that starts from single timestamps and iteratively refines the temporal supervision until convergence.

As we saw in Chapter 6, using single timestamps and the proposed training algorithm it is possible to achieve performances comparable to those obtained with full temporal supervision. Compared to other weakly supervised approaches, which require less labelling effort but are challenged by complex video dynamics, single timestamps appear to be the sought compromise between annotation endeavour and accuracy. Additionally, single timestamps also alleviate ambiguity, since annotators do not have to decide when the action starts and ends, but only to mark one frame within or close to the action. Extending single timestamp supervision to other tasks such as video localisation concludes the list of possible future directions suggested here. The following summarises the main questions posed in this Thesis, discussing the offered answers and their limitations, as well as giving some insight into what could be explored next.

* * *

This Thesis started questioning the arbitrariness and subjectivity of temporal boundaries for action recognition. We saw that delimiting actions in videos is a subjective task, and that accurate temporal labelling is important. While the study shed some light onto this overlooked issue, the Thesis has not explored how models can be designed to be robust.

As a rule, supervised approaches assume temporal boundaries to be precise when learning from videos. Moving from such assumption is the next step towards robust models that can learn from inaccurate annotations. That is, to devise approaches that are able to discard noisy training data, rather than simply trust the given supervision and force an incorrect representation.

This aligns with the current state of the play. Weakly supervised approaches optimise self-attention modules to exploit weak temporal cues, i.e. are able to learn discriminative features without temporal boundaries. This introduces the following question: considering the recent spread and success of weakly supervised models, as well as the scale of current datasets, are full temporal boundaries a necessary and viable labelling?

The single timestamps approach proposed in this Thesis is an attempt to answer this question. The proposed training algorithm is able to achieve high performances, however a good overall location of the timestamps is required. Alleviating this limitation is the next goal. Timestamps and training supervision could also be refined combining visual inputs with other modalities, such as audio. Indeed, sound is a precious cue that has been receiving more attention lately. For example, action timestamps could be automatically positioned or adjusted by analysing auditory data, which can signal the presence of an object or the occurrence of an action, as showed in recent research [3, 51].

Before finishing this Thesis, the next Section provides an overview of how far action recognition has gone and what are the next objectives in video understanding from a broader perspective.

Action recognition: where are we and what's next?

Action recognition approaches have long been based on object recognition models or other systems borrowed from the image domain. Indeed, at the beginning of deep learning, action recognition in videos was accomplished by applying image classification models on video frames. Two-stream architectures combine RGB and optical flow modalities, either by means of simple late fusion (e.g. [95]) or by actively learning correspondences between appearance and motion features (e.g. [29]). This was a success. In fact, the two-stream architecture was the first CNN framework that marked the decline of hand-crafted solutions and the beginning of the deep learning era. Popular models such as TSN [110] enlarge the receptive field of video frameworks, allowing CNNs to learn from longer portions of the action. Nevertheless, the underlying approach and the backbone networks of most current models remain fundamentally image classification ones. CNNs employing 3D convolutions are a natural way to model time. After a long time of disuse due to the difficulties in training 3D architectures, Inflated 3D CNNs [11] are now a successful reality. Still, 3D CNNs can currently capture only short range dynamics, due to their heavy training requirements. Recent research (e.g. TSM [61], R(2+1)D [104] and S3D [116]) attempts to alleviate such burden, which seems to be a promising direction.

The datasets commonly used for nearly a decade (e.g. UCF-101 [98] and HMDB-51 [54]) have been a precious benchmark to develop new architectures and push video understanding forward. Current CNNs have become powerful enough to achieve very high accuracy on such datasets, where actions are strongly characterised by their static appearance and can be sufficiently recognised with little temporal modelling. However, as datasets are now becoming more complex, with videos showing more intricate dynamics (e.g. the Something-Something dataset [39]), the image classification based paradigm is reaching its limitations.

Approaches able to perform sophisticated temporal reasoning are the next goal. Understanding how entities interact and evolve over time is key to build intelligent recognition systems that go beyond image classification. Indeed, recent works are already following this direction (e.g. TRN [123]). At the same time, there is a need for large video datasets capturing unconstrained actions in the wild. That means unscripted actions in native and open environments, as well as actions that go beyond simple object interactions. The Something-Something and EPIC Kitchens [21] datasets are a step forward towards this direction.

The sheer amount of data available nowadays and in the days to come sets another ambition, i.e. to deviate from fully supervised approaches and embrace weak and self supervision. This has received a good degree of attention lately, which led to models being able to learn actions from untrimmed videos, for example. Eventually, this is perhaps the ultimate objective of computer vision, to reach a time when machines learn themselves how to see and what to look for, a time when machines can actually be called intelligent. Fortunately, there is still a long way to go.

...the future is untrimmed...

BIBLIOGRAPHY

[1] TwentyBN-jester dataset: a hand gesture dataset.
`https://20bn.com/datasets/jester`, 2017.

[2] H. Alwassel, F. Caba Heilbron, V. Escorcia, and B. Ghanem.
Diagnosing error in temporal action detectors.
In *European Conference on Computer Vision (ECCV)*, 2018.

[3] R. Arandjelovic and A. Zisserman.
Look, listen and learn.
In *International Conference on Computer Vision (ICCV)*, 2017.

[4] H. Bay, T. Tuytelaars, and L. Van Gool.
Surf: Speeded up robust features.
In *European Conference on Computer Vision (ECCV)*, 2006.

[5] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei.
What's the point: Semantic segmentation with point supervision.
In *European Conference on Computer Vision (ECCV)*, 2016.

[6] Y. Bengio, J. Louradour, R. Collobert, and J. Weston.
Curriculum learning.
In *International Conference on Machine Learning (ICML)*, 2009.

[7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri.
Actions as space-time shapes.
In *International Conference on Computer Vision (ICCV)*, 2005.

[8] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic.
Weakly supervised action labeling in videos under ordering constraints.
In *European Conference on Computer Vision (ECCV)*, 2014.

[9] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid.
Weakly-supervised alignment of video with text.
In *International Conference on Computer Vision (ICCV)*, 2015.

[10] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles.
ActivityNet: A large-scale video benchmark for human activity understanding.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] J. Carreira and A. Zisserman.
Quo vadis, action recognition? a new model and the kinetics dataset.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] J. Carreira, E. Noland, C. Hillier, and A. Zisserman.
A short note on the kinetics-700 human action dataset.
In *arXiv preprint arXiv:1907.06987*, 2019.

[13] C.-Y. Chang, D.-A. Huang, Y. Sui, L. Fei-Fei, and J. C. Niebles.
D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] G. Chéron, J.-B. Alayrac, I. Laptev, and C. Schmid.
A flexible model for training action localization with varying levels of supervision.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[15] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville.
Recurrent batch normalization.
In *arXiv preprint arXiv:1603.09025*, 2016.

[16] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar.
Movie/script: Alignment and parsing of video and text transcription.
In *European Conference on Computer Vision (ECCV)*, 2008.

[17] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray.
Visual categorization with bags of keypoints.
In *European Conference on Computer Vision Workshops (ECCVW)*, 2004.

[18] N. Dalal and B. Triggs.
Histograms of oriented gradients for human detection.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[19] N. Dalal, B. Triggs, and C. Schmid.
Human detection using oriented histograms of flow and appearance.
In *European Conference on Computer Vision (ECCV)*, 2006.

[20] 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.

In *British Machine Vision Conference (BMVC)*, 2014.

[21] 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.

[22] 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.
In *Robotics Institute*, 2008.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei.
Imagenet: A large-scale hierarchical image database.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[24] L. Ding and C. Xu.
Weakly-supervised action segmentation with iterative soft boundary assignment.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman.
The pascal visual object classes (VOC) challenge.
In *International Journal of Computer Vision (IJCV)*, 2010.

[27] A. Fathi and J. Rehg.
Modeling actions through state changes.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[28] A. Fathi, Y. Li, and J. Rehg.
Learning to recognize daily actions using gaze.
In *European Conference on Computer Vision (ECCV)*, 2012.

[29] C. Feichtenhofer, A. Pinz, and A. Zisserman.
Convolutional two-stream network fusion for video action recognition.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] B. Fernando, C. T. Y. Chet, and H. Bilen.
Weakly supervised gaussian networks for action detection.
In *arXiv preprint arXiv:1904.07774*, 2019.

[31] M. A. Fischler and R. C. Bolles.
Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.
In *Communications of the ACM*, 1981.

[32] A. Gaidon, Z. Harchaoui, and C. Schmid.
Temporal localization of actions with actoms.
In *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

[33] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann.
Devnet: A deep event network for multimedia event detection and evidence recounting.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia.
Turn tap: Temporal unit regression network for temporal action proposals.
In *International Conference on Computer Vision (ICCV)*, 2017.

[35] J. Gao, Z. Yang, and R. Nevatia.
Cascaded boundary regression for temporal action detection.
In *British Machine Vision Conference (BMVC)*, 2017.

[36] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin.
Convolutional sequence to sequence learning.
In *International Conference on Machine Learning*, 2017.

[37] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell.
Actionvlad: Learning spatio-temporal aggregation for action classification.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] P. M. Gollwitzer.
Action phases and mind-sets.
In *Handbook of motivation and cognition: Foundations of social behavior*, 1990.

[39] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al.
The" something something" video database for learning and evaluating visual common sense.
In *International Conference on Computer Vision (ICCV)*, 2017.

[40] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber.
Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.
In *International Conference on Machine learning (ICML)*. ACM, 2006.

[41] K. He, X. Zhang, S. Ren, and J. Sun.
Deep residual learning for image recognition.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] D. Huang, L. Fei-Fei, and J. C. Niebles.
Connectionist temporal modeling for weakly supervised action labeling.
In *European Conference on Computer Vision (ECCV)*, 2016.

[43] N. Hussein, E. Gavves, and A. W. Smeulders.
Timeception for complex action recognition.
In *Conference on Computer Vision and Pattern Recognition (CVPR*, 2019.

[44] N. Inkawhich, M. Inkawhich, Y. Chen, and H. Li.
Adversarial attacks for optical flow-based action recognition classifiers.
In *arXiv preprint arXiv:1811.11875*, 2018.

[45] S. Ioffe and C. Szegedy.
Batch normalization: Accelerating deep network training by reducing internal covariate shift.
In *International Conference on Machine Learning (ICML)*, 2015.

[46] T. S. Jaakkola, D. Haussler, et al.
Exploiting generative models in discriminative classifiers.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 1999.

[47] L. Jiang, X. Ma, S. Chen, J. Bailey, and Y.-G. Jiang.
Black-box adversarial attacks on video recognition models.
In *arXiv preprint arXiv:1904.05181*, 2019.

[48] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar.
THUMOS challenge: Action recognition with a large number of classes.
`http://crcv.ucf.edu/THUMOS14/`, 2014.

[49] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei.
Large-scale video classification with convolutional neural networks.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[50] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al.
The kinetics human action video dataset.
In *arXiv preprint arXiv:1705.06950*, 2017.

[51] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen.
Epic-fusion: Audio-visual temporal binding for egocentric action recognition.
In *International Conference on Computer Vision (ICCV)*, 2019.

[52] D. P. Kingma and J. Ba.
Adam: A method for stochastic optimization.
In *International Conference on Learning Representations (ICLR)*, 2014.

[53] B. Korbar, D. Tran, and L. Torresani.
Scsampler: Sampling salient clips from video for efficient action recognition.
In *arXiv preprint arXiv:1904.04289*, 2019.

[54] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre.
HMDB: a large video database for human motion recognition.
In *International Conference on Computer Vision (ICCV)*, 2011.

[55] H. Kuehne, A. Arslan, and T. Serre.
The language of actions: Recovering the syntax and semantics of goal-directed human activities.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[56] H. Kuehne, A. Richard, and J. Gall.
Weakly supervised learning of actions from transcripts.
In *Computer Vision and Image Understanding (CVIU)*, 2017.

[57] I. S. Kwak, D. Kriegman, and K. Branson.
Detecting the starting frame of actions in video.
In *arXiv preprint arXiv:1906.03340*, 2019.

[58] T. A. Lampert, A. Stumpf, and P. Gançarski.
An empirical study into annotator agreement, ground truth estimation, and algorithm evaluation.
In *Image Processing*, 2016.

[59] I. Laptev and T. Lindeberg.
Local descriptors for spatio-temporal recognition.
In *International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[60] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld.
Learning realistic human actions from movies.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[61] J. Lin, C. Gan, and S. Han.
Temporal shift module for efficient video understanding.
In *International Conference on Computer Vision (ICCV)*, 2019.

[62] T. Lin, X. Zhao, and Z. Shou.

Single shot temporal action detection.
In *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017.

[63] D. Liu, T. Jiang, and Y. Wang.
Completeness modeling and context separation for weakly supervised temporal action localization.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[64] J. Liu, J. Luo, and M. Shah.
Recognizing realistic actions from videos "in the wild".
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[65] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang.
Multi-granularity generator for temporal action proposal.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[66] M. Marszałek, I. Laptev, and C. Schmid.
Actions in context.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[67] M. Marszalek, I. Laptev, and C. Schmid.
Actions in context.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[68] P. Matikainen, M. Hebert, and R. Sukthankar.
Trajectons: Action recognition through the motion analysis of tracked features.
In *International Conference on Computer Vision Workshops (ICCVW)*, 2009.

[69] R. Messing, C. Pal, and H. Kautz.
Activity recognition using the velocity histories of tracked keypoints.
In *International Conference on Computer Vision (ICCV)*, 2009.

[70] P. Mettes, J. C. van Gemert, and C. G. Snoek.
Spot on: Action localization from pointly-supervised proposals.
In *European Conference on Computer Vision (ECCV)*, 2016.

[71] P. Mettes, C. G. Snoek, and S.-F. Chang.
Localizing actions from video labels and pseudo-annotations.
In *arXiv preprint arXiv:1707.09143*, 2017.

[72] G. Miller.
Wordnet: a lexical database for english.
In *Communications of the ACM*, 1995.

[73] D. Moltisanti, M. Wray, W. Mayol-Cuevas, and D. Damen.
Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video.
In *International Conference on Computer Vision (ICCV)*, 2017.

[74] D. Moltisanti, S. Fidler, and D. Damen.
Action recognition from single timestamp supervision in untrimmed videos.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[75] P. Nguyen, T. Liu, G. Prasad, and B. Han.
Weakly supervised action localization by sparse temporal pooling network.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[76] J. C. Niebles, C.-W. Chen, and L. Fei-Fei.
Modeling temporal structure of decomposable motion segments for activity classification.
In *European Conference on Computer Vision (ECCV)*, 2010.

[77] NIST.
Trecvid multimedia event detection.
`https://www.nist.gov/itl/iad/mig/med-2014-evaluation`, 2014.

[78] S. Paul, S. Roy, and A. K. Roy-Chowdhury.
W-TALC: Weakly-supervised temporal activity localization and classification.
In *European Conference on Computer Vision (ECCV)*, 2018.

[79] H. Pirsiavash and D. Ramanan.
Detecting activities of daily living in first-person camera views.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[80] K. K. Reddy and M. Shah.
Recognizing 50 human action categories of web videos.
In *Machine Vision and Applications*, 2013.

[81] A. Richard, H. Kuehne, and J. Gall.
Weakly supervised action learning with RNN based fine-to-coarse modeling.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[82] A. Richard, H. Kuehne, A. Iqbal, and J. Gall.
Neuralnetwork-viterbi: A framework for weakly supervised video learning.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[83] M. Rodriguez, J. Ahmed, and M. Shah.
Action mach a spatio-temporal maximum average correlation height filter for action recognition.

In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[84] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek.
Image classification with the fisher vector: Theory and practice.
In *International Journal of Computer Vision (IJCV)*, 2013.

[85] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap.
A simple neural network module for relational reasoning.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[86] S. Satkin and M. Hebert.
Modeling the temporal extent of actions.
In *European Conference on Computer Vision (ECCV)*, 2010.

[87] K. Schindler and L. Van Gool.
Action snippets: How many frames does human action recognition require?
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[88] I. Sergey and S. Christian.
Accelerating deep network training by reducing internal covariate shift.
In *CoRR*, 2015.

[89] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun.
Overfeat: Integrated recognition, localization and detection using convolutional networks.
In *International Conference on Learning Representations (ICLR)*, 2013.

[90] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang.
Autoloc: Weakly supervised temporal action localization in untrimmed videos.
In *European Conference on Computer Vision (ECCV)*, 2018.

[91] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta.
Hollywood in homes: Crowdsourcing data collection for activity understanding.
In *European Conference on Computer Vision (ECCV)*, 2016.

[92] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta.
Asynchronous temporal fields for action recognition.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[93] G. A. Sigurdsson, O. Russakovsky, and A. Gupta.
What actions are needed for understanding human actions in videos?
In *International Conference on Computer Vision (ICCV)*, 2017.

[94] 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.

In *arXiv preprint arXiv:1804.09626*, 2018.

[95] K. Simonyan and A. Zisserman.
Two-stream convolutional networks for action recognition in videos.
In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[96] K. Simonyan and A. Zisserman.
Very deep convolutional networks for large-scale image recognition.
In *CoRR*, 2014.

[97] K. K. Singh and Y. J. Lee.
Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization.
In *International Conference on Computer Vision (ICCV)*, 2017.

[98] K. Soomro, A. R. Zamir, and M. Shah.
Ucf101: A dataset of 101 human actions classes from videos in the wild.
In *arXiv preprint arXiv:1212.0402*, 2012.

[99] E. Spriggs, F. De La Torre, and M. Hebert.
Temporal segmentation and activity classification from first-person sensing.
In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2009.

[100] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus.
Intriguing properties of neural networks.
In *International Conference on Learning Representations*, 2013.

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

[102] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman.
Learning from noisy labels by regularized estimation of annotator confusion.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[103] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri.
Learning spatiotemporal features with 3d convolutional networks.
In *International Conference on Computer Vision (ICCV)*, 2015.

[104] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri.
A closer look at spatiotemporal convolutions for action recognition.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[105] J. C. Van Gemert, M. Jain, E. Gati, C. G. Snoek, et al.
Apt: Action localization proposals from dense trajectories.
In *British Machine Vision Conference (BMVC)*, 2015.

[106] G. Varol, I. Laptev, and C. Schmid.
Long-term temporal convolutions for action recognition.
In *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[107] H. Wang and C. Schmid.
Action recognition with improved trajectories.
In *International Conference on Computer Vision (ICCV)*, 2013.

[108] H. Wang and C. Schmid.
Action recognition with improved trajectories.
In *International Conference on Computer Vision (ICCV)*, 2013.

[109] H. Wang, A. Kläser, C. Schmid, and C. Liu.
Action recognition by dense trajectories.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[110] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool.
Temporal segment networks: towards good practices for deep action recognition.
In *European Conference on Computer Vision (ECCV)*, 2016.

[111] L. Wang, Y. Xiong, D. Lin, and L. Van Gool.
UntrimmedNets for weakly supervised action recognition and detection.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[112] X. Wang, A. Farhadi, and A. Gupta.
Actions transformations.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[113] X. Wei, J. Zhu, and H. Su.
Sparse adversarial perturbations for videos.
In *arXiv preprint arXiv:1803.02536*, 2018.

[114] M. Wray, D. Moltisanti, W. Mayol-Cuevas, and D. Damen.
SEMBED: Semantic embedding of egocentric action videos.
In *European Conference on Computer Vision Workshops (ECCVW)*, 2016.

[115] M. Wu and M. Kwiatkowska.
Robustness guarantees for deep neural networks on videos.
In *arXiv preprint arXiv:1907.00098*, 2019.

[116] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy.
Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification.
In *European Conference on Computer Vision (ECCV)*, 2018.

[117] H. Xu, A. Das, and K. Saenko.
R-c3d: Region convolutional 3d network for temporal activity detection.
In *International Conference on Computer Vision (ICCV)*, 2017.

[118] X. Yang and Y. Tian.
Effective 3D action recognition using eigenjoints.
In *Journal of Visual Communication and Image Representation*, 2014.

[119] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei.
Every moment counts: Dense detailed labeling of actions in complex videos.
In *International Journal of Computer Vision (IJCV)*, 2018.

[120] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici.
Beyond short snippets: Deep networks for video classification.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[121] C. Zach, T. Pock, and H. Bischof.
A duality based approach for realtime tv-l 1 optical flow.
In *Joint Pattern Recognition Symposium*. Springer, 2007.

[122] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin.
Temporal action detection with structured segment networks.
In *International Conference on Computer Vision (ICCV)*, 2017.

[123] B. Zhou, A. Andonian, A. Oliva, and A. Torralba.
Temporal relational reasoning in videos.
In *European Conference on Computer Vision (ECCV)*, 2018.