Universidad de Alcalá

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Teoría de la Señal y Comunicaciones

Online Analysis of Streaming Videos for Human Action Understanding

Dissertation by

Marcos Baptista Ríos

For the degree of

Doctor of Philosophy in Information and Communication Technologies

Supervisor

Roberto Javier López Sastre

2020

Agradecimientos

Bueno, pues ya he terminado! Me he pasado la última pantalla!

Con esto, cierro una etapa. Una etapa larga. Una etapa que no habría sido posible comenzar sin las personas que me dieron la oportunidad de meterme en todo esto.

Para empezar, Roberto. Mi tutor. Gracias por darme la oportunidad de hacer el doctorado. Estricto y metódico. Pero resolutivo. Gracias por hacerme entender el juego de la investigación. Es verdad que mi segundero marcha infinitamente más lento que el tuyo, pero siempre has ido empujando.

Sin embargo, la etapa no empezó con el doctorado. Empezó mucho antes. De hecho, de no ser así, no estaría escribiendo esto hoy. Y por ello, tengo que darles las gracias a Marta Marrón y Cristina Losada. Fueron siempre un gran apoyo.

Esta etapa, no sólo es universitaria. La siento también como una etapa personal. Durante muchos años, se pasa mucho tiempo con gente que ve lo mismo que tú, sufre lo mismo que tú y lucha como tú. Buscando el mismo objetivo. Y siguiendo, inevitablemente, el mismo camino. Y eso, crea una unión que, independientemente de lo que depare el futuro, es bastante fuerte. Vosotros sois José, Carol, David “El Jefe”, Casillas, Letis y Juanma.

Y todavía me quedan, en un plano aún más personal, dos agradecimientos muy importantes. Uno es para mis padres. Gracias por estar siempre al lado, de manera incondicional. Por el apoyo y el esfuerzo para que yo haga lo que quiera y para que tenga los medios para ello.

Y el otro, para Irene. Gracias por darme mi tiempo y mi espacio para poder conseguir esto. Por acompañarme siempre, junto con los Pupus y mi Lolita, formando el mejor equipo que se puede tener.

Abstract

This thesis is part of the PREPEATE research project, conducted in the GRAM research group of the University of Alcalá, which aims to develop an assistive robotic platform based on advanced artificial intelligent techniques. The robot will analyse the human behaviour by processing live video content. To this end, this work tackles the topics of Temporal Action Proposals (TAP) and Online Action Detection (OAD).

For the first problem, state-of-the-art approaches address it following an offline and supervised setting, which implies having access to the whole video beforehand and a fully annotated dataset. In the robotic platform scenario, the video must be processed as it is collected and labels are not always available. For this reason, an unsupervised online solution is introduced. It generates action proposals through a Support Vector Classifier used as a clustering module to identify action candidates. To refine them it employs rank pooling over feature dynamics as a filter, removing those proposals that belong to the background of the video. An experimental evaluation is conducted on ActivityNet and THUMOS14 datasets, achieving more than 41% and 26% of the recall performance of the best supervised models, respectively.

Regarding OAD, unlike traditional offline action detection approaches, where the evaluation metrics are clear and well established, the OAD setting presents very few works and no consensus on the evaluation protocols to be used. This thesis proposes to rethink the OAD scenario, clearly defining the problem itself and the main characteristics that the models which are considered online must comply with. Additionally, the thesis also introduces a novel metric: the Instantaneous Accuracy (IA), which exhibits an online nature and solves most of the limitations of the previous metrics. A thorough experimental evaluation on 3 challenging datasets is conducted, where the performance of various baseline methods is compared to that of the state of the art. Results confirm the problems of the previous evaluation protocols, and suggest that an IA-based protocol is more adequate to the online scenario.

Resumen

Esta tesis forma parte del proyecto de investigación PREPEATE, llevado a cabo en el grupo de investigación GRAM de la Universidad de Alcalá. En él se pretende desarrollar una plataforma de robótica asistencial basada en técnicas avanzadas de inteligencia artificial. El robot estudiará el comportamiento humano mediante el análisis de vídeo. Para ello, la tesis aborda los problemas de Propuestas Temporales de Acciones (PTA) y Detección de Acciones Online (DAO) en vídeos.

En cuanto al primer problema, las soluciones más recientes lo abordan con un proceso "offline" y supervisado, que implica tener el vídeo con anterioridad y datos completamente anotados. En el escenario definido por el robot, el vídeo se procesa según se recoge y las anotaciones no siempre están disponibles. Por ello, se presenta una solución "online" y no supervisada. Ésta genera propuestas de acción mediante un "clustering" basado en Máquinas de Vectores Soporte, y utiliza "Rank Pooling" sobre las dinámicas de las características para eliminar propuestas que no pertenezcan a un segmento de acción. El modelo se evalúa en las bases de datos Activitynet and THUMOS14, alcanzando el 41% y el 26%, respectivamente, del rendimiento de los mejores modelos supervisados.

En cuanto a DAO, a diferencia de los enfoques offline de detección de acciones, donde las métricas están bien establecidas, el problema de DAO presenta pocos trabajos y apenas consenso sobre los protocolos de evaluación. Esta tesis propone repensar el escenario de DAO, definiéndolo claramente y detallando las principales características que deben cumplir los modelos "online". Se introduce también una nueva métrica llamada *Instantaneous Accuracy* (IA), la cual es "online" y resuelve las limitaciones de las métricas anteriores. La tesis realiza una evaluación exhaustiva en 3 conjuntos de datos y se compara el rendimiento de varios métodos de referencia con el de los del estado del arte. Los resultados confirman los problemas de los protocolos de evaluación anteriores y sugieren que un protocolo basado en la IA es más adecuado.

Contents

List of Figures

List of Tables

Acronyms

AA Action Anticipation.

AD Action Detection.

AR Action Recognition.

cAP calibrated Average Precision.

IA Instantaneous Accuracy.

IoU Intersection over Union.

maIA mean average Instantaneous Accuracy.

mAP mean Average Precison.

OAD Online Action Detection.

ODAS Online Detection of Action Start.

OffAD Offline Action Detection.

SGD Stochastic Gradient Descent.

STAL Spatio-temporal Action Localisation.

STAP Spatio-temporal Action Proposals.

SVC Support Vector Classifier.

SVM Support Vector Machine.

SVR Support Vector Regression.

TAD Temporal Action Detection.

TAL Temporal Action Localisation.

TAP Temporal Action Proposals.

tIoU temporal Intersection over Union.

wIA weighted Instantaneous Accuracy.

Chapter 1

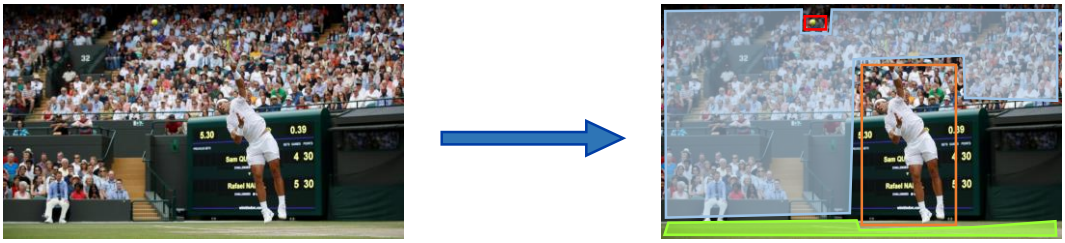
Introduction

Providing computers with sufficient intelligence to analyse and understand the information collected by cameras is something that human beings have been pursuing for many years. All the techniques and algorithms that have been developed for this purpose are grouped under the name of Computer Vision.

In today's world, video content is very much a part of people's daily lives, whether it is for entertainment or information, such as social networks or TV, or to assist in the workplace, such as surveillance cameras in retail stores or in the public transport to control capacity. And for this reason, video is the source of data that best allows studying how people interact with the world around them.

However, how complex is this task? If one looks at Figure 1.1, how long does the brain take to extract and arrange the information from the image in Figure 1.1.(a) as in Figure 1.1.(b)? At a glance, one could probably spot the ball and localise the person to determine that someone is playing tennis. Also at the same glance, the court and the grandstand can be easily differentiated. Finally, if the image is analysed a bit further, with the pose and the body shape of the player, it could also be determined that the ball has just been hit and the player has just served to start the point.

Even though it seems to be a simple and quick task for the human brain, each one of the above details extracted from the image corresponds to some of the most relevant research topics in Computer Vision: object detection, action localisation, semantic segmentation and temporal action detection. To understand especially the complexity of the last topic, while a computer vision system needs to analyse the video, the human brain could easily guess, in some situations, the stage of the action by observing



(a) (b)

Figure 1.1: **The complexity of Computer Vision.** How long does the brain take to arrange the information from image **(a)** as in **(b)**? At a glance, one can immediately determine that someone is playing tennis by just spotting the ball and localising the person. As quickly, the court and the grandstand can be easily differentiated. If the image is analysed a bit further, it could also be determined that the player has just served to start the point. Even though the process is very simple, each detail extracted from the image corresponds to some of the most relevant research topics in Computer Vision: object detection (spotting the ball), action detection (localising the person which is playing), semantic segmentation (differentiating court from grandstand) and temporal action localisation (determining the player has just served).

only a single frame and, in this way, guess also the duration of it. And it is this very problem what this thesis focuses on. More specifically, on localising those parts of a certain video which can contain a human action of interest.

The described problem is characterised by *untrimmed videos*, a type of videos in which action segments coexist with irrelevant ones, *i.e.* background segments, appearing the latter more frequently. However, this thesis explores a perspective that is not very common, yet can be very useful in certain real scenarios (see Section 1.1): the *online* analysis of untrimmed videos. In this particular setting, a certain system has to analyse the video as it grows, and make predictions at the instant of execution. Such predictions can only be based on current and previous processed video content, but never on future content, as it is not known. In contrast to traditional offline methods that have access to the whole video beforehand, within the online video processing scenario, actions must be discovered only with partial observations. This limitation, together with the fact that both the segments containing actions and those containing background can share a great appearance, being sometimes difficult to distinguish, makes the problem a real challenge.

1.1 Motivation

The thesis presented here is part of the PREPEATE research project, conducted in the GRAM research group of the University of Alcalá. The aim of this project is to develop a robotic platform to assist people with disabilities by applying advanced artificial intelligent techniques. To this end, the robot must interact with the environment, *i.e.* process the information, understand it and react consequently. In the case of this project, information refers to the visual content collected with a camera. Therefore, the first two tasks are addressed with Computer Vision algorithms.

Most of the task of understanding the environment involves human behaviour analysis. The Computer Vision community has opened several research lines to address this. Some examples of these lines are Action Detection (AD), which consists in localising in an image the area where the action is being carried out, or Action Recognition (AR), which refers to the task of determining the action that a person is performing in a video. And, as Figure 1.2 clearly shows, the interest in the topic is rapidly growing each year.

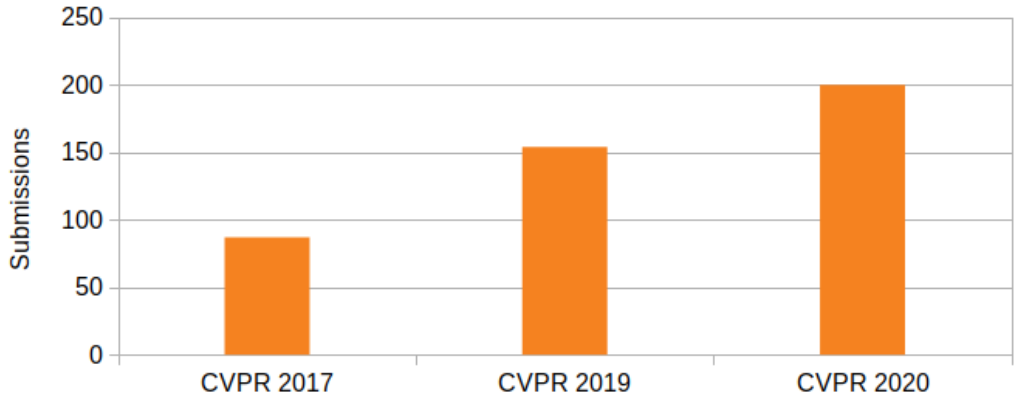


Figure 1.2: **Human Behaviour Analysis papers submitted to IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) in the last years.** The number of submissions are estimated from the official statistics of CVPR'17; CVPR'19 and CVPR'20. Those of the year 2018 and of previous years to 2017 were not easily available. It can be observed that the topic is gaining interest.

In the last five years, the problem of localising those parts of a certain video that can contain an action, known as Temporal Action Localisation (TAL), has become of great interest. Proof of this is the considerable amount of contributions of the scientific community in the topic, *e.g.* (Heilbron et al., 2016; Shou et al., 2016; Escorcia

et al., 2016; Buch et al., 2017; Gao et al., 2017; Chao et al., 2018). Typically, all the proposed solutions consist of a Temporal Action Proposals (TAP) generator and an action classifier module. While the former is responsible for generating video segments of high action probability (also known as action proposals), the latter is in charge of predicting the category of the action. In all works, the TAP generator is applied to the entire video at once, in such a way that the video has always to end before the TAP module can be executed. This property becomes a limitation for certain real applications (*e.g.* the the robotic platform previously described) in which the intelligent system requires to know the content of the video at each instant, *i.e.* in an online fashion, to make its decisions. Offering the action segments to the classifier as soon as possible would, in these cases, be much more appropriate. However, this option has never been explored. The thesis that is presented here tackles this research line and proposes a TAP approach that is capable of generating the action segments as they occur in the video and, thus, accelerating the TAL task. Additionally, in contrast to all state-of-the-art approaches, the one designed here is unsupervised, which means that labels are not needed during training.

Besides, it has also been recently opened by De Geest et al. (2016) the problem of recognising at each instant the class of action that it is being performed in the video, which is called Online Action Detection (OAD). Different from the TAL topic, the philosophy of Online Action Detection (OAD) does fit with those cases in which the information is needed at every moment. Nevertheless, the problem has been little explored and there are some conditions and properties that require a more concrete definition, as this work will show. This thesis also seeks to solve this by thoroughly studying the topic to redefine in a clearer way the conditions of the scenario that is considered in the topic, as well as establishing the properties which all proposed Online Action Detection (OAD) solutions must show. Additionally, as the performance of the solutions is measured with a metric that is inherited from Offline Action Detection (OffAD) topics, this work also introduces a new metric that is completely in line with the conditions and properties of the Online Action Detection setting.

Beyond the research project that motivates this thesis, the ultimate goal of the work presented in this document is to be useful for many other real-world applications. In this sense, the current work can be applied to a wide variety of situations. Some of these, but not all, are listed below:

- **Human-robot interaction.** In a scenario where robots and people have to

interact with each other, it is necessary to provide the former with intelligent systems based on what is proposed in this thesis so that they are able to analyse human behaviour online and react accordingly as quickly as possible.

- **Medical applications.** For example, as an aid for the nursing staff of a hospital. With a camera in the room, the patient's behaviour can be automatically monitored. If any action is out of the ordinary, the system would alert the staff as soon as possible.

- **Supervision of manufacturing processes.** Some manufacturing processes require a high level of concentration from the operator, which can sometimes lead to failures when fatigue or stress appear. In such situations, an intelligent camera could monitor the operator's actions and stop the process chain when it discovers that the operator is starting to behave mistakenly, so that the failure is not propagated.

- **Video surveillance.** The methods that are proposed in this thesis could help the surveillance personnel, for example in retail stores, by raising an alarm when a person is behaving anomalously.

- **Live sport statistics.** With a system trained on the actions that players typically perform in a certain sport, a lot of statistics could be automatically collected. In those sports that allow it, these online statistics could be accessible by coaches so that they can adapt the strategy if necessary.

- **New era video games.** In this new era of video games to come, not only will virtual reality glasses be used, but also an exoskeleton. The player is thus, totally immersed in the virtual world. With the online models that are proposed, the game could analyse the actions that the player is carrying out so that to act in the virtual world in consequence.

1.2 Contributions of the Thesis

The contributions derived from this thesis are specified below:

- A deep study of the literature of the TAP, TAL and OAD topics. Thanks to this, the reader can have a clear picture of the current state of the art in these topics

and contextualise the objectives of this thesis as well as the solutions that are chosen.

- A novel online, unsupervised method for the TAP problem. Unlike all other state-of-the-art approaches, the proposed method generates action proposals in an online fashion, *i.e.* as soon as they appear in the video. On top of that, it is totally unsupervised, which means that during training it does not have access to any kind of label from the dataset.

- A solid definition of the OAD problem. The one made by De Geest et al. (2016) is revisited to clarify it and new key conditions are added to improve its completeness.

- A new set of conditions for OAD methods. The way in which methods should deal with the untrimmed streaming videos that are considered in the OAD topic is ambiguous. Hence, this thesis establishes a set of new conditions that all OAD methods must comply with.

- A new evaluation protocol with a novel metric for OAD. The metric that is used to measure the performance of the methods, known as calibrated Average Precision (cAP), is a small variation of the mean Average Precison (mAP) metric, which is used in several Offline Action Detection topics. Therefore, it is not appropriate for the problem. To solve this, a new evaluation protocol is offered along with a novel metric called Instantaneous Accuracy (IA), which is capable of measuring the performance of methods at the time of execution.

1.3 Thesis Structure

The structure of the document is as follows:

- Chapter 2 puts this thesis in context by reviewing all the topics in the field of video action understanding that are closely related to those which the work here presented contributes to. These topics are Temporal Action Proposals (TAP), Temporal Action Localisation (TAL) and Online Action Detection (OAD).

- Chapter 3 introduces the new online and unsupervised algorithm for the Temporal Action Proposals (TAP) problem. It starts with a comparison between

the common offline supervised approaches and the new online unsupervised method. Afterwards, all the modules that comprise the approach are formally described. At last, the novel solution is tested with the typical experimental set-up for TAP.

- Chapter 4 focuses on the Online Action Detection problem. The first part of the study reveals the main weaknesses of the current state of the topic and clearly redefines its properties as well as establishes the set of conditions that any OAD method should comply with. The second part of the chapter contains the new evaluation protocol with the novel metric. All the findings in both parts of the chapter are proven and evaluated in the final experiments section.

- Chapter 5 summarises the conclusions of this thesis. Additionally, it also shows several future directions that could be explored after concluding the present work.

Chapter 2

Related Work

In the extensive field of video understanding, the problem of temporally localising actions in untrimmed videos by giving their labels and temporal boundaries has gained massive interest in the recent years.

More specifically, the offline perspective of the problem, named Temporal Action Localisation (TAL), has received a great increase in contributions. Currently, the best and most common way to solve this problem involves two stages: Temporal Action Proposals (TAP) generation and action classification. Since the former is crucial for the whole problem, it is considered a task itself. Regarding the online version, Online Action Detection (OAD), it is recently becoming of interest in the field, due to its many real-world applications, such as human-robot interaction or autonomous vehicles (see Section 1.1).

The thesis presented in this work contributes to the topics of TAP and OAD. For this reason, this section summarises the recent contributions on the concerned topics, as well as those from the field of video understanding that are closely related.

2.1 The Story of Video Action Understanding

This section puts into context the topics which this thesis contributes to. Not only does it describe them, but it also situates them historically in order to know how and when they were originated and to which other topics they are related.

As the purpose is to understand the main motivations that the authors have found to make the topics evolve to the state they are in nowadays, the section does not delve

into the details of the works that are mentioned.

2.1.1 From Images to Videos, from Objects to Actions

The Object Recognition topic has traditionally considered two problems: Image Classification and Object Detection. While the former refers to inferring the label of the object that a certain image contains, the latter consists in locating the region of the image where the object is. In the past, the typical schema to address Object Detection was based on applying exhaustively an object classifier around all the image through windows of different size (sliding window technique) in order to find regions with high probability of containing the object.

Derived from the Object world, the field of video action understanding considers the counterpart problems of Image Classification and Object Detection: Video Classification and Temporal Action Localisation (TAL), respectively. The former, also named as Action Recognition, consists in inferring the label of what is happening in a certain trimmed video (*e.g.* Wang et al. (2011); Simonyan & Zisserman (2014); Tran et al. (2015); Carreira & Zisserman (2017)). The latter seeks to find those temporal segments within a video where the action is happening.

At the beginning, the TAL task was interpreted in two different ways. On the one hand, works such as those proposed by Junsong Yuan et al. (2009), Tran & Yuan (2011) or Oikonomopoulos et al. (2011), considered the problem as an extension of Object Detection to videos and thus, they proposed to exhaustively process them with a 3D version of the sliding window (the sliding sub-volume) used to detect objects in images in order to spatio-temporally locate the action, *i.e.* to get a bounding box indicating in each frame where the action is being performed and find its extension in time. In the majority of their experiments, the videos were segmented to the length of the action, so that in all frames the action was present. On the other hand, several authors have addressed the problem as a retrieval task where given an action query, the method had to search throughout all the video for the segments in which the action was being performed. In this case, the videos involved were untrimmed, *i.e.* irrelevant information could appear at some parts of the video. A classical work representing this interpretation of TAL is the one proposed by Gaidon et al. (2013). Later on, this interpretation evolved to just temporally locate all the instances of a set of actions of a certain dataset without using any query, which is the way the task is understood today (*e.g.* Oneata et al. (2014); Shou et al. (2016); Zeng et al. (2019)). The

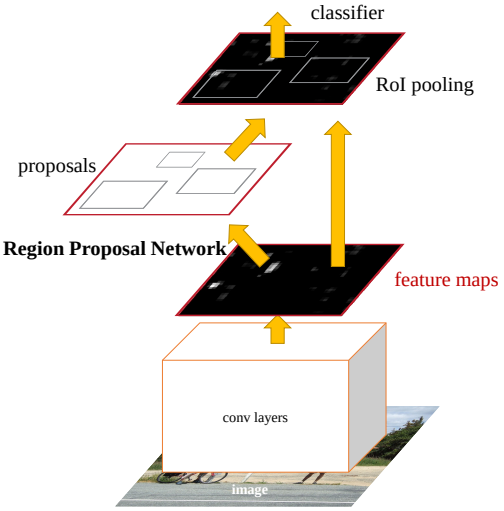
work written by Jiang et al. (2013) is a good survey to find more information about the state of the art on all these tasks back in those years.

These interpretations led to having two well-differentiated problems, though they share the goal of inferring the temporal location of the action in the video. The first one was coined as Spatio-temporal Action Localisation (STAL) and the second kept the name of Temporal Action Localisation (TAL). Following the ideas applied for Object Detection until that time, the researchers explored, for both topics, methods based on the sliding window technique, *i.e.* the video was exhaustively analysed with 3D or 2D windows of different lengths to spatio-temporally or temporally locate the action, depending on the problem that was being solved.

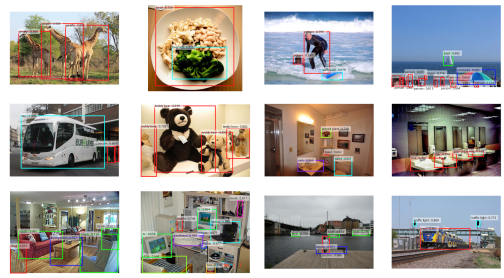
Coming back to the Object Detection problem, the sliding window based approaches were computationally very heavy. So, the Computer Vision community started using the concept of object proposals, which are defined as regions in the image with high probability of containing an object, regardless of its class. These object proposals were in turn passed to a classifier to infer the class of the object they might contain. Due to the good results of proposal-based solutions, this configuration is still utilised nowadays. In fact, object proposals became a problem itself. One of the most representative methods of this proposal-based style is the Faster R-CNN proposed by Ren et al. (2015) (see Figure 2.1). For a deeper understanding of the Object Detection problem, and more specifically of the works that utilise object proposals, the reader is encouraged to review the work of Hosang et al. (2016).

Driven by the success of the proposals concept for detecting objects in still images, some authors proposed to extend it to the STAL problem. The boost in performance that this concept offered made it become also a well-known task within the field of video understanding called Spatio-temporal Action Proposals (STAP). Here, an action proposal corresponds to a temporal series of bounding boxes, known as tubes or action tubes, which can potentially surround a human action. This definition of proposals is more challenging since, in contrast to the object detection version which only seeks to solve the *where*, it also requires to answer *when*. Some traditional examples are the works done by Jain et al. (2014); Gkioxari & Malik (2015); Yu & Yuan (2015); van Gemert et al. (2015). Far from being solved, nowadays STAP is still an open problem and of much interest because of its usefulness for STAL, such as the works of Zhu et al. (2017) and Escorcia et al. (2020).

Nevertheless, in 2016, Heilbron et al. (2016) realised of two important facts: i)



(a) Faster R-CNN architecture.

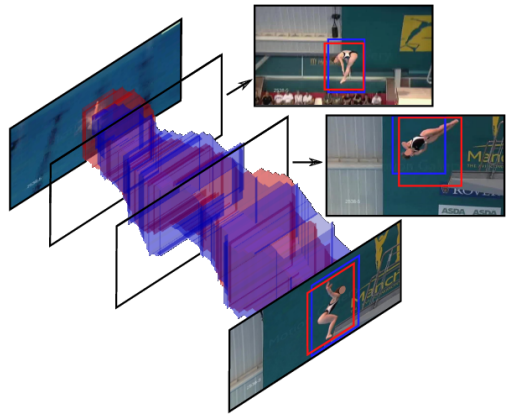


(b) Faster R-CNN output example.

Figure 2.1: The Faster R-CNN method is one of the most representative works for Object Detection. Instead of analysing the image with the sliding window technique, it applies an algorithm called Region Proposal Network (RPN) that finds the bounding boxes and their associated probability of containing an object, regardless of its class. Then, this bounding boxes are passed to a classifier to infer the class. Figures from Ren et al. (2015).

proposals were increasing the performance of both Object Detection and STAL methods, yet they had never been applied to TAL; and ii) although STAP approaches could have a good performance in the spatial dimension, they were not able to achieve high recall when trying to *temporally* localise the actions in untrimmed videos. The time dimension for them was not strictly addressed, but remained an optional output derived from the construction of the action tubes. Knowing this, Heilbron et al. (2016) raised the following question: *Why not to detect proposals only in time?* And therefore, they proposed the new topic of Temporal Action Proposals (TAP), in which methods were only designed to find, in long untrimmed videos, those segments that can contain with high probability a human action. These segments are the temporal action proposals, and, different from those of STAP, they are only defined by their initial and ending times, or so called boundaries. With the years, solving this problem has become a crucial step to also solve the TAL task. The vast majority of TAL approaches utilise a pipeline whose first step is a good method to obtain temporal action proposals that will, in turn, be passed to a classifier to know the label of the action (*e.g.* Gao et al. (2018); Lin et al. (2018); Xu et al. (2020))

(a) Temporal Action Proposals.



(b) Spatio-temporal Action Proposals.

Figure 2.2: **Comparison between Spatio-temporal and Temporal Action Proposals.** Temporal Action Proposals is the problem of localising, in an untrimmed video, those temporal segments that contain the action. In this case, proposals are defined by their initial and ending times. Spatio-temporal Action Proposals not only seeks to find the action segments but also the spatial location or region of each frame in which the action is being performed. Therefore, proposals are a series of bounding boxes which can potentially surround a human action. Both problems are crucial steps for the Temporal Action Localisation and Spatio-temporal Action Localisation tasks, respectively, where the proposals that are generated will then be passed to a classifier to infer the class of the action that is happening. Figures from Heilbron et al. (2016) and van Gemert et al. (2015), respectively.

2.1.2 The Sooner the Better. The Online Paradigm for Video Analysis

By the year 2010, parallel to the research lines previously described, but with less attention, some Computer Vision scientists realised that there was a different research question that could be of much interest for certain real-world scenarios: *How long does it take for a method to recognise the action being performed on a video?* To interpret it in a bit more technical way, if one considers a trimmed video containing only the concerned action, and the restriction of analysing the video from the beginning to the end as if future frames were not known, *How many frames would the method require to be able to recognise the action?* or *How much of the action would the method require to see to recognise the action?*

With the aim of answering these questions, some authors, such as Ryoo (2011); Yu et al. (2012); Cao et al. (2013); Lan et al. (2014); Kong et al. (2014), proposed an experimental setting with trimmed videos in which to evaluate the amount of video that a given method needs to discover the action that is taking place. This was coined as Action Prediction.

However, Hoai & De la Torre (2012) stated that despite the significant relevance of Action Prediction, the problem of finding events or actions in videos was far from solved. They added new conditions to have a more realistic setting than that of Action Prediction. First, events in videos are important situations that happen between irrelevant parts of the video, *i.e.* the background. Hence, experiments should be done with *untrimmed* videos. Second, since an event is defined by its initial and ending times, these instants are what methods should find. This new setting described the problem of Early Action Detection. Given all these conditions, they developed a system that first analyses the video and then decides the initial and ending times of the event over the information that has been extracted. In their experimental set-up, they used videos where only one kind of event was taking place and with few background.

It is not only until 2016 that De Geest et al. (2016) claimed that what had been done up to that date was not realistic enough. In the real world, actions happen between long periods of background, or irrelevant situations. In a video, these situations have a large variability and cover most part of it. Therefore, a proper dataset should show these characteristics. Regarding actions, their initial and ending times must be detected online, and not after the video has been analysed. These facts motivated them to define the problem of Online Action Detection (OAD) as another task in the field of Action Detection in videos. As described by the authors, given an untrimmed streaming video whose end is unknown, OAD is the problem of detecting actions that are happening *now* without any information from the future (because it is not known).

Nowadays, although the topic has few contributions, it is gaining some attention due to its applicability to many interesting fields that need systems that can interact with their environment and make decisions *now*, such as robotics, video surveillance or medical applications, among others.

2.1.3 Concluding Remarks

As described in this section, the concept of *proposals* has transformed the way in which several important Computer Vision tasks are undertaken, such as Object Detection or Spatio-temporal Action Localisation. The same has happened in the Temporal Action Localisation task since they have been used, in such a way that generating proposals has become a problem itself, known as Temporal Action Proposals. And it is this particular problem to which this thesis contributes to. As it is explained in

the following sections, TAP has been only addressed by fully supervised approaches. However, this thesis proposes to explore an unsupervised setting.

Regarding the early stage of the Online Action Detection problem, this thesis also seeks to solve the ambiguities this topic still has in its definition, such as the way in which the background is treated or the metric that is used to evaluate the performance of the methods.

2.2 Temporal Action Proposals

Along nearly all the literature published up to date, two styles of solutions for TAP can be identified according to their initialisation:

(a) Segment-based methods (Heilbron et al. (2016); Shou et al. (2016); Escorcia et al. (2016); Buch et al. (2017); Gao et al. (2017); Chao et al. (2018)).

(b) Actionness-based methods (Yuan et al. (2017); Zhao et al. (2017); Lin et al. (2018); Lin et al. (2019)).

Besides, it is important to note that all these previous works are trained with strong supervision. They are, typically, composed of several modules that must be trained independently and making use of all the available labels (*e.g.* action categories, temporal annotations, etc.). Some of them, like the method designed by Shou et al. (2016), go a bit further and not only they use the temporal boundaries to differentiate action and background frames in training but also implement a loss function that depends on the Intersection over Union between ground truth and generated temporal action segments. Only two works have currently tried to tackle the problem with less supervised methods: Ji et al. (2019), with their semi-supervised training procedure; and Khatir et al. (2019), who proposed some extensive experiments with an online clustering method and a rank-pooling based Fernando et al. (2016) filtering.

2.2.1 Initialisation Philosophies for Temporal Action Proposals

This section presents the different initialisation schemas that have been used for the TAP task, along with their most representative works.

Segment-based initialisation

Methods based on this kind of initialisation generate thousands of varied-length overlapped segments, which can then be confirmed or discarded as valid proposals. Once the proposals are confirmed, their starting and ending times (boundaries) might be refined.

A commonly used technique to generate these candidate segments is the sliding window strategy in which videos are densely sampled several times with temporal windows of different lengths. After this sampling, each video is transformed into a bag of overlapped candidate segments. The works by Heilbron et al. (2016) and Shou et al. (2016) initialise their TAP methods with this bag of segments approach. Concretely, the former explores the possibility of learning a dictionary over the features of ground truth action segments. At test time, all candidate segments generated by the sliding window are tried to be reconstructed through the learned dictionary. Those segments with low reconstruction error are confirmed as actual proposals, being the rest discarded. On the other hand, Shou et al. (2016) rely on a multi-stage C3D (Tran et al. (2015)) network for determining whether a candidate segment is an action proposal or not. Figure 2.3 shows the whole pipeline of their work for Temporal Action Detection. First, they propose to uniformly sample frame video segments of different lengths. Then, they solve the TAP problem in the first stage by feeding with the frame segment a C3D network that has been re-purposed to discriminate between action and background.

Different from the sliding window approach, TURN-TAP authors (Gao et al. (2017)) decided to generate segments through a grouping features strategy. They first decompose the video in short contiguous windows (*e.g.* 16 or 32 frames) called *units*. Afterwards, they extract the features of these *units* and pool sets of those that are contiguous to form clips. In order to represent different time scales, they vary the amount of unit features that are pooled. The collected clips are considered candidate segments for action proposals. Finally, inspired by the Faster R-CNN architecture (Ren et al. (2015)), they train two siblings fully connected layers which are fed with the features of the candidate segments. The first one determines whether they belong to an action and the second regresses two offsets to refine their boundaries. In the work following TURN-TAP, CBR (Gao et al. (2017a)), the authors use a cascaded approach over the TURN-TAP module with two stages: 1) all segment-based candidates are analysed; and 2) those segments that have been confirmed as a proposal are fed

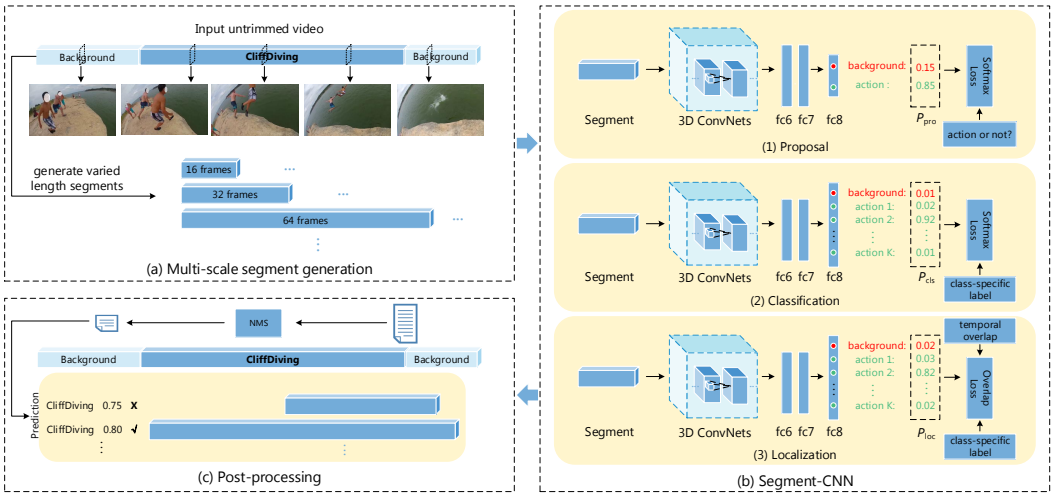


Figure 2.3: **Multi-stage CNN for Temporal Action Localisation.** It consists of 3 stages in which a C3D network is trained for different purposes: 1) Proposals, 2) Classification and 3) Localisation. While the first stage generates the proposals, the second and third classify them and refine their initial and ending times, respectively. Figure from Shou et al. (2016).

back to the system to refine their scores and boundaries. TURN-TAP and CBR are not the only ones which have taken inspiration from the Faster R-CNN. Xu et al. (2017) came up with R-C3D, a method that is a re-purposed version of the Faster R-CNN. Nevertheless, this framework presents several issues when it is directly applied to the new problem. Chao et al. (2018) noted these issues and designed TAL-Net to overcome them. First, they proposed to add different features scales to handle large variations of action durations. And second, they extended information from those portions of the video during which action/background transition is taking place.

Among all the segment-based initialisation approaches, there are also methods such as those proposed by Escorcia et al. (2016) and Buch et al. (2017), named DAPs and SST, respectively. They are capable of analysing the videos in a single pass. Both works share the same goal: to generate varied length proposals passing only one time over the video stream. Despite the fact that DAPs and SST are based on a C3D+RNN architecture, they exhibit one key difference that makes SST more efficient. DAPs considers only one frame window that is slid over the video stream in such a way that part of its frames are always overlapped. Consequently, all the frames are analysed several times. The frame window of SST is, in the other hand, slid contiguously but without overlapping any frame. In this way, each frame is analysed only once, turning into a more efficient approach than DAPs.

All these methods that rely on anchor segments to be initialised present some relevant drawbacks. First, the fixed-length frame windows that are used as anchors are created arbitrarily, without taking into account any information from the video, *e.g.* frame/video features or shot transitions. As a consequence, a considerable amount of candidate segments can be incoherently generated, which forces the methods to create thousands of segments to find the good ones. Second, although this strategy of generating candidates can make some approaches to exhibit a good recall performance because of the many opportunities to succeed in finding an segment overlapping the ground truth, the precision is strongly sacrificed, as it is demonstrated in Chapter 3 of this thesis. Lastly, since the initial and ending times of each proposal depends on those of the anchor segment, they are neither flexible nor related to the video content, which leads to have inaccurate boundaries.

The method presented in Chapter 3 of this thesis is, in contrast to those cited in this section, online, *i.e.* the video is processed as it is generated in a way that proposals are created as the action part is found in the video. For this reason, it naturally creates far fewer action proposals, making it more efficient without sacrificing the precision. Since it is based on an unsupervised classification-based clustering procedure over the video features, boundaries are created depending on the video content.

Actionness-based initialisation

Approaches that are based on this initialisation consist in finding the temporal boundary points (starting and ending times). These are then related to each other to build candidate proposals.

Actionness is the main concept used in all methods of this group. It was introduced by Chen et al. (2014) for the topic of spatial action detection. The authors defined *actionness* as the quantification of the likelihood of containing a generic action (no matter the label) at a specific location within an image. Typically, actionness is shown through a heat map like those of Figure 2.4. However, if the concept is taken to the TAP topic, it could be redefined as the quantification of the likelihood of containing an action instance at a specific moment (unit, instant, frame, . . .) of the video.

The first alternative to segment-based approaches and that relied on the concept of actionness was TAG, which was proposed by Zhao et al. (2017). The first step of TAG consists in dividing the video into contiguous snippets. TAG collects the snippet-level representation of a video after extracting Two-stream CNN features (Simonyan

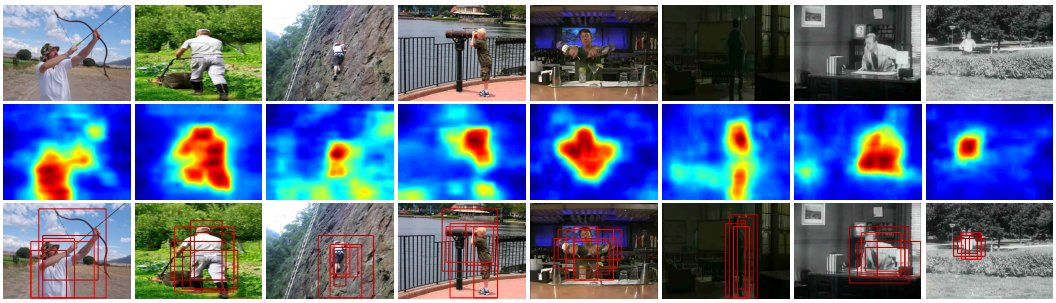


Figure 2.4: **Actionness for Spatial Action Proposals.** Spatial Action Proposals refers to the task of localising within an image those regions that are likely to contain an action. An actionness map shows the likelihood of each location in the image of belonging to an action. This figure contains examples of actionness maps over images (middle row) to extract spatial action proposals (bottom row). Figure from Wang et al. (2016).

& Zisserman (2014)). Afterwards, it processes each snippet to obtain an actionness curve using the TSN (Wang et al. (2016)) classifier. More specifically, this curve is built upon the scores of the action classifier. Finally, the classic watershed algorithm (Roerdink & Meijster (2000)) is applied to the 1D actionness complement curve to group temporal segments with high action score and hence generate the set of action proposals. A visual explanation of this algorithm is depicted in Figure 2.5.

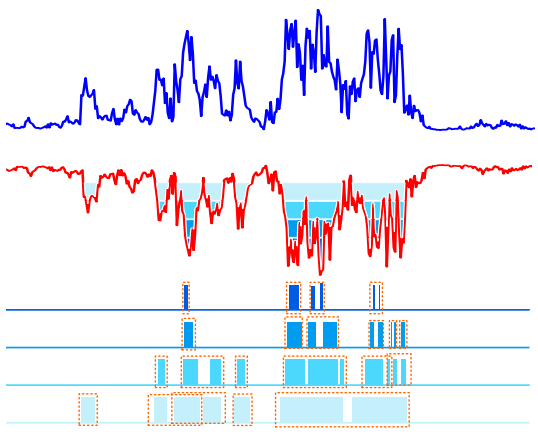
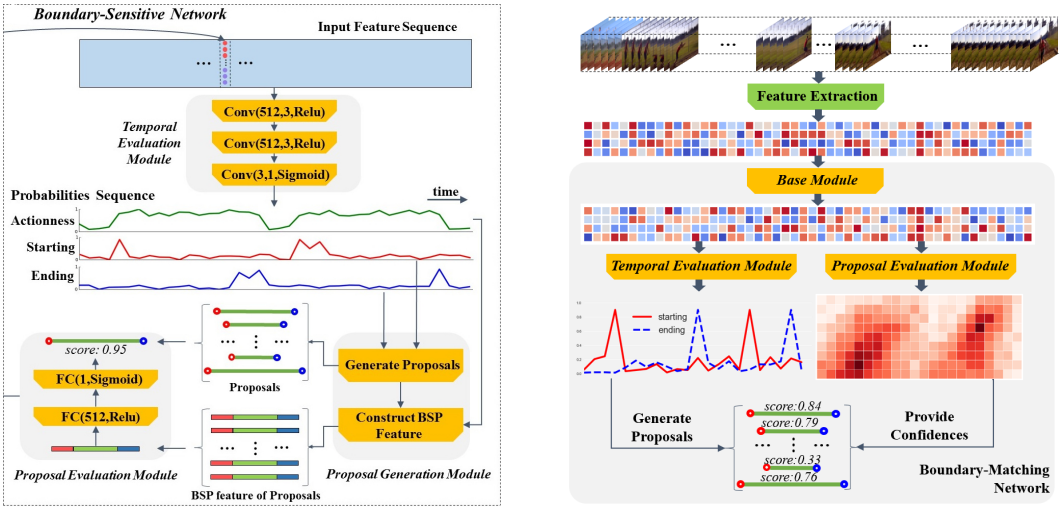


Figure 2.5: **Temporal actionness grouping for Temporal Action Proposals.** As stated by the authors, the algorithm works as if it flooded the 1D signal with different water levels. Some regions of it would be filled and form the action proposals, while others would remain empty and correspond to the background. Different lengths of proposals are obtained when using different levels of water. Figure from Zhao et al. (2017).

Lin *et al.* proposed in their recent publications that working with the boundary points (starting and ending instants) that are found along the actionness curve is more beneficial than creating directly the segments by thresholding the actionness

curve. In this direction, they came up with the approaches BSN (Lin et al. (2018)) and BMN (Lin et al. (2019)). Same as in TAG, both approaches collect first a snippet-level representation of each video with the Two-stream CNN feature extractor. Then, they slide a non-overlapped window across the snippets to obtain, with a temporal convolutional network, three curves or sequences with starting, ending and actionness probabilities. Lastly, all boundary points are related to each other and assigned a certain score. BSN obtains this score by feeding a 2-fully-connected layer module with a pooled version of the actionness sequence over the proposal duration. On the other hand, BMN builds a confidence map over all possible combinations of boundary points and searches it for the score corresponding to a given proposal.



(a) Boundary sensitive network (BSN). (b) Boundary matching network (BMN).

Figure 2.6: **Boundary sensitive network (BSN) and boundary matching network (BMN) for Temporal Action Proposals.** Both of them share the same structure except in the way of assigning the scores to the proposals. While BSN regresses a score for each proposal by feeding a 2-layer fully connected module with a pooled version of the actionness over the proposal duration, BMN builds a confidence map covering all possible combinations of boundary points and searches for the score corresponding to a given proposal. Figures from Lin et al. (2018) and Lin et al. (2019), respectively.

Different from the previous style, these approaches build candidate segments upon information from videos: the actionness. Their boundary points are not dependent to any anchor segment and so they can be more accurate. However, since these detected boundaries are matched to each other without much reasoning, lots of incoherent candidates can still appear, which means the problem of precision remains unsolved. Furthermore, the score of each candidate is generally based on the

actionness present along its duration. If some parts of the video containing an action that is difficult to be detected show low actionness, the associated score will be low and the method could miss them.

Combined approaches

Having observed the previously mentioned drawbacks in methods from both styles, some authors have proposed to leverage both types of initialisations by joining them, such as Gao et al. (2018) and Liu et al. (2019) with their CTAP and MGG methods, respectively. Apart from their particular details, CTAP and MGG share the same structure. First, unit level features from videos are extracted. Afterwards, the following independent sets are built: a) a set of window-based candidate segments; and b) a set of high-actionness segments. Finally, each candidate segment is refined by comparing it to those of the second set.

As for the specific aspects of CTAP, two-stream features with TSN (Wang et al. (2016)) strategy are used as unit-level features. Then a 2-layer temporal convolutional network over unit features is used to obtain actionness. Finally, TAG (Zhao et al. (2017)) and TURN-TAP (Gao et al. (2017)) methods extract the actionness and window-based proposals, respectively. The main contribution is the complementary filter designed to collect high quality complementary proposals from sliding windows to fill the omitted actionness proposals. This filter is composed of two consecutive modules. The first one assigns to all window-based proposals a probability whether they can be detected by actionness. The second module filters those proposals with low probability of being detected through TAG.

Regarding MGG, it utilises a two-stream CNN for the unit-level features. At the beginning, these features are combined with the position embedding of each unit and fed into a 2-layer temporal convolutional network, whose outputs are evaluated through a bilinear matching model to obtain a set of video matching representations. From this point, MGG is divided into two branches: segment-based proposal generator and actionness-based proposal generator. Different from CTAP, the segment-based proposal generator is not based on a sliding window. Instead, the temporal features are delivered to a U-shape conv-deconv architecture that forms a pyramid of varied-length candidate action segments. The second branch generates three frame probability curves: starting, ending and middle probabilities. TAG is applied to the curves to obtain a set of actionness-based proposals. Since these types of proposals

tend to be more accurate, those boundaries from any of the segment-based proposals whose temporal Intersection over Union (tIoU) meets a certain threshold with any actionness-based segment are adjusted to the actionness boundaries.

2.2.2 Level of Supervision

It is important to note that all the works that have been mentioned so far tackle the problem from a supervised perspective. This means that all the available training data is used during training, as well as all the labels for each video (action/background label and starting/ending times). Additionally, those methods that are composed of multiple stages, such as BSN, BMN, CTAP or MGG, need an independent training procedure for each of their modules, which makes them non end-to-end models where a strong supervision is needed.

As an exception to this, the work of Khatir et al. (2019) is the first attempt of bringing the the unsupervised setting to the TAP problem, with a simple online clustering based on the euclidean distance between frame features. The AlexNet (Krizhevsky et al. (2012)) network pre-trained on ImageNet (Deng et al. (2009)) for the task of Image Classification is used as the feature extractor. Then, each video is processed so to group the frames into clusters depending on the euclidean distance between their features. Each one of the resulting clusters is considered a candidate action proposal. With this, the whole video is covered by candidates. To remove those that could represent background, they use the Rank Pooling method (Fernando et al. (2016)) over the dynamics of each cluster.

This unsupervised approach is of special interest for the work presented in this thesis, as the TAP method (SVC-UAP) introduced in Chapter 3 is designed with a similar schema: unsupervised clustering and Rank-Pooling-based proposal filtering. However, the cited approach presents a significant weakness in a key aspect in an unsupervised setting: the information extracted from the video. Instead of a classic and shallow 2D convolutional network, the model of this thesis utilises C3D (Tran et al. (2015)), a 3D convolutional network to capture not only the spatial but also the temporal features. Its weights are pre-trained from the action recognition task, which is a task closer to TAP than Image Classification.

A different perspective is also the recent semi-supervised approach proposed by Ji et al. (2019). This work is not a new method but a novel training setting which adapts the Mean Teacher semi-supervised procedure, proposed by Tarvainen & Valpola

(2017), to the BSN (Lin et al. (2018)) framework. The authors in this work claim that reducing the training set to a small amount of training samples in complicated frameworks, such us BSN or CTAP, leads to overfitting due to the strong level of supervision that they require. To mitigate this, they propose a training procedure where two sets of training samples are used: a) a smaller set of labelled training samples; and b) a set of unlabelled videos. In the Mean Teacher semi-supervised training, the base network is replicated to have two identical models: the teacher and the student. Only the student network is updated via back-propagation. The weights of the teacher model are updated by averaging those of the student within a certain number of previous iterations. When the training samples are labelled, the weights are updated with the supervised classification loss based on the softmax over the student output. If the input corresponds to unlabelled videos, the student is encouraged to have the same output as the teacher with a consistency loss. The intuition is that the teacher model is more stable and its predictions are better. Figure 2.7 depicts the Mean Teacher semi-supervised training designed by Tarvainen & Valpola (2017).

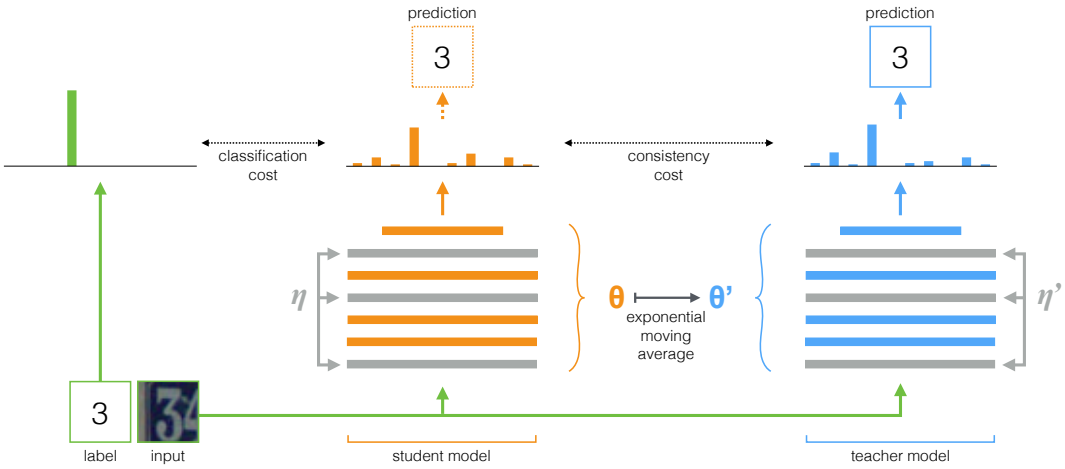


Figure 2.7: **Mean Teacher framework for semi-supervised learning.** The student and the teacher model evaluate the input after applying noise (η, η') to it. The output of the student model is compared to the labels with a softmax-based classification loss and with the teacher output using a consistency loss. After the weights of the student model have been updated with gradient descent, the teacher model weights are updated as an exponential moving average of the student weights. The training iteration with an unlabelled example would be similar but without including the classification loss. Figure from Tarvainen & Valpola (2017).

Tarvainen & Valpola (2017) also suggest adding noise to the input features to prevent the network from overfitting. With this aim, Ji et al. (2019) apply two perturbations to the features: Time Warping and Time Masking. The former resamples

the feature sequence so it serves as data augmentation. The latter works as a dropout for features instead of neurons: given a probability, a certain feature is removed from the sequence. The whole approach is shown in Figure 2.8.

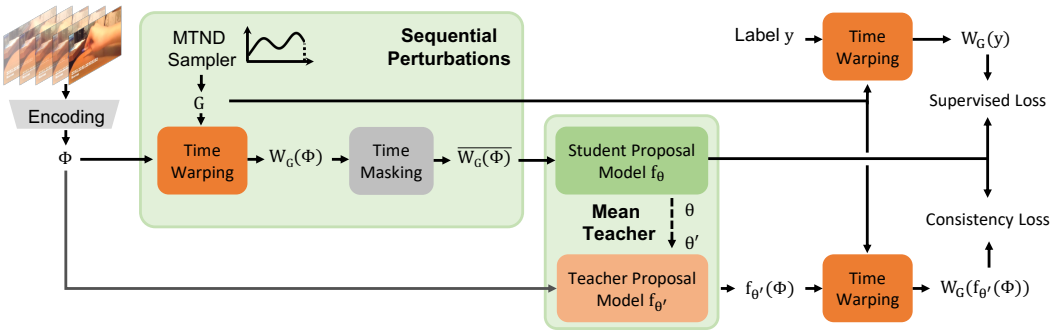


Figure 2.8: **Semi-supervised setting for Temporal Action Proposals.** The authors of this work propose to adapt the Mean Teacher training framework to the BSN model to have a semi-supervised TAP approach. As noise, they apply Time Warping and Time Masking over the video features: the first resamples the feature sequence so it also serves as data augmentation and the second works as a dropout for features instead of neurons. Figure from Ji et al. (2019).

Different from this semi-supervised, the setting chosen in this thesis is not an adaptation of a training procedure but a dedicated solution. It does not need any annotation of any kind (initial/ending times of actions, durations or action labels) and it works in an online fashion.

2.2.3 Concluding remarks

The objective of this task is to find actions segments in videos, which can be simply seen as differentiating action from background. Beyond the specific technical drawbacks that the works cited above present, there is a fundamental downside derived from their design philosophy: allowing methods to generate such a large amount of potential action segments regardless of the precision and seeking only to maximise the recall, removes the relevance that the background class should really have. And even though such approaches could be interesting tools for Video Summarisation or Highlight Retrieval, they are inefficient for the situations in which this thesis is contextualised (see Section 1.1). Within the online scenarios that are considered, where decisions are to be taken *now*, the system must not try multiple times to find the relevant part of the video (the action in this problem), but do it as fast and soon as possible and with the least possible number of failures. This downside is the major

motivation for changing the paradigm and developing an online system such as the one presented here.

2.3 Temporal Action Localisation

The problem of Temporal Action Localisation (TAL), also called Temporal Action Detection (TAD), involves temporally locating those segments of a certain video that contain the action *and determining their action labels*. In contrast to the Temporal Action Proposals task, in Temporal Action Detection methods go one step further, providing the action category.

As stated by Alwassel et al. (2018) in their diagnosis of TAL models, the best performing methods are, generally, based on two-stage pipelines (Lin et al. (2018); Zeng et al. (2019); Xu et al. (2020)): 1) Temporal Action Proposals and 2) Action Classification. More specifically, they consist of a temporal action proposals module (any of those described in Section 2.2) and an action classifier. While the former is responsible for generating segments with high probability of containing an action, the latter assigns a label to them. From the introduction of the TAL problem to date, this type of pipeline has been the preferred one among the authors, and also the one which has generally obtained the best results (Ghanem et al. (2018)). On the other hand, there are those methods that perform temporal action detection in just one stream (*e.g.* Shou et al. (2017); Shyamal Buch & Niebles (2017); Lin et al. (2017)), *i.e.* without the Action Proposals stage.

These top performer methods have an important commonality: they are fully supervised, *i.e.* they make use of all the available annotations during training. Recently, however, more work that uses a weakly supervised approach is emerging, though with not the same effectiveness (Wang et al. (2017b); Nguyen et al. (2018); Paul et al. (2018); Narayan et al. (2019)), where only the label of the action happening in the video is available as annotation. In short, these methods typically obtain, for each frame, a probability value of being representative of the action which the video is labelled with. Low probability frames are discarded and considered background.

The thesis presented here is not focused on this task. However, it addresses the Temporal Action Proposals problem, which is crucial for TAL. Beyond competing to have the best performance, the TAP module introduced in Chapter 3 supposes a new method for a paradigm barely unexplored: online unsupervised Temporal Action Pro-

posals. The objective of this online unsupervised solution is to be fast and efficient in generating proposals, so they can be quickly delivered to the action classifier to perform the TAL task as soon as possible.

2.3.1 Two-stage Temporal Action Localisation

Much of the success of the methods that follow this strategy is due to the flexibility that they offer as they can combine different modules to get the best final result. The majority of two-stage TAL works have obtained state-of-the-art results by especially contributing to one of the stages. Moreover, TAP authors typically extend their experiments by applying a state-of-the-art classifier to the proposals that their methods have previously generated. Concretely, the most used classifiers are SCNN (Shou et al. (2016)) and UntrimmedNets (Wang et al. (2017b)).

Figure 2.3 contains the whole pipeline that Shou et al. (2016) designed for Temporal Action Localisation. As it was said in Section 2.2.1, the authors proposed a C3D-based method built with three stages: 1) Proposal, 2) Classification and 3) Localisation. The first and second stages would be sufficient for TAL as it generates labelled proposals. However, they added the Localisation stage to score again each proposal according to the potential IoU with the ground truth. As this was one of the first two-stage temporal action detector solutions, many TAP works have used its classification branch (SCNN-cls) to categorise their proposals. As an example, TURN-TAP (Gao et al. (2017)), CTAP (Gao et al. (2018)), BSN (Lin et al. (2018)), BMN (Lin et al. (2019)), MGG (Liu et al. (2019)) and DBG (Lin et al. (2020)) have utilised SCNN-cls over their proposals.

As for UntrimmedNet, described in Figure 2.9, it is one of the first weakly-supervised approaches for Temporal Action Localisation. This approach is trained without having access to the temporal annotations of actions. Instead, the category of the action instances that appear in a video is considered the video-level label, and each proposal is to be classified as if it was an action instance of that category. The input video is first divided into contiguous clips (candidate proposals) whose features are then extracted and fed into a classification stage, which assigns a vector of logits to each candidate. Those k segments with highest score for a class are considered the actual action proposals. Lastly, to perform video classification, those logits from the k best proposals are either directly averaged or averaged with attention-based weights. The class with the maximum score is chosen as the video category. When UntrimmedNet

is used as a classification stage over other method's proposals, the proposal is treated as the video and the video category as the detected action. This is the way in which TAP methods, such as BSN, BMN, MGG or DBG, make use of the UntrimmedNet, and typically, achieving better results than with SCNN-cls.

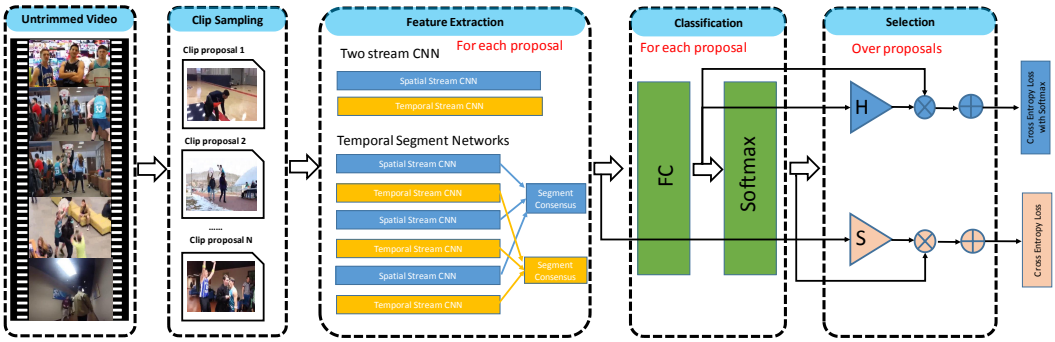


Figure 2.9: **UntrimmedNets for weakly-supervised Temporal Action Localisation.** During training, the label of the action appearing in the video is considered as the label for the whole video, boundary annotations are not used. The video is first divided in a set of contiguous clips. These clips serve as the initial candidate proposals. Once the features are extracted, each candidate is assigned a label and a score by the classification module. The final selection module ranks the proposals to select them according to the classification score of each proposal. Figure from Wang et al. (2017b).

Given the huge interest that graphs have lately raised, Zeng et al. (2019) designed their PGCN method for TAL, which is based on Graph Convolutional Networks. With this work, the authors contributed to the second stage of a TAL solution: the action classifier. Their intuition is that previous approaches were not exploiting the relationship between proposals and the benefits that this relationship could have to improve the classification. To explore this, they propose the possibility of modelling that relationship through a Graph Convolutional Network in which nodes are proposals and edges represent the relationship between them. As shown in Figure 2.10, the system is built on top of the proposals of a previous TAP model, concretely BSN. To initialise the nodes, two-stream I3D (Carreira & Zisserman (2017)) features of each proposal are extracted and pooled. Two identical graphs are set to determine the action category of the proposal and refine its boundaries, respectively. To date, PGCN is the top performer method for the Temporal Action Localisation task on THUMOS14 (Idrees et al. (2017)) and ActivityNet (Heilbron et al. (2015)) datasets.

For these two-stage systems, which are heavily dependent on the temporal action proposal stage, much of the criticism is in the same direction as for TAP approaches:

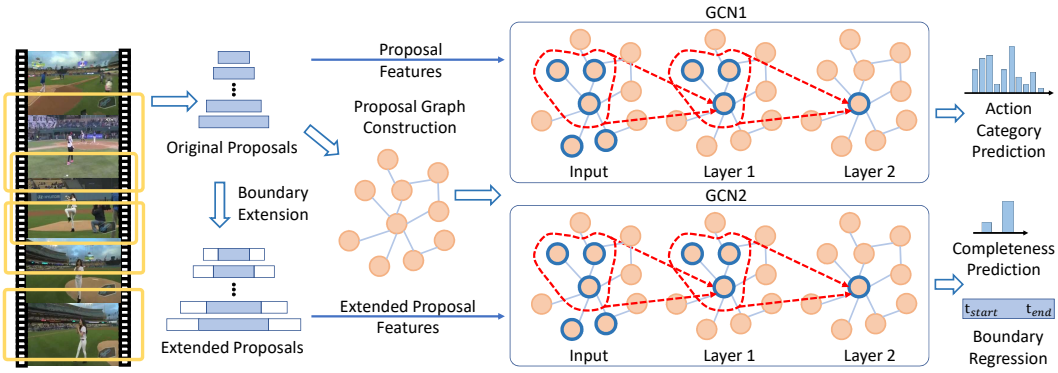


Figure 2.10: **PGCN: Graph Convolutional Networks for Temporal Action Localisation**. In this work, the authors want to exploit the benefits of modelling the relationship between proposals to improve the classification of action proposals. The features of each proposal are used to initialise each node. Two identical parallel graphs are placed to determine the action category of the proposal and refine its boundaries, respectively. Figure from Zeng et al. (2019).

precision, accuracy of the boundary points and efficiency. Additionally, there are situations that require detections to be obtained as soon as possible and, due to their architecture, these mentioned TAL approaches are not suitable. All the work presented in this thesis is focused on analysing videos in an online fashion. Specifically, the approach presented in Chapter 3 proposes to change the paradigm. Different from the approaches that have been mentioned in this section, it is an online TAP solution which is able to deliver all action proposals to the action classifier as soon as they are generated so to accelerate the whole process of temporal action detection.

2.3.2 Single-stream Temporal Action Detectors

If TAP approaches and action classifiers can already be complicated individually, bringing them together, as in a two-stage temporal action detector, may result in a cumbersome system. Some authors have noticed this fact and have proposed a single-stage TAD philosophy (Yeung et al. (2016); Shou et al. (2017); Shyamal Buch & Niebles (2017); Lin et al. (2017); Long et al. (2019)). In contrast to previous methods, these are based on end-to-end approaches whose modules are not optimised separately, albeit they still need to perform a post-processing of their outputs.

The work of Yeung et al. (2016) was the earliest end-to-end solution, and it currently is the only one that has introduced Reinforcement Learning to address the problem. In a nutshell, the authors came up with a LSTM-based system that at each

time step outputs: 1) segment boundaries and categories; 2) a probability of being a correct candidate segment; and 3) the index of the next frame to observe to confirm the segment. Iteratively, the system finds all action segments by refining in each step the boundaries that it regresses. Once the probability of correct segment is sufficiently high, it confirms the detected segment and continues searching for more.

On the other hand, the best single-stream model as of now is GTAN, which was proposed by Long et al. (2019). This method first split the video in consecutive segments and extract their Pseudo-3D (Qiu et al. (2017)) features. Then, the features are fed into a series of 8 cascaded 1D convolutional layers that reduce the number of features to create different scales. At each scale, the network produces Gaussian kernels which span according to the length of the predicted action proposal. Finally, all proposals are classified.

Although the performance of the works that are cited here may not reach the same level as that of two-stage methods, their structure eases the application, and in some situations they could be more appropriate. Additionally, they are more efficient in the sense that fewer proposal segments are needed.

2.4 Online Action Detection

As previously stated in Section 2.1.2, the online paradigm is a recently proposed way of analysing videos in which the only content available to recognise the action are past and present frames, *i.e.* the video is not supposed to be available beforehand, as it happens in all works mentioned in the previous sections. Consequently, this scenario needs methods that are capable of detecting actions as soon as they occur. Online analysis of video content shows great usefulness in some real-world scenarios. For example, one can imagine a video surveillance application that has to raise an alarm when certain action is happening. In this particular situation, the application cannot wait for the video to end, so any previous method can be applied. On the other hand, it needs a solution that can analyse the video content as it is generated so to detect and raise the alarm as soon as the action appears.

In 2016, De Geest et al. (2016) claimed that what had been proposed up to that date to analyse video in an online way was not realistic enough (see Section 2.1.2). This motivated the authors to define a new problem: Online Action Detection (OAD). As described by the authors, given an untrimmed streaming video whose end is un-

known, OAD is the problem of detecting actions that are happening *now* without any information from the future (because it is not known). Since they considered themselves the first OAD work, they contributed with a new dataset, named TVSeries, that complies with the aforementioned characteristics. A new metric and several baseline methods were thoroughly evaluated in TVSeries. Regarding the metric, they came up with the calibrated Average Precision (cAP), which is an adaptation of the mAP metric used for OffAD. Finally, they offer several baselines based on SVMs, CNNs and LSTMs. Furthermore, in their follow-up work (De Geest & Tuytelaars (2018)) they introduced a two-stream LSTM approach, like the one seen in seen in Figure 2.11. While the first layer encodes previous and present frame representations, the second is used to reintroduce, after some delay, the class probabilities so that the network is aware of the sequence of detections that have been made at past video frames. It is important to note that neither of these first approaches explicitly discriminates the background from the action categories as if it was one more class. Although it could have been used during training, at testing, the output only shows the distribution of probabilities for only the actions that are annotated.

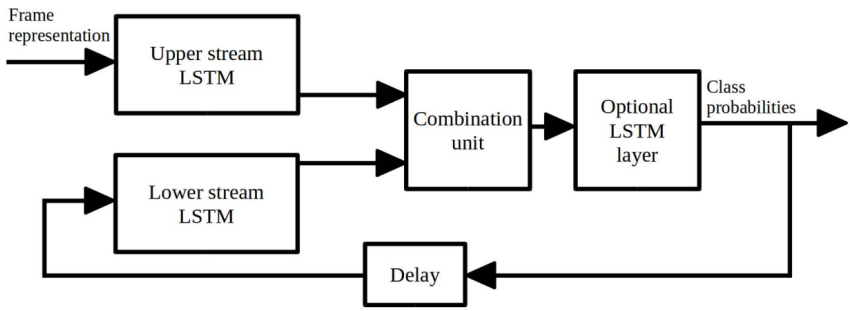


Figure 2.11: **Two-stream feedback LSTM for Online Action Detection.** While the first layer encodes the previous and present frame representations, the second is used to reintroduced, after some delay, the class probabilities so that the network is aware of the sequence of detections that have been made at past video frames Figure from De Geest & Tuytelaars (2018).

Besides, the authors of RED (Gao et al. (2017b)) designed an action anticipation system based on a reinforced encoder-decoder structure (Figure 2.12). The encoder-decoder part of the model learns to anticipate both the frame features and the class which they belong to through a squared loss and a cross entropy loss, respectively. The output sequence of predicted labels is introduced into a reinforcement learning module that calculates a reward according to the correctness of the sequence. They claim that OAD is a especial case of action anticipation when the anticipated time

is zero. In this direction, they conducted an experiment where the anticipated time is reduced to a minimum value of 0.25 seconds and compared their results to other state-of-the-art methods. Similar to De Geest et al. (2016) and De Geest & Tuytelaars (2018), the background is also not treated as another class, in addition to all actions. Instead, the cross entropy loss applied to the decoder outputs during training only considers the specific annotated classes, and the reinforce module utilises the background during training to make the network learn in which frame of the given sequence is the background/action change.

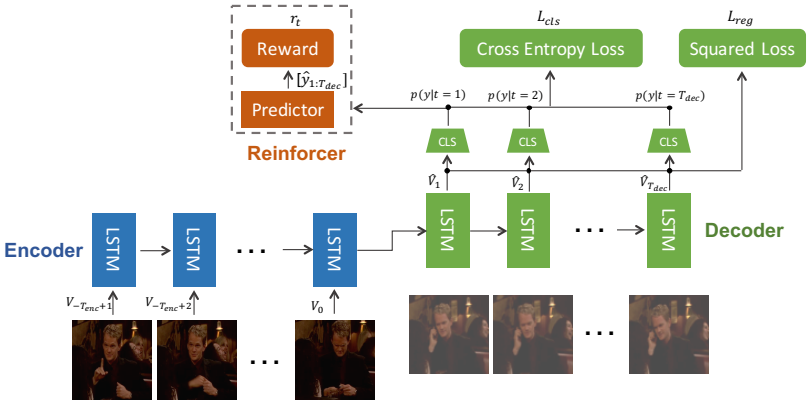


Figure 2.12: **Reinforced Encoder-Decoder for Action Anticipation.** The encoder-decoder part of the model learns to anticipate both the frame features and the class which it belongs to through a square loss and a cross entropy loss, respectively. The output sequence of predicted labels is introduced to a reinforcement module that calculates a reward according to the correctness of the sequence. Figure from Gao et al. (2017b).

More recently, Xu et al. (2019b) designed the TRN Cell described in Figure 2.13. As input, they collect groups of 6 consecutive frames and extract their appearance features from the central frame and their motion features from the optical flow computed over the whole group of frames. Both features are concatenated and fed into the TRN Cell at each iteration. As seen in Figure 2.13, the TRN Cell receives the features from the current group of frames and anticipates several consecutive future frame representations. These representations are pooled to form a single vector and then concatenated to the present frame. The concatenated vectors and that of the present representation are the inputs of a following RNN cell. Finally, a FC layer will output class probabilities considering all classes plus the background. With the TRN Cell, a smart but controversial use of the future is introduced. In the OAD setting, using future information is not allowed, simply, because it is unknown. But in this case, the authors claimed that future is predicted and hence, actual future is not

used.

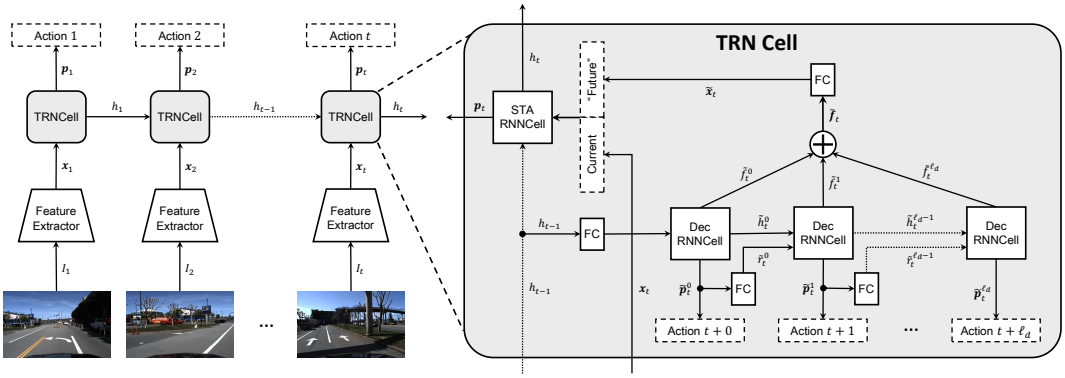


Figure 2.13: **TRN Cell for Online Action Detection.** In this work, the authors (Xu et al. (2019b)) proposed to use predicted future frame representations to improve present predictions. To this aim, they designed the TRN Cell, which outputs a vector that combines the present frame features with predicted future frame features. Figure from Xu et al. (2019b).

Out of the OAD topic but very closely related is the Online Detection of Action Start (ODAS) problem. Although the scenario and its conditions are the same, the aim is to detect only those instants in which actions start. It is a very recent topic and only two existing works have currently contributed to it: Shou et al. (2018) and Gao et al. (2019).

Overall, given the early stage of the OAD topic, there are some ambiguities regarding the background and the evaluation that must be cleared up. This thesis proposes that the background must be treated as another class of same importance as the action classes. Therefore, methods should detect it explicitly. In the cAP metric, proposed by De Geest et al. (2016), the ability of the methods to discriminate between actions and background is not measured. Additionally, it must be applied at the end of the execution, once the whole video has been analysed, so the performance of methods cannot be evaluated in an *online fashion*. Chapter 4 of this thesis analyses thoroughly all these ambiguities and proposes several solutions. First, the properties of the OAD topic are revisited and new ones are established to improve its completeness. Second, a novel online evaluation protocol is introduced and, in contrast to the currently used, it is able to show the online performance of methods and measure their ability to discriminate all actions and background. Additionally, everything is proved through an experimental set-up that includes some baselines from the state of the art, as well as new ones that comply with all the new conditions.

Chapter 3

Unsupervised Action Proposals

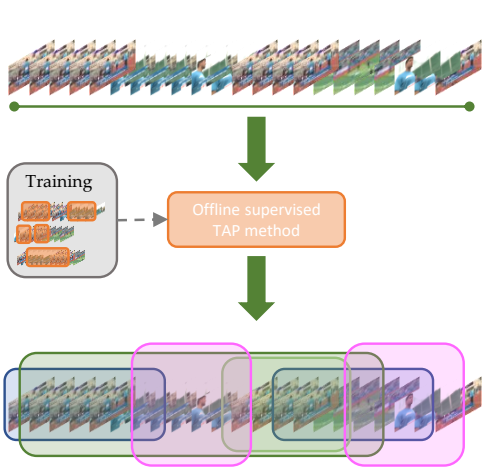
As seen in the Related Work of Chapter 2, the problem of localising in videos temporal segments that are likely to contain an action, named Temporal Action Proposals, is crucial for the Temporal Action Localisation task. This is evidenced by the large body of work that have been published up to date, *e.g.* (Shou et al., 2016; Wang et al., 2017b; Lin et al., 2018; Ji et al., 2019; Lin et al., 2020), among others.

However, all the proposed TAP approaches address the problem from a supervised learning paradigm and following an offline setting, as the one shown in Figure 3.1a. These conditions imply that: 1) during training, the complete set of temporal annotations is used; and 2) the video must be processed before generating the action proposals. This setting leads to having lots of overlapped proposals which usually are overestimated to maximise the recall but sacrifice the precision of the models.

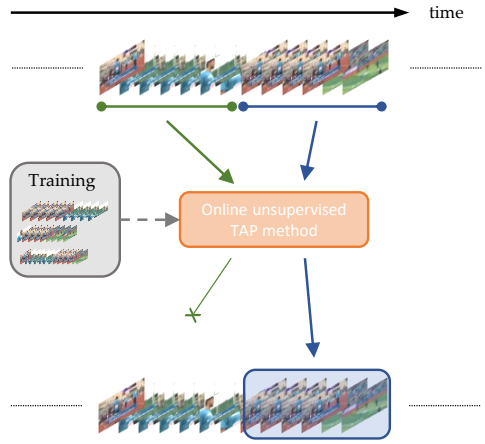
Despite the good performance that standard approaches offer, there are some real situations where they are not the best choice. From the point of view of training, fully-supervised approaches typically consist of several modules that have to be thoroughly trained independently. Given this strong level of supervision and the need of all the available labels, it makes these solutions impractical for fast-changing scenarios. As for the way of analysing the video, methods that need access to the whole video cannot be used in environments that are characterised by streaming videos. Additionally, these environments may require decisions to be made according to the video content at any given moment, therefore, online. Very clear examples of these realistic situations can be a scenario where a robot has to interact with a human, an intelligent video surveillance service designed to raise an alarm when a certain action

is detected, or autonomous vehicles, for instance.

Taking this into consideration, the work presented in this chapter aims to explore a new direction, described in Figure 3.1b. First, the solution is totally *unsupervised*: the model is not allowed to use any labelled data during training nor any feature pre-trained on the evaluated dataset. Second, it is *online*: it is able to generate action proposals as soon as they occur in the video stream. This restriction forces the model to work with partial observations of the video, instead of having access to the whole video to generate and refine the proposals. Although this new setting seems much more challenging, it is undoubtedly more appropriate for the situations above mentioned.



(a) Offline supervised setting for Temporal Action Proposals.



(b) Online unsupervised setting for Temporal Action Proposals.

Figure 3.1: **Comparison between standard offline supervised approaches and the new online unsupervised setting that is proposed for Temporal Action Proposals.** In the standard setting (**right**), methods are trained with all the annotations available in a dataset and, during test, they have always access to the whole video. All the approaches that are based on this setting generate typically lots of overlapped proposals. In contrast to this, in the new online setting that is proposed in this chapter (**left**), methods are trained with unlabelled videos and have only access to partial observations of the video stream at test time. Action proposal segments must be generated or discarded as soon as the video arrives to the system.

Chapter's contributions

Figure 3.2 shows an overview of the proposed approach. The main contributions of this chapter can be summarised as follows:

- As of now, this work is the first in addressing the Temporal Action Proposals problem with a novel unsupervised solution, which is based on two main modules: a Support Vector Classifier (SVC) and a filter based on dynamics. While the former discriminates between contiguous sets of video frames to generate sets of candidate segments, the latter computes the dynamics of these segments and applies a distance criterion between each segment dynamics and a randomised version of them.
- Unlike all state-of-the-art approaches, this is the first model that operates completely online. The video is processed as it arrives to the system, without accessing any information from the future nor modifying any past decision.
- Comparing to the state of the art, the best unsupervised configuration achieves more than 41% and 59% of the performance of the best supervised model for ActivityNet and THUMOS14 datasets, respectively.

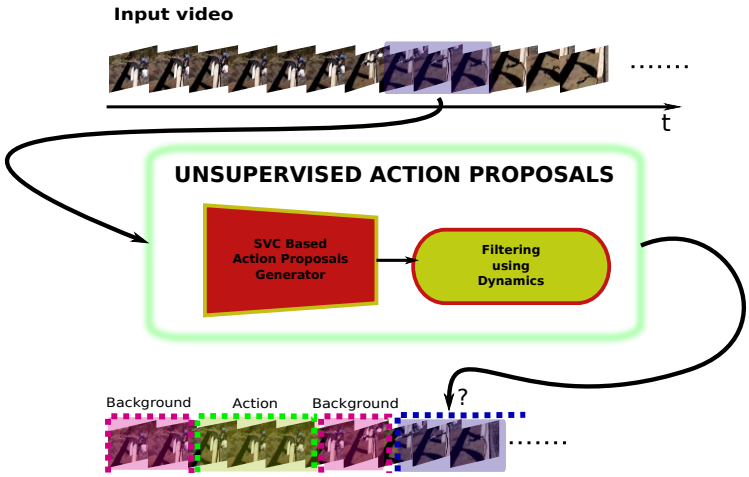


Figure 3.2: **Overview of the new unsupervised online method for Temporal Action Proposals.** The new proposed framework utilises recursively two modules: a Support Vector Classifier (SVC) and a rank-pooling-based filter. While the former discriminates between contiguous sets of frames to generate candidate action proposal segments, the latter computes the dynamics of these segments and applies a distance criterion between each segment dynamics and a randomised version of them to confirm or discard them as actual action proposals.

3.1 Proposed Approach

As seen in Figure 3.2, the framework that is proposed in this chapter is composed of two modules: a SVC-based proposal generator and a dynamics-based filter. Each of them rely on the two following hypothesis, respectively:

(H1): Frames that belong to different parts of a video are separable by classifiers, once they have been projected to a set of deep features.

(H2): Features from frames that belong to an action have a temporal structure, while those of background do not. Hence, background segments can be discarded when this temporal structure is not found.

As depicted in Figure 3.3, a particular video is analysed in an online fashion to find different segments on it. Technically, at each instant of execution, deep features of frames are obtained and organised in two consecutive groups. Following hypothesis H1, an online classification-based procedure based on SVC (Boser et al. (1992)) is used due its demonstrated ability to separate sets of features. When this module confuses the groups, the features are considered similar and hence from the same group. Conversely, features are considered from different groups when the SVC is capable of separating them. The process iterates throughout the video to generate all candidate action proposal segments.

The dynamics of a certain video can be defined as the video-wide temporal evolution of the appearance of its frames. This type of meta-features has been used to address the problem of action recognition in several works, such as (Fernando et al., 2016; Bilen et al., 2016; Wang et al., 2017a; Cherian et al., 2017; Fernando et al., 2017; Cherian et al., 2018). This work, instead, proposes using this concept to find and discard the background segments in an unsupervised way. According to hypothesis H2, if the dynamics of a segment are compared to those of the same segment but when its features have been *randomly* disordered, their similarity will be high only for background. Conversely, if the same is done with an action segment, the effect on the dynamics will be much more visible as the randomised version of the segment will miss the structure of the action.

The described pipeline is named as SVC-UAP. It works according to the setting shown in Figure 3.1b: it is *unsupervised*, since the the deep model is pre-trained to recognise actions in datasets that are different from the concerned one and no

annotation is used during training; and it is *online* because only the part of the video that has been seen is accessible. A detailed overview of the approach is shown in Figure 3.3.

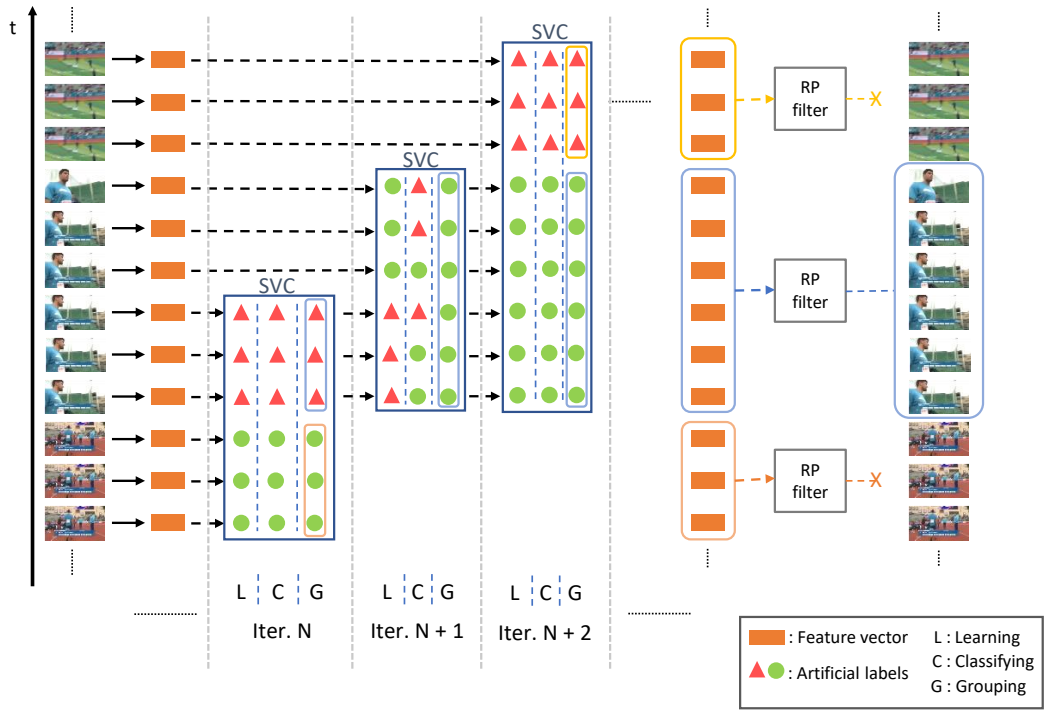


Figure 3.3: **Online unsupervised action proposals generation process.** At each iteration, the SVC module is, first, fed with a certain number of contiguous frame features. Afterwards, they are artificially labelled for the learning step of the SVC module (note that the labels are artificial, independent at each step and do not correspond to any actual category). Finally, the SVC will decide on which groups to make after evaluating the classification error. A high value means that the features are not easily separable and, thus, they belong to the same segment. Conversely, features are considered from different groups when the SVC is capable of separating them. The rank pooling module will determine whether the candidate segments generated by the SVC are actual proposals. The model operates completely online, accessing only to the video frames available until time t.

3.1.1 Definition of the Problem

It is assumed that an untrimmed video can be denoted as a frame sequence $\mathcal{V}^i = \{v_n^i\}_{n=1}^{l_i}$, where l_i encodes the duration of the video, and v_n^i is the n-th RGB frame. This kind of videos contain portions without actions. Therefore, the set of temporal action annotations for $\mathcal{V}^i$ is defined as $A_{\mathcal{V}^i} = \{a_k = (t_{1,k}, t_{2,k})\}_{k=1}^{K_{\mathcal{V}^i}}$, being $t_{1,k}$ and $t_{2,k}$

the starting and ending times for the action instance k, respectively, and $K_{\mathcal{V}^i}$ the total number of actions instances in the video $\mathcal{V}^i$.

The objective of a TAP method is to generate a set of proposals $AP_{\mathcal{V}^i} = \{ap_k\}_{k=1}^{Kp}$ that correctly overlaps with the set $A_{\mathcal{V}^i}$. While traditional supervised methods are allowed to use the action annotations during learning, an unsupervised approach cannot. So, the objective is to generate the set $AP_{\mathcal{V}^i}$ without using any temporal annotation.

3.1.2 Learning Unsupervised SVC for TAP

If a certain video is represented as the frame sequence $\mathcal{V}^i = \{v_n^i\}_{n=1}^{l_i}$, the first stage of the pipeline is to extract any state-of-the-art feature representation for every frame or set of frames, as it can generalise to these two types of representations. Formally, the sequence $\mathcal{V}^i$ is converted to a set of visual features $\mathcal{F}^i = \{f_n^i\}_{n=1}^{l_i}$, where $f_n^i \in \mathbb{R}^d$.

Given the obtained features, the model begins processing the video $\mathcal{V}^i$ by accessing the first $2 \times N$ features in $\mathcal{F}^i$ to split them into two sets of N consecutive features, $\mathcal{S}_{t=1}^+$ and $\mathcal{S}_{t=1}^-$, i.e. $\mathcal{S}_{t=1}^+ = \{f_1^i, f_2^i, \ldots, f_n^i\}$ and $\mathcal{S}_{t=1}^- = \{f_{n+1}^i, f_{n+2}^i, \ldots, f_{2n}^i\}$. Note that $t = 1$ because it is in the first iteration of the process and that for every new iteration N new features are evaluated. The results in Section **??** show the great importance of this parameter to the unsupervised model.

The following step consists in finding whether these two sets belong to the same segment. To do so, the two sets are artificially identified with two different labels $\mathcal{Y}_{t=1}^+ = \{+1\}_{n=1}^N$ and $\mathcal{Y}_{t=1}^- = \{-1\}_{n=1}^N$, and the SVC proceeds to learn to separate them according to the labels. Mathematically, it is defined the tuple (w_t, b_t) which represents the max-margin hyperplane to separate $\mathcal{S}_t^+$ and $\mathcal{S}_t^-$. The SVC solves the following primal problem to find (w_t, b_t):

$$\min_{w_t, b_t, \zeta} \frac{1}{2} w_t^T w_t + C \sum_k \zeta_k \,, \tag{3.1}$$

$$\begin{aligned} \text{subject to } & y_k (w_t^T \phi(f_k) + b_t) \geq 1 - \zeta_k \,, \\ & \zeta_k \geq 0 \; \forall k \,, \end{aligned} \tag{3.2}$$

where $f_k \in \mathbb{R}^d$ are the given K feature vectors in the sets $\mathcal{S}_t^+$ and $\mathcal{S}_t^-$ with the labels of $\mathcal{Y}_t^+$ and $\mathcal{Y}_t^-$ concatenated in vector y. In Equation (3.2), the function $\phi()$ implicitly

maps the training feature vectors to a higher dimensional space. If a linear kernel is used, then $\phi(f_k) = f_k$. If a different kernel function $K(f_k, k_l)$ is used, then $K(f_k, k_l) = \phi(f_k)^T \phi(f_l)$. Overall, no constraint is imposed to the kernel to be used.

At iteration t, once the learning phase of the SVC is finished and the tuple (w_t, b_t) has been obtained, the algorithm classifies the provided features and measures its performance by computing the classification error rate (Cer_t). Lastly, it evaluates Cer_t to decide whether to join or separate the initial groups of features. A high Cer_t means that the SVC is not able to correctly separate the two sets. Hence, the two sets of features $\mathscr{S}_t^+$ and $\mathscr{S}_t^-$ should be joined in the same candidate proposal for the next iteration of the algorithm. On the other hand, a low Cer_t implies that the set $\mathscr{S}_t^+$ is different from $\mathscr{S}_t^-$ and can be considered a different proposal. A threshold α is defined to evaluate these conditions: if $Cer_t \geq \alpha$ then $\mathscr{S}_{t+1}^+ = \mathscr{S}_t^+ \bigcup \mathscr{S}_t^-$, the proposal size is increased for the next iteration; if $Cer_t < \alpha$, then $\mathscr{S}_{t+1}^+ = \mathscr{S}_t^-$ and a new action proposal ap_k is generated from the set $\mathscr{S}_t^+$.

Apart from measuring the performance of the SVC module, the Cer_t is also used in Equation 3.3 to obtain a score for each of the generated action proposals.

$$s_k = e^{-Cer_t} \tag{3.3}$$

The standard evaluation metrics for TAP require the models to assign a score for every generated proposal. With the solution described so far, it is possible to evaluate the recall over the annotations by just ordering the generated candidate proposals and picking the set of n best proposals, according to this score.

The whole process is shown in Figure 3.3. It can be seen how in each iteration the SVC module decides on which groups to make after learning and classifying based on the initial artificial labels. Each of the candidate segments that are generated has to also be evaluated by the next step: the Rank-pooling filter. This module, which is described in the next section, will confirm or discard the candidate as an actual proposal.

3.1.3 Rank-Pooling Filtering

After having collected a candidate action proposal, it must be determined whether it is part of an action. As the setting proposed in this chapter is unsupervised, the algorithm cannot have access to any kind of annotation from the dataset. Instead, it

can only work with the information that can be extracted from the video. This is done with the rank-pooling-based filter that is introduced here.

Let ap_k be a candidate action proposal generated by the SVC module. First, a set $\mathscr{F}^{ap_k} = \{f_n\}_{n=1}^{l_{ap_k}}$ is built, where $f_n \in \mathbb{R}^d$ and l_{ap_k} encodes the size of the proposal. $\mathscr{F}^{ap_k}$ contains the ordered set of features for the video frames included in ap_k. Then, the set of features $\tilde{\mathscr{F}}^{ap_k}$ is generated, which is a randomly disordered version of $\mathscr{F}^{ap_k}$. Finally, a rank-pooling model similar to the one proposed by Fernando et al. (2017) is used to compute the dynamics of $\mathscr{F}^{ap_k}$ and $\tilde{\mathscr{F}}^{ap_k}$.

As in the rank-pooling model, the dynamics of a set of features is summarised as the parameters of a curve in the input space that captures the frame temporal order via linear projections. This is done by optimising a pairwise-learning-to-rank problem based on Support Vector Machine (SVM). In particular, a rank-SVM with a linear Support Vector Regression (SVR) based formulation is implemented, which is known to be a robust point-wise ranking method (Liu (2009)).

Given any set of features $\mathscr{F} = \{f_t\}_{t=1}^{l} = \{f_1, f_2, \ldots, f_l\}$, a direct projection of the input vectors f_t to a time variable t is obtained by employing a linear model with parameters $\omega^{\mathscr{F}}$, as follows:

$$\omega^{\mathscr{F}} = \arg\min_{\omega^{\mathscr{F}}} \sum_t |t - \omega^{\mathscr{F}} \cdot f_t| \,. \tag{3.4}$$

$\omega^{\mathscr{F}}$ summarises the sequence of dynamics, becoming the pooled dynamics descriptor for $\mathscr{F}$, which compactly encodes a sequence of features into a single vector.

For the implementation, the following operations are applied to the feature vectors in $\mathscr{F}$ before computing the rank-pooling dynamics $\omega^{\mathscr{F}}$: (1) a temporal smoothing with time varying mean vectors; (2) a point-wise non-linear operator $\Phi(\cdot)$; and (3) an L2 normalisation. For the experiments, the non-linear function is: $\Phi(f_t) = \text{sgn}(f_t)\sqrt{|f_t|}$.

The rank-pooling filtering mechanism computes the dynamics for $\mathscr{F}^{ap_k}$ and $\tilde{\mathscr{F}}^{ap_k}$, being them $\omega^{\mathscr{F}^{ap_k}}$ and $\omega^{\tilde{\mathscr{F}}^{ap_k}}$, respectively. As described above, the distance between these two dynamics vectors allows the model to identify action proposals, discarding candidates that might include background information. For a candidate that does not represent to any action, the distance between its dynamics and the dynamics of its randomised version should not be high. The Euclidean distance is used to implement this filtering mechanism by measuring it between these two vectors and using

a threshold r to discard background segments: if $d(\omega^{\mathcal{F}^{ap_k}}, \omega^{\tilde{\mathcal{F}}^{ap_k}}) < r$, the candidate proposal is rejected.

It is important to emphasise again that the whole filtering process based on these dynamics, obtained through rank-pooling, works in a fully unsupervised way. Their computation does not require access to any type of annotation. Additionally, the implementation based on linear SVR is efficient, which allows the model to work with online video streams.

Apart from the visual description of Figure 3.3, the implementation is also described in Algorithm 1. Specifically, it shows the procedure that the SVC-UAP solution follows to obtain the action proposals within a certain video.

Algorithm 1 Pseudocode with the implementation of the proposed SVC-UAP method on a certain video to obtain its action proposals. It is worth noting that no labels of any kind are used (unsupervised) and frames are processed as they arrive to the method.

Given a certain video $\mathcal{V}^i$

Input: Incoming frames of a certain video: $\mathcal{V}^i = \{v_0^i, v_1^i, v_2^i, ..., v_{l_i}^i\}$
Features to collect in each iteration: N
Cer threshold: α
Rank Pooling filter threshold: r

Output: Set of action proposals: $AP_{\mathcal{V}^i} = \{ap_k\}_{k=1}^{Kp}$

$AP = \{\}$
$f = 0$
while not end of video **do**
  if first iteration **then**
    $frames = \{v_f^i, v_{f+1}^i, v_{f+2}^i, ..., v_{f+2*N}^i\}$ ▷ First incoming video frames
    $features = FeatExtr(frames)$ ▷ Feature extraction
    $S_1 = features[0:N]$ ▷ Split in two set of features
    $S_2 = features[N:2*N]$
    $f = 2*N+1$
  else
    $frames = \{v_f, v_{f+1}, v_{f+2}, ..., v_{f+N}\}$ ▷ Next incoming video frames
    $features = FeatExtr(frames)$
    $S_2 = features$
    $S_1 = S_{previuous}$
    $f = f+N+1$
  end if
  $Cerr = SVC(S_1, S_2)$ ▷ SVC: Learn, classify and get $Cerr$
  if $Cerr \leq \alpha$ **then** ▷ No action proposal found
    $S_{previous} = Join(S_{previous}, S_1)$ ▷ Join the two sets
  else ▷ Possible action proposal found
    $S_{randomised} = randomise(S_{previous})$ ▷ Random shuffling
    $distance = L2(RP(S_{previous}), RP(S_{randomised}))$ ▷ Get distance
    if $distance \leq r$ **then**
      $discard(S_{previous})$ ▷ If similar, it is background. Thus, discarded
    else
      $AP = append(AP, S_{previous})$ ▷ Proposal confirmed
      $S_{previous} = \{\}$
    end if
  end if
end while
return AP

3.2 Experiments

This section thoroughly evaluates the new SVC-UAP method that was previously proposed. First, an ablation study where the hypotheses H1 and H2 are confirmed is presented. With the same study, it is also explained how the parameter configuration is chosen. Finally, given the typical TAP set-up, a comparison with the fully-supervised methods of the state of the art is offered.

3.2.1 Experimental Set-up

Datasets

As of now, it is the first time an unsupervised solution is evaluated on the two main datasets for the TAP problem: ActivityNet (Heilbron et al. (2015)) and THUMOS14 (Idrees et al. (2017)). This fact offers the opportunity to compare how far the unsupervised solution is from the state of the art.

ActivityNet. It is, in its 1.3 version, a large-scale dataset that provides more than 19k annotated untrimmed videos (648 hours of video). This means that not all the video frames belong to actions, but also to background. Note that this is not the case in other recent large-scale video dataset, e.g., Kinetics-600 (Carreira & Zisserman (2017)) or Moments in Time (Monfort et al. (2020)).

THUMOS14. Not as large as ActivityNet, this dataset comprises more than 400 untrimmed videos, where 20 action categories have been annotated. Following the common set-up: train and test are conducted on the validation and test subsets, respectively.

Evaluation Metric

Following the standard, the Average-Recall versus Average Number of Proposals per Video (AR-AN) metric is used. The objective of any AP method is to produce temporal segments where an action might be occurring. Given a set of proposals, this metric considers a true positive (t_p) if the proposal segment has a temporal intersection over union $tIoU$ with the annotated action that is greater than a certain threshold. The recall is computed following the next equation:

$$R = \frac{t_p}{t_p + f_n}, \tag{3.5}$$

where f_n stands for false negative.

As in the official ActivityNet challenge (Heilbron et al. (2020)), for the TAP task the Average-Recall (AR) is defined as the mean of the recall values computed for the set of $tIoU$ thresholds $[0.5, 0.95]$, using a step of 0.05. Note that for THUMOS14, $tIoU$ thresholds are $[0.5, 0.9]$, using a step of 0.1. On the other hand, the Average Number of Proposals per video (AN) is defined as the total number of action proposals that is allowed to collect for each video. When plotting the AR-AN curve, the maximum AN value is set to 100. For the comparison of methods, it is also reported the area under the AR-AN curve (AUC).

Implementation Details

The SVC-UAP solution is entirely implemented using Python. Concretely, with the Scikit-Learn library (Pedregosa et al. (2011)).

As a fully unsupervised approach, annotations of any kind are not used during the training phase. An unsupervised training typically involves finding the parameter configuration that offers the best results. More precisely, during training, the set of trainable parameters of the SVC-UAP are varied. For each dataset, the configuration that has obtained the best performance, in terms of the AR-AN metric, is saved. Afterwards, the best SVC-UAP setting is directly run on the test set.

As it has been described in Section 3.1.2, the approach is agnostic to the backbone kernel used in the SVC. Specifically, linear and RBF kernels are used during the experiments. The rank-pooling-based filter is implemented by applying the temporal smoothing to the input data, as well as the non-linearity function detailed in Section 3.1.3. Moreover, before learning the linear regressor to obtain the corresponding dynamics, the data is l2 normalised. Table 3.1 shows all the SVC-UAP variants that have been used in the experiments. They are named depending on the components that are incorporated.

Table 3.1: Variants of the SVC-UAP method used in the experiments.

	Linear Kernel	**RBF Kernel**	**Rank Pooling**
SVC-UAP-linear	✓	✗	✗
SVC-UAP-linear-rp	✓	✗	✓
SVC-UAP-RBB	✗	✓	✗
SVC-UAP-RBF-rp	✗	✓	✓

As a sanity check baseline, a random method RAND, which generates random action proposals, is also implemented. It technically generates a set of random time intervals as proposals within a video and assigns random scores to them.

For all approaches and datasets, the C3D network designed by Tran et al. (2015) serves as the feature extractor. Precisely, the model is configured to have a temporal resolution of 16 frames with an 8-frame overlap between consecutive inputs and features consists of the activations of the second fully-connected layer (named fc7). For ActivityNet, the dimensionality of the features is reduced with PCA from 4096 to 500, as proposed in the ActivityNet challenge (Heilbron et al. (2020)). For THUMOS14, the vector sized is not varied. It is very important to note that the features used are fine-tuned on the Sports1M dataset (Karpathy et al. (2014)) and not on any of the concerned datasets. This way, any model/features selection for specific data is avoided. Otherwise, it could be considered a violation of the unsupervised character of this work.

3.2.2 Experimental Evaluation in Activitynet

Ablation Study

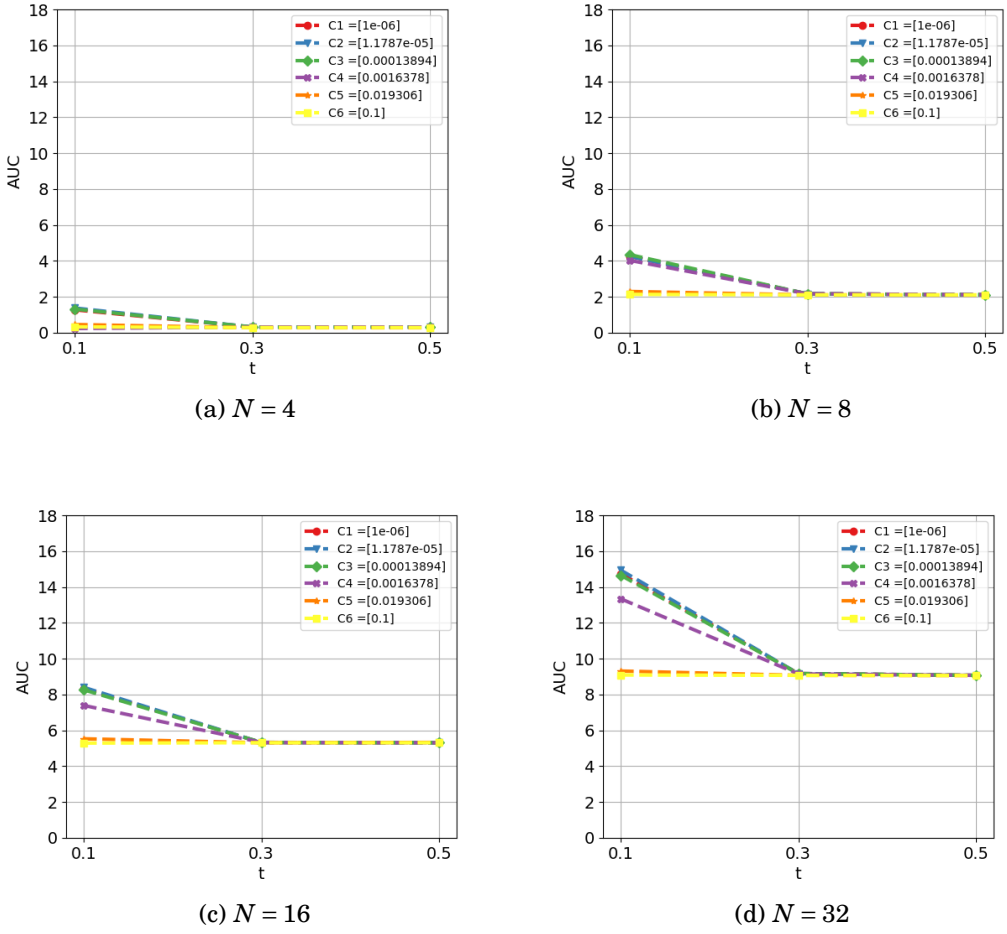
This section analyses the influence of each of the parameters and parts of the SVC-UAP model. All the experiments that have been conducted, and the involved parameters, are described in Table 3.2.

Table 3.2: This table contains a brief description of each of the experiments that have been conducted in the ablation study. The three experiments are consecutive and the best parameter configuration in each is included in the following.

	Modified Parameters	**Fixed Parameters**	**SVC-UAP Variant**
Experiment 1	$t = [0.1, 0.3, 0.5]$ $N = [4, 8, 16, 32]$ $C = [1e^{-6}, 1.1787e^{-5}, 1.3894e^{-4}, 1.6378e^{-3}, 1.9306e^{-2}, 1e^{-1}]$	-	SVC-UAP-linear
Experiment 2	$r = [1, 2, 3, 4, 5, 7.5, 10, 20, 30]$	$t = 0.1$ $C = 1.1787e^{-5}$ $N = 32$	SVC-UAP-linear-rp
Experiment 3	$N = [32, 64, 128, 256, 512, 1024]$	$t = 0.1$ $C = 1.1787e^{-5}$ $r = 1$	SVC-UAP-linear-rp

Experiment 1. Influence of main parameters. The first experiment studies the influence of the main parameters of the model beginning from a basic configuration with a linear kernel for the SVC module and no rank-pooling filter. The evaluated

parameters are: the number of samples added in each iteration (N), the threshold for the online clustering/aggregation (t) and the regularisation parameter of the linear kernel (C) of the SVC. The range of values used for each parameter is shown in the Experiment 1 description in Table 3.2. Figure 3.4 shows the evolution of the AUC when varying the parameters.



(a) $N = 4$ (b) $N = 8$

(c) $N = 16$ (d) $N = 32$

Figure 3.4: Ablation study on ActivityNet dataset. Performance in terms of AUC of the AR-AN curve when varying the parameters N, t and C. Basic configuration: linear kernel for the SVC module and no rank-pooling filter.

From Figure 3.4, several important conclusions can be drawn. First, the influence of the threshold t remains stable for all N and C: the higher this threshold is, the more restrictive the model is when generating action proposals. Specifically, $t = 0.1$ offers optimal results. Second, the analysis reveals the best performance is obtained

when $C = 1.1787e^{-5}$. Interestingly, this holds for all the evaluated combinations of parameters. Finally, the parameter N makes the difference. The performance increases for higher values of N, i.e., incorporating bigger groups of features at each iteration appears more beneficial. This phenomenon is deeply discussed below, after including the rank-pooling filter. The ablation study continues using the winner configuration of parameters from the previous analysis, that is: $C = 1.1787e^{-5}$, $N = 32$ and $t = 0.1$.

Experiment 2. Adding the rank-pooling filter module. The next objective is to validate the effectiveness of the rank-pooling-based filter module. Figure 3.5 showcases the variation in the AUC of the AR-AN curve when the rank-pooling is integrated. Concretely, the influence of the rank-pooling parameter r is evaluated. High values of this threshold lead to a more restrictive rank-pooling filter which discards more candidate proposals. Conversely, lower values mean that fewer proposals are discarded. As it is an unsupervised scenario, it is only possible to trust in the features and assume that action and background are differentiable enough, though there may exist action features similar to that of background. As this cannot be controlled, the balance situation should be found. Such situation appears when $r \in [1,4]$: the rank-pooling filter is able to discard proposals without dramatically losing performance. Table 3.3 shows the difference in the performance when adding this module. Although there is a slight improvement in AUC, the filter also improves the approach from the perspective of efficiency. The fact of having the same or better performance with fewer proposals suggests that the precision of the method is increasing.

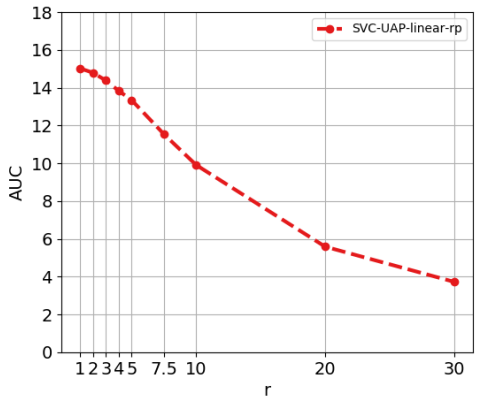


Figure 3.5: Evolution of the AUC of the AR-AN curve when varying the threshold r of the rank-pooling filter. The model is configured to use a linear kernel for the SVC module.

Table 3.3: Comparison between SVC-UAP-linear (without rank-pooling) and SVC-UAP-linear-rp (with rank-pooling).

	SVC-UAP-Linear	SVC-UAP-Linear-rp
AUC	14.94	**15.02**

Experiment 3. Influence of parameter N. As pointed above, in light of the results in Figures 3.4 and 3.6 configuring the parameter N seems very relevant to have a good performance. Specifically, Figure 3.6 shows the performance of the SVC-linear-rp model for $N = [32, 64, 128, 256, 512, 1024]$, where the AUC reaches a plateau for $N \geq 256$. Analysing the statistics of the ActivityNet dataset, one discovers that: (a) each video has on average 3 annotated actions; and (b) annotations cover 55% of the duration of the videos. These two facts mean that actions in this dataset are long. Therefore, generating longer proposals results more beneficial. This parameter controls how big the new group of features that is incorporated in each iteration is and, thus, it is indirectly controlling the duration of the proposals.

Overall, the use of each of the modules that have been proposed in this work is supported by the results reported in this ablation study. For the rest of the experiments on this dataset, the discovered optimal parameters are used.

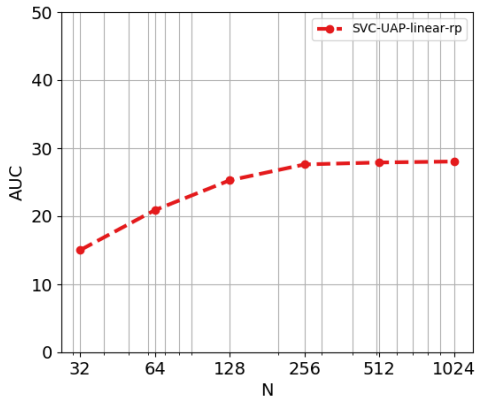


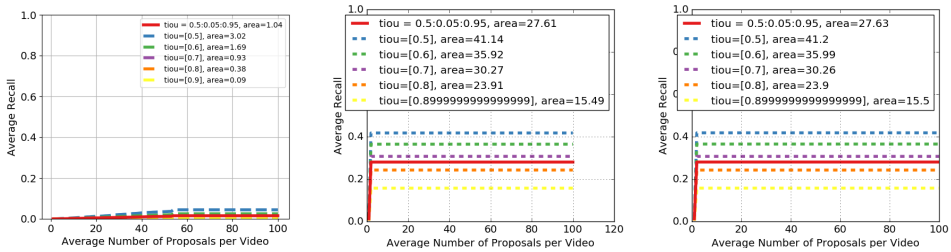
Figure 3.6: Evolution of AUC of AR-AN curve when N increases.

Overall Results on ActivityNet

This section presents the main results of the SVC-UAP approach with the configuration derived from the ablation study. It can be compared to the fully-supervised

approaches that are in the state of the art of the TAP task. Figure 3.7 includes the performance of the approach proposed in this chapter when using different kernels for the SVC module, as well as that of the RAND baseline. This last random model achieves only an AR of 1.04, which means that TAP on ActivityNet is a complex task. From Figures 3.7b and 3.7c one can conclude that for this dataset the RBF and linear kernels offer the same performance. For this reason, and based on efficiency, it is more practical to use the approach with the linear kernel, especially for an online setting.

It is worth recalling that the proposed approach works in an online fashion, which means that the generated proposals do not overlap. The number of proposals generated is typically lower than that of the offline proposal methods. This leads to always having a plateau after a few AN, as seen in Figures 3.7b and 3.7c.



(a) Random Baseline (b) SVC-UAP-linear-rp (c) SVC-UAP-RBF-rp

Figure 3.7: Performance in terms of AR-AN for the problem of TAP on AcivityNet dataset. Comparison of the proposed approach with different kernels for the SVC module and the random baseline.

As for the state-of-the-art comparison, Table 3.4 shows the performance in terms of AUC in the AR-AN curve for the current TAP state-of-the-art supervised models. It is observed that the best supervised model achieves 67.10 of AUC@100 proposals. Thus, the proposed unsupervised models are able to recover 41% of the performance of this model, showing a promising direction for unsupervised approaches.

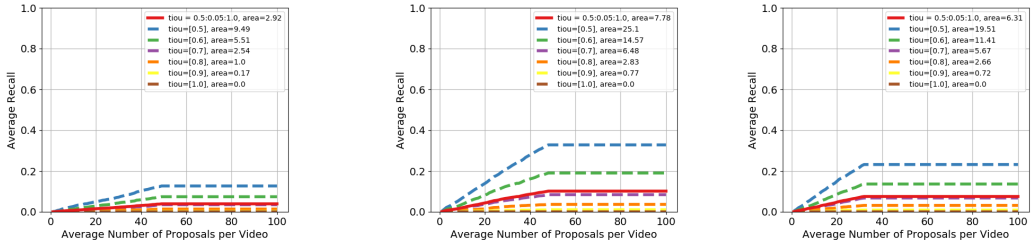
Note that all state-of-the-art models present an offline setting where the full video must be analysed to cast proposals. Additionally, these methods generate thousands of overlapped AP per video. On the contrary, all SC-UAP variants work completely online, generating proposals as videos evolve and in a more efficient way since the number of AP is, by far, smaller.

Table 3.4: Comparison with the state-of-the-art for the problem of TAP on ActivityNet. The superscript *s* indicates the method is supervised. The best supervised model achieves 67.10 of AUC@100. The best unsupervised model achieves 27.63, so it is able to recover 41% of the best performing method.

	AUC
Dai et al. (2017)^s	59.58
Lin et al. (2017)^s	64.40
CTAP (Gao et al. (2018))^s	65.72
BSN (Lin et al. (2018))^s	66.17
BMN (Lin et al. (2019))^s	67.10
SVC-UAP-linear-rp	27.61
SVC-UAP-RBF-rp	27.63

3.2.3 Experimental Evaluation on THUMOS14

For the sake of a thorough experimental evaluation, the performance of the SVC-UAP is also reported on the THUMOS14 dataset. Both linear and RBF kernels are used for the SVC module, and the parameter configuration is: $N = 8$, $t = 0.09$, $C = 0.019306$ and $r = 0.1$. As with the previous experiments for ActivityNet, Figure 3.8 the AR-AN curves of the approach when using different kernels, as well as that of the random baseline. Additionally and following what is done in the literature, Table 3.5 presents the results in terms of AR-AN with a maximum of 50 and 100 proposals per video (AR@50 and AR@100).



(a) Random Baseline (b) SVC-UAP-linear-rp (c) SVC-UAP-RBF-rp

Figure 3.8: Performance in terms of AR-AN for the problem of TAP on THUMOS14 dataset. Comparison of the proposed approach with different kernels for the SVC module and the random baseline.

Figure 3.8 shows that the gap in the performance between the SVC-UAP variants and the baseline is smaller than that reported for ActivityNet. This means that: (i) the random model is adequate as a baseline for the problem and (ii) this dataset is

Table 3.5: Comparison with the state-of-the-art for the problem of TAP on THUMOS14. [s] indicates the method is supervised. The best method achieves a 10.16 % of recall with only 50 proposals. This represents 26% of the performance of the better (BMN by Lin et al. (2019)) supervised state-of-the-art model.

	AR@50	**AR@100**
SCNN-prop (Shou et al. (2016))[s]	17.22	26.17
SST (Buch et al. (2017))[s]	19.90	28.36
CTAP (Gao et al. (2018))[s]	32.49	42.61
BSN (Lin et al. (2018))[s]	37.46	46.06
BMN (Lin et al. (2019))[s]	39.36	47.72
SVC-UAP-linear-rp	10.16	10.16
SVC-UAP-RBF-rp	7.53	7.53
Random baseline	3.96	3.96

even more challenging than ActivityNet.

After analysing some of the statistics of the dataset to better understand the results, first, the average duration of the annotated actions is less than 5 seconds, as it is shown in Figure 3.9. Second, there are about 15 instances per video, covering only 20% of the content. These numbers indicate that THUMOS14 is a much more sparse dataset with shorter action segments than ActivityNet. The sparsity of the dataset is consistent with the shape of the curves in Figures 3.8b and 3.8c. This fact suggests that TAP methods need to throw more proposals to improve the recall, but the online condition of the proposed approach clearly increases the challenge. While it cannot apply any post-processing to the generated proposals, many offline state-of-the-art methods do it to maximise the recall. In an offline setting, the approaches have access to the whole video. In this work instead, this maximisation is absolutely dangerous, as precision can be forgotten, which in an online scenario would be disastrous.

As for the comparison to the state of the art, the solution with a linear kernel achieves a 10.16 of recall value with only 50 proposals per video and *without* any supervision. This represents 59% and 26% of the performance of the worse (SCNN-prop by Shou et al. (2016)) and the better (BMN by Lin et al. (2019)) supervised state-of-the-art models, respectively. These results are sufficiently motivating to continue investigating on the unsupervised paradigm for action proposals.

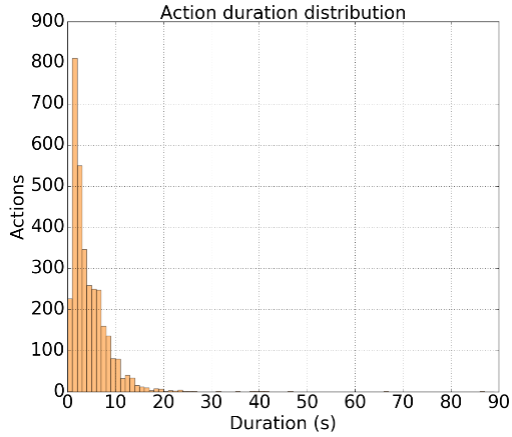


Figure 3.9: Ditribution of the duration of the actions in THUMOS14.

3.3 Conclusions

Throughout this chapter it has been presented a simple, unsupervised, online and efficient classification-based method for the problem of TAP. This approach generates candidate action proposals through an SVC, capable of grouping consecutive sets of frame features in a certain video to create time boundaries that define action candidate segments. It has also been proposed a filtering module which uses the rank-pooling over the dynamics of the candidates segments to discard those that belong to the background of the video. It is important to note that supervision of any level is not applied to the model during training: no action annotations are used, as well as the feature extraction network has not been pre-trained in any of the concerned datasets.

As far as it is known, this is the first time that a thorough experimental evaluation of an unsupervised approach is presented on the two main TAP benchmarks: ActivityNet and THUMOS14. The ablation study that has been conducted on the ActivityNet dataset justifies the integration of each part of the approach and supports the working hypotheses upon which each module is based. Although the datasets show different natures in their annotations, all proposed SVC-UAP variants are capable of adapting to both of them. When comparing to the state of the art, the best SVC-UAP configuration achieves more than 41% and 26% of the recall performance of the best supervised models for ActivityNet and THUMOS14 datasets, respectively.

Video datasets are growing enormously and, consequently, their labelling becomes

a very expensive task. Having systems that can work without requiring labels of any kind, as the one it was proposed here, is of great interest. This fact, coupled with the online nature of the SVC-UAP method, makes it possible to conclude that the work described in this chapter is a promising new paradigm for the TAP task. Different from the current state-of-the-art approaches, SVC-UAP can provide the proposals as they are obtained to a certain action classifier so the action can be detected as soon as it occurs.

The code and results are publicly available[1] so that others can reproduce them and explore this novel unsupervised TAP perspective.

[1]URL of the repository: https://github.com/gramuah/svc-uap

Chapter 4

Understanding Online Action Detection

This chapter is focused on the problem of localising actions in untrimmed streaming videos. More specifically, on the online perspective of it, where actions must be detected as soon as they happen. This particular task was coined as Online Action Detection (OAD) by De Geest et al. (2016).

As seen in Chapter 2, the broad and challenging topic of Action Detection has been studied by the Computer Vision community with a special interest in recent years (Shou et al. (2016); Yeung et al. (2016); Shou et al. (2017); Buch et al. (2017); Gao et al. (2017a); Shyamal Buch & Niebles (2017); Zhao et al. (2017); Gao et al. (2017); Dai et al. (2017); Chao et al. (2018); Xu et al. (2019a)). However, all these cited works share an important aspect: they assume that the whole video is available beforehand to make predictions on it. Thus, they perform Offline Action Detection (OffAD).

If one thinks of a robotic platform that must interact with humans in a realistic scenario or an intelligent video surveillance application designed to raise an alarm when a particular action is detected, it would be difficult to use any offline method in these applications as they would detect relevant situations once they have occurred. These cases need OAD approaches.

In OAD, any method should be able to work with partial observations of a video stream to detect actions. Note that these action segments are likely to be the exception rather than the rule, compared to non-relevant (background) segments. Ad-

ditionally, this online definition allows for an important property: action prediction. Ideally, an action should be detected even before it is fully performed. Observing Figure 4.1, any OAD method should recognise both the background and the action and reduce the background-to-action transition time (δ in the Figure) to the minimum possible.

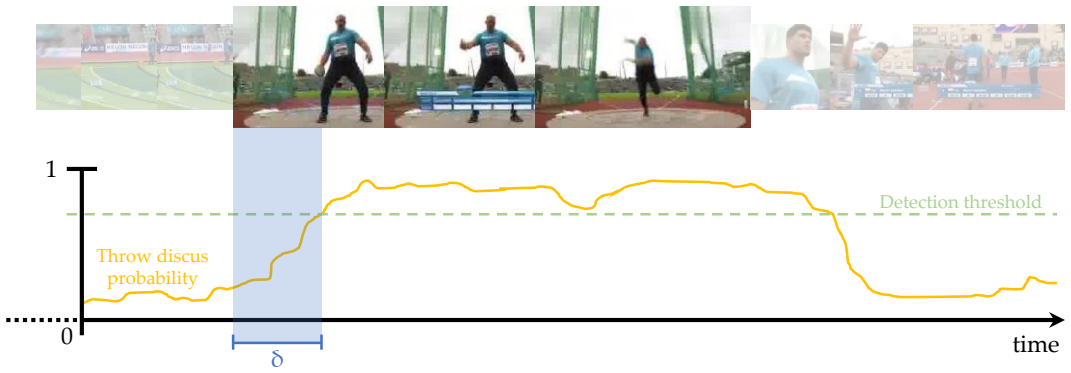


Figure 4.1: **Predicting actions at early stages with Online Action Detection.** Action Prediction refers to the problem of inferring the label of the action that is being performed on a certain trimmed video with the minimum action observed. The OAD task is more challenging since it not only requires the methods to be able to differentiate the action from the background, but also to detect the action as soon as it appears and with few observations of it. Therefore, the objective for a certain OAD model is to recognise both the background and the action and reduce the background-to-action transition time (δ in the Figure) to the minimum possible.

There exists few works that addressed this OAD setting, *e.g.* De Geest et al. (2016); Gao et al. (2017b); De Geest & Tuytelaars (2018). However, all of them present major weaknesses in two key aspects: a) the evaluation metric; and b) the way in which background is considered by both models and evaluation protocols.

Regarding the evaluation, the common experimental set-up includes two datasets: THUMOS14 (Idrees et al. (2017)) and TVSeries (De Geest et al. (2016)). Despite the fact that the problem remains the same for each dataset, two different metrics are proposed for each one of them: the mean Average Precison (mAP) and the calibrated Average Precision (cAP), respectively. This lack of consensus hinders interpreting the general performance of the method, since each metric generates different information. Moreover, the metrics cannot be said to be of an online nature. They do not provide information about the instantaneous performance over time of the solutions. Given a test video, the whole set of action detections has to be accessed and sorted

(by their score) after the method is executed. So they need to be computed entirely offline.

In relation to the second aspect, an OAD setting is characterised by untrimmed videos where actions appear sparsely and the background (non-relevant segments) predominates. Consequently, this setting should demand this last category to be handled equally important to all other actions. However, almost all the online methods published to date have been designed to cast a specific action prediction even for those parts of the video with non-relevant content. De Geest et al. (2016) proposed solving this issue by calibrating the mean average precision to mitigate the effects of wrong predictions during background segments. Their strategy does not consider the background as another class so methods are not encouraged to learn to deal with it. With this situation, it is interesting that when the background class is not considered in the evaluation but in the annotations, all proposed metrics cannot saturate to the maximum they have been designed for. In other words, the maximum of a precision-based metric will never be of 100% even if the method cast the correct category for every action frame.

Chapter's contributions

The objective of the current chapter is to better understand the problem of Online Action Detection, from its definition to its limitations. To this end, the following contributions have been made:

- A solid definition of the OAD problem is offered. The one made by De Geest et al. (2016) is revisited to clarify it and new key conditions are added to improve its completeness.

- A deep study of the currently used evaluation protocols, their metrics and their limitations is carried out. Additionally, a set of new conditions are stated as well as a new, more adequate evaluation protocol is introduced. This protocol is based on a novel *online* metric: the Instantaneous Accuracy (IA). An overview of this new protocol and its comparison to the previous ones is given in Figure 4.2.

- Both the metrics and the current state-of-the-art methods as well as the IA are tested on THUMOS14 (Idrees et al. (2017)), TVSeries (De Geest et al. (2016))

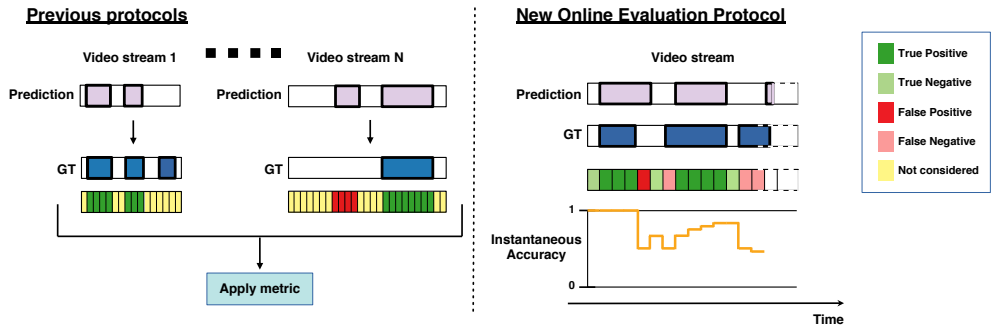


Figure 4.2: **Online Evaluation Protocol**. Previously used evaluation protocols for Online Action Detection (OAD) were based on: 1) running the online methods through all the videos in the dataset; 2) applying the offline metric to the obtained results. Additionally, offline metrics proposed so far do not consider the background in their evaluation. In this work, an Online Evaluation Protocol based on the new Instantaneous Accuracy (IA) metric is proposed. With it, OAD approaches are evaluated online, considering the background and regardless of the length of the video.

and ActivityNet1.3 (Heilbron et al. (2015)) datasets. The experiments demonstrate all the mentioned limitations and how the new metric helps to overcome them.

4.1 Online Evaluation Protocol for Online Action Detection

It is quite interesting that the Online Action Detection (OAD) topic has been barely explored considering the wide range of real-world situations in which an OAD-based system could be deployed, such as human-robot interaction, medical applications, manufacturing processes, video surveillance, sports or even video games (see Chapter 1).

In 2016, the pioneer work of De Geest et al. (2016) stated that OAD needs a solid definition and a strong evaluation protocol. As of today, this work demonstrate that the topic does still need both a better definition and a stronger evaluation protocol. These two aspects are revisited along this section.

4.1.1 Online Action Detection

The task of OAD in realistic scenarios is clearly defined by the following basic properties:

1. **Streaming videos** are assumed. This means that neither the length of the video nor the content is known. If one thinks of a surveillance scenario, the video would be recorded and analysed simultaneously, until an unknown end.

2. Actions must be **detected as soon as they happen**, ideally in real-time. They must be captured with the minimum time lag possible with respect to their initial instants.

3. **Detections must be causal.** For this task, the future time is absolutely not known. Therefore, it cannot be used to make any present prediction.

Despite the fact that OAD is naturally characterised by untrimmed streaming videos where actions appear sparsely, it can be found state-of-the-art models which do not take into account the background as a category **?**. Instead, these methods consider the task as a per-frame labelling problem where detecting action ground truth segments is what only matters. Mislabelled background segments are dismissed. Consequently, these methods will not be able to achieve the maximum of a precision-based metric even if the method cast the correct category for every action segment. This effect will be shown in Section 4.2.

In addition to redefine and revisit the properties of the OAD task itself, this work also proposes the following properties that any OAD-based method should comply with:

1. Both action and background segments must be explicitly discriminated.

2. No post-processing or posterior thresholding to action label scores can be applied. Decisions on frames are to be made at the moment of execution.

3. Methods cannot revisit past detections. Once the decision is made, it cannot be changed.

4.1.2 Online Evaluation Protocol: The Instantaneous Accuracy

An evaluation protocol is usually in line with the task for which it was designed. However, this is not the case in Online Action Detection. The few works on this topic have used evaluation metrics from the different task of Offline Action Detection. Particularly, De Geest et al. (2016) came up with a new yet insufficient protocol, as it will be shortly shown. Consequently, a new evaluation protocol for OAD is needed. This protocol must meet the following conditions:

(**C1**): **An online video-level metric is needed.** With it, the performance can be evaluated as a video grows without having to wait to an unknown end.

(**C2**): **Background detection ability must be measured.** If an OAD method has to be able to detect background, the evaluation protocol must also measure such ability.

(**C3**): **Dynamic action/background ratio.** The value of a *true* factor –true positive (action) and true negative (background)– should be conditioned to the negatives vs. positives ratio, which must be dynamic and based only on the seen portion of the video.

Previous metrics

All previous evaluation protocols use class-level metrics which have to be applied offline, at the end of the test time, accessing the whole set of action annotations in a given test video. Hence, condition (C1) is directly violated. These protocols are mainly based on using the per-frame mean Average Precison (mAP) or its calibrated version: the calibrated Average Precision (cAP). As an example, one can check how the works of De Geest et al. (2016), Gao et al. (2017b) and De Geest & Tuytelaars (2018) use them in their experiments. Further investigation on how these protocols are employed is given in the experiments section of this chapter (Section 4.2).

Regarding mAP, it measures the precision (Equation 4.1), across all classes. As can be seen in its definition, only positives factors (actions) are considered and their value is always the same regardless of any ratio. This means that conditions (C2) and (C3) are not complied.

Precision in cAP is expressed as in Equation 4.2. This metric was introduced by De Geest et al. (2016). It balances the precision with the w parameter, which is the

ratio between negative and positive frames. It is basically a modification of mAP metric where conditions (C1) and (C2) are still not complied. It would solve condition (C3), but w is computed a priori (not dynamically) using previous information about all videos and action categories.

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{4.1}$$

$$\text{cPrec} = \frac{w\text{TP}}{w\text{TP} + \text{FP}}. \tag{4.2}$$

Besides, due to the nature of these metrics, coupled with the fact that some methods (*e.g.* De Geest et al. (2016); Gao et al. (2017b)) do not detect the background, it is not possible for the performance to reach a 100%, even if all action frames are correctly classified, as it will be shown in the experiments section (Section 4.2).

To solve these issues, the thesis presented here introduces a new evaluation protocol that satisfies all the conditions and utilises a novel metric, called Instantaneous Accuracy (IA).

Instantaneous Accuracy: mathematical formulation

Considering a set of $\mathcal{N}$ test streaming videos, for each video $\mathcal{V}_i$, where $i = 1,\ldots,N$; an OAD method generates decisions about the label (an action category or background) of the video at each instant of execution. Usually, any previously used metric would collect first the set of decisions (action detections) defined by their initial and ending times, sort them by any kind of scoring and evaluate them according to a particular equation.

This work introduces a new metric which meets all the aforementioned conditions: the Instantaneous Accuracy (IA(t)). In contrast to the common described protocol, this metric is designed to evaluate methods online, *i.e.* during execution.

Time is a continuous variable, and so Instantaneous Accuracy should be as well. However, such an ideal assumption is not valid for a real scenario. Same as online methods show a time lag between predictions (frames, chunk of frames,...), IA is also defined by a delta time (Δt) which reflects how often it is computed throughout the video.

The metric takes as input the set of decisions made by the method from the be-

ginning of the video up to the current instant of execution. With this information, the IA is calculated every Δt along the considered time range. Different factors are then generated according to the relationship between predictions and ground truth at each instant: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). In this new protocol, action categories are positives while background is negative; and the possibility of multiple actions occurring simultaneously is also evaluated with this metric, though most datasets do not show this situation, nor most state-of-the-art methods can hardly give multiple predictions.

For a particular instant of time $0 < t \leq T_i$ during a video $\mathcal{V}_i$, where T_i encodes the video temporal length, the IA(t) is expressed as the instant-level classification accuracy:

$$\mathrm{IA}(t) = \frac{\sum_{j=0:\Delta t:t'} \mathbf{tp}(j) + \sum_{j=0:\Delta t:t'} \mathbf{tn}(j)}{\mathrm{K}}, \tag{4.3}$$

where vectors **tp** and **tn** encode the true positives (action) and true negatives (background), respectively. The term K represents the total population considered until time t, which is dynamically obtained as follows:

$$\mathrm{K} = \left\lfloor \frac{t}{\Delta t} \right\rfloor. \tag{4.4}$$

As described so far, IA is a video-level metric that considers both action and background in its formulation. Thus, conditions C1 and C2 are fulfilled. To meet condition C3, it is also proposed the weighted version of the IA: the wIA. It has the same formulation as that of IA, but scaling the *true* factors by the ratio between accumulated background and action instants at each Δt:

$$\mathrm{wIA}(t) = \frac{\sum_{j=0:\Delta t:t} \mathrm{w}(t) \cdot \mathbf{tp}(j) + \sum_{j=0:\Delta t:t} \frac{1}{\mathrm{w}(t)} \cdot \mathbf{tn}(j)}{\mathrm{K}}, \tag{4.5}$$

with the term w(t) representing the dynamic ratio between background and action instants accumulated at each Δt until time t in the ground truth, *i.e.* in $\mathcal{V}_i(0:t)$.

Both IA and wIA versions only use information from the past and adapt their parameters in each iteration. Although they would be sufficient to evaluate an OAD method on a single video stream of any length, it is also introduced in this work the mean average Instantaneous Accuracy (maIA), shown in equation 4.6. It summarises the IA or the wIA performance across a dataset so researchers can compare their

methods:

$$\text{maIA} = \frac{1}{\text{N}} \sum_{i=1:\text{N}} \left(\frac{\Delta t}{T_i} \sum_{j=0:\Delta t:T_i} \text{IA}(j) \right). \tag{4.6}$$

Instantaneous Accuracy: implementation details

Despite the fact that the mathematical formulation of IA is clear and simple, certain conditions and restrictions come up when implementing it. The whole metric presented in this work has been developed in two consecutive versions: IA-v1 and IA-v2.

The two of them share the way of collecting both the results and the ground truth information. However, the second introduces some features that make it more fair, configurable and suited to the problem. All the features are described below.

Slot. It is the basic unit of the metric in both versions. It corresponds to the parameter Δt. When the metric receives, at a certain instant of execution, the predictions and the annotations, it builds a grid such that it represents the video from the beginning to the current instant. Each element of the grid will contain the corresponding factor (TP, TN, FP or FN) depending on the relationship between predictions and annotations. This way, the slot represents the theoretical instant of evaluation.

Weighting policy. This feature is implemented in IA-v1 and IA-v2 and, as explained in the previous section, it tries to adjust the difference between the number of ground truth action and background instants encountered up to the moment of execution. During implementation, these weights are the ratio between action and background slots. However, this ratio can only exist when both action and background exist and so, the weighting begins to be applied when actions and background slots are present in the video. Otherwise, the metric values calculated at each grid when only one kind of slot is seen are not weighted. This particular situation happens at the beginning of the videos, where only actions or background are shown during several slots.

Slot coverage. The slot coverage is introduced in IA-v2 to deal with the situation of having only part of the observed slot occupied by an action annotation, typically the transition moment between action and background. An action will be considered to be happening within a slot if its coverage is above a certain configurable threshold in the form of a percentage.

Multi-action mode. IA-v2 considers the situation of having several actions in the ground truth or in the predictions. Each of them will be analysed independently and

Figure 4.3: **Instantaneous Accuracy implementation.** Illustrative example of how the slot, the grid and the factors are actually built before computing the Instantaneous Accuracy during a certain part of a video. Here, the slot is configured to 1 second and the slot coverage threshold is 50%.

will generate the correspondent factor in the grid.

A summary of the implemented features in both versions as well as an illustrative example of the metric are shown in Table 4.1 and Figure 4.3, respectively.

	Slot	Weighting policy	Slot coverage	Multi-action
IA-v1	✓	✓	✗	✗
IA-v2	✓	✓	✓	✓

Table 4.1: Summary of implemented features in both versions of the Instantaneous Accuracy (IA) metric.

4.2 Experiments

This section aims to both prove the weaknesses of the commonly proposed metrics for Online Action Detection and show the advantages of using the new Instantaneous Accuracy metric introduced in the current chapter. It also experimentally demonstrates the ambiguities of some state-of-the-art methods when evaluated with different metrics, as well as how IA can harmonise the results so they can be better understood.

4.2.1 Experimental Set-up

Details on the datasets, metrics and baselines used to obtain all experimental results reported here.

Datasets

Three different datasets were used for the experiments: THUMOS14 (Idrees et al. (2017)), TVSeries (De Geest et al. (2016)) and ActivityNet v1.3 (Heilbron et al. (2015)). All of them provide untrimmed videos where action and background segments coexist, suiting the Online Action Detection scenario.

THUMOS14. It has temporal annotations for a set of 413 videos, covering 20 sport classes. On average, every video contains 15 action annotations. The 200 videos from the validation set are used for training, while the remaining 213 from the test set are used at test time.

TVSeries. This dataset was specifically designed for OAD. It contains 27 episodes from 6 popular TV series with 30 realistic action categories annotated. Its large variability (occluded, multiple persons or non-relevant actions, among others), makes it a really challenging dataset.

ActivityNet v1.3. Due to its popularity in the field of Action Detection, this work also integrates it in the experiments. It is a large scale dataset specifically designed for Temporal Action Localisation, which contains about 20K untrimmed videos for 200 action classes. The average number of action instances per video is of 1.5. For this dataset,the training set and the validation set are used during training and test, respectively. While both THUMOS14 and TVSeries have been already used within the OAD context, the current work is the first in using this dataset for the online setting.

Evaluation Metrics

To compare the performance of the baselines to that of the state-of-the-art methods, the set-up used by Gao et al. (2017b) is followed: the results on THUMOS14 are analysed with the per-frame mAP, described by Equation 4.1, while those on TVSeries with the cAP (De Geest et al. (2016)) metric defined by Equation 4.2.

To evaluate all possible methods according to the new protocol proposed in this

thesis, first, the IA is computed for all videos in each dataset. Afterwards, the maIA is calculated to have a comparison of the performance across each dataset. The two modes of the Instantaneous Accuracy (weighted and non-weighted) are used in the experiments with the parameter Δt set to 0.5 seconds, as well as the two implemented versions: IA-v1 and IA-v2.

Baselines

Apart from state-of-the-art methods, the work is also supported by the evaluation of three baseline models.

All background **(All-BG)**. This baseline simulates a model which never outputs an action category but it always predicts background. Apart from revealing some limitations in the metrics, it also helps to understand the degree of complexity of the datasets.

Ignoring Background Model (**IBM**). To show the importance of detecting the background, as well as to demonstrate why previous metrics do not encourage this fact, this work makes use of a model which always assigns correct labels to ground truth action frames. Basically, it produces a random action label for every background frame. Thus, it is acting as if the background did not exist during the model's training. If one thinks of this method working in a realistic surveillance scenario, it will always be rising false alarms. It can be also understood as the counterpart of the **All-BG** baseline.

3D Convolutional Network (**3D-CNN**). As it is shown in Figure 4.4, this baseline consists in a 3D convolutional network trained to discriminate between all labels, *i.e.* all action categories and the background. The goal is to establish baseline results for the new online evaluation protocol for OAD with a model capable of explicitly detecting actions and background.

3D-CNN is based on the C3D network designed by Tran et al. (2015). Technically, the dimension of the last fully connected layer is modified so that it coincides with the number of action categories *plus* the background. The architecture is fed with 16-frame-length chunks.

For training, the chunks are extracted contiguously. Those whose intersection with the ground truth is greater than the 80% are marked as action chunks, otherwise they are considered as background chunks. As this kind of videos usually contain

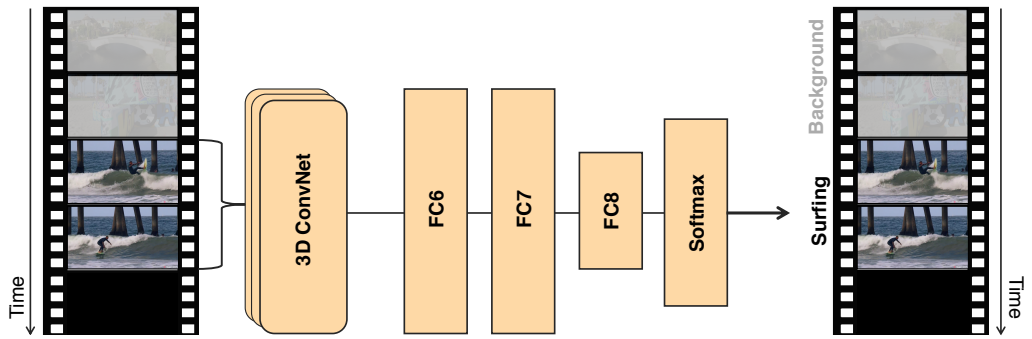


Figure 4.4: **3D-Convolutional Network baseline**, based on the C3D network by Tran et al. (2015). The model is trained to discriminate between all actions and background. The dimension of the last fully connected layer is modified so that it coincides with the number of action categories plus the background. It makes predictions in an online fashion, avoiding to peek into the future for adjusting or post-processing the detections.

more background, the training data $\mathcal{T}$ that results from extracting all chunks is unbalanced. This is solved by matching the number of samples in each class: $N_{\mathcal{T}} = \frac{N_{chunks}}{C}$, being C the total number of classes including background.

C3D model weights from Sports-1M (Karpathy et al. (2014)) dataset are used to initialise the network. SGD is configured with learning rates 10^{-3}, 10^{-4}, 10^{-5} for THUMOS14, ActivityNet v1.3 and TVSeries, respectively. Momentum is fixed to 0.9 for all datasets. The model is trained for 15 epochs with the learning rate decreasing every 2 epochs by a factor of 10.

During test, the online process is simulated on each video by gathering 16 non-overlapping frames. They are the input to the network, which will cast a prediction. To make the decision on the chunk, if the softmax value corresponding to the background class is above 0.8, it is considered as background. Otherwise, the detection will be the action class with highest softmax score.

One can notice that 3D-CNN not only is simple but does not require either refinement or post-processing and, additionally, it can run in real-time (at more than 100 fps). The experimental evaluation shows that it is a strong baseline. The framework Caffe (Jia et al. (2014)) was used for its implementation.

4.2.2 Comparison to Previous Metrics

The experiments in this section demonstrate the main weaknesses of the previous evaluation metrics. To this end, the work presents an analysis using THUMOS14

and TVSeries datasets, where the performance of different methods with each metric is compared. Beyond comparing numbers, the objective is to discuss the conclusions that can be drawn from them to determine how informative the metrics are, as well as to highlight the significant role that the background plays in OAD.

To begin, Table 4.2 shows the per-frame mAP performance on THUMOS14 dataset for the currently supposed state-of-the-art models: RED and MultiLSTM, proposed by Gao et al. (2017b) and Yeung et al. (2018), respectively. It is also shown the performance of the IBM and 3D-CNN baselines.

	State-of-the-Art		**Baselines**	
	MultiLSTM	RED	3D-CNN	IBM
mAP(%)	41.3	45.3	30.1	57.0

Table 4.2: Per-frame mAP performance of THUMOS14

The results of 3D-CNN prove that the model is a strong baseline for this task. Additionally, although the state-of-the-art models are considered online, they do not comply with the online conditions stated in this work (see section 4.1.1).

RED (Gao et al. (2017b) was specifically designed for Action Anticipation (AA). However, the authors claimed that Online Action Detection is a particular case of AA where the anticipation time is zero. The model was trained without using background information and in order to force to keep the sequence of anticipated actions (background not included) by using Reinforcement Learning. For the OAD experiments, they use results when the anticipating time is 0.25 seconds. For all these reasons, while RED can certainly be a good method for AA, it should not be treated as a pure OAD method.

Regarding MultiLSTM (Yeung et al. (2018)), this solution is designed for frame-level action labelling. It is fed with multiple contiguous frames and the output is a per-frame prediction over multiple frames. At each time step, the model predicts the label of the corresponding frame and of a certain amount of previous frames. These overlapped predictions help to refine past predictions. Refining means revisiting previous labels to change them. Since this effect is not allowed in OAD, MultiLSTM can not be considered as a pure online method.

The IBM model shows that even with a metric that does not explicitly consider the background to compute the performance, this kind of methods that are not able to discriminate actions from background will never achieve a 100%. This case is further

discussed with the next table.

For the TVSeries dataset, Table 4.3 shows the results for all the baselines and the state-of-the-art model designed by De Geest et al. (2016) (CNN). Three different conclusions can be derived from the table.

	CNN	All-BG	3D-CNN	IBM
mAP (%)	1.9	0	1.6	30.9
cAP (%)	60.8	0	10.8	96.9
maIA-v1 (%)	3.51	78.31	71.90	-
maIA-v2 (%)	2.45	78.37	71.79	-
weighted maIA-v1 (%)	12.46	22.91	28.95	-
weighted maIA-v2 (%)	8.36	22.86	29.27	-

Table 4.3: Analysis of all the metrics on TVSeries.

First, the fact that IBM casts action categories for background frames affects negatively to the precision. If every frame that belongs to the background counts as a False Positive (FP) when it is assigned an action category, the FP value in Equation 4.1 reaches its maximum. Consequently, a mAP of 30.9% is the maximum value for any IBM-type OAD model. Regarding the results with the cAP metric, the performance of a model which does not care about the background, such as the IBM baseline, in the massively unbalanced TVSeries dataset (sparse action segments) seems to be surprisingly good. This effect is caused by the weight parameter w in Equation 4.2, which rather than balancing the results, it is hiding the errors made by the model in the background segments. Overall, IBM confirms that using methods and metrics that are not capable of managing the background category is not appropriate for Online Action Detection.

Second, as mentioned in Section 4.2.1 and shown by mAP and cAP values, All-BG is a baseline which does not recover any action during any video. This fact also means that no False Positive was generated. However, the ability of not generating these unwanted factors is not measured by the common metrics; so they do not encourage methods to really discard background from action. Additionally, considering background segments as important as those of action is one of the key ideas introduced in this chapter. The IA metric not only makes this possible, but also is able to measure the relevance of each detection (action or background) through its weighted version.

Finally, in terms of mAP, 3D-CNN offers a competitive performance when com-

pared to that of the state-of-the-art CNN and it confirms it to be a proper baseline for the OAD problem. The main difference between the CNN model and the 3D-CNN is that the former does not cast predictions of background category, while the latter does. As demonstrated before, IBM-type methods (such as CNN) benefit from the w parameter of the cAP metric. This fact leads to have a high performance in terms of this metric, even when that of mAP is not so strong. On the other hand, 3D-CNN is trained to also detect background. This model generates less false positives during background segments, but it can also cast background when an action is being performed. This situation is reflected in the gap with the cAP performance of CNN.

In addition to all the above conclusions, it is essential to point out that neither mAP nor cAP are *online* metrics. The results in Tables 4.2 and 4.3 for these two metrics can only be reported once the methods have been executed on all the videos. Instead, the IA metric introduced in this work can perform a true online comparison between OAD solutions, as it is shown in the next section.

4.2.3 Evaluation with Instantaneous Accuracy

This section investigates on the evaluation of the performance of methods in an *online* fashion with the IA metric. The experiments were conducted on the three datasets and using the 3D-CNN baseline and the CNN approach by De Geest et al. (2016).

No result files or code were found for state-of-the-art methods other than CNN and LSTM, both proposed by De Geest et al. (2016). Since their performances are similar, only the results of the first were generated for its simplicity, reproducing the code provided by the authors. Note that these methods, although specifically designed for OAD, do not recognise background.

Instantaneous Accuracy: parameter configuration

As explained in Section 4.1.2, the Instantaneous Accuracy metric relies on three parameters. Figures 4.5 and 4.6 help graphically understand the parameter configuration and the effects on the performance that they may cause.

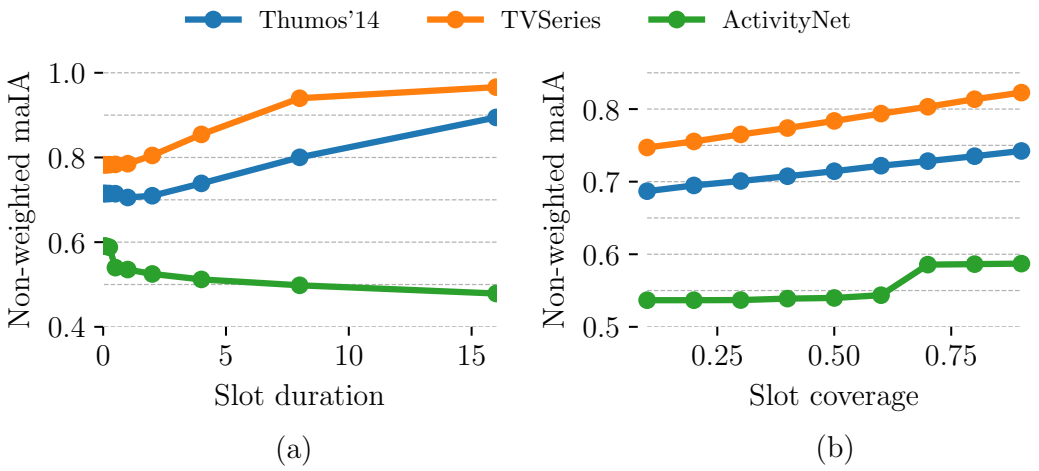


Figure 4.5: **Varying slot parameters.** This figure shows the variation of the non-weighted maIA performance of the All-BG baseline at different values of slot duration (a) and slot coverage (b). The proper slot duration value is found when it is shorter than that of any action. As for the slot coverage, it controls how restrictive should the metric be at transition instants (action to background or vice-verse). Higher level of coverage means that more slot has to be covered to be considered as action.

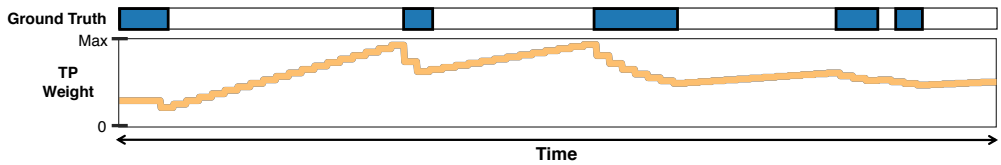


Figure 4.6: **Dynamic weights**. This figure corresponds to a certain video. The upper part of it shows its annotated actions (blue segments). It can be seen how the value of a True Positive (TP) is weighted according to the ratio of negative/positive slots seen up to the current instant of evaluation. The weights of the True Negative factors are inverse to those of TP.

The slot duration corresponds to Δt and represents how often the evaluation is applied. This parameter should be zero, yet such an ideal duration is not achievable. Not all actions have the same duration within a dataset, therefore the slot duration has to be short enough so no action throughout a video is left unevaluated. The effect of this parameter on the performance can be investigated using the All-BG baseline. Concretely, Figure 4.5a shows the performance of this baseline when different values for the slot duration are set. In this experiment, the slot coverage is fixed to 50%, which means that a slot contains an action if it is covered at least by a 50%. In THUMOS14 and TVSeries datasets, the majority of ground truth instances last

no more than 4 seconds. In these cases, the longer the duration of the slot, the more difficult it is for an action to cover it. Consequently, more background is considered and thus, the performance for this baseline increases. On the other hand, action durations in ActivityNet are more diverse and longer, so the performance of the baseline decreases. Not all the actions are being missed during the evaluation. Despite these differences, these three datasets have in common that a balance is found when its duration is shorter than that of any action. Such balance value, which is 0.5 seconds in the experiments, is considered the appropriate. For the rest of the experiments, this parameter is set to the appropriate value.

As well as the slot duration, the slot coverage can also be studied with the All-BG baseline. Figure 4.5b depicts the variation of the performance when different levels of slot coverage are set and when the slot duration is fixed to 0.5 seconds. This parameter controls how restrictive should the metric be in transition instants (action to background or vice-verse). Higher levels of coverage means that more slot has to be covered to be considered as action. Hence, the performance of the All-BG baseline will increase as more slots will be set as background. If necessary, this parameter would allow to configure the importance given to the background. Since this work states that background and action categories are equally relevant, the slot coverage has been configured to 50%.

In the wIA, the value of a true prediction is dynamically adjusted according to the ratio of negative/positive slots seen until the current instant of evaluation time. Figure 4.6 shows this dynamic behaviour in the weights of True Positives (TP) (True Negative weights are the inverse) throughout the video. Action and background are not always balanced at each evaluation instant during the video stream, and so the weight of a TP increases in those portions of the video in which there is no action annotated. This fact represents how the metric is modulating the importance of a correct prediction and it is a very relevant difference in relation to previous evaluation protocols.

Figure 4.6 also demonstrates at the beginning of the video how the weighting policy works: as long as actions and background do not coexist, all weights (TP and TN) are fixed to 1, *i.e.* no weighting is applied. Just as soon as both actions and background are present in the ground truth, the weighting starts to be used.

Online evaluation in streaming videos with Instantaneous Accuracy

In a nutshell, the novel IA measures *in an online way* how accurate a certain OAD method is being along the video stream, based only on what has been seen up to the instant of evaluation.

Figure 4.7 showcases the wIA-v1 for a given portion of two THUMOS14 videos. It is important noticing that an accuracy value for a certain slot does not depend on that of the previous slot. These values rely only on the predictions an weights for each correct prediction. The dynamic weighting can lead to situations where the accuracy value decreases (not much) even when a method is offering right predictions. However, this is how it has to be: since nothing about the video is known before, the importance of the detection must vary throughout the streaming. This effect is seen in the upper example of Figure 4.7, in the background part in between ground truth action annotations.

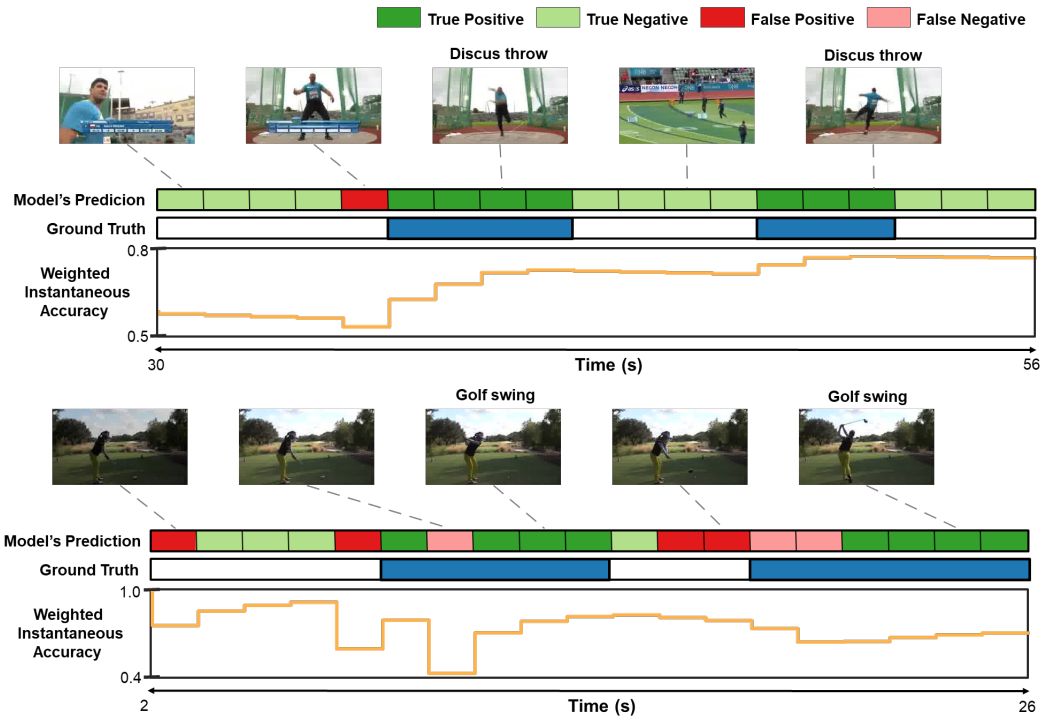


Figure 4.7: **Detail of Instantaneous Accuracy (IA)**. The figure showcases the evolution of the wIA-v1 on two different THUMOS14 videos. IA metric is an online video-level metric which measures the ability of methods to discriminate all categories, including the background. Each instant of evaluation depends on the current model's prediction.

The wIA-v1 for CNN (De Geest et al. (2016)) and 3D-CNN methods on videos of the test set of the TVSeries dataset is shown in Figure 4.8. In this case, it is shown how the metric allows for a true *online* comparison between OAD models. In other words, the evolution of the accuracy of each method for each video can be compared completely online. This is an important feature, which characterises the proposed evaluation metric, and which sets it apart from the rest.

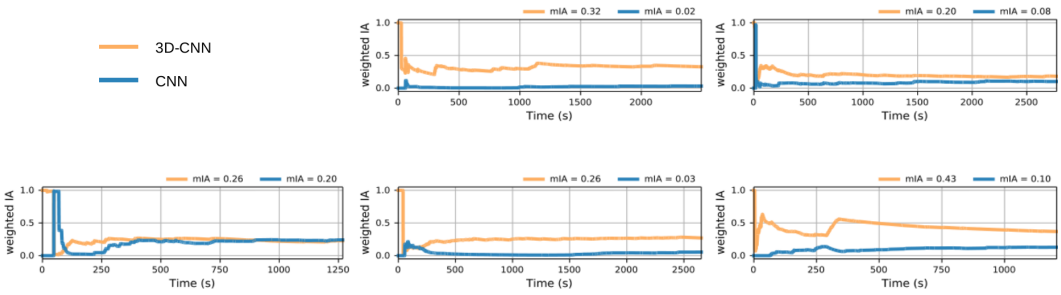


Figure 4.8: **Online video-level comparative with IA-v1.** IA-v1 evaluation in 5 videos of the TVSeries dataset.

maIA as IA consolidation for evaluation across a dataset

The IA metric can be directly used to compare methods at the video level, though in research the performance of methods is typically analysed across a dataset. To this end, the mean average Instantaneous Accuracy (maIA) (defined by Equation 4.6) is introduced. Table 4.4 presents the performance with the weighted and non-weighted versions of maIA on the three datasets.

	THUMOS14		**TVSeries**			**ActivityNet**	
	All-BG	3D-CNN	All-BG	3D-CNN	CNN	All-BG	3D-CNN
maIA-v1 (%)	71.19	72.64	78.31	71.90	3.51	40.19	21.72
maIA-v2 (%)	71.44	73.64	78.37	71.79	2.45	40.83	21.98
weighted maIA-v1 (%)	41.79	58.10	22.91	28.95	12.46	53.65	27.40
weighted maIA-v2 (%)	42.29	60.28	22.86	29.27	8.36	53.98	27.63

Table 4.4: Weighted and non-weighted maIA on THUMOS14, TVSeries and ActivityNet.

The consistency of the proposed metric can be observed through the results of the All-BG baseline. THUMOS14 and TVSeries are very imbalanced datasets and when introducing the weighting, its performance drops a lot. On ActivityNet, All-BG performs similar regardless of the weighting due to the fact that the dataset is

more balanced. Thus, the metric is capable of making a fair evaluation in all kind of datasets.

The low numbers of 3D-CNN on TVSeries and ActivityNet are caused by different reasons. TVSeries is an imbalanced and challenging dataset. With such a lot of background, a model as simple as 3D-CNN is not able to learn well to discriminate action from background. ActivityNet is balanced but has many classes to distinguish. Finally, the reproduced CNN of De Geest et al. (2016) performs poorly according to maIA due to it does not handle background. Its performance is alleviated when weighted with the positive/negative ratio.

Despite the challenge that THUMOS14 and TVSeries suppose, the 3D-CNN model performs better than the All-BG baseline. This proves that, although not perfectly, it is able to differentiate between action categories and background.

Regarding the two versions of the metric, their numbers are similar for all datasets. As it was shown in Table 4.1, version 2 introduces the slot coverage parameter and the multi-class evaluation. Since these do not have multi-class segments in their videos (only a few videos of the THUMOS14 dataset and involving not many classes), the different in the numbers are caused by the slot coverage parameter. As the action durations are shorter in THUMOS14 and TVSeries datasets, the difference between the v1 and v2 is more noticeable.

4.3 Conclusion

Online Action Detection (OAD) in untrimmed streaming videos is a challenging problem with few contributions. This work reveals three main unsolved necessities:

- The problem itself lacks of a solid definition of its properties.
- There is no clear consensus on how methods should deal with the kind of videos that are used.
- A proper online evaluation protocol has not been defined.

The OAD problem is fairly new and an agreement on its properties cannot be found among the few contributions made so far to it. This issue is solved by clearly defining them:

1. Streaming videos are assumed, where neither the length nor the content of the video is known.
2. Actions must be detected as soon as they happen.
3. Detections must be causal. Future time is not known, hence it cannot be used to make any present prediction.

As with the OAD properties, those of methods have not a consensus either. They are established in this work as follows:

- Both actions and background segments must be explicitly discriminated.
- No post-processing or posterior thresholding to detection scores can be applied.
- OAD methods cannot revisit past detections.

Regarding the third aspect, a proper online evaluation protocol should meet the following conditions:

- It has to be online, for consistency with the metric.
- It must measure the ability of methods to discriminate both actions and background.
- It must be based only on the seen portion of video

Since none of the previously used metrics complies with these conditions, this work introduces the novel Instantaneous Accuracy (IA). IA is an online video-level metric which computes the accuracy at every instant of evaluation. The experiments prove the limitations of the previous metrics as well as the robustness of the new metric.

A python toolkit with the implementation of the Instantaneous Accuracy along with a user manual is publicly available here[1].

[1]URL of the repository: https://github.com/gramuah/ia

Chapter 5

Conclusion

The work presented in this thesis focused on the analysis of streaming videos for human action understanding. Concretely, on how to analyse *untrimmed* streaming videos in an *online* fashion to find and classify those video parts (or segments) that contain a human action. The *untrimmed* nature of the videos is fundamental, as it implies that: 1) not all the video is relevant to the action recognition system; and 2) systems must be sensitive to the information considered as background, differentiating it from the temporal segments that may contain actions. On the other hand, the condition of having to process videos *online* defines a much more challenging scenario than the offline standard.

These conditions, which separate this work from traditional offline action detection and recognition, are imposed by a specific application scenario: the one where the action recognition system has to recognise the action as soon as it happens. In the particular case of this thesis, the scenario is determined by the implementation of action recognition solutions in autonomous robotic platforms that must interact with real users.

Overall, the topics that this thesis contributed to are Temporal Action Proposals (TAP) and Online Action Detection (OAD). TAP refers to the problem of localising segments of video that can contain an action, regardless of its category. This topic is of great utility for the case of a robotic platform, since it would allow the robot to understand the environment *before* interacting with it. However, the way in which the problem has been tackled before this thesis was not adequate for such robotic application. In all previous methods, the whole video content was supposed to be

known beforehand, *i.e.* following an *offline* setting. In contrast to this, the solution introduced here addresses the task without such assumption, but only analysing the video content as it is generated, *i.e. online*. Regarding OAD, it consists in detecting actions as soon as they happen. In this case, the topic itself assumes that: i) videos are untrimmed; and ii) future video content is not available. Therefore, detections mus be given only based on the information seen until the present. As it can be noticed, the conditions of the task suits perfectly to the real-world scenario with the robotic platform. In addition to understanding the environment, as with TAP, OAD could be utilised to directly interact with the people involved in the scenario since different reactions could be set depending on the action detected.

This chapter summarises the scientific contributions derived from the research work carried out in this thesis to the mentioned topics, as well as describes several further improvements to tackle the encountered limitations. Additionally, it also provides some related future research lines that could be explored.

5.1 Contributions

5.1.1 Contributions to Temporal Action Proposals

The task of finding in videos temporal segments which have high probability of containing an action, also known as Temporal Action Proposals (TAP), has shown to be crucial to solve the TAL problem. The contributions this thesis made to the TAP task are listed below:

- A thorough review of the literature on the TAP task as well as on the TAD problem was offered to contextualise the solutions that were proposed. As a result, three important facts were identified:

    1. The best and common way to solve the TAL problem involves two stages: Temporal Action Proposals generation and action classification. The latter just classifies the gerated proposals through a standard classifier, *e.g.* SCNN-cls (Shou et al. (2016)) or UntrimmedNet (Wang et al. (2017b)). For this reason, having a TAP module that is capable of generating high quality proposals is essential.
    2. The option of generating action proposals in an online way has not been explored.

3. All the state-of-the-art approaches are strongly supervised, remaining the unsupervised set-up unaddressed.

- Considering the facts found in the review of the literature, this thesis introduced the first unsupervised and online approach for the TAP task: the SVC-UAP method. This solution iteratively uses two modules: a SVC and a filter based on Rank Pooling dynamics (Fernando et al. (2017)). The former is responsible for grouping consecutive sets of frame features to create time boundaries that define candidate action proposals. Then, these candidates are analysed by the second module, which will discard candidates whose feature dynamics and those of the randomised version of the features are very similar, therefore assuming they belong to background. In contrast to the state of the art, the whole pipeline is executed online, having access only to present and past frames, and not to the whole video. Additionally, since the model is unsupervised, annotations from the datasets are not needed during training.

- A deep evaluation of the SVC-UAP method on the two main TAP benchmarks: ActivityNet (Heilbron et al. (2015)) and THUMOS14 (Idrees et al. (2017)). First, an ablation study confirmed each part of SVC-UAP works as expected. Second, although SVC-UAP does not, obviously, perform at the same level as current state-of-the-art *supervised* approaches, the work proposed is a promising new paradigm for the TAP task.

5.1.2 Contributions to Online Action Detection

The following list describes the contributions this thesis made to the OAD problem.

- A deep review of the state of the art was conducted to identify the necessities of this recent topic, as well as the weaknesses of the few methods that have been proposed up to date. This has revealed that: i) OAD needs a more solid definition of the problem; ii) there is a lack of consensus on how the methods should deal with the kind of videos involved in this task; and iii) a proper (online) evaluation protocol consistent with the problem is needed.

- This thesis first redefined in a clear way the properties of the problem: 1) streaming videos are assumed; 2) actions are detected as soon as they happen; and 3) action detections must be causal.

- Regarding the OAD methods, it has been established that: 1) both action and background must be explicitly discriminated; 2) post-processing is not allowed to refine action detections; and 3) revisiting past detections is forbidden.

- As for measuring the performance, a slightly modified metric inherited from Offline Action Detection (OffAD) (the calibrated Average Precision (cAP)) topics is proposed by De Geest et al. (2016), which is offline and does not consider the background. Therefore, it is not in line with the new definition of the problem. To solve this, this work introduced a new evaluation protocol for Online Action Detection with a new metric known as Instantaneous Accuracy (IA). This metric is an online video-level metric which computes the accuracy at every instant of execution. It explicitly considers the background and it is capable of evaluating multi-class datasets. Since methods are evaluated as the video grows, the ratio between background and action content is different at each instant. To overcome this, the IA is capable of dynamically weighting the detections.

- A thorough experimental evaluation on THUMOS14 (Idrees et al. (2017)), TVSeries (De Geest et al. (2016)) and ActivityNet (Heilbron et al. (2015)) was proposed. Thanks to the use of strong baselines specifically designed for the problem, the results proved both the ambiguities found in the state-of-the-art works and the limitations of the previous metric. The experiments also demonstrated that the new evaluation protocol based on the novel Instantaneous Accuracy metric is the most adequate way of measuring the performance.

5.2 Discussion and Further Improvements

In addition to providing the concrete solutions previously described, this research work also raises several further questions that could be addressed to improve the work here proposed. While some of them come from limitations, others represent only a few examples among all the possibilities that are still unexplored. The following lines explain several of these directions.

- The unsupervised online TAP solution proposed in Chapter 3 utilises the C3D (Tran et al. (2015)) network as feature extractor, which is a 3D convolutional network. Experimenting with newer 3D architectures, such as I3D (Carreira & Zisserman (2017)) or R(2D+1) (Tran et al. (2018)), would be a very next step.

Additionally, most recent TAP tend to use frame-level features (Lin et al. (2019); Liu et al. (2019). It would be interesting to have a comparison of the performance of the SVC-UAP method when using both type of features, *i.e.* volumetric and frame-level.

- The SVC-UAP method for TAP introduced in this work is supported on two hypotheses (see Section 3.1). Concretely, hypothesis H1 suggests that frame features from different parts of video are separable by classifiers. However, even if H1 is true, is the method able to reason that different video shots from the same action belong to the same proposal? Typically, actions are performed always in the same environment, so features may not change much. However, there are some special cases, where the same action is seen but from a different perspective, which will generate different features. Under these situations, our SVC-UAP could fail, splitting the action segment. To avoid this, it would be worth trying to use features from object detectors, as objects are always in the action, no matter the perspective of the video. For instance, Furnari & Farinella (2019) proposed a method for Action Anticipation that processes the video considering three modalities: appearance (RGB), motion (optical flow) and object-based features. The object-based features are obtained from applying the Faster R-CNN (Ren et al. (2015)) method and forming a vector for each frame with the number of detected objects and their score.

- SVC-UAP is an unsupervised method. This means that the features to be used by the classifier cannot be pre-trained with the dataset on which experiments are conducted. Moreover, the categories of the training dataset should ideally not coincide with the categories used in the test dataset. In this work the C3D network was used, because it was pre-trained on a dataset (Sports1M) that does not share categories with the datasets used in the experiments. Overall, these constraints imply that one must trust in that these auxiliary datasets contain enough representative information of the problem that is to be solved. However, the option of working with visual inductive prior knowledge extracted from the concerned dataset, such as the work proposed by Oyallon et al. (2019), has not been explored, though it would be of great interest for such an online approach. The reasons to use visual inductive priors are: i) they are efficient; and ii) they provide a totally unsupervised representation as no pre-training of any kind is used.

- Regarding the Online Action Detection problem, the new evaluation protocol introduced in this thesis requires collecting several parameters to compute the new Instantaneous Accuracy metric. However, beyond computing the performance, these parameters can also be used to conduct a diagnose of the OAD methods to study, for example, their ability to discriminate background or the time of action they need to rise a detection, among others. By doing this, the community would be able to discover the weaknesses of the methods and thus, improve them.

- Until now, OAD has only been addressed by methods that only work with visual features. But, why not incorporate audio? Fusing visual and audio features is gaining interest lately, for example in action recognition (Kazakos et al. (2019)) or in active speaker detection (Alcazar et al. (2020)).

5.3 Future Research Lines

The ultimate goal of all the work described in this thesis is to implement it on a robotic platform and prepare it to interact within a real environment. Apart from all the required research to make the robot interact in such an environment, *i.e.* move, navigation or collision avoidance; more issues will appear when trying to run the algorithms with real input data from a camera. For instance, the data will be different from that on which the methods have been trained. This means that, at least, they will need to be trained again. In the case of the SVC-UAP model for TAP, thanks to its unsupervised nature, adapting to a new environment will require no effort. However, the OAD model is supervised, *i.e.* it needs labelled data during training. To avoid the tedious process of annotating some representative data from the real environment, several techniques that prepare the model to be used on unlabelled data, such as Unsupervised Domain Adaptation or Zero-shot Learning, could be tried.

Besides, there are also other topics that could benefit from some of the ideas on which this work is built. For example, the very rapidly growing topic of Instructional Video Analysis (Zhukov et al. (2019); Tang et al. (2019)), which focuses on discovering the steps (or sub-tasks) needed to carry out a certain task, *e.g.* repairing something or cooking something. A potential application of this topic is the creation of an instruction database, so that when a task is required, the system automatically offers the necessary instructions to complete it. An OAD model would considerably accel-

erate the process of building that database since it would be capable of indexing the instructions as soon as they are detected.

Another topic that could profit from OAD is Video Captioning (Krishna et al. (2017); Zhou et al. (2018)). This topic refers to the problem of locating those segments during a video where something of interest is happening, to then, describe them with text. An OAD model would help in detecting those segments as they appear, and hence accelerate the captioning task by generating directly the associated caption. An online video captioning setting could be of interest for captioning live web content using visual features, as it is currently based only on audio.

These are only a few examples of all the possible future directions that can be explored. Fortunately for the Computer Vision community, there is still much to discover and a lot of work to be done.

5.4 Scientific Contributions

During the course of this Thesis, it has been possible to make scientific contributions to the main topics of the Thesis, as well as others resulting from side projects or collaborations with other research groups. All of them are indicated below.

Contributions directly related to the Thesis

- **Participation in the project PREPEATE (TEC2016-80326-R)** from the Spanish Ministry of Economy, Industry and Competitiveness (see project site here)

- **Embarrassingly Simple Model for Early Action Proposal** (see paper here)
  Marcos Baptista Ríos, R. J. López-Sastre, F. J. Acevedo-Rodríguez, S. Maldonado-Bascón
  In European Conference on Computer Vision Workshops (ECCVW) - 2018

- **The Instantaneous Accuracy: a Novel Metric for the Problem of Online Human Behaviour Recognition in Untrimmed Videos** (see paper here)
  Marcos Baptista Ríos, R. J. López-Sastre, Fabian Caba Heilbron, Jan C. van Gemert, F. J. Acevedo-Rodríguez, S. Maldonado-Bascón
  In International Conference on Computer Vision Workshops (ICCVW) - 2019

- **Rethinking Online Action Detection in Untrimmed Videos: A Novel Online Evaluation Protocol** (see paper here)
  Marcos Baptista Ríos, R. J. López-Sastre, Fabian Caba Heilbron, Jan C. van Gemert, F. J. Acevedo-Rodríguez, S. Maldonado-Bascón
  In IEEE Access - 2019

- **Unsupervised Action Proposals Using Support Vector Classifiers for Online Video Processing** (see paper here)
  Marcos Baptista Ríos, R. J. López-Sastre, F. J. Acevedo-Rodríguez, Pilar Martín-Martín, S. Maldonado-Bascón
  In Sensors - 2020

Side contributions

- **Learning to Exploit the Prior Network Knowledge for Weakly-Supervised Semantic Segmentation** (see paper here)
  Carolina Redondo-Cabrera, **Marcos Baptista Ríos**, R. J. López-Sastre
  IEEE Transactions on Image Processing - 2019

- **Combining Online Clustering and Rank Pooling Dynamics for Action Proposals** (see paper here)
  N. Khatir, R. J. López-Sastre, **Marcos Baptista Ríos**, S. Nait-Bahloul, F. J. Acevedo-Rodríguez
  Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) - 2019

- **Organising the 1st Visual Inductive Priors for Data-Efficient Deep Learning Workshop 2020** (see website here)
  Jan C. van Gemert, Anton van den Hengel, Attila Lengyel, Robert-Jan Bruintjes, Osman Semih Kayhan, **Marcos Baptista Ríos**
  In conjunction with European Conference on Computer Vision (ECCV) - 2020

References

Alcazar, J. L., Caba, F., Mai, L., Perazzi, F., Lee, J.-Y., Arbelaez, P., and Ghanem, B. Active speakers in context. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Alwassel, H., Caba Heilbron, F., Escorcia, V., and Ghanem, B. Diagnosing error in temporal action detectors. In *European Conference on Computer Vision (ECCV)*, September 2018. doi: 10.1007/978-3-030-01219-9_16.

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., and Gould, S. Dynamic image networks for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3034–3042, June 2016. doi: 10.1109/CVPR.2016.331.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *ACM Annual Workshop on Computational Learning Theory*, pp. 144–152, July 1992. ISBN 089791497X. doi: 10.1145/130385.130401.

Buch, S., Escorcia, V., Shen, C., Ghanem, B., and Niebles, J. C. Sst: Single-stream temporal action proposals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6373–6382, July 2017. doi: 10.1109/CVPR.2017.675.

Cao, Y., Barrett, D., Barbu, A., Narayanaswamy, S., Yu, H., Michaux, A., Lin, Y., Dickinson, S., Siskind, J. M., and Wang, S. Recognize human activities from partially observed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2658–2665, June 2013. doi: 10.1109/CVPR.2013.343.

Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, July 2017. doi: 10.1109/CVPR.2017.502.

Chao, Y., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., and Sukthankar, R. Rethinking the faster r-cnn architecture for temporal action localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1130–1139, June 2018. doi: 10.1109/CVPR.2018.00124.

Chen, W., Xiong, C., Xu, R., and Corso, J. J. Actionness ranking with lattice conditional ordinal random fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 748–755, June 2014. doi: 10.1109/CVPR.2014.101.

Cherian, A., Fernando, B., Harandi, M., and Gould, S. Generalized rank pooling for activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1581–1590, July 2017. doi: 10.1109/CVPR.2017.172.

Cherian, A., Sra, S., Gould, S., and Hartley, R. Non-linear temporal subspace representations for activity recognition. In *IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2197–2206, June 2018. doi: 10.1109/CVPR.2018.00234.

CVPR'17. Cvpr'17. Official statistics. URL http://vision.cse.psu.edu/people/chrisF/cvpr_2017/primary_graph.html.

CVPR'19. Cvpr'19. Official statistics. URL http://cvpr2019.thecvf.com/files/CVPR%202019%20-%20Welcome%20Slides%20Final.pdf.

CVPR'20. Cvpr'20. Official statistics. URL http://cvpr2020.thecvf.com/sites/default/files/CVPR2020_opening.pdf.

Dai, X., Singh, B., Zhang, G., Davis, L. S., and Chen, Y. Q. Temporal context network for activity localization in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5727–5736, October 2017. doi: 10.1109/ICCV.2017.610.

De Geest, R. and Tuytelaars, T. Modeling temporal structure with lstm for online action detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1549–1557, March 2018. doi: 10.1109/WACV.2018.00173.

De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., and Tuytelaars, T. Online action detection. In *European Conference on Computer Vision (ECCV)*, pp. 269–284, October 2016. doi: 10.1007/978-3-319-46454-1_17.

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

Escorcia, V., Caba Heilbron, F., Niebles, J. C., and Ghanem, B. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision (ECCV)*, pp. 768–784, October 2016. doi: 10.1007/978-3-319-46487-9_47.

Escorcia, V., Dao, C. D., Jain, M., Ghanem, B., and Snoek, C. Guess where? actor-supervision for spatiotemporal action localization. *Computer Vision and Image Understanding (CVIU)*, 192:102886, 2020. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2019.102886.

Fernando, B., Anderson, P., Hutter, M., and Gould, S. Discriminative hierarchical rank pooling for activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1924–1932, June 2016. doi: 10.1109/CVPR.2016.212.

Fernando, B., Gavves, E., Oramas M., J., Ghodrati, A., and Tuytelaars, T. Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(4):773–787, April 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2558148.

Furnari, A. and Farinella, G. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6251–6260, October 2019. doi: 10.1109/ICCV.2019.00635.

Gaidon, A., Harchaoui, Z., and Schmid, C. Temporal localization of actions with actoms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2782–2795, Nov 2013. ISSN 1939-3539. doi: 10.1109/TPAMI.2013.65.

Gao, J., Yang, Z., and Nevatia, R. Cascaded boundary regression for temporal action detection. In *British Machine Vision Conference (BMVC)*, pp. 521–5211, September 2017a. doi: 10.5244/C.31.52.

Gao, J., Yang, Z., and Nevatia, R. RED: reinforced encoder-decoder networks for action anticipation. In *British Machine Vision Conference (BMVC)*, pp. 921–9211, September 2017b. doi: 10.5244/c.31.92.

Gao, J., Yang, Z., Sun, C., Chen, K., and Nevatia, R. Turn tap: Temporal unit regression network for temporal action proposals. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3648–3656, October 2017. doi: 10.1109/ICCV.2017.392.

Gao, J., Chen, K., and Nevatia, R. Ctap: Complementary temporal action proposal generation. In *European Conference on Computer Vision (ECCV)*, pp. 70–85, September 2018. doi: 10.1007/978-3-030-01216-8_5.

Gao, M., Xu, M., Davis, L., Socher, R., and Xiong, C. Startnet: Online detection of action start in untrimmed videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5541–5550, October 2019. doi: 10.1109/iccv.2019.00564.

Ghanem, B., Niebles, J. C., Snoek, C., Caba-Heilbron, F., Alwassel, H., Escorcia, V., Khrisna, R., Buch, S., and Duc-Dao, C. The activitynet large-scale activity recognition challenge 2018 summary. arXiv:1808.03766, 2018.

Gkioxari, G. and Malik, J. Finding action tubes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 759–768, June 2015. doi: 10.1109/CVPR.2015.7298676.

Heilbron, F. C., Escorcia, V., Ghanem, B., and Niebles, J. C. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–970, June 2015. doi: 10.1109/CVPR.2015.7298698.

Heilbron, F. C., Niebles, J. C., and Ghanem, B. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1914–1923, June 2016. doi: 10.1109/CVPR.2016.211.

Heilbron, F. C., Ghanem, B., Niebles, J. C., and Snoek, C. Activitynet challenge 2020, 2020. URL http://activity-net.org/challenges/2020/index.html.

Hoai, M. and De la Torre, F. Max-margin early event detectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2863–2870, June 2012. doi: 10.1109/CVPR.2012.6248012.

Hosang, J., Benenson, R., Dollár, P., and Schiele, B. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*

(TPAMI), 38(4):814–830, April 2016. ISSN 1939-3539. doi: 10.1109/TPAMI.2015.2465908.

Idrees, H., Zamir, A. R., Jiang, Y.-G., Gorban, A., Laptev, I., Sukthankar, R., and Shah, M. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1 – 23, 2017. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2016.10.018.

Jain, M., v. Gemert, J., Jégou, H., Bouthemy, P., and Snoek, C. G. M. Action localization with tubelets from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 740–747, June 2014. doi: 10.1109/CVPR.2014.100.

Ji, J., Cao, K., and Niebles, J. C. Learning temporal action proposals with fewer labels. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7072–7081, October 2019. doi: 10.1109/ICCV.2019.00717.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia (ACMM)*, pp. 675–678, November 2014. ISBN 9781450330633. doi: 10.1145/2647868.2654889.

Jiang, Y.-G., Bhattacharya, S., Chang, S.-F., and Shah, M. High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval (IJMIR)*, 2(2):2:73–101, June 2013. doi: 10.1007/s13735-012-0024-2.

Junsong Yuan, Zicheng Liu, and Ying Wu. Discriminative subvolume search for efficient action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2442–2449, June 2009. doi: 10.1109/CVPR.2009.5206671.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732, June 2014. doi: 10.1109/CVPR.2014.223.

Kazakos, E., Nagrani, A., Zisserman, A., and Damen, D. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5491–5500, October 2019. doi: 10.1109/ICCV.2019.00559.

Khatir, N., López-Sastre, R. J., Baptista-Ríos, M., Nait-Bahloul, S., and Acevedo-Rodríguez, F. J. Combining online clustering and rank pooling dynamics for action proposals. In *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, pp. 77–88, July 2019. ISBN 978-3-030-31332-6. doi: 10.1007/978-3-030-31332-6_7.

Kong, Y., Kit, D., and Fu, Y. A discriminative model with multiple temporal scales for action prediction. In *European Conference on Computer Vision (ECCV)*, pp. 596–611, September 2014. ISBN 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_39.

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Niebles, J. C. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, Oct 2017. doi: 10.1109/ICCV.2017.83.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105, 2012.

Lan, T., Chen, T.-C., and Savarese, S. A hierarchical representation for future action prediction. In *European Conference on Computer Vision (ECCV)*, pp. 689–704, September 2014. ISBN 978-3-319-10578-9. doi: 10.1007/978-3-319-10578-9_45.

Lin, C., Li, J., Wang, Y., Tai, Y., Luo, D., Cui, Z., Wang, C., Li, J., Huang, F., and Ji, R. Fast learning of temporal action proposal via dense boundary generator. In *AAAI Conference on Artificial Intelligence (AAAI)*, February 2020.

Lin, T., Zhao, X., and Shou, Z. Single shot temporal action detection. In *ACM International Conference on Multimedia (ACMM)*, pp. 988–996, 2017. ISBN 9781450349062. doi: 10.1145/3123266.3123343.

Lin, T., Zhao, X., Su, H., Wang, C., and Yang, M. Bsn: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision (ECCV)*, pp. 3–21, Cham, September 2018. ISBN 978-3-030-01225-0. doi: 10.1007/978-3-030-01225-0_1.

Lin, T., Liu, X., Li, X., Ding, E., and Wen, S. Bmn: Boundary-matching network for temporal action proposal generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3888–3897, October 2019. doi: 10.1109/ICCV.2019.00399.

Liu, T.-Y. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, March 2009. ISSN 1554-0669. doi: 10.1561/1500000016.

Liu, Y., Ma, L., Zhang, Y., Liu, W., and Chang, S. Multi-granularity generator for temporal action proposal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3599–3608, June 2019. doi: 10.1109/CVPR.2019.00372.

Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., and Mei, T. Gaussian temporal awareness networks for action localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 344–353, June 2019. doi: 10.1109/CVPR.2019.00043.

Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., and Oliva, A. Moments in time dataset: One million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(2):502–508, Feb 2020. ISSN 1939-3539. doi: 10.1109/TPAMI.2019.2901464.

Narayan, S., Cholakkal, H., Khan, F. S., and Shao, L. 3c-net: Category count and center loss for weakly-supervised action localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8678–8686, October 2019. doi: 10.1109/ICCV.2019.00877.

Nguyen, P., Han, B., Liu, T., and Prasad, G. Weakly supervised action localization by sparse temporal pooling network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6752–6761, June 2018. doi: 10.1109/CVPR.2018.00706.

Oikonomopoulos, A., Patras, I., and Pantic, M. Spatiotemporal localization and categorization of human actions in unsegmented image sequences. *IEEE Transactions on Image Processing (TIP)*, 20(4):1126–1140, April 2011. ISSN 1941-0042. doi: 10.1109/TIP.2010.2076821.

Oneata, D., Verbeek, J., and Schmid, C. The lear submission at thumos 2014, 2014.

Oyallon, E., Zagoruyko, S., Huang, G., Komodakis, N., Lacoste-Julien, S., Blaschko, M., and Belilovsky, E. Scattering networks for hybrid representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(9):2208–2221, September 2019. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2855738.

Paul, S., Roy, S., and Roy-Chowdhury, A. K. W-talc: Weakly-supervised temporal activity localization and classification. In *European Conference on Computer Vision (ECCV)*, pp. 588–607, Cham, 2018. ISBN 978-3-030-01225-0.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research (JMLR)*, 12(85):2825–2830, 2011.

PREPEATE. Prepeate research project (tec2016-80326-r). Spanish Ministry of Science, Innovation and Universities. URL http://agamenon.tsc.uah.es/Investigacion/gram/projects/prepeate/index.html.

Qiu, Z., Yao, T., and Mei, T. Learning spatio-temporal representation with pseudo-3d residual networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5534–5542, October 2017. doi: 10.1109/ICCV.2017.590.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, December 2015. doi: 10.1109/tpami.2016.2577031.

Roerdink, J. B. T. M. and Meijster, A. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1–2): 187–228, January 2000. ISSN 0169-2968. doi: 10.3233/fi-2000-411207.

Ryoo, M. S. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1036–1043, November 2011. doi: 10.1109/ICCV.2011.6126349.

Shou, Z., Wang, D., and Chang, S. Temporal action localization in untrimmed videos via multi-stage cnns. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1049–1058, June 2016. doi: 10.1109/CVPR.2016.119.

Shou, Z., Chan, J., Zareian, A., Miyazawa, K., and Chang, S. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1417–1426, July 2017. doi: 10.1109/CVPR.2017.155.

Shou, Z., Pan, J., Chan, J., Miyazawa, K., Mansour, H., Vetro, A., Giro-i Nieto, X., and Chang, S.-F. Online detection of action start in untrimmed, streaming videos. In *European Conference on Computer Vision (ECCV)*, pp. 551–568, September 2018. ISBN 978-3-030-01219-9. doi: 10.1007/978-3-030-01219-9_33.

Shyamal Buch, Victor Escorcia, B. G. and Niebles, J. C. End-to-end, single-stream temporal action detection in untrimmed videos. In *British Machine Vision Conference (BMVC)*, pp. 931–9312, September 2017. doi: 10.5244/c.31.93.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 568–576, December 2014.

Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., and Zhou, J. Coin: A large-scale dataset for comprehensive instructional video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1207–1216, June 2019. doi: 10.1109/CVPR.2019.00130.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1195–1204, December 2017. ISBN 9781510860964.

Tran, D. and Yuan, J. Optimal spatio-temporal path discovery for video event detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3321–3328, June 2011. doi: 10.1109/CVPR.2011.5995416.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, December 2015. doi: 10.1109/ICCV.2015.510.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6459, June 2018. doi: 10.1109/CVPR.2018.00675.

van Gemert, J. C., Jain, M., Gati, E., and Snoek, C. G. M. Apt: Action localization proposals from dense trajectories. In *Proceedings of the British Machine Vision*

Conference (BMVC), pp. 177.1–177.12, September 2015. ISBN 1-901725-53-7. doi: 10.5244/C.29.177.

Wang, H., Klaser, A., Schmid, C., and Liu, C. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176, June 2011. doi: 10.1109/CVPR.2011.5995407.

Wang, J., Cherian, A., and Porikli, F. Ordered pooling of optical flow sequences for action recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 168–176, March 2017a. doi: 10.1109/WACV.2017.26.

Wang, L., Qiao, Y., Tang, X., and Van Gool, L. Actionness estimation using hybrid fully convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2708–2717, June 2016. doi: 10.1109/CVPR.2016.296.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pp. 20–36, Cham, October 2016. ISBN 978-3-319-46484-8. doi: 10.1007/978-3-319-46484-8_2.

Wang, L., Xiong, Y., Lin, D., and Van Gool, L. Untrimmednets for weakly supervised action recognition and detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6402–6411, July 2017b. doi: 10.1109/CVPR.2017.678.

Xu, H., Das, A., and Saenko, K. R-c3d: Region convolutional 3d network for temporal activity detection. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5794–5803, October 2017. doi: 10.1109/ICCV.2017.617.

Xu, H., Das, A., and Saenko, K. Two-stream region convolutional 3d network for temporal activity detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(10):2319–2332, October 2019a. doi: 10.1109/TPAMI.2019.2921539.

Xu, M., Gao, M., Chen, Y., Davis, L., and Crandall, D. Temporal recurrent networks for online action detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5531–5540, October 2019b. doi: 10.1109/ICCV.2019.00563.

Xu, M., Zhao, C., Rojas, D. S., Thabet, A., and Ghanem, B. G-tad: Sub-graph localization for temporal action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Yeung, S., Russakovsky, O., Mori, G., and Fei-Fei, L. End-to-end learning of action detection from frame glimpses in videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2678–2687, June 2016. doi: 10.1109/CVPR.2016.293.

Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., and Fei-Fei, L. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision (IJCV)*, 126(2):375–389, April 2018. doi: 10.1007/s11263-017-1013-y.

Yu, G. and Yuan, J. Fast action proposals for human action detection and search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1302–1311, June 2015. doi: 10.1109/CVPR.2015.7298735.

Yu, G., Yuan, J., and Liu, Z. Predicting human activities using spatio-temporal structure of interest points. In *ACM International Conference on Multimedia (ACMM*, pp. 1049–1052, October 2012. ISBN 9781450310895. doi: 10.1145/2393347.2396380.

Yuan, Z., Stroud, J. C., Lu, T., and Deng, J. Temporal action localization by structured maximal sums. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3215–3223, July 2017. doi: 10.1109/CVPR.2017.342.

Zeng, R., Huang, W., Gan, C., Tan, M., Rong, Y., Zhao, P., and Huang, J. Graph convolutional networks for temporal action localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7093–7102, October 2019. doi: 10.1109/iccv.2019.00719.

Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., and Lin, D. Temporal action detection with structured segment networks. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2933–2942, October 2017. doi: 10.1109/ICCV.2017.317.

Zhou, L., Zhou, Y., Corso, J. J., Socher, R., and Xiong, C. End-to-end dense video captioning with masked transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8739–8748, June 2018. doi: 10.1109/CVPR.2018.00911.

Zhu, H., Vial, R., and Lu, S. Tornado: A spatio-temporal convolutional regression network for video action proposal. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5814–5822, October 2017. doi: 10.1109/ICCV.2017.619.

Zhukov, D., Alayrac, J., Cinbis, R. G., Fouhey, D., Laptev, I., and Sivic, J. Cross-task weakly supervised learning from instructional videos. In *IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3532–3540, June 2019. doi: 10.1109/CVPR.2019.00365.