Detection and characterization of salient moments for automatic summaries

Laura Melissa Sanabria Rosas

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Détection et Caractérisation des Moments Saillants pour les Résumés Automatiques

Laura Melissa SANABRIA ROSAS

Inria, Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)

Présentée en vue de l'obtention du grade de docteur en Informatique **d'**Université Côte d'Azur
Dirigée par : Frédéric PRECIOSO / Thomas Menguy, Wildmoka
Soutenue le : 03/12/2021

Devant le jury composé de :
Rainer Lienhart, Professeur, University of Augsburg, Allemagne
Vasileios Mezaris, Chercheur Senior, ITI/CERTH, Grèce
Stefano Melacci, Professeur, University of Siena, Italie
Catherine Achard, Professeur, Sorbonne Université, France
François Bremond, Chercheur Senior, Inria, France

Région Provence Alpes Côte d'Azur

UNIVERSITÉ CÔTE D'AZUR

WILDMOKA

Détection et Caractérisation des Moments Saillants pour les Résumés Automatiques

Jury :

Président du Jury :
François Bremond, Chercheur Senior, Inria, France

Rapporteurs :
Rainer Lienhart, Professeur, University of Augsburg, Allemagne
Vasileios Mezaris, Chercheur Senior, ITI/CERTH, Grèce

Examinateurs :
Catherine Achard, Professeur, Sorbonne Université, France
Stefano Melacci, Professeur, University of Siena, Italie

Directeur :
Prof. Frédéric PRECIOSO, Professeur des universités, Université Côte d'Azur, France

Invités :
Thomas Menguy, Wildmoka, France

Abstract

Video content is present in an ever-increasing number of fields, both scientific and commercial. Sports, particularly soccer, is one of the industries that has invested the most in the field of video analytics, due to the massive popularity of the game. Although several state-of-the-art methods rely on handcrafted heuristics to generate summaries of soccer games, they have proven that multiple modalities help detect the best actions of the game. On the other hand, the field of general-purpose video summarization has advanced rapidly, offering several deep learning approaches. However, many of them are based on properties that are not feasible for sports videos.

Video content has been for many years the main source for automatic tasks in soccer but the data that registers all the events happening on the field have become lately very important in sports analytics, since these event data provide richer information and requires less processing.

Considering that in automatic sports summarization, the goal is not only to show the most important actions of the game, but also to evoke as much emotion as those evoked by human editors, we propose a method to generate the summary of a soccer match video exploiting the event metadata of the entire match and the content broadcast on TV. We have designed an architecture, introducing (1) a Multiple Instance Learning method that takes into account the sequential dependency among events, (2) a hierarchical multimodal attention layer that grasps the importance of each event in an action and (3) a method to automatically generate multiple summaries of a soccer match by sampling from a ranking distribution, providing multiple candidate summaries which are similar enough but with relevant variability to provide different options to the final user.

We also introduced solutions to some additional challenges in the field of sports summarization. Based on the internal signals of an attention model that uses event data as input, we proposed a method to analyze the interpretability of our model through a graphical representation of actions where the x-axis of the graph represents the sequence of events, and the y-axis is the weight value learned by the attention layer. This new representation provides a new tool for the editor containing meaningful information to decide whether an action is important. We also proposed the use of keyword spotting and boosting techniques to detect every time a player is mentioned by the commentators as a solution for the missing event data.

Keywords: Video Summarization, Multimodal data, Event data, Deep networks, Multiple Instance Learning

Résumé

Le contenu vidéo est présent dans un nombre toujours plus grand de domaines, tant scientifiques que commerciaux. Le sport, en particulier le football, est l'une des industries qui a le plus investi dans le domaine de l'analyse vidéo, en raison de la popularité massive de ce sport.

Bien que plusieurs méthodes de l'état de l'art utilisent des heuristiques pour générer des résumés de matchs de football, elles ont prouvé que de multiples modalités aident à détecter les meilleures actions du match. D'autre part, le domaine du résumé vidéo à usage général a progressé rapidement, offrant plusieurs approches d'apprentissage profond. Cependant, beaucoup d'entre elles sont basées sur des hypothèses qui ne sont pas réalisables pour les vidéos sportives.

Le contenu vidéo a été pendant de nombreuses années la principale source pour les tâches automatiques dans le football, mais les données qui enregistrent tous les événements qui se produisent sur le terrain sont devenues dernièrement très importantes dans l'analyse du sport, car ces données d'événements fournissent des informations plus riches et nécessitent moins de traitement.

Considérant que dans le résumé automatique de sports, l'objectif n'est pas seulement de montrer les actions les plus importantes du jeu, mais aussi d'évoquer autant d'émotions que celles évoquées par les éditeurs humains, nous proposons une méthode pour générer le résumé d'une vidéo de match de football en exploitant les métadonnées d'événement de tout le match et le contenu diffusé à la télévision. Nous avons conçu une architecture, introduisant (1) une méthode d'apprentissage d'instances multiples qui prend en compte la dépendance séquentielle entre les événements, (2) une couche d'attention multimodale hiérarchique qui saisit l'importance de chaque événement dans une action et (3) une méthode pour générer automatiquement plusieurs résumés d'un match de football en choisissant parmi une distribution de rangs, fournissant plusieurs résumés candidats qui sont suffisamment similaires mais avec une variabilité pertinente pour fournir différentes options à l'utilisateur final.

De plus, nous avons proposé des solutions à certains défis supplémentaires dans le domaine du résumé des sports. À partir des signaux internes d'un modèle d'attention qui utilise des données d'événements comme entrée, nous avons introduit une représentation graphique des actions où l'axe des x du graphique représente la séquence d'événements et l'axe des y est la valeur du poids appris par la couche d'attention. Cette nouvelle représentation fournit un nouvel outil à l'éditeur contenant des informations significatives pour décider si une action est importante. Nous proposons également l'utilisation de techniques de repérage de mots-clés et de boosting pour détecter chaque fois qu'un joueur est mentionné par les commentateurs.

Mots clés:Résumés vidéo, données multimodales, données d'événements, réseaux profonds, apprentissage à instances multiples.

Acknowledgements

First of all I would like to thank my supervisor Frédéric Precioso. Thank you for trusting me even when I did not. I think I would never be able to pay you back for everything you have done for me. I admire your kindness, patience and humbleness. I have learned so much from you as a person, as a researcher and as a professor, you are a true role model.

Thanks to Thomas Menguy and all the people in Wildmoka, you were always very nice and welcoming even though I was a stranger going to your offices from time to time. And to all the members of the jury, thank you for accepting so kindly to be part of this, for the time and for the useful comments.

To my family, because you were always there even with the time difference. I can write pages describing all the things I want to thank you, mom and dad, but for now I will just thank you for pushing me to wake up early every Saturday to go to my English courses, for being my support all the time, for instilling in me all those moral values and for teaching me that we need to appreciate the small things in life. To Caro, for always believing in me. To Lucy, for listening all my weird stories and explanations, even when I was not making any sense and for being my best friend when I needed it the most. Juanito and Santi, it is sad I can't see you growing but I appreciate all the laughs and calls.

I want to thank my friends, because they made the PhD life much easier. You guys are my family, we have so many great memories together that it is very difficult to mention only one. Eva, thank you for teaching me the real meaning of sorority. Ninad, thank for being my partner in the never-say-no club. Thomas, thank you for staying with me all the times I was tired of these crazy people. Fernando, thank you for all the nice food, drinks, moments and laughs. Miguel, even though our story changed as no one expected, I want to thank you for all your help and support during all the difficult moments.

Tita and Carlos, thank you for being so kind even when you did not know us, for always making me feel like a member of your beautiful family and for allowing me being part of the life of those wonderful children. Lola, you were the first one believing in me, you were my first role model, thank you for introducing me this beautiful city.

And to all the people I met during these four years, if I do not mention you is because I am afraid of missing someone, but thank you for your company, advice, help, time and all the things I got from you. Dingge, sharing the office with you made the time much funnier.

And last but not least, Yannis, the best unexpected moment at the end of this thesis journey. Thank you for all your love and for being so sweet with me even when I did not have the time or the energy to be a good company.

This thesis has been partially supported by Région Provence Alpes Côte d'Azur (PACA), Université Côte d'Azur (UCA), and Wildmoka Company.

Contents

List of Figures

List of Tables

List of Acronyms

RNN Recurrent Neural Network

LSTM Long Sort-Term Memory

HMA Hierarchical Multimodal Attention

MIL Multiple Instance Learning

CNN Convolutional Neural Network

LSE Log-Sum-Exp function

mAP mean Average Precision

XML Extensible Markup Language

C3D 3D Convolutional Neural Networks

IFAB International Football Association Board

Chapter 1

Introduction

1.1 Context

The volumes of current video content are exploding, which has intensified the research in the areas of time segmentation, storage, search, and navigation in video content. Another major change in the field of video is the way the new generations are consuming the content, mostly in mobile phones and very short clips, due to the increased use of social networks. These new usages require the emergence of new tools for creating, editing, managing, and distributing videos. Those tools are the main services provided by Wildomoka, a company based in Sophia Antipolis, France. Wildmoka provides a platform where the broadcasters and content owners can clip the best moments from live TV and then share with their users in social networks.

Sports, particularly soccer, is one of the sectors that has invested the most in the video analysis field, due to the massive popularity of the game. In a professional league such as Premier League, for example, with 10 matches per weekend, video from every stadium and multiple camera angles can quickly add up to dozens of hours of footage. And some companies manage the broadcasting of several competitions at the same time. In addition, the fans expect the summaries and highlights to be available as soon as the match is finished. Yet, most of the process for producing summary videos in broadcasting companies is still labor-intensive, time-consuming, and not scalable.

1.2 Challenges

Most of the state of the art methods for detecting soccer video actions rely on handcrafted rules, including dominant color of the field, logo detection and goalmouth detection [11, 8, 12, 13, 14]. Other rules assume that when there is not much excitement in the match the producers mostly show a global view to convey the whole status of the game and when there is an important action, zoom-ins and close-ups tend to be the majority as they can show the cause and effect of the action. Based on these assumptions, the video is divided in shots to later use a classifier. Although most recent works propose deep learning approaches, they are still based on these implicit producing

rules [15, 9]. One of the main reasons for these rule-based approaches is that the definition of what is an action in sports is far from being as well defined as it is in general-purpose videos. This is even more evident for soccer since there is not any fixed phase a priori in a game as you can find in baseball or cricket, and since the players from both teams can move anywhere on the field unlike tennis or volleyball.

Producing summaries almost immediately after it ended, causing as much emotion as the ones made by human operators and witnessing the course of the match, is a relevant challenge for automatic soccer video summarization. Since a summary is indeed not only a sequence of goals, even if goals are important events. Current summaries are produced by professional operators who try to render the story of the match, reflecting the dramaturgy of the match, and possibly connect this story with the last news about the players involved. These editorial decisions show the subjectivity of the task since there is not a unique and perfect ground-truth summary for a match, it might depend on the platform where the video will be published, the league, the country, the length constraint, etc.

Despite the huge amount of sports video content available online, there are not public datasets for summarization. The copyright, subjectivity and different time-consuming tasks that involve the processing of long video matches like soccer, are among the main causes for the lack of available data, making more difficult the evaluation and comparison of any method proposed to tackle the task.

1.3 Motivation

There is no fully automatic solution to produce summaries in a short time (targeting real-time), on real sport content (one soccer match to be summarized is at the very least a 90-minute-long video), considering different modalities. The current solution for sport broadcasters is thus to rely on human operators to generate in live, highlights, summaries, specific content for social networks, and any extra content that will build viewer loyalty. Our motivation lies in the multiple solutions that can rise, benefiting not only the multimedia community but also the sports broadcasting industry.

A soccer match is at least 90 minutes long and its summary is only 5 minutes, processing all the video in one step might not be feasible in a technical aspect and designing summarization techniques that analyzes the entire match to select such sparse number of actions is not very optimal. A possible solution is to split the summarization process in two tasks, first detect all the actions of the match that could be selected to be in the summary, and then decide which of these actions indeed belong to the summary.

For many years automatic video summarization in sports mainly has relied on hand-crafted heuristics using video content as the main source. However, event data acquired in live during the matches is a very important source of information which very recently started to be exploited in the machine learning field for sports. The event data provide specific details of all events

happening on the field like the type of action, the position on the field, the players involved, the part of the body with which the player touched the ball, etc. Processing event data is significantly faster than video frames since a soccer match contains in average around 1700 events compared with more than 130k frames. In addition, video frames lead to several issues like occlusions, resolution, or subjectivity.

In the sports field it is very common to exploit different modalities since many of them provide important information of the match. As it was mentioned before, the video frames are the most popular modality, however the audio also plays a very important role since it records the emotions and reactions expressed by the crowd and the commentators. A main set of solutions to the combination of multiple modalities in sport tasks relies on rules which have a strong justification based on experts' knowledge and experience but are clearly dependent on the sport or the data [16]. An alternative solution are the multimodal fusion approaches proposed in non-sports related tasks. These approaches can be cast into two categories, one is the naive concatenation of the modalities to create a single feature vector as input and the other is the attention-based models to dynamically determine the relevance of each modality [17].

1.4 Contributions and Organization of the Manuscript

In this work we propose two solutions for Soccer Video Summarization. The two solutions present three stages: first the generation of action proposals, then the use of different modalities to decide which actions belong to the summary and a final summary generation. With the first solution we propose a fully automatic way of generating a single video summary per match using event data, audio and video features. And the second solution describes a semi-automatic method that proposes different candidate summaries to human operators. In the Chapter 4, we describe the fully automatic video summarization system, by exploring the use of Multiple Instance Learning (MIL) and multimodal features. In the Chapter 5, we depict the semi-automatic summaries generator, proposing our own methods to deal with MIL for sequential data, merging of multimodal features, and subjectivity. Chapter 6 presents additional challenges we faced in this thesis, trying to provide some interpretability of our system and investigating briefly how to deal with missing event data. Finally, Chapter 7 concludes this dissertation.

Fully automatic video summarization system (Chapter 4)

We develop a multimodal approach to automatically generate video summaries of soccer matches that consider event, audio, and video features. The event features get a shorter and better representation of the match, and the audio helps detect the excitement generated by the game. Our method consists of three consecutive stages: Proposals, Summarization and Content Refinement. The first one generates summary proposals, using Multiple Instance Learning to deal with the similarity between the events inside the summary and the rest of the match. The Summarization stage uses event and audio features as input of a hierarchical Recurrent Neural Network (RNN) to decide which proposals should indeed be in the summary. And the last stage,

takes advantage of the visual content to create the final summary. This approach leads to the following contributions:

• The use of event data instead of video to extract the main information of the match, significantly reducing the amount of information and the time to process it.

• We propose to combine event data with audio features to choose the parts of the match that belong to the summary.

• Based on the variability among similar actions and, the similarity between the actions that belong to the summary and the ones that do not, we propose to use Multiple Instance Learning to generate action proposals.

Semi-automatic summaries generator (Chapter 5)

We use the experience acquired in the development of the first solution and propose another method that instead of providing a single video summary, it provides several candidate summaries to deal with subjectivity and length constraint. We hence made three contributions:

• A Multiple Instance Learning approach that exploits LSTM sequentiality to process time-dependent instances and generate proposals.

• A HMA mechanism that, unlike the state-of-the-art methods, in the first stage learns the importance of each modality (event data and audio) at the event level and in the second stage learns the importance of each event inside the action. In terms of F1-score, HMA outperforms by 7% the state-of-the-art methods in multimodal attention and by 15% the soccer baselines.

• A method for automatically generating multiple summaries of a soccer match by sampling from a ranking distribution. We provide different options to the editor solving two relevant issues in sports summarization, subjectivity, and time constraints. Our method outperforms by 6% the state-of-the-art. And in terms of generalization, it outperforms by 9% in the prediction without fine-tunning of a different soccer competition.

Additional challenges on interpretability and missing event data (Chapter 6)

One of the main challenges of machine learning is the well-known "black box" concept, where the model performs very well but it does not provide information about how it made the decisions. Currently, in broadcast companies human editors decide which actions belong to the summary based on multiple rules they have created using different sources of information but mainly relying on the metadata describing the match. These rules define different action profiles that help the editor to generate better customized summaries. Thus, we proposed a method to analyze the interpretability of our model and verify if it learns similar rules to the ones created by human editors:

• We propose to create a graphical representation of these action profiles from the weights learned by a neural network model with an attention layer. The results in soccer matches show the capacity of our approach to transfer knowledge between datasets from different broadcasting companies and the ability of the attention layer to generate meaningful action profiles.

As previously mentioned, event data provides relevant information of the match. However, these data are not always publicly available, they are sold by private companies and as it is

explained later in Section 2.1.3, they are not available for all the competitions. On the other hand, commentators play a very important role in sports, since they tell the story of the match in real time, providing also important details such as the name of the players or the type of the action. As a starting point to tackle the issue of missing event data, we propose:

• The use of the audio signal to identify the involvement of the player in a specific time of the match. Leveraging from the techniques employed in voice assistant systems, we use a keyword spotting algorithm to detect all the times a commentator mentions the name of the player in the audio signal of broadcast matches. This would help to complete the event data, when they are produced by the aforementioned companies, adding the players who are "invisibly involved" in the action. And this detection would also help to produce partial event data for leagues or matches where they are missing, by adding the timestamp of all the moments where each player is involved.

Chapter 2

Related Work

In this chapter, we briefly describe the terminology used throughout this thesis, the three most prominent types of data extracted from soccer matches, the difference between action recognition and action detection, and some explanations about video summarization including sports and multimodal techniques. We also provide an overview of the state of the art in the different topics this thesis has an impact.

2.1 Data sources of soccer matches

A soccer match generally is 90 minutes long, the game is divided in two halves of 45 minutes each and there is a break of 15 minutes between them. Unlike other sports, the clock never stops during the match, even if the ball is not in play. To recover some lost time, the referee can add some extra time at the end of each half. Then the duration of a soccer match varies from 105 to 115 minutes. Because of the great popularity of this sport there are several companies in charge of registering everything that happens in those minutes, such as broadcasting companies producing video footage from several cameras and microphones, and sport analytics companies recording event logs and tracking players.

2.1.1 Video

Depending on the country or the competition, there is one or few companies who have the rights to broadcast the video of the matches. These companies have access to the footage of several cameras, and they have professional editors who decide, according to their own production styles or editing patterns, which are the best scenes to depict the course of the game. Then the final video match is not always the same, it depends on the country, the producer, the game, the competition and many other factors.

In terms of video production, several authors [7, 18, 19, 20, 21] agree that there are three main types of views in soccer: long-shot, medium-shot and close-up. A long-shot displays the global view of the field as shown in Figure2.1a. A medium-shot is a zoomed-in view of a specific

(A) Long-shot (B) Zoom-in (C) Close-up

FIGURE 2.1: Types of views in Soccer videos.

part of the field where a whole human body is usually visible as in Figure2.1b. A close-up shot might show the above-waist view of one person, the audience or the coach (Figure2.1c).

In terms of video content, the broadcast video does not show all the time the current state of the match. There might be video/image advertisements, editors add replays of relevant moments of the match in order to provide a better user experience or they also broadcast reactions of the players and crowd. Therefore, a broacast video of a soccer match does not contain only images of the field, we can find images of players, crowd, coach, logos, among others.

In terms of video structure, videos can be considered linear if we see it as frames stacked together. Videos can also be considered as having hierarchical structure, if we see frames, shots, and scenes as three different levels. A video shot consists of a consecutive sequence of frames where there are not camera (view) changes. Video scene can be defined as combination of several shots stitched together which represents a relatively complete semantic content like a goal action.

As mentioned before, the clock of the match never stops during the 45 minutes of each half. Therefore, a popular method to define the state of the match is called *play and break* [7, 18, 22, 9, 8, 15]. This method is inspired by the definition set by the International Football Association Board (IFAB) where it says that the ball is out of play (*break*) when "it has wholly passed over the goal line or touchline on the ground or in the air" or "play has been stopped by the referee" and the ball is in play (*play*) at all other times [23]. However, dividing a soccer video into play and break is not an easy task because unlike sports like tennis where volleys are always preceded by a serve, in soccer there is not exact temporal structure for the different transition of events. Also, in soccer there is not a canonical scene like in tennis (when a serve starts, the scene is usually switched to the court view) or in baseball (when a pitch starts, there is a pitching view taken by the camera behind the pitcher). The video sequence of each play in a soccer game typically contains multiple shots with similar characteristics.

2.1.2 Audio

During live sport broadcasts, microphones are strategically located around the pitch to recreate the stadium atmosphere. There are other microphones placed in the commentators' room that is usually semi-isolated in an upper level of the stadium. Even though each microphone records a different audio track, the broadcast video fusions all this audio sources in one.

The final audio track of the match is then composed mainly by foreground commentary coexisting with background sounds. The background sounds include ambient crowd noise, sparse segments of crowd noise, coach reactions (indications to the players, conversations with their staff or arguments with the referees), the voice of the players, the whistle and clapping. The audio signal is therefore more complex to analyze than in other sports, and this is possibly one of the explanations to the fact that there are very few significant examples where audio is used as main source for content characterization in soccer [24, 25, 26, 27, 28].

For audio classification, instead of directly using the raw audio signal, several features are extracted from it. In this thesis we focus on features in time, frequenc,y and cepstral domains.

The time-domain audio features are extracted directly from the samples of the audio signal. By visualizing a signal in time domain, we may analyze how the signals evolve with time. Figure 2.2 depicts a signal in time domain where the y-axis represents the amplitude of the signal. This figure also shows the outcome of the *Zero-crossing rate (ZCR)* feature, which is the number of times the signal changes its sign from positive to negative and vice versa. It can be interpreted as a measure of the noisiness of a signal, it exhibits higher values in the case of noisy signals.

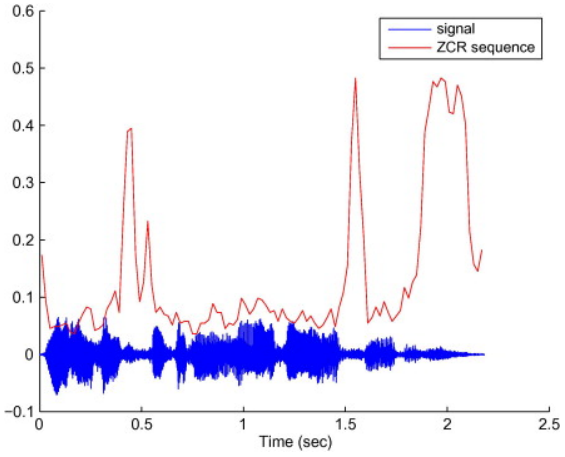


FIGURE 2.2: Example of a speech segment and the respective sequence of ZCR values. Image from [1]

To analyze a signal in terms of frequency, the time-domain signal is converted into frequency-domain (or spectral) by using Fourier transform. From this you can extract features like the *Spectral centroid*, which describes the brightness of a sound signal by locating the center of the mass of the spectrum. Also the *Spectral spread*, which measures how the spectrum is distributed around its centroid, being generally widely spread for environmental sounds and narrowly for speech like sounds. *Spectral entropy* measures the uniformity or flatness. *Spectral flux* points the sudden changes in the frequency energy distribution of sounds, being higher for speech due the rapid alternation among phonemes. And the *Spectral rolloff* is the frequency below which a certain percentage (usually around 90%) of the magnitude distribution of the spectrum

is concentrated, it can be used for discriminating between voiced and unvoiced sounds or to discriminate between different types of music tracks.

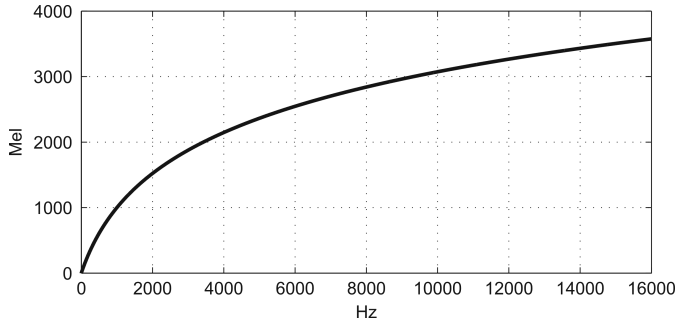


FIGURE 2.3: Mel-scale. Frequency warping function for the computation of the MFCC. Image from [1]

A cepstrum is obtained by tacking the inverse Fourier transform of the logarithm of the spectrum signal. The cepstrum features are mainly used in the field of speech processing. A very popular feature in this domain are the *Mel-Frequency Cepstrum Coefficients (MFCC)*, where the frequency bands are distributed according to the mel-scale, instead of the linearly spaced approach. The mel-scale (see Figure 2.3) introduces a frequency warping effect in an attempt to conform with certain psychoacoustic observations which have indicated that the human auditory system can distinguish neighboring frequencies more easily in the low-frequency region.

Sharma et. al [29] provide an extended explanation of the audio features described in this thesis and many others.

2.1.3 Event Data

Due to the amount of money invested in this sport, soccer clubs collect a significant amount of data during matches. Hareen et al. [2] broadly divide these data into three types: Match sheet data, Event stream data and Tracking data. Match sheet data provides a more general information such as score, cards, and line-ups. Event stream data describes all the atomic actions the players perform with the ball such as passes, shots, tackles and throw-ins. Tracking data captures the positions of all players and the ball at all times.

As shown in Figure 2.4, the three types of soccer data differ in availability and granularity. Match sheet data are available officially or non-officially for almost all professional and semi-professional soccer matches but it provides a high-level summary of the match. In the other extreme, Tracking data provides a high level of detail but it is available only for a small number of competitions, where the richest clubs compete. Event stream data tries to find a balance between the limited granularity of the match sheet data and the detailed tracking information.

Event stream data are produced and sold by companies such as Wyscout[30], Opta Sports[31] and StatsBomb[32]. These companies have expert video analysts, who are trained and focused

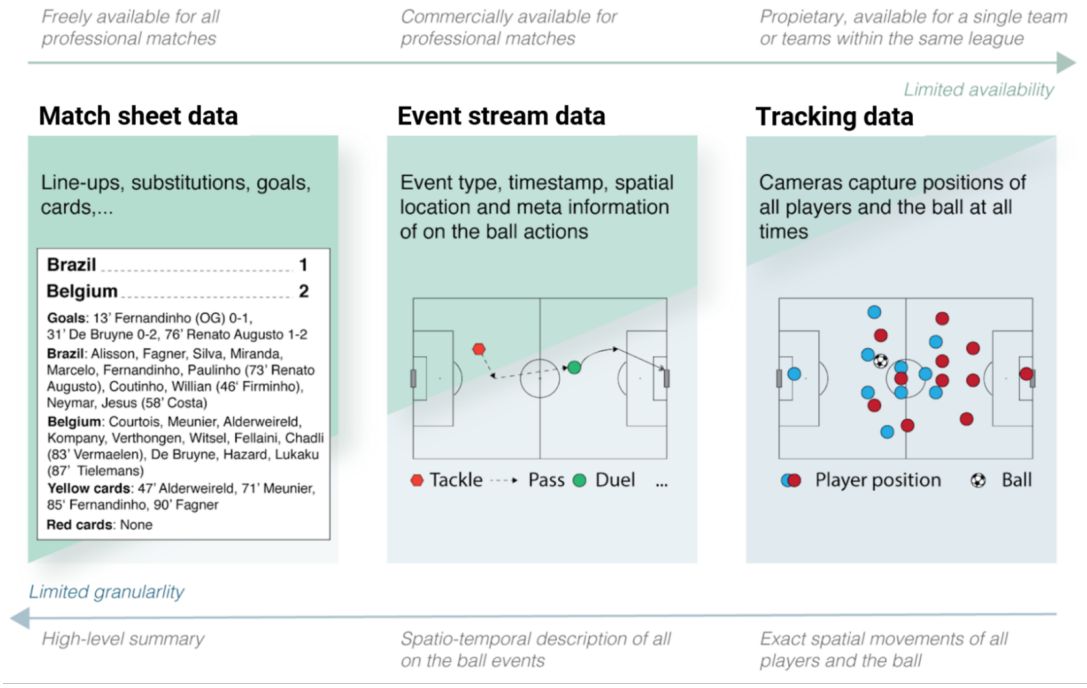


FIGURE 2.4: Different data collected from soccer games [2]. To the left the data which is easier to find but with lesser granularity. To the right the data which is less available but with a higher level of granularity.

on data collection for soccer. The tagging of events in a match is usually performed in live during the match by three analysts, one per team and another acting as supervisor of the output of the whole match. This tagging describes the match events happening on the field, each containing information about its type (shot, pass, foul, tackle, etc.), a timestamp, the player(s), the position on the field and additional information (e.g., success of the event). For a more detailed explanation in the procedure of data collection, the reader can refer to the work published by Pappalardo et al. [33] for Wyscout data and the one published by Liu et al.[34] for Opta data.

```
<Event id="1589317905" event_id="261" type_id="1" period_id="1"
    min="26" sec="34" player_id="54911" team_id="148" outcome="1"
    x="73.0" y="33.6" timestamp="2017-05-20T20:26:58.976"
    last_modified="2017-05-20T20:27:03" version="1495308423220">
  <Q id="1862571302" qualifier_id="213" value="5.0" />
  <Q id="1819292845" qualifier_id="155" />
  <Q id="1615989244" qualifier_id="141" value="4.9" />
  <Q id="1145325238" qualifier_id="212" value="20.7" />
  <Q id="1942384419" qualifier_id="56" value="Right" />
  <Q id="1262641701" qualifier_id="140" value="79.5" />
</Event>
```

FIGURE 2.5: Example of event provided by Opta. It is a tag from an xml file.

Figure 2.5 and 2.6 are examples of events registered by Opta and Wyscout respectively. Each event contains additional information, e.g., whether the card was yellow or red or whether the freekick was direct or indirect. Opta encodes this additional information as *qualifiers* (*Q* in

```
{
    "eventId": 8,
    "subEventName": "Simple pass",
    "tags": [
        {"id": 1801}
    ],
    "playerId": 122671,
    "positions": [
        {"y": 50, "x": 50},
        {"y": 53, "x": 35}
    ],
    "matchId": 2057954,
    "eventName": "Pass",
    "teamId": 16521,
    "matchPeriod": "1H",
    "eventSec": 1.6562140000000004,
    "subEventId": 85,
    "id": 258612104
}
```


FIGURE 2.6: Example of event provided by Wyscout. It is an entry of a json file.

Figure 2.5) and Wyscout encodes them as *tags*. The event stream data of Wyscout has a simpler structure, however the information provided by this company tends to be less accurate and is not as thorough as the one provided by Opta. Each vendor uses different definitions and terminology. For example the action of a shot on target faced by the goalkeeper, even if a goalkeeper does not touch the ball and the ball is going into the net or saved by a defender, is denoted as *Save attempt* by Wyscout, while Opta has *Save*, *Miss*, *Post* to describe the different situations of this event.

While in this thesis we use event stream data to extract event metadata for soccer summarization, several approaches use it for other tasks like analyzing advantage of playing on the home field [35], recognizing teams [36], automatically discovering patterns in offensive strategies [37, 38], predicting passes [39], detecting tactics [40], predicting the chance to score the next goal [41], evaluating the performance or contributions of the players [42, 43, 44] and modeling ball possession [45]. In all existing works, event data has been used to provide sports analytics. To the best of our knowledge our work is the first to consider event data as a core modality to be combined with more standard content-oriented data such as audio or video.

2.2 Action Recognition and Detection in Videos

The field of video content analysis can be divided in two main tasks: action recognition and action detection. Action recognition (also known as action classification) aims at classifying trimmed video clips into fixed set of categories. On the other hand, action detection also needs to predict the start and end times of the activities within untrimmed videos. In other words, with action recognition we can know if an action is present in a video and with action detection, we can also know in which exact part of the video the action is happening.

The success of Convolutional Neural Network (CNN) in image recognition have driven the progress in action recognition. One popular method that tries to apply image approaches on videos is the Two-Stream network that employs separate CNN to process different modalities (e.g., RGB frames and optical flow) [46, 47]. However, processing the video rather than only images, allows to capture the spatial and temporal features. This motives researchers to create models that take as input groups of frames instead of only one frame, such as 3D Convolutional Neural Networks (C3D) [48, 49], Temporal Segment Networks [50], Long-term Temporal Convolutions [51] and Non-Local Networks [52].

Compared to action recognition, action detection is a more challenging problem, and it is closer to real life scenarios since most videos in real life are untrimmed and they might contain multiple actions or information not relevant to any action. Many methods tackle this task in two steps, first to generate a number of candidate temporal windows and then an action classifier discriminates each window independently into one of the actions of interest. To create the different segments some methods use sliding windows [53, 54, 55], dictionary learning [56], graph convolutional networks [57] or recurrent neural networks [3, 58]. Recently, several works have implemented a strategy inspired by the method proposed in [59] for object detection in images. Instead of splitting the input and predicting if there is an action in each of the resulting split segment, the network predicts temporal proposals in a single pass of the video [60, 61, 62, 63, 64].

2.2.1 SST: Single-Stream Temporal Action Proposals

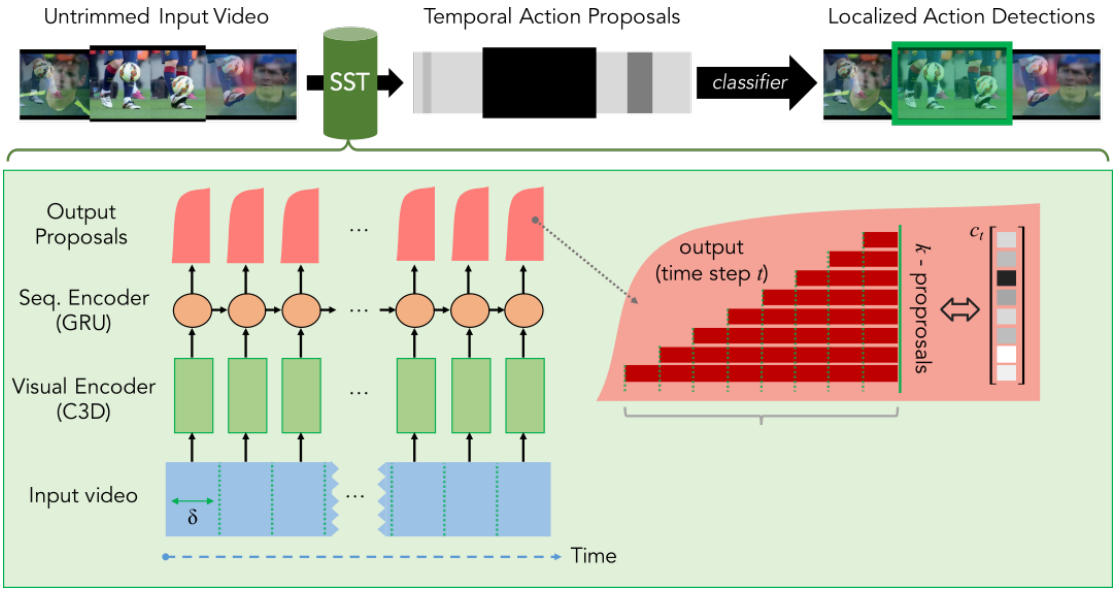


FIGURE 2.7: Schema and model architecture of SST approach. C3D features are extracted from the input video. These features are the input to a GRU-based model, which outputs k proposals at each time step. Image from [3]

In chapter 5, we use SST [3] to compare with our generation of action proposals method. It is a deep architecture for the generation of temporal action proposals in untrimmed video sequences.

A graphical representation is shown in Figure 2.7. The model takes as input an untrimmed video sequence and feeds it through a C3D, with a time resolution of $\delta = 16$ frames. Then each time step t of a GRU-based model receives the corresponding encoded C3D feature vector. And finally, the output are the confidence scores of multiple proposals at each time step t.

Concretely, at each time step t the model outputs confidence scores $\{c_t^j\}_{j=1}^k$ that correspond to k proposals $P_t = \{(b_{t-j}, b_t)\}_{j=1}^k$, where the tuple (b_{t-1}, b_t) indicates a proposal with start and end at frames b_{t-1} and b_t, respectively. All proposals considered at time t have a fixed ending boundary at t, and the model considers proposals of sizes $1, 2, ..., k$ time steps.

2.2.2 Multiple Instance Learning

This paradigm was first described by Dietterich et al. [65] to predict drug activity. Then it was introduced in many other fields like object tracking [66], object detection [67, 68] and image tagging [69]. This paradigm is very popular in medical images [70, 71, 72], where an entire image of the organ or tissue is labeled as malign but only a small portion of it is actually malign.

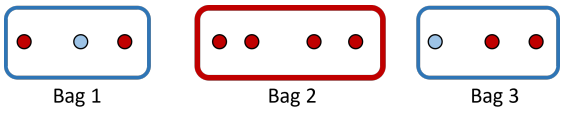


FIGURE 2.8: An illustration of the concept of Multiple Instance Learning. Dots are instances and rectangles are bags. Blue corresponds to positive and red corresponds to negative.

In the classical supervised learning problem the objective is to find a model that predicts a target value $y \in \{0, 1\}$, for a given instance. In the case of MIL, instead of a single instance there are groups of instances called *bags*. There is also a single binary label Y associated with the bag. Furthermore, it assumes that individual labels exist for the instances within a bag, i.e., $y_1, ..., y_k$ and $y_k \in \{0, 1\}$, however there is no access to those labels and they remain unknown during training. Then as it is illustrated in Figure 2.8, in MIL a bag is labeled as negative if all the instances inside the bag are negative and a bag is labeled as positive if at least one instance of the bag is positive:

$$Y = \begin{cases} 0, & \text{iff } \sum_k y_k = 0, \\ 1, & \text{otherwise} \end{cases} \tag{2.1}$$

Wang et. al propose MI-Net, a neural network performing MIL in an end-to-end manner, which take bags with a various number of instances as input and directly output the labels of bags. MI-Net has three fully-connected layers and one MIL Pooling Layer (See Figure 2.9). The network focuses on learning bag representation, rather than predicting instance probability. No matter how many input instances there are, the MIL Pooling Layer aggregates them into one

feature vector as a bag representation. Finally, a fully-connected layer with only one neuron and sigmoid activation takes the bag representation as input and predicts bag probability.

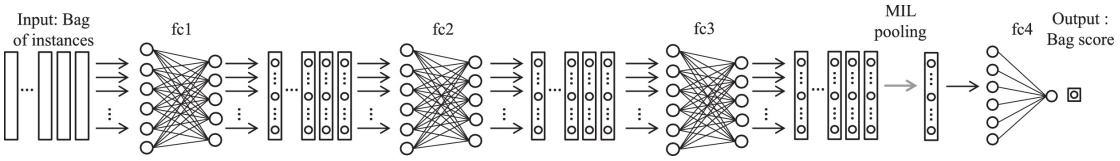


FIGURE 2.9: A MI-Net with three fully-connected layers and one MIL pooling layer. Image from [4]

To the best of our knowledge our work is the first to consider MIL to generate action proposals.

2.3 Video Summarization

The terms of video summary and video highlights are used in several approaches as the same task, but the video highlights normally involve only the detection of important events in the video while the summary also preserves important structural and semantic information [73]. An example of this difference in soccer is that a summary might contain events that are not considered very important for a match, such as a goal attempt, but they might be relevant to show that even though the team did not score any goal, it had clear opportunities during the match.

Video summarization can be defined as a technique to create a short version of the original video while preserving the main story/content. The produced video summary is usually composed of a set of representative frames (key frames), or video shots that are stitched in chronological order to form a shorter video [74, 75]. When the summary contains only key frames it is know as video storyboard, which is just a slide show of frames. And the summary composed of shots is known as video skim, which usually also includes the audio.

2.3.1 Summarization of general-purpose videos

The task of finding the most representative parts (either frames or shots) of a video has been tackled in many ways. For instance, the observation that similar videos share similar summary structures [76, 77, 78]. Taking as inspiration semantic segmentation, Rochan et al. [79] use a fully convolutional network across time, where the output is a mask showing the relevant frames for the video summary. On the other hand [80, 81] use a combination of objectives like interestingness, uniformity, representativeness to identify the most appealing moments. Owing to the fact that it is very difficult to create a video summarization dataset and the recent success of Generative Adversarial Networks, there are several works based on unsupervised approaches [82, 83, 84]. Zhang et al. [5] were the first ones using LSTM for video summarization, their method is a bidirectional LSTM followed by a Multi-Layer Perceptron. Although LSTM can model long-range structural dependencies, Zhao et al. [6] propose a hierarchical LSTM to help the model to handle particularly long sequences. For a more extensive list of the methods on

video summarization using deep neural networks, the reader can read the survey provided by Apostolidis et al. [75].

In chapters 4 and 5, we use vsLSTM [5] and H-RNN [6] as comparison of general-purpose summarization and frames-based methods.

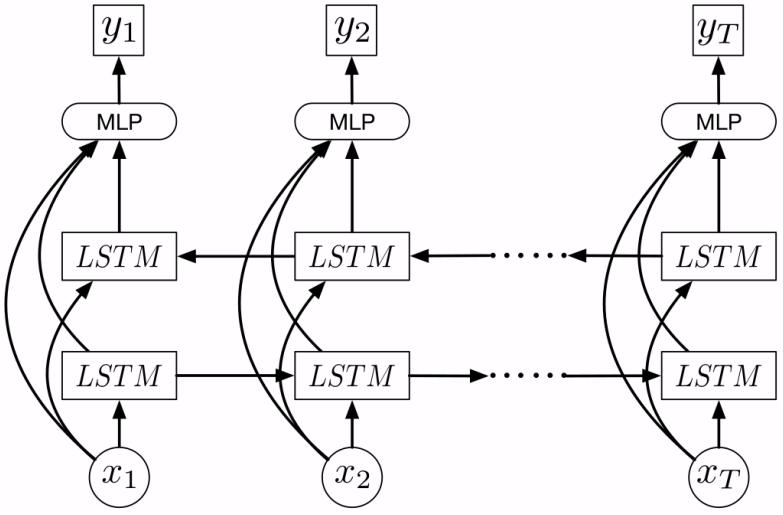


FIGURE 2.10: vsLSTM schema. The architecture is composed of a bidirectional LSTM, followed by a multi-layer perceptron. Image from [5]

2.3.1.1 vsLSTM: Video Summarization with Long Short-term Memory

The architecture is composed of two LSTM layers, one layer models video sequences in the forward direction and the other the backward direction. This representation is also called bidirectional LSTM. The forward and backward chains model temporal inter-dependencies between the past and the future. The inputs of the layers x_t are visual features extracted at the t-th frame. The outputs combine the LSTM layers' hidden states and the visual features with a multi-layer perceptron, representing the likelihood of whether the frames should be included in the summary.

2.3.1.2 H-RNN: Hierarchical Recurrent Neural Network

This method propose a hierarchical recurrent neural network with two layers (see Figure 2.11). The first layer is an LSTM that encodes short video subshots cut from the original video, then the final hidden state of each subshot is input to a bidirectional LSTM and the output of this second layer is used to predict the confidence of each subshot to be selected in the summary. The two layers exploit the intra-subshot and inter-subshot temporal dependency, respectively.

For a clearer explanation, the frame sequence is separated into several fix-sized sequences denoted as subshots $(f_1, f_2, ..., f_s), (f_{s+1}, f_{s+2}, ..., f_{2s}), ..., (f_{m*s+1}, f_{m*s+2}, ..., f_T)$ where f_i is the feature representation of frame i, T denotes the total frames in the video, m is the number

of subshots, and s is the length of each subshot. Then, all the subshots of the video are input to the first LSTM layer. τ_i denotes the final hidden state of the i-th subshot and the sequence $(\tau_1, \tau_2, ..., \tau_m)$ is input to the second layer. Finally, the output of this last layer is used to predict the confidence of a certain subshot to be selected into the video summary.

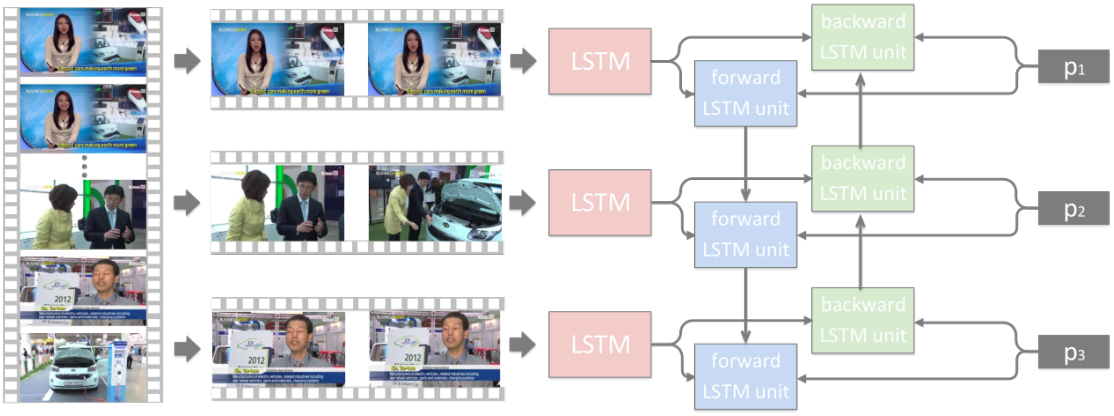


FIGURE 2.11: H-RNN schema. The model contains two layers, where the first layer is an LSTM and the second layer is a bidirectional LSTM. Image from [6]

2.3.2 Video Summarization for Sports

A possible solution to automatically generate summaries of sport videos is to use the previously described methods for the summarization of general-purpose videos. However, they are not usually suited for sports domain and all the specific challenges that this task entails. Some methods [79, 85, 86] need to load the whole video in memory as input sample, which is not feasible for a 90-minute video such as a soccer match. Several approaches are based on the maximization of diversity, trying to minimize the number of similar shots [87, 5, 82, 88, 89, 90] but the maximization of diversity is not an optimal approach for sports summarization. For instance, in soccer actions like goals, corners or free-kicks are visually very similar since they are located in the same area of the field. And this situation holds for many sports.

Early works in video summarization for sports mainly rely on hand-crafted heuristics. Figure 2.12 depicts different steps usually followed to create a video summary. They exploit the characteristics of the field (lines, goal mouth), cinematographic properties like the camera motions, slow motion or zooming, and also specific edition patterns like the replays, to select representative parts of the video [7, 91, 92, 8]. Many of these rules are indeed based on knowledge acquired by experts but there are usually many aspects that constraint the quality of the output. For instance, Figure 2.13 shows an example of the steps followed to detect the penalty box. It uses image processing techniques assuming that the field is green, there are three parallel lines, and the field is the largest green area of the image. However, it ignores possible drawbacks like

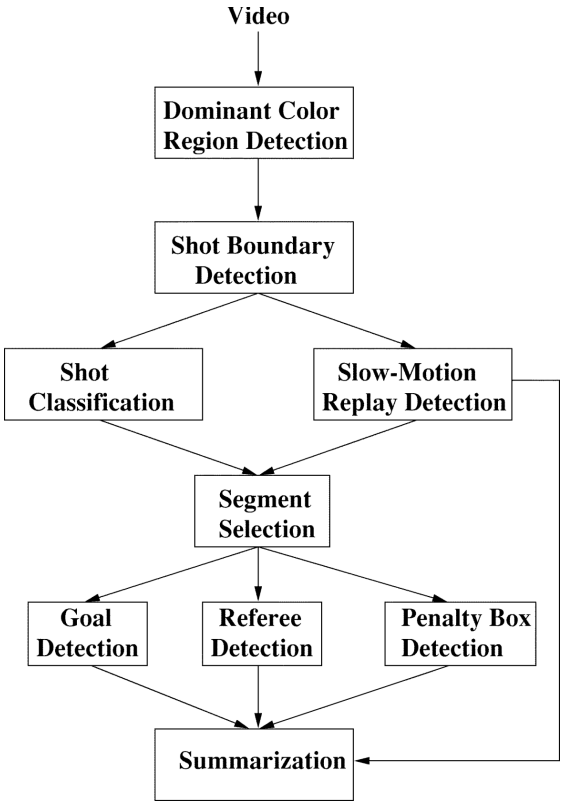


FIGURE 2.12: Diagram of video summarization based on heuristics. Image from [7]

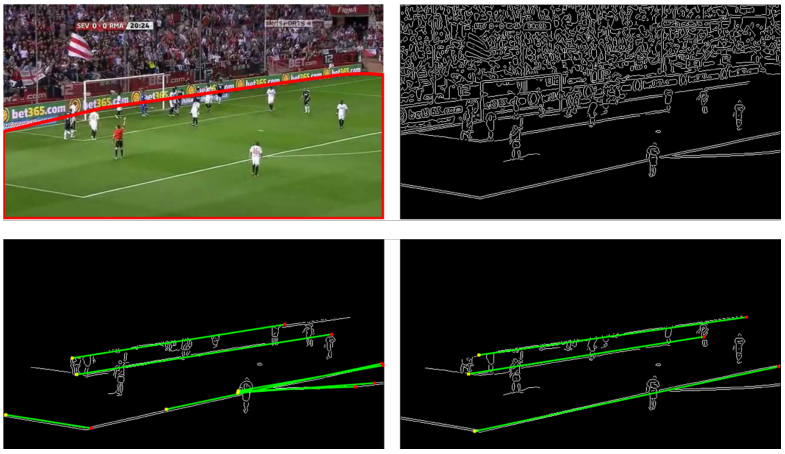


FIGURE 2.13: Example of penalty box detection. Using image processing techniques, it detects boundaries of the grass field and lines to finally choose three parallel lines. Image from [8]

the change of grass color in some stadiums, green uniforms, or different angle of the camera. And we can describe similar examples for the techniques of almost every box of Figure 2.12.

Even though other methods use sophisticated techniques as deep learning, they still exploit sports related characteristics. Most of them following the play and break technique described in Figure 2.14. Jiang et al. [15] use play and break technique to split the video into shots to then detect soccer events using RNN and frame features extracted from a CNN. Liu et al. [9] propose to first use 3D convolutional networks to locate the actions and then use play and break

technique to segment the actions. Yu et al. [93] find the frames containing the competition logo to locate the replays in the match, then VGG-features are extracted from the video frames before the replay to use them as the input of an LSTM that classifies soccer events. Javed et al. [94] mix heuristics knowledge based on the replay information and an extreme learning machine to detect key-events. More recently, Agyeman et al. [95] proposed to use features extracted from 3D convolutions to then train an LSTM for action classification.

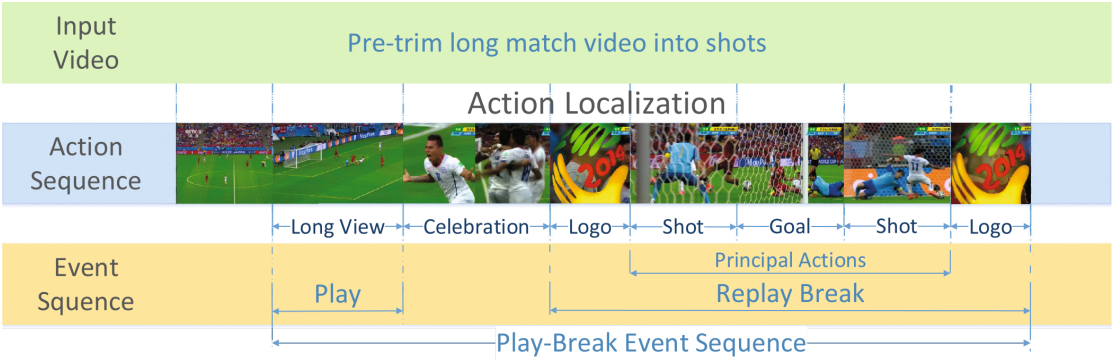


FIGURE 2.14: Play and break example. A long view scene (Play) followed by non-long view actions (break), and shot sequences surrounded by a pair of replay logos. Image from [9]

The main limitations in the state-of-the-art of video sport summarization is the lack of standardization in the evaluation process and the assumptions based on heuristics. Most of the methods do not evaluate their results with the commonly used summarization metrics, they usually focus on the accuracy detecting the most important actions of the match, such as goals.

2.3.3 Multimodal Sports Summarization

There are several works in the state-of-the-art which prefer to tackle the problem of sports summarization as a multimodal task instead of using only the video, since multiple modalities play an important role to choose the best moments of sports videos.

Some methods propose to use the interactions in social networks like the tweet streams during the game. Corney et al. [96] identify the team that each Twitter user supports and produce a subjective summary of events as seen by the fans. EpicPlay [97] leverage the fact that American football is a highly structured game to annotate Twitter streams and then select video highlights from the game. Huang et al. [98] analyze Twitter streams and use document summarization approaches to generate textual summaries of Basketball matches. Tang et al. [99] use deep learning to classify soccer actions from the text timeline found in several web pages.

Other methods detect the intervals with highest motion from the optical flow. For instance, Pandya et al. [14] exploits optical flow techniques to overcome challenges on frame-based approaches such as illumination and ground conditions. And Mendi et al. [100] propose to select key-frames using motion analysis in Rugby 7s videos.

Audio is another relevant modality in sports, since it helps to identify the excitement of the commentators and the crowd, sometimes the ball hit like for tennis or baseball, and the whistle.

Several approaches based their decisions on only audio features in Baseball, Rugby, Golf and among other sports [101, 102, 103].

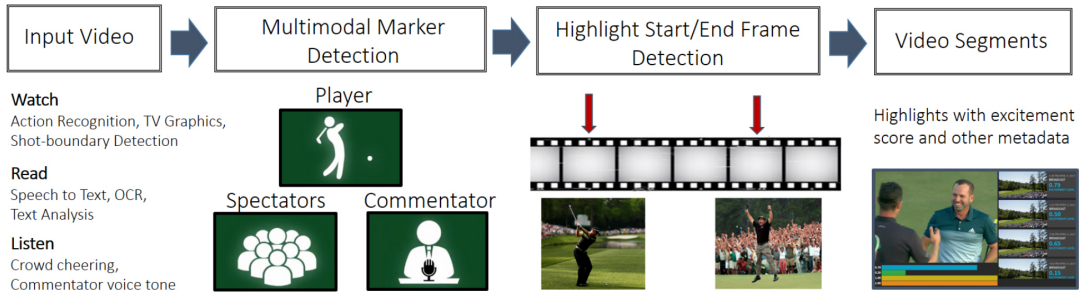


FIGURE 2.15: Highlights detection in golf using multiple features extracted from different modalities. Image from [10]

Several methods merge different modalities like the sound energy, the score, camera motions, players' reactions, referee whistle, etc [104, 10, 16, 105, 106, 107]. Figure 2.15 shows an example of the detection of highlights in golf, using multimodal features such as audio tone and speech to text analysis to obtain a commentator excitement score, and frame features to detect player celebration. The merging is usually performed based on heuristics, thresholds, and sports-related rules.

Chapter 3

The challenge of summarizing sport videos

Analyzing video content to produce summaries and extracting highlights in sport videos has been of great interest for decades. Sports is one of the domains that has invested the most in the video analysis field, owing to the massive popularity of sports in different content platforms, to its accordingly huge business market, and lately to the emergence of sport bet companies in many countries. Despite this popularity, there are still many issues to solve in the automatic generation of summaries in sport videos.

In this chapter, we analyze the challenges of summarizing sport videos, describing first the differences between general-purpose videos and sport videos. Then, we try to explain how event data provide richer information and is more efficient than video content. Finally, we set the definition of event and action, explaining also how to detect the actions of a match based on video summaries.

3.1 General-Purpose Videos vs Sport Videos

The field of video summarization of general-purpose content has progressed rapidly providing promising results. However, there are several differences between summarizing general-purpose videos and sport videos. In this section we will describe the main differences in terms of benchmark datasets and some cases of subjectivity we can find in sport videos, and then we will provide the numerical performance in soccer matches of a method designed to detect actions in general-purpose videos.

3.1.1 Benchmark Datasets

In terms of benchmark datasets for video summarization [111, 112, 113, 114, 115], the length of general-purpose videos varies from 6 to 10 minutes and the summary length is around 15% of the original video (56 to 90 seconds). On the other hand, for sport videos a match can vary from

TABLE 3.1: Comparison between General Purpose dataset and Sports videos. *Soccer matches* represent a sample of 100 soccer matches from the Premier League competition, the action duration is taken from the actions posted by official broadcasters. Action ratio is the ratio between the action duration and the video duration.

Dataset	Number of videos	Action duration (seconds)	Video duration (seconds)	Action Ratio
ActivityNet [108]	20K	49.21	116.7	49.12
THUMOS-14 [109]	412	4.6	213	31.01
HACS [110]	1.5M	40.6	156	30.84
Soccer matches	100	30.84	6300	22.16

one to several hours and in sports like soccer the summaries are about few minutes, for instance a medium-large summary can be 5 minutes long which represents less than 6% of the original video of a soccer game.

When comparing in terms of all the actions in a video, instead of only the actions belonging to a summary, there are also clear differences. The main benchmark datasets for action recognition in general-purpose videos [109, 108] have a mean video duration between 100 and 200 seconds, where the actions represent at least 30% of the video and less than 20% of the videos contain more than one type of action. However, in soccer datasets [116] the mean video duration is more than one hour long, the actions represent at most 20% of the video and all the videos contain more than one type of action (see Table 3.1).

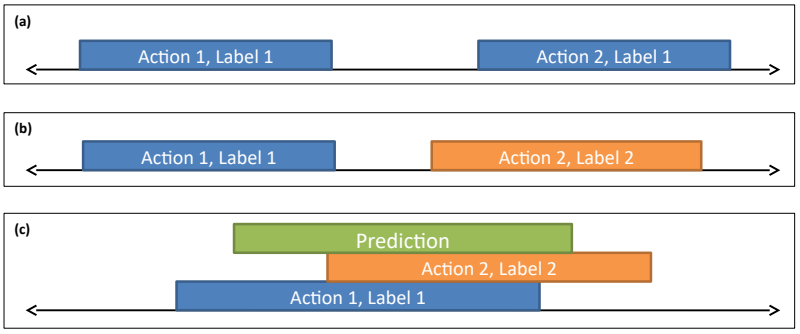


FIGURE 3.1: Types of temporal locations of ground-truth actions. **(a)** Multiple temporal locations of the same action. **(b)** Multiple temporal locations of different action. **(c)** Overlapped temporal location.

Even though these datasets provide multiple temporal locations within the same video, it is very common to find the case **(a)** of Figure 3.1, where all the actions of the video belongs to the same class. In ActivityNet, 3679 videos contain multiple temporal locations of actions but only 13 of them contain at least two different action classes, as the case **(b)** of Figure 3.1. For THUMOS-14 the situation corresponds to 61 videos over 412.

Other characteristic to analyze is the overlapped actions. In THUMOS-14, 13% of the videos (55 videos) contain at least one temporal location that overlaps with other temporal location but

for ActivityNet is only 0.9% (185 videos). These are important characteristics of sport video content like soccer, where we often find several temporal locations for actions such as *goal*, *substitution* and *yellow card* within the same video match. It is also very likely that temporal locations of actions such as *goal* and *corner* overlap.

Considering that our samples are entire video matches, a significant part of the videos is background (no-action). The actions represent only 22% of the soccer dataset, compared with 31% in THUMOS-14, 31% in HACS and 49% in ActivityNet (see Table 3.1).

There are also differences in terms of the content of the videos. In general-purpose datasets the videos are usually very different from one class to another (e.g., skiing and rowing are two classes in THUMOS-14 dataset) since their objective is to cover a broad range of activities. Therefore, these videos usually present different backgrounds and landscapes. In contrast, in sports like soccer, where the long-shots are very common and players are mainly located near the goal mouth, most of the actions are very visually similar.

3.1.2 Subjectivity

Besides the differences with the benchmark datasets of general-purpose videos, the subjectivity is also an important issue in the context of sport videos. Broadcasting companies aim at extracting video segments, traditionally named "actions", that should be attractive for their users. Therefore, the objective is not only to find the exact time of the climax of the action, but also to identify the context that led to this action and what happened after. The actions in this specific context can be seen as very little stories. Then the definition of when starts the action and when the conclusion of the action is reached is subjective.

Figure 3.2 illustrates for instance three actions of the same match labeled as *Goal* but with different beginnings and ends. The frames of the center show when the ball goes into the goal mouth, the ones on the left show how the actions start with a corner, a goalkeeper's pass and a throw-in, and the ones on the right show how they finish with the frames that depict the new score and the beginning of a replay.

This previous explanation leads us to another challenge related to the metrics. In temporal action detection, methods are often evaluated using mean Average Precision (mAP). However, as Colin et al. [117] stated, the mAP metric is less relevant than for instance F1-score which *"does not penalize for minor temporal shifts between the predictions and ground truth, which may have been caused by annotator variability"* while with *"mAP detection scores, if there is more than one correct detection within the span of a single true action then only one is marked as a true positive and all others are false positives"*.

Therefore, using mAP to evaluate temporal action detection in sport videos would not be very accurate. In real-life scenarios the temporal locations of the actions might vary according to different aspects such as the operator, the broadcasting company, the platform where the action is posted, the country, the history of the team, among others. Similarly, if there are multiple

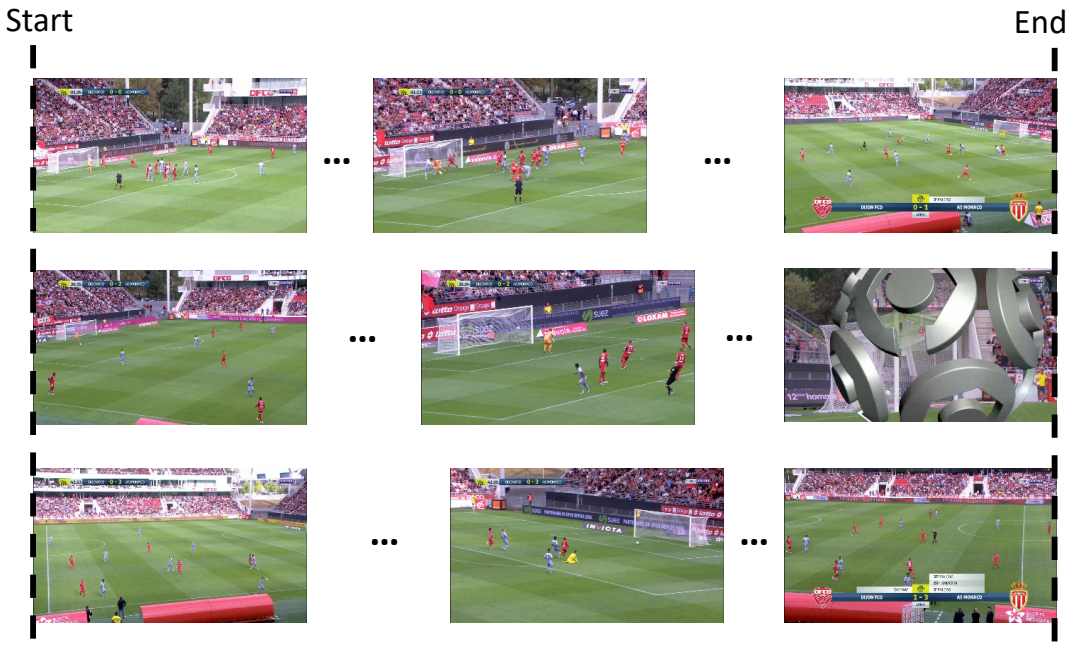


FIGURE 3.2: Subjective definition of start and end in ground truth. Each row shows a different example of Goal in the same match. The pictures on the left are the start frames (set by a human operator), on the right are the end frames (also set by a human operator) and in the middle are the frames where the ball goes into the goal mouth.

predicted temporal locations matching one ground truth, using mAP there will be only one True Positive and the rest will be False Positives while all these candidate temporal locations should be proposed to the user to offer more flexibility with respect to the subjectivity of the temporal action location boundaries.

It is important to notice that this last challenge makes even more difficult the comparison between existing methods for general-purpose videos and methods for sport videos.

3.1.3 Action Detection in Sport videos

So far, we have described the main differences between general-purpose videos and sport videos where we can argue that a model built to detect actions in general-purpose videos will not have a good performance in sport videos, due to the subjectivity, the sparsity of the actions or the difficulty to differentiate between action and no-action. However, we consider that it is relevant to evaluate the performance of a method, which was originally proposed to detect actions in general-purpose videos, using sport videos.

We created a soccer action dataset with 20 games. The temporal annotations were made in real-time by professional editors i.e., while the matches were broadcast. There are annotations for 20 different actions: *Start Match, End Match, End First Half, Start Second Half, Saved Field, Corner, Shot on Target, Shot not Target, Penalty Missed, Goal on Penalty, Goal on Field, Yellow*

Card, Free-kick, Substitution, Post, Offside, Red Card, Yellow Card, Chance, Injury. These actions are in average 30 seconds long but they can vary from 7 to 85 seconds.

We chose R-C3D [61] since it is one method for action detection with competitive performance in terms of speed and mAP. We used the code and weights published by the authors. We decided to initialize the network with the weights trained on THUMOS-14 because among the datasets they have explored, THUMOS-14 is the dataset with the closest properties to ours (i.e. ratio actions-vs-Background). We have randomly chosen 80% of our 20 matches for training and the remaining 20% for testing.

To perform a fair comparison, we just analyze the action location task, evaluating the ability of the method to distinguish between action and no-action, since for the task of video summarization the type of action is not always necessary. Then we do not take into account the prediction of the class, we only measure the number of missing actions. An action is considered as missing if the method did not predict any temporal location that overlaps with it. We have found that R-C3D misses 84 actions from the 266 total test actions of the soccer action dataset, which represents 32%. This is a significantly high percentage of missing actions considering that we are evaluating the model with very easy conditions, i.e., not considering the intersection between the real interval and the predictions nor the misclassification of the type of action.

3.2 Video Content vs Event Data

One of the main challenges for the broadcasting companies in the recent years is the speed at which the customers expect the content to be available. Broadcasters need to provide summaries as soon as the match is finished. Despite this need, broadcasting companies usually do not fully rely on automatic algorithms to generate these summaries, instead they mainly rely on human editors aided by algorithms. These editors use the video content to build a summary but most of their work is based on event data since using the video content as main source is very time consuming and not easy to come back to retrieve information.

Automatized algorithms might help to reduce the work of human operators, however there is still a lot of information to process. In the case of soccer, the match duration is around 90 minutes, at a rate of 25 frames per second it corresponds to 135K frames of at least 112x112 pixels (C3D [49] input size). In addition, there are still some important challenges for these algorithms in terms of video content. The homogeneity of the field and the visual similarity between different types of actions make more difficult the differentiation of relevant and non-relevant information. The quick movement of the players around the entire field, the low resolution and the occlusions hinders their detection and tracking. The subjectivity in editorial decisions can make a replay or person-related closeup skip information of the current state of the match which can cause a disruption in time continuity on the video.

On the other hand, as mentioned in Section 2.1.3 there exist event stream data provided by companies like Prozone, GeniusSports, Opta, WyScout, and many others, from which we can extract event data. Table 3.2 shows an example of the data collected for a *pass* event.

TABLE 3.2: Example of information collected for a *pass* event.

Feature Name	**Value**
Type	Pass
Qualifier	Short
Start Location	(10,30)
End Location	(15,35)
Time	322
Period	1H
Team	Chelsea FC
Outcome	1 (accurate)
Player	Lukaku

The number of events in a match is significantly smaller than the number of frames. In the case of soccer, a match has about 1500 events overall. In terms of information to process, instead of 112x112 values each event is represented by few values like the type, location, and the time. In addition, this event data contain richer information and reduces significantly the subjectivity. The events only register sport-related actions, unlike the video they do not contain replays, celebrations, crowd reactions or advertisements. For instance, Figure 3.3 depicts an example of a goal summary clip. This clip contains frames that are not directly related to soccer events such as the preparation of the player to kick the ball in the corner of the field, the players' celebration, the reaction of the crowd, and the replay. On the other hand, the editorial decision of showing the replay hides some events that are happening on the field.

FIGURE 3.3: Example of summary clip and the events inside its time interval. On top there are frames sampled from the summary clip and on bottom the events with an approximation of the time location in the video clip.

3.3 Event and Action

In the multimedia community, the concept of event is generally vague and overlaps with the concept of action and activity. Chen et al. [118] define the concept of action and actionness with

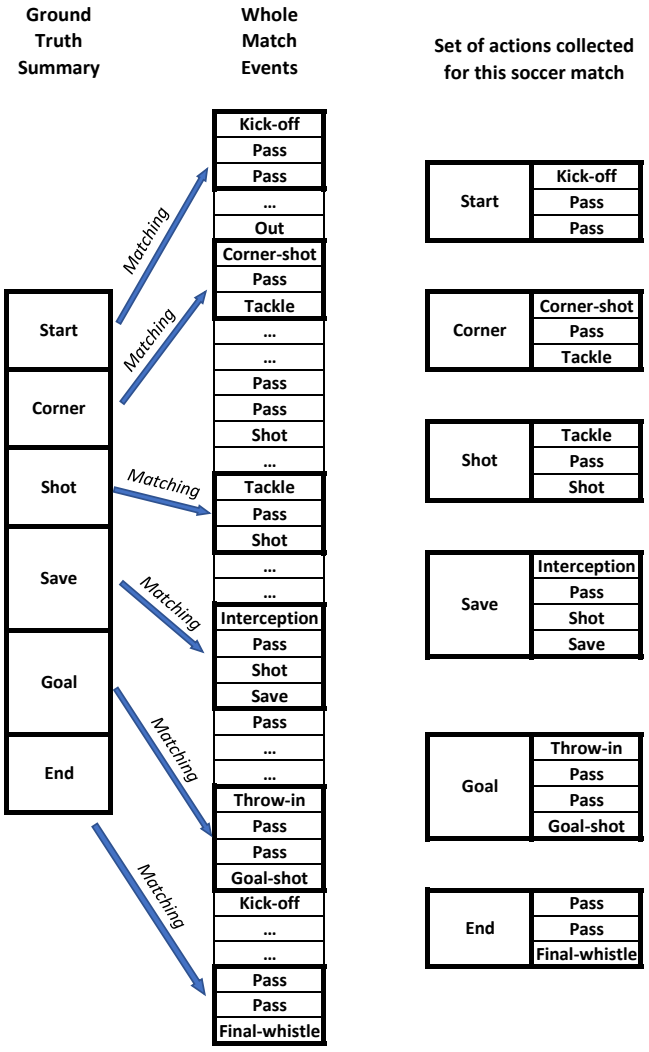


FIGURE 3.4: Example of the event representation of a whole soccer match, displayed in the middle column. Events are all the atomic activities happening on the field during the given match. The first column represents the ground-truth summary of the given whole match. An action is a set of consecutive events that belongs to a summary.

4 aspects that define an action: an agent, an intention, a bodily movement, and a side-effect. Dai et al. [60] define an activity as a set of events or actions, with a beginning and an ending time. Sigurdsson et al. [119] argue that temporal boundaries in activities are ambiguous.

In this thesis, we define the concept of *event* as any of the atomic activities happening on the field such as *Pass, Tackle, Out, Goal-shot, etc*. This definition is very similar to the one described by Giancola et al. [116], where an event is anchored in a single time instance. Thus, the events correspond to the previously described metadata collected by Opta, Wyscout, and other companies. Then, an *action* is a continuous set of consecutive *events*. This way of defining actions allows to disambiguate their temporal boundaries.

After clarifying the difference between event and action, there are still two main challenges, in the context of soccer it is unclear when a given action such as *scoring a goal* begins and ends,

and there is no available dataset for action detection in soccer videos. For this reason, we propose to use weak supervision principles to detect the actions in our soccer matches. We assume that if a set of consecutive events belong to a summary, then they were implicitly labeled as action by an editor. Therefore, we use as reference the different video clips of the summary videos.

However, these video clips were selected based on video content, they do not contain only events. For instance, the clip containing the event *Goal-shot* might also show some events leading to the goal, then the celebration of the players, the reaction of the crowd or coaches and the replay. In our context, we want to reduce as much as possible the subjectivity, later to be added by editorial decisions, hence we consider only sports-related events and define an *action* as the set of consecutive *events* that might belong to a summary video clip.

To give a more concrete example, we illustrate how we define events and actions in Figure 3.4. On the left, we show the *actions* of the summary of the given match. Then, we match these *actions* with the event sequences which correspond the best (comparing the timestamp of the events and the action time in the match) in the whole soccer match. We thereby obtain a set of event sequences that belong to a true summary. As aforementioned *events* are all the atomic activities happening on the field during this match.

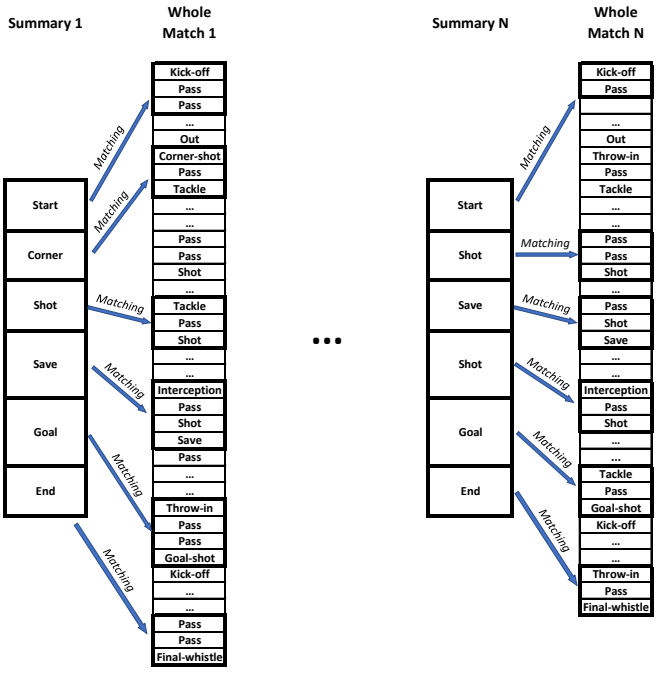


FIGURE 3.5: Example of collecting actions, i.e. sequences of events from all training summaries.

As displayed on Fig. 3.5, each summary provides a set of actions thus collecting the corresponding sequences of events from the corresponding whole matches. This process results in a vocabulary of actions that is going to be the basis for detecting potential summary actions (see Figure.3.6).

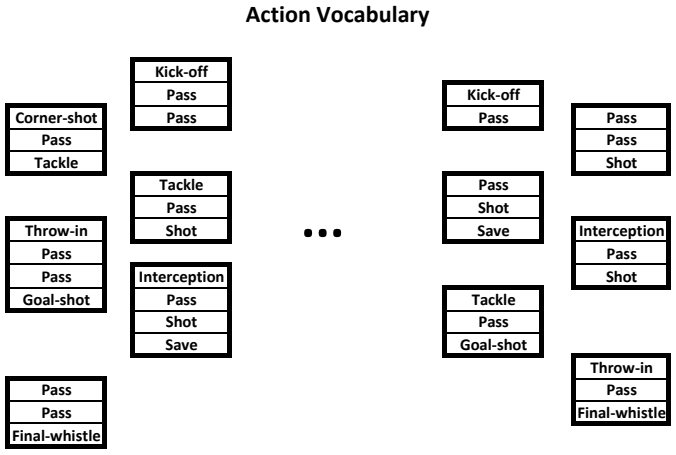


FIGURE 3.6: All the actions collected from summaries in the training set and the corresponding sequences of events in all the matches in the training set.

Let us analyze a more concrete example, we illustrate the process in the first two columns of Figure 3.7. In the first column on the left, we present an example of the event representation of a given soccer match. In the second column, we show the *actions* of the given match, thus the event sequences (corresponding to these actions) belong to the summary of this match or to the summary of any other match in the training set. This explains also the diagonal stripes area which represent an action, here the sequence of events {*Pass, Pass, Shot*}, which is not part of the final Ground Truth Summary of that given match (produced by a human operator) but the exact same sequence is part of the Ground Truth Summary of another match in the training set.

It is thus important to note that following this definition, many events do not belong to any action. For instance, in the first column on the left of the Figure 3.7, the event *Out* happening before the event *Corner-shot* or the event *Kick-off* right after the event *Goal-shot*, are not part of any action since they do not belong to an event sequence of the current summary as it can be seen in the third column corresponding to the Ground Truth Summary for that match, but they do not belong either to any other sequence from any other summary in the training set.

The next step to identify the actions of our soccer matches is to label as action any sequence of events that is identical to any of the actions vocabulary.

A more detailed explanation is given by a joint analysis of the columns *Whole Match Actions* and *Ground Truth Summary* from Figure 3.7. All the actions of the ground-truth summary are part of the action proposals of the match. And even though the sequence of events {*pass, pass, shot*} is not in the ground-truth summary of the corresponding match (Blue cell), it is still considered as a potential action proposal for this match because this sequence of events was found in the ground-truth summary of another match from the training set. It is important to notice that to find the actions in a match, we only look for actions present in matches from the training set and not in any match from the test set or the validation set.

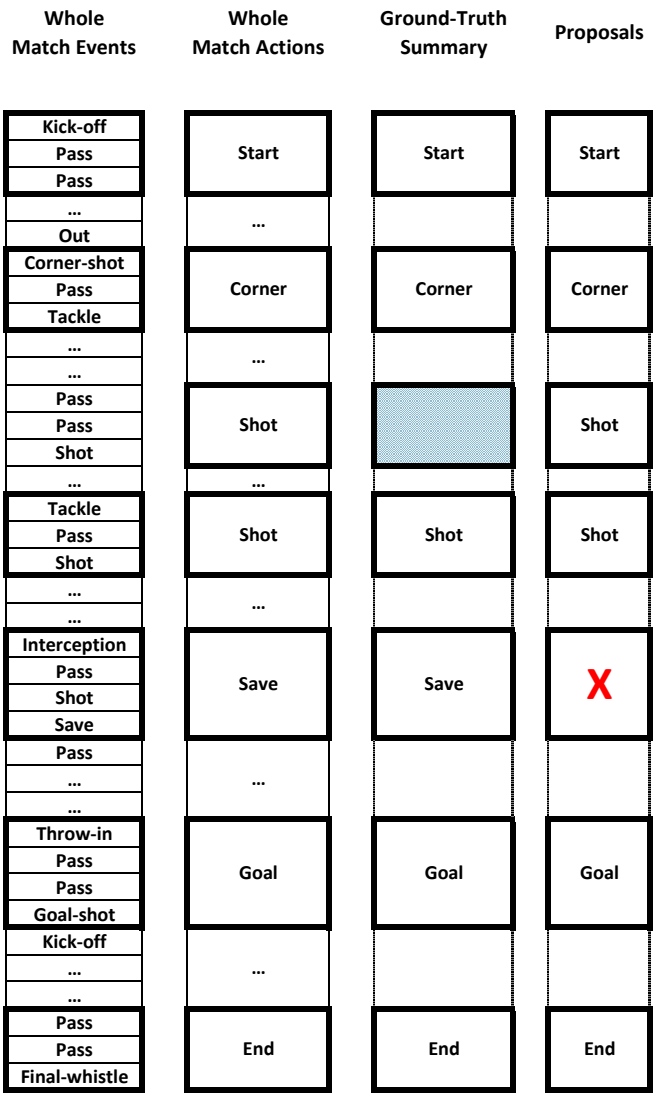


FIGURE 3.7: Example of the event representation of a whole soccer match, displayed in the first column. Events are all the atomic activities happening on the field during the given match. An action is a set of consecutive events that might belong to a summary. The third column represents the ground-truth summary of the given whole match, the blue cell indicates the action does not belong to the summary of the given match. A proposal is a set of consecutive positive events predicted by our Action Proposal Generation stage. The fourth column describes thus the set of proposed actions to be possibly in the final summary. The red cross X indicates that the action was not predicted as proposal.

Chapter 4

Fully automatic video summarization system

Many state-of-the-art methods aim at summarizing a video in only one step, trying to identify the key frames that increase the diversity of the resulting summary. However, that approach might lead to three main limitations in the summarization of sport matches. First, as we mentioned before, the diversity is not a very accurate objective for a sports summary, especially in soccer where the actions are very visually similar. Second, the summary of a soccer match is not just sparse key frames spread along the video, usually each clip of the summary represents an action, containing the main event and its context. And third, a soccer match is at least 90 minutes long, processing all the video in one step might not be feasible in a technical aspect.

Therefore, we split the summarization process in two tasks, first detect all the actions of the match that could be selected to be in the summary, and then decide which of these actions indeed belong to the summary. Considering that the amount of information to process using event data is significantly less than the one using video frames, our approach use event data instead of video frames for these two tasks. However, the events do not consider editorial decisions that are helpful to provide a better user experience (such as replays, celebrations and reactions), we propose a third task that uses the video frames to define the beginning and end of the actions.

Our first approach for automatic soccer video summarization is illustrated in Figure 4.1. It combines precision and relevance of event data with the expressiveness of multimedia content. It consists of three stages:

- The *Proposals* stage deals with the similarity of inter-categorical actions. Two very similar sets of events $\{pass, tackle, pass\}$ can be parts of two different actions goal-opportunity and corner, the former being in the summary while the latter not. This issue is addressed by a Multiple Instance Learning (MIL) network providing a score for each event, further concatenated to end up with consecutive positive events as proposals.

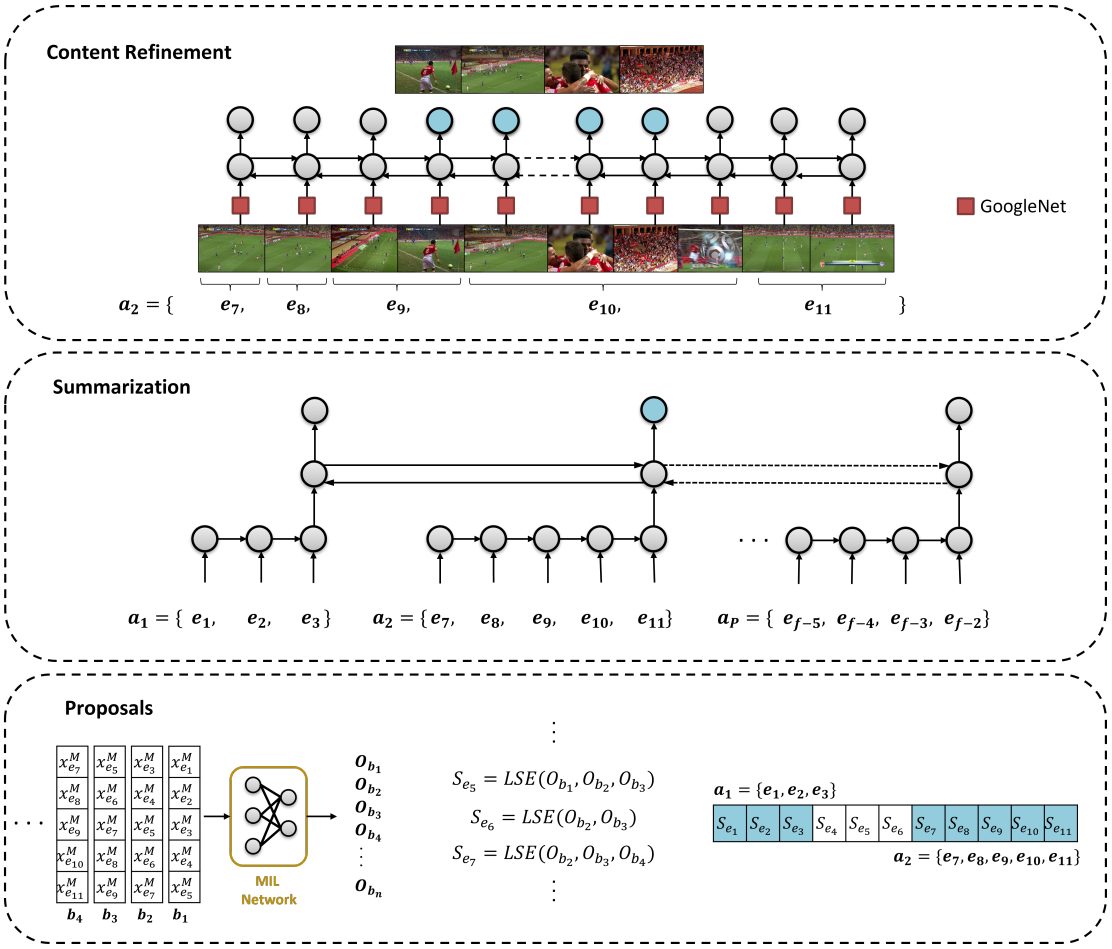


FIGURE 4.1: Our fully automatic video summarization system. The Proposals stage takes as input events grouped into bags b_n to be processed by a Multiple Instance Learning (MIL) Network, then LSE function is applied on all the predicted values of each event to finally group the consecutive positive events into Proposals. The Summarization stage is composed by a hierarchical LSTM: bottom LSTM layer creates a representation of each proposal a_p and an upper bidirectional LSTM decides whether the proposal is part of the summary. The Content Refinement stage takes the frame features of the positive proposals and decide which ones of them indeed belong to the summary.

- The *Summarization* stage consists of a multimodal Hierarchical LSTM with two levels. The LSTM of the first level accumulates in each proposal from previous stage, the emotion and excitement information of every concerned event using metadata-based feature vectors concatenated with audio features. The second level is a bidirectional LSTM capturing the forward-backward temporal dependencies among proposals in order to predict the probability of each proposal to be selected into the summary.

- The *Content refinement* stage exploits the visual information to refine the boundaries of the clips predicted as being part of the summary so that the resulting clips are not anymore restricted to start and/or end of event boundaries. A final bidirectional LSTM network hence predicts which frames among the ones belonging to the pre-selected proposals should

be preserved in the final summary. This last stage allows also to focus on visually salient frames.

4.1 Multimodal Features

TABLE 4.1: Metadata Features Description for our first approach

Name	**Description**
x position	Position where the event occurs, along the wider side of the field
y position	Position where the event occurs, along the shorter side of the field
Time elapsed	The time difference between the current event and the previous one
Type	Type of event
Qualifier	Descriptor of each type of event
Outcome	Event success indicator

For the metadata we use the information provided by Opta. The description of the features extracted from the event data for our first approach is presented in Table 4.1. Event type and qualifiers are categorical features, and the others are real-valued features. Table 4.2 shows the 40 event types and Table 4.3 describes the 34 qualifiers used in this approach.

We use one-hot encoding representations for the categorical features, i.e., for the event type there is a vector of size 40 with all its elements as zero except the one that corresponds to the type of the event, and similarly for the qualifier. The difference between these two features is that an event can be represented by several qualifiers thus, the qualifiers vector might have several non-zero values, in contrast to the event type that has only one non-zero value.

The metadata feature vector $x_{e_f}^M$ is then the concatenation of: one hot-encoding vector for the type of event, one hot-encoding vector for the event qualifier, outcome value, x position, y position, and the time passed from the previous event in seconds.

On the other hand, audio plays a very important role in sports, where crowd cheering and excitement in the commentators' tone are usually indicators of an important action. For this reason, our Summarization stage not only use the event data features but also the energy of the audio. The audio signal is extracted from the broadcast videos of the dataset. We use only the first channel of the audios and a sampling rate of 48000. As proposed by Rui et al.[101], we use sub-band short-time energies. Considering the perceptual property of human ears, we can divide into four sub-bands the critical bands that represent cochlear filters in the human auditory model [120]. These sub-bands are:

- En_1: 0-630Hz
- En_2: 630-1720Hz
- En_3: 1720-4400Hz
- En_4: 4400Hz and above

TABLE 4.2: Overview of the 40 event types used in our first approach.

Event Type	**Description**
Pass	Any pass from one player to another
Offside	Attempted pass made to a player who is in an offside position
Take On	Attempted dribble past an opponent
Foul	Foul is committed resulting in a free-kick
Out	Ball goes out of play for a throw-in or goal kick
Corner	Ball goes out of play for a corner kick
Tackle	Dispossess an opponent of the ball
Interception	Player intercepts any pass event between opposition players and prevents the ball reaching its target
Save	Goalkeeper saves a shot on goal
Punch	Goalkeeper punches the ball
Keeper pick-up	Goalkeeper picks up the ball
Penalty faced	Goalkeeper faces penalty by opposition
Sweeper	Goalkeeper comes off his line and/or out of his box to clear the ball
Cross not claimed	Cross not successfully caught
Smother	Goalkeeper comes out and covers the ball in the box winning possession
Clearance	Player under pressure hits ball clear of the defensive zone or/and out of play
Miss	Any shot on goal which goes wide or over the goal
Post	Whenever the ball hits the frame of the goal
Attempt Saved	A player made a shot, and it was blocked
Goal	All goals
Card	All cards
Substitution	Player comes on as a substitute
Start delay	When there is a stoppage in play such as a player injury
End delay	When the stoppage ends and play resumes
Start	Start of a match period
End	End of a match period
Aerial	Aerial duel when the ball is in the air
Challenge	Player fails to win the ball as an opponent successfully dribbles past them
Ball recovery	When a player takes possession of a loose ball
Dispossessed	Player is successfully tackled and loses possession of the ball
Error	Mistake by player losing the ball
Offside provoked	When an offside decision is given against an attacker
Shield ball	Defender uses his body to shield the ball from an opponent
Foul throw in	A throw-in not taken correctly resulting in the throw being awarded to the opposing team
Penalty faced	Goalkeeper faces a penalty by opposition
Chance missed	A player does not actually make a shot on goal but was in a good position to score and only just missed receiving a pass
Bad touch	A player makes a bad touch on the ball and loses possession
Contentious referee decision	Any major talking point or error made by the referee (including VAR decisions)
Injury time announcement	Injury Time awarded by Referee
Blocked pass	Like interception but player already very close to ball

TABLE 4.3: Overview of the qualifiers used in our first approach. Event type column specifies the event type to which the qualifier is associated.

Event Type	**Qualifier**	**Description**
Pass	Long ball	Long pass over 32 meters
	Cross	A ball played in from wide areas into the box
	Head pass	Pass made with a player's head
	Through ball	Ball played through for player making an attacking run to create a chance on goal
	Free-kick	Any free-kick, direct or indirect
	Corner	Corner
	Player caught outside	Player who was in an offside position when pass was made
	Goal disallowed	Pass led to a goal disallowed for a foul or offside
	Chipped	Pass which was chipped into the air
	Lay off	Pass where player laid the ball into the path of a teammates run
	Launch	Pass played from a player's own half up towards front players. Aimed to hit a zone rather than a specific player
	Flick on	Pass where a player has flicked the ball forward using their head
	Pull back	Player in opposition's penalty box reaches the by-line and passes (cuts) the ball backwards to a teammate
	Switch of play	Any pass which crosses the center zone of the pitch and in length is greater than 60 on the y axis of the pitch
	Assist	The pass was an assist for a shot. The type of shot then dictates whether it was a goal assist or just key pass
	Blocked pass	Like interception but player already very close to ball instead of touch event in past
	Kick off	Starting pass
Miss, Post, Attempt saved, Goal	Penalty	Attempt on goal was a penalty kick
	Own goal	A Goal scored by the player of a conceding team
	Volley	Shot on the volley (ball doesn't bounce before the shot)
	Strong	Shot was subjectively classed as strong
	Weak	Shot was subjectively classed as weak
	Swerve	Shot which swerves to the left/right - from attackers perspective
	Deflection	Shot deflected off another player
	Hit woodwork	Any shot which hits the post or crossbar
	Big change	Shot was deemed by Opta analysts an excellent opportunity to score
	Individual play	Player created the chance to shoot by himself, not assisted
	Second assisted	Indicates that this shot had a significant pass to create the opportunity for the pass which led to a goal
	Own shot blocked	Player blocks an attacking shot unintentionally from their teammate
Foul	Yellow Card	Player shown a yellow card
	Second yellow	Player receives a 2nd yellow card which automatically results in a red card
	Red card	Player shown a straight red card

For an audio signal A with a sampling rate d, the short-time energy for each sub-band l at any second s is defined as

$$En_l^s = \frac{1}{2 * s * d} \sum_{j=s*d}^{3*s*d} A(j) \tag{4.1}$$

Since each event has a timestamp corresponding to the video time when the event occurs, the energy of the event e_f is the energy of the second s corresponding to its timestamp. We set the audio features into a vector $x_{e_f}^A$ concatenating En^s, En_1^s, En_2^s, En_3^s, and En_4^s.

For the video features, each frame is represented by the output of the penultimate layer (pool 5) of GoogleNet [121], which is a 1024-dimension feature vector

4.2 Proposals

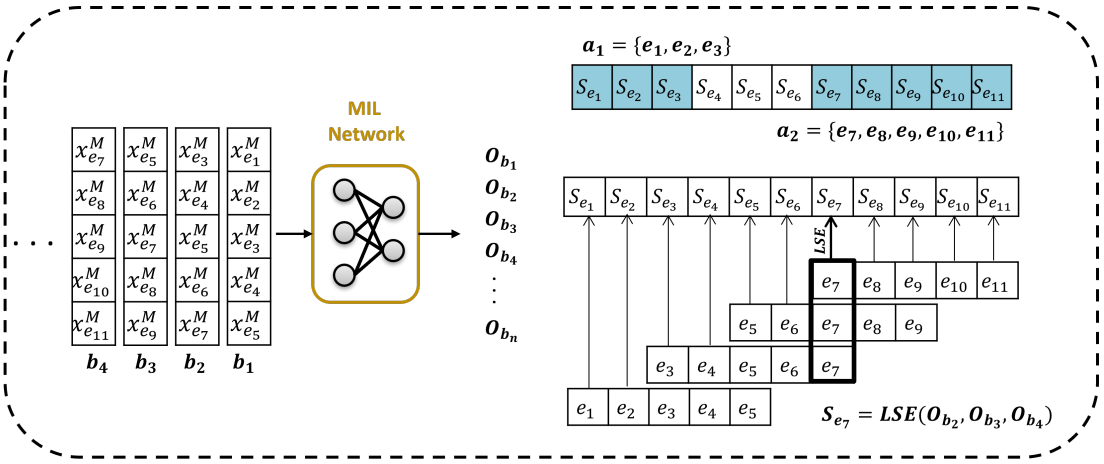
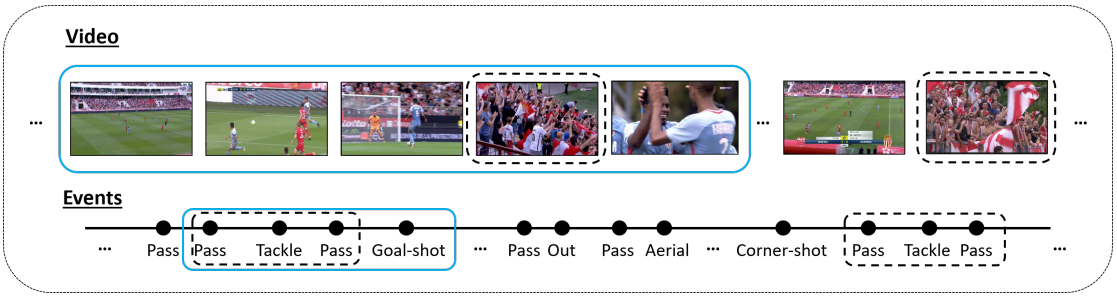


FIGURE 4.2: Proposals stage. The input of this stage are bags of events, in this specific example the bag size is 5. Then a MIL network outputs a score O_{b_n} per bag. In order to get a value per event S_{e_f}, LSE function is applied on all the scores of the event. Finally, the consecutive positive events are grouped to form action proposals. The positive events are in blue.

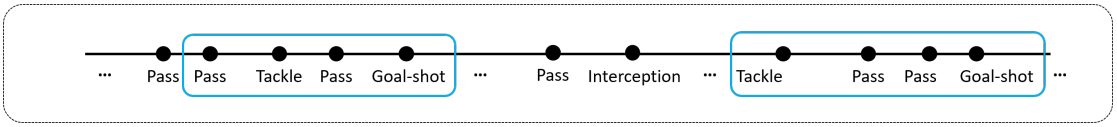
The goal of the first block of our method is to identify the action proposals of the match, that is to say consecutive relevant events. For instance, an action of a *goal* might be corresponding to the sequence {*pass, interception, pass, goal*}. Such groups of events are considered as proposals if they are parts of the match that might belong to the summary.

The idea of extracting pieces of the input as proposals to then in a second stage decide which of these proposals are indeed classified as positive has been widely used in object detection [59, 122, 123, 124, 125] and action detection [3, 126, 61, 60, 127]. The main goal of the proposals extraction is to filter as much as possible the relevant and non-relevant information by identifying the negative parts of the sample (i.e., background in the case of object detection and non-action in the case of action detection), in order to be discarded for the following classification stage.

However, in sport matches there is a high similarity between positive (i.e. action) and negative (i.e. not action) samples. In the case of soccer, for instance the sequence {*pass, tackle, pass*} can



(A) Inter-categorical Similarity



(B) Intra-categorical Diversity

FIGURE 4.3: Examples of inter-categorical similarity and intra-categorical diversity. (a) Example of inter-categorical similarity of actions. In the bottom part there are video frames that represent clips of video. The black dots are the events of the match. Blue line represents a goal action. The dashed line indicates that similar parts of the match can be both inside an action and outside an action. (b) Example of intra-categorical diversity of actions. Blue line represents a goal action. Two goal actions might be formed by different sequences of events.

be the beginning of a goal action but the same sequence can belong to some section of the match where nothing relevant is happening (see Figure 4.3a). This inter-categorical similarity is not the only issue we have to face, since simultaneously our system should be able to deal with a high intra-category diversity where two instances of the same action can only partially match when considering their event sequences (see Figure 4.3b). For these reasons, we believe that in the context of soccer matches, a MIL approach is more suitable than a traditional learning method (See Section 2.2.2 for the description of MIL).

For the Proposals stage, we use the MI-Net architecture proposed by Wang et al [4]. This network consists of three fully connected layers followed by one MIL Pooling layer, where the latter aggregates all instance features in order to learn a bag representation. This network receives bags of instances as inputs, and outputs a score per bag (See Figure 4.4).

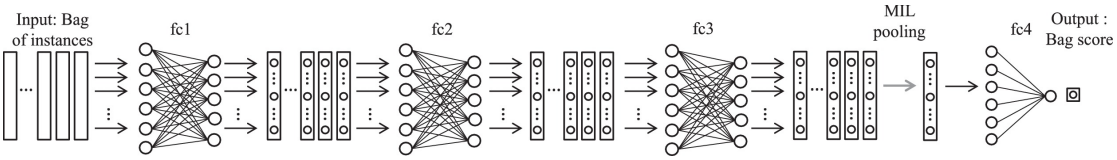


FIGURE 4.4: A MI-Net with three fully-connected layers and one MIL pooling layer. Image from [4]

We represent a match as a sequence of events $E = \{e_1, e_2, ..., e_F\}$. These events are atomic soccer actions like pass, tackle, out, interception, post, throw-in, head, etc. If there are three consecutive passes in the game, you will have three similar events *pass* in a row.

$X = \{x^M_{e_1}, x^M_{e_2}, ..., x^M_{e_F}\}$ represents the set of instances, where $x^M_{e_f}$ is the metadata feature vector characterizing the f-th event of the match. We denote the set of bags by $B = \{b_1, b_2, ..., b_n\}$, where a bag refers to a consecutive subset of instances from X. The difference between action proposal and bag is that an action proposal is a set of consecutive events that might belong to the summary, while a bag is a set of any consecutive events.

It is important to notice that we do not have access to the ground-truth actions, then the bags are created in a class-agnostic way. We use a sliding window across the whole match events, with a stride such that there is an overlap between two consecutive windows (Figure 4.2 shows an example of sliding window of size 5 and stride 2). The overlap on the sliding window leads some events to belong to more than one bag and thus obtaining several scores. In order to define a score per event instead of per bag, we merge all the possible scores associated to the given event.

We denote O_{b_n} as the MIL Network output, i.e. the score for the bag. Since an event might belong to several bags owing to the sliding window overlap, we need a method to obtain the accumulated score per event S_{e_f} integrating all the predictions of the bags this event belongs to (Black bold rectangle in Figure 4.2 shows that event $e7$ belongs to three different bags b_2, b_3, b_4). Wang et al. [4] evaluated Log-Sum-Exp, Max and Min functions as pooling methods to define the score per bag. We have empirically compared these methods to obtain the score per event and we have found that Log-Sum-Exp function, given in Equation (4.2), provides the best results.

$$S_{e_f} = r^{-1} \cdot log\left[\frac{1}{|\{b_n \mid x_{e_f} \in b_n\}|} \sum_{b_n | x_{e_f} \in b_n} r \cdot O_{b_n}\right] \tag{4.2}$$

Once we obtain S_{e_f} for each event, we use a threshold T_{ps} to select the positive events. Then we group the consecutive positive events into proposals $A = \{a_1, a_2, ..., a_P\}$, where $a_p = \{e_f \mid S_{e_f} >= T_{ps}\}$. Thus, proposals are most of the time related to what is commonly called actions in sport commentaries.

We manually decide the end of an action only in one specific case, when one of the events inside the action is a goal-shot. For instance, we found that $\{e_7, e_8, e_9, e_{10}, e_{11}\}$ is a set of positive consecutive events and both e_6 and e_{12} are negative events. If one of these events let us say e_9, is goal-shot, then we would define two different actions, $\{e_7, e_8, e_9\}$ and $\{e_{10}, e_{10}\}$.

4.3 Multimodal Hierarchical LSTM

Hierarchical LSTM has shown to be very efficient for video summarization since it helps to model longer dependencies than traditional LSTM [128, 6, 129]. However, previous works use the entire video as input of the first level of the hierarchy but for much shorter videos. Instead, because of the size of each video we consider here, we propose to take as input only the relevant parts of the match extracted from our previous Proposals stage.

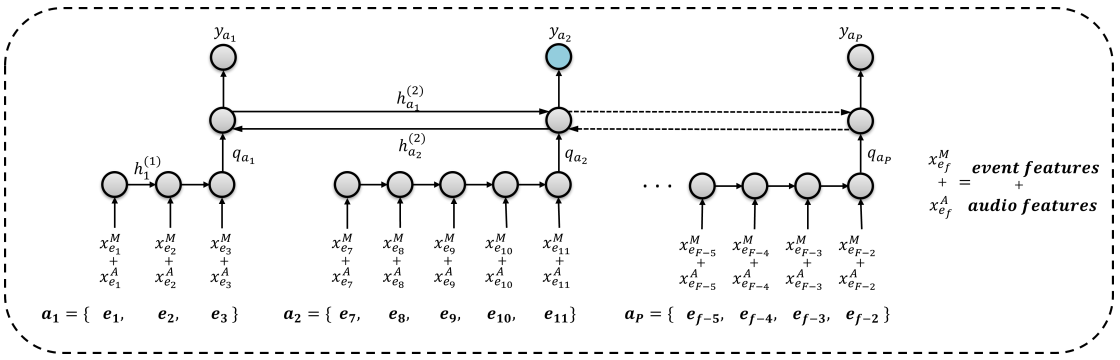


FIGURE 4.5: Summarization stage. It is composed by a hierarchical LSTM: bottom LSTM layer creates a representation of each proposal q_{a_p} and an upper bidirectional LSTM decides whether the proposal a_p is part of the summary. The $+$ symbol represents concatenation.

As it is depicted in Figure 4.5, the Summarization stage is a two-level Hierarchical LSTM. The first level creates a representation of each proposal by accumulating in each proposal the emotion and excitement information of every concerned event; and the second level captures the forward-backward temporal dependencies among proposals to predict the likelihood of each proposal to be part of the summary.

For the Summarization stage, a multimodal instance is then represented by the concatenated feature vector $\{x_{e_f}^M + x_{e_f}^A\}$. Where $x_{e_f}^M$ is the metadata feature vector and $x_{e_f}^A$ is the audio feature vector. This multimodal instance is the input of the first level. Assuming e_f belongs to the proposal a_p, the hidden state of this level's LSTM unit is $h_{e_f}^{(1)}$, encoding all events in the proposal a_p up to the $f-th$ event by computing over the current feature vector $\{x_{e_f}^M + x_{e_f}^A\}$ and the previous hidden state $h_{e_{f-1}}^{(1)}$.

After processing an entire proposal, we denote the final hidden state of the LSTM unit as q_{a_p}, the encoding vector for the proposal a_p. The LSTM unit memory and the initial hidden state are then reset to zero.

After all the proposals are processed, we end up with a sequence of encodings $Q = \{q_{a_1}, q_{a_2}, ..., q_{a_P}\}$. We construct a bidirectional-LSTM over Q which grasps the temporal dependencies between proposals in the summary.

Indeed, in a game with a largely unbalanced score (8 to 1 for instance) the summary may not present all the goals; or in a game with not much action, with many shots-not-on-target, the summary may contain only some of them maybe the first ones or maybe some evenly distributed with respect to other actions present in the summary. These choices are directly related to the storyline of the summary that we target to learn with this bidirectional-LSTM.

The output of this second level is denoted as $Y = \{y_{a_1}, y_{a_2}, ..., y_{a_P}\}$, where y_{a_p} indicates the likelihood of whether the proposal a_p should be included in the summary.

We use a threshold T_{ss} to select which are the proposals that comprise the predicted summary. We denote this summary by $Summ = \{a_p \mid y_{a_p} >= T_{ss}\}$

4.4 Content Refinement

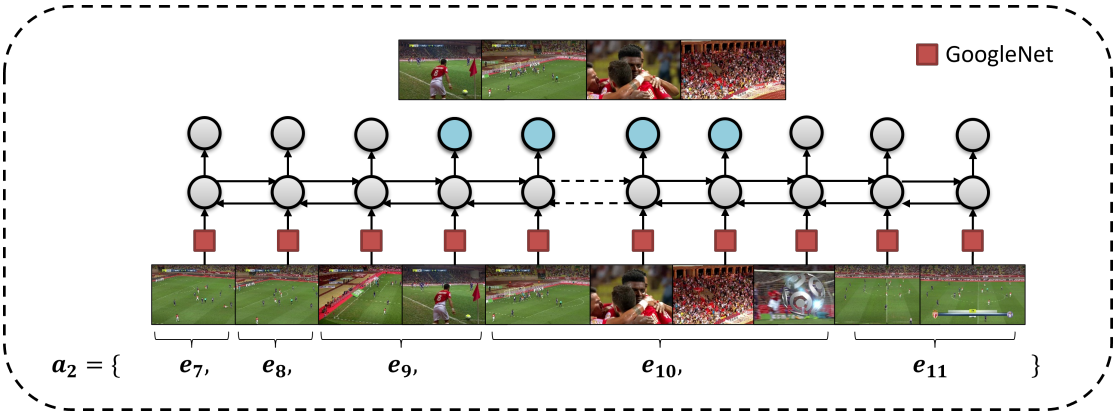


FIGURE 4.6: Content Refinement stage. It takes the frame features of the proposals predicted as positive by the Summarization stage and then, using a bidirecional-LSTM, it decides which of these frames indeed belong to the summary. The input of this stage are GoogleNet features.

Representing a match as a sequence of events significantly reduces the amount of information to process compared to analyze the content at frame level. However as mentioned before, the events are occurring on the field since they are acquired by a person watching the game in the stadium and not behind a TV screen. Hence, most of the times the boundaries of the clips (on TV) are not completely aligned with on-field events, since clips boundaries are decided by the producer who might cut the content in the middle of an event. Thus, the final clips cannot be restricted to start and/or end of event boundaries.

For this reason, we decided to exploit the visual content broadcast on TV by training a final bidirectional-LSTM to decide which frames of the selected events through $Summ$ really belong to the final summary (see Figure 4.6). This last stage introduces visual features in the process and thus accounts for new features to capture possible interestingness and representativeness within the proposal.

Let us define $G = \{g_1, g_2, ..., g_n\}$, where each g_n is a positive proposal extracted from the Summarization stage. The beginning and the end of each g_n is given by the timestamp of the first and last events of g_n. We represent the frame corresponding to this beginning and end events by $R_{g_n}^{beg}$ and $R_{g_n}^{end}$ respectively.

Each input sample of the LSTM corresponds to the feature vectors of the frames inside the interval $[R_{g_n}^{beg}, R_{g_n}^{end}]$. And for each of these frames there is an output, which corresponds to the likelihood of this frame to be part of the final summary.

4.5 Experiments

We first introduce the experimental setting, describing the dataset, features and metrics. We then present the quantitative results to demonstrate the advantages of the proposed approach over

a frames-based method and the comparison of our model with and without audio features. We further perform a qualitative comparison.

4.5.1 Setup

Summary-based Dataset. Our dataset consists of 20 complete soccer games from 2017-2018 season of French Ligue 1. The ground truth video summaries were made by professional editors of a sports broadcast company. The matches were played at different times of the day, in multiple fields and with 19 different teams. We have manually detected the corresponding temporal intervals of each summary clip in the corresponding original matches. We create 5 folds where each fold has 16 games for training and 4 games for testing.

The first step to create the dataset for the Proposal stage is to detect the action proposals. Based on the assumption that each clip inside a summary follows a logical time sequence decided by a human editor where a main event (e.g., goal-shot, card, free-kick) is shown along with its context (e.g. the foul that led to the card), we need a specific dataset where summary actions are present with their context. As we previously mentioned, to the extent of our knowledge, there are no publicly available datasets for soccer video summarization. We therefore create our own. The details of the process is described in Section 3.3.

The second step is to create the bags, we slide a window of 5-events size and stride 2. The bag is considered as positive if at least 3 of the 5 events belong to a ground truth action proposal. It is important to notice that to find the ground truth proposals of the training set, we only look for event sequence patterns present inside this set (not in the test set).

For the Summarization stage, a proposal a_p is considered as part of the summary if at least one event of the proposal is overlapped with a clip of the ground truth video summary.

Networks specifications. The Multiple Instance Learning network has 256, 128 and 64 neurons in its three fully connected layers. Each of the two LSTM layers of the Summarization stage have 128 units. The LSTM of the Content Refinement stage has 256 units. To train our model, we adopt a stage-wise routine, using Adam optimizer and binary cross-entropy as loss function.

Evaluation. As in previous works on video summarization [128, 76, 130], we evaluate our generated summary U computing the Precision, Recall and F-score against V, the summary created by the editors:

$$\begin{aligned} Precision &= \frac{\text{overlapped duration of } U \text{ and } V}{\text{duration of } U} \\ Recall &= \frac{\text{overlapped duration of } U \text{ and } V}{\text{duration of } V} \\ F_{score} &= 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \end{aligned} \tag{4.3}$$

4.5.2 State-of-the-art Summarization Methods

One challenge in our context lies in comparing our approach with the state-of-the-art, relevant datasets are under copyright infringement and as far as we know no other method considers a full soccer game as a single sample.

As mentioned in Section 2, other methods are not suitable for comparison. Methods [76, 5, 130, 128, 80, 81] optimize summary diversity which is not convenient for soccer videos since a summary could contain several similar actions; methods in [131, 97, 132, 96, 10, 106, 99] use different input multimedia data (text or comments from social networks) not easily reachable.

vsLSTM [5] and H-RNN [6] appeared to be some of the few summarization methods for general-purpose videos that we have found where input samples are full videos, optimization does not rely on diversity and input data are similar to ours. Therefore, we trained these two methods from scratch using the frame features extracted from our video dataset.

In addition to the original models, we define additional ones that take the same ideas of H-RNN and vsLSTM but modified to be compliant with the inputs of our approach, they are described in the following subsections with the prefix *event-*. In other words, we keep the same architectures proposed in the original papers, but we perform some modifications for the model to use our event and audio features as input.

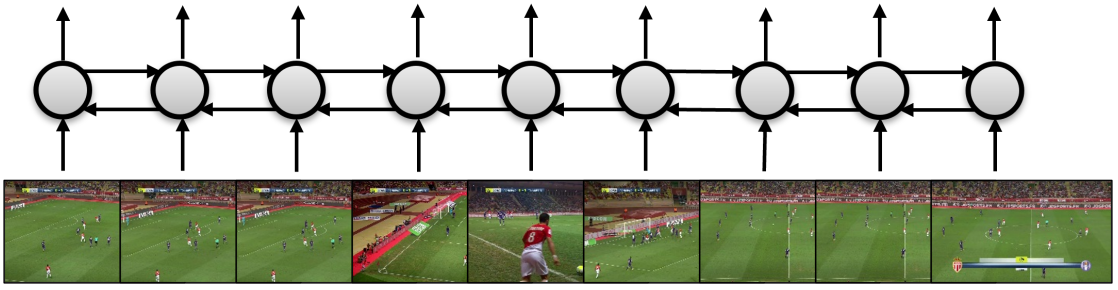


FIGURE 4.7: vsLSTM schema. The model is composed of a bidirectional LSTM that predicts at each time step.

The schema of vsLSTM is shown in Figure 4.7. The model is a bidirectional-LSTM followed by a multi-layer perceptron that makes a prediction per frame. The inputs are frame features extracted from GoogleNet.

event-vsLSTM. It is a vsLSTM architecture but instead of using frames features, it takes the same input as our approach, either $x_{e_f}^M$ or $\{x_{e_f}^M + x_{e_f}^A\}$ (in that last case we precise "with audio" in the experiments).

The schema of H-RNN is shown in Figure 4.8. The first layer is an LSTM, which processes video subshots generated by cutting the whole video into fixed-size segments. Then the representation of each subshot is the input to a second layer. Specifically, the second layer is a bidirectional LSTM, which exploits the inter-subshot temporal dependency and determines whether a certain subshot is valuable to be in the summary.

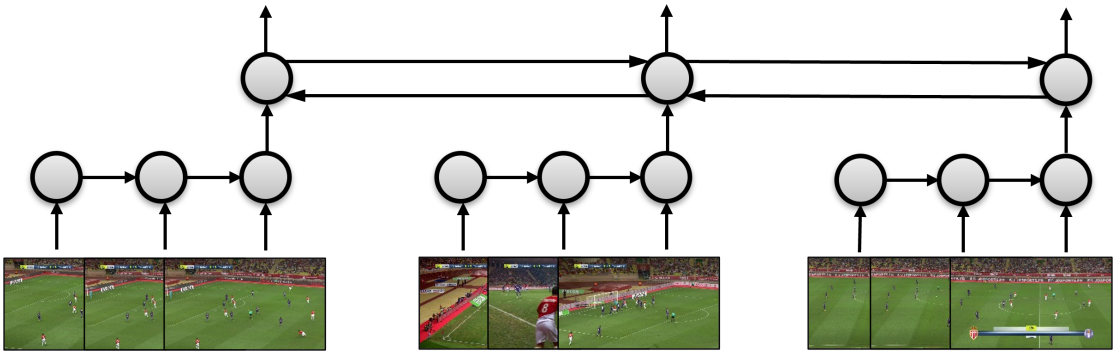


FIGURE 4.8: H-RNN schema. It contains two layers, where the first layer is an LSTM and the second layer is a bi-directional LSTM

The input of this model are frame features extracted from GoogleNet. We set the segment size to 40, as recommended by the authors.

event-H-RNN. As the original H-RNN we use fixed-size segments but instead of using frame features as inputs, this new model takes the same input as our approach, either $x_{e_f}^M$ or $\{x_{e_f}^M + x_{e_f}^A\}$ (in that last case we precise "with audio" in the experiments). Although the authors of H-RNN advise to use 40 as segment size, we have empirically found that for our event-based approach is better to choose a significantly smaller size, most likely due to the fact the average number of events on a ground truth clip is 7. For this reason and to perform a fair comparison, we have used as segment size the bag size used in our Proposals stage.

4.5.3 Performance Results

TABLE 4.4: Multimodal Performance Comparison of our fully automatic video summarization system. All the models were trained with event and audio features.

Method	Precision	Recall	F-score
event-vsLSTM (with audio)	0.414	0.389	0.384
event-H-RNN (with audio)	0.257	0.594	0.355
Ours (with audio)	0.470	0.457	0.459

All the scores reported in this section correspond to the results of the 20 games of our dataset, we gather the test sets results of the 5 folds.

Comparison in a multimodal context. In Table 4.4, we compare our approach with event-H-RNN and event-vsLSTM, since they are our multimodal models. To obtain these results, the event models were trained with the concatenation of the events and audio features $\{x_{e_f}^M + x_{e_f}^A\}$ as input, and our approach was trained as explained in the previous section. The input for the Proposals stage is the event feature $x_{e_f}^M$, for the Summarization stage it is$\{x_{e_f}^M + x_{e_f}^A\}$ and for the Content Refinement stage it is the frame features. Event-H-RNN shows the highest Recall, but the Precision is the lowest, which means this method has issues to identify the events that do not

belong to the summary. Our approach clearly outperforms in terms of F-score and Precision, it also shows a good trade-off between Precision and Recall.

The results of the state of the art show that H-RNN performs better than vsLSTM [128, 6, 129], however with our data event-vsLSTM obtains better F-score and Precision than event-H-RNN. Probably the fixed size of the segment and the overlap to decide if the segment is positive, have to be carefully analyzed. We have tried different segment sizes and overlap ratio, and report the best results.

TABLE 4.5: Performance comparison with frames based models. H-RNN [6] and vsLSTM [5] were trained with frames features, as it was proposed originally in the papers. Our approach was trained without audio features.

Method	**Precision**	**Recall**	**F-score**
vsLSTM	0.553	0.241	0.296
H-RNN	0.406	0.295	0.335
Ours	0.436	0.401	0.415

Comparison with frame-based models. To verify that our approach is better than frame-based methods, the results of H-RNN and vsLSTM are provided in Table 4.5. To be fair, we make the comparison with our approach that was trained without audio features. Although vsLSTM has higher Precision, its Recall is the lowest. One possible interpretation for this case (that we also checked visually on the resulting summaries) is that the method only learned to correctly predict the most common actions like shots on target. The Recall and F-score of our method are the highest, and it is the method with the best trade-off between Precision and Recall. This clearly shows that even without the audio features our event-based method can extract the most accurate summaries.

TABLE 4.6: Comparison on undetected parts of ground-truth summary. Missing clips represent the percentage of clips which were completely missed. And False Negatives represent the percentage of all seconds that were not detected.

Method	**Missing Clips**	**False Negatives**	**Recall**
vsLSTM	0.509	0.759	0.241
H-RNN	0.548	0.705	0.295
event-vsLSTM (with audio)	0.364	0.611	0.389
event-H-RNN (with audio)	0.398	0.406	0.594
Summarization stage (with audio)	0.267	0.355	0.644

Focus on missing clip and false negative rates. Since the last stage of our approach is only in charge of the refinement of clips predicted as summary, it is very important that the Summarization stage misses the least number of clips belonging to the summary. The column *Missing clips* of Table 4.6 represents the ratio between the number of clips that where not detected

at all and the total number of clips in the summary. Our method has the lowest missing clips ratio, with almost 10% less than the second best on this column. Table 4.6 also reports the Recall and false negatives, where the latter is the ratio between the seconds that were not detected and the total duration of the ground truth summary. Our approach gets the lowest ratio of false negatives and highest recall. All these results prove that our combination of Multiple Instance Learning and hierarchical LSTM is the best choice because no matter how good our Content Refinement stage is, if we replace our first two stages by any of the state-of-the-art models, these models will always provide less and possibly shorter positive proposals.

TABLE 4.7: Performance comparison for models with and without audio features.

Method	**Precision**	**Recall**	**F-score**
event-vsLSTM	0.435	0.351	0.381
event-H-RNN	0.249	0.567	0.337
Ours	0.436	0.401	0.415
event-vsLSTM (with audio)	0.414	0.389	0.384
event-H-RNN (with audio)	0.257	0.594	0.355
Ours (with audio)	0.470	0.457	0.459

Impact of audio features. It is worth mentioning that audio features play an essential role to detect important actions. If we compare the results between the models trained with audio and the ones trained without audio (shown in Table 4.7), we can see that adding the audio features usually improves the scores, especially the Recall. In addition, our approach performs the best in terms of F-score, even without these additional features.

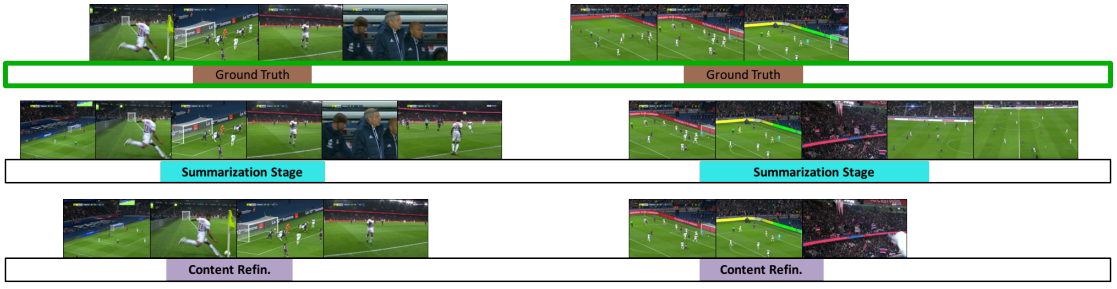


FIGURE 4.9: Video prediction example of our method. Pictures on the top are sampled from the ground-truth summary, the ones in the middle are from the Summarization stage and in the bottom are from the final summary prediction of our method. The color bars below the images represent time intervals and the green rectangle represents the ground-truth.

4.5.4 Qualitative Results

We illustrate an example of our video summarization results in Figure 4.9 in order to show how important is the Proposals and Summarization stages to obtain good results in our Content

Refinement stage. The rectangles below the images represent the timeline across the match, the inner colored rectangles are time intervals and the images above the rectangles are frames sampled from the match video. The top picture with the green timeline shows two different examples of ground truth summary intervals. The middle and bottom picture depict the Summarization and Content Refinement predictions that are closest to the ground truth intervals.

The example on the left side shows a good behavior of our model, where the Proposals stage detects all the events surrounding a ground truth interval, the Summarization stage classifies as positive this proposal and finally the Content Refinement stage can detect the most relevant part of the clip. The example on the right side of the figure presents a case where the first stage of our method creates a proposal that misses the beginning of the ground truth interval; however, the last stage is able to predict that the beginning of the proposal is relevant for the summary.

The sampled frames of Figure 4.9 are also important to show that the borders of the ground truth intervals might be subjective. On the left-side example, the beginning of the clip is when the player kicks the ball from the corner and the end is the coach reaction, our prediction starts before when the camera shows a wide angle of the corner shot and finishes when the player approaches for a throw-in. One could argue that both the predicted and the true borders are valid. A similar situation occurs at the end of the right-side example, the ground truth clip ends on the team celebration and the prediction ends the clip on the audience celebration.

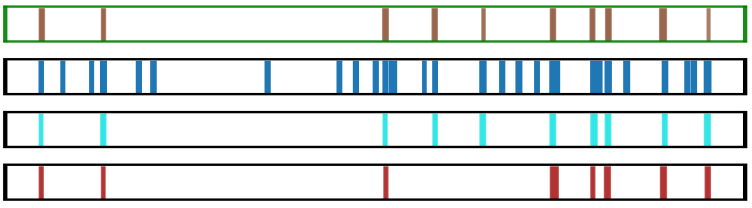


FIGURE 4.10: Comparison of intervals prediction of one entire match. The topmost row shows the ground-truth intervals. Results of the Proposals and Summarization stage are the second and third rows respectively. The bottom row shows the intervals prediction of event-H-RNN model.

Figure 4.10 depicts the prediction intervals for an entire match, where the top picture with the green rectangle represents the ground truth intervals. As shown on the second row (from top to bottom) the Proposals stage can detect all the intervals from the real summary. And the third row demonstrates that although the Proposals stage outputs multiple false positives, the Summarization stage can detect which proposals indeed belong to the summary. The bottom picture represents the prediction intervals of event-H-RNN model, where we can visually confirm the high rate of missing clips and false negatives, shown in previous section.

Another important property of our model that is worth emphasizing is the fact that the Multiple Instance Learning Approach significantly helps to extract meaningful proposals. There is a particular interval that corresponds to a substitution, which is a very uncommon action in our summaries. Our Proposals stage is able to capture this event even when there is no substitution interval in the training set.

4.6 Discussion

Even though we showed that the use of audio features improves the summarization results, there are still opened questions in this topic. Previous works have explored additional features (such as MFCC, Zero Crossing Rate, Spectral centroid, among others), in order to extract important information in sport audio signals [133, 134, 135, 136]. Also, there are other methods to combine different modalities besides the concatenation of features, which were not explored in this chapter.

It is important to mention that we had to find, download, and process our own videos since there are not available datasets for sports summarization. We decided to create a dataset as realistic as possible, then the summaries are videos created by professional broadcasters. However, dealing with real-life data involves several challenges. The main limitation is that we do not have the time location of each summary clip into the original match video. Therefore, we need to manually analyze each video summary to first, extract each clip and then find the exact corresponding time in the original match video. In addition to the time-consuming task of generating the time intervals for the video summarization, we need to deal with the processing of the event data. The event data are stored in Extensible Markup Language (XML) files that are not related with the video. For instance, the timestamp of each event is relative to the beginning of the half (first half or second half of the match) then we must carefully identify in the match video the exact time where each half starts.

4.7 Conclusion

In this chapter we presented an approach for automatic generation of summaries from soccer videos based on multimodal features, including audio energy, event features from sports analytics and visual information from video frames. The proposed approach consists of three consecutive stages: a Proposals stage deals with the similarity between the events inside the summary and the rest of the match, a Summarization stage accumulates the emotion and excitement information of each proposal to capture the temporal dependencies in order to decide which proposals are part of the summary and, finally a Content Refinement stage exploits the visual information to predict which frames among the ones belonging to the pre-selected proposals should be preserved in the final summary. Our model outperforms by an 8% margin not only the video processing state-of-the-art methods but also methods that use event and audio features. There are several key contribution factors: the capacity of Multiple Instance Learning to deal with similar inter-categorical actions, our idea to complement hierarchical LSTMs' strength with the generation of proposals and, the use of audio features to improve the detection of important events. While other methods propose to use frames, we have demonstrated that to base the summarization on events get a shorter and better representation of longer videos as soccer matches.

Chapter 5

Semi-automatic summaries generator

In the previous chapter we described the first solution we proposed, a fully automatic method for the summarization of soccer matches. From this solution we had important outcomes like the relevance of multimodal features, splitting the problem in different stages, MIL for inter-categorical similarity and intra-categorical diversity, and the use of event data instead of video frames.

Our previous solution outperforms state-of-the-art methods and does not require human intervention to generate a video summary. However, it provides a unique video summary per match which is not a perfect solution in a real-life scenario where editors prefer to have several options to choose according to personal predilections, platform limitations, user criteria or fan preferences. In this chapter we describe our second solution which is a semi-automatic approach aiming to tackle two relevant issues in the field of video summarization of soccer matches, subjectivity, and length constraint.

Our Semi-automatic summaries generator is also composed of three stages (see Figure 5.1).

- The *Generation of Action Proposals* stage gets as input the event data of the match and uses Multiple Instance Learning for sequential data to detect all the action proposals of the match.
- The *Multimodal Summarization* stage takes the event data and audio features of the action proposals and using a hierarchical multimodal attention model it decides which of these action proposals indeed belong to the summary.
- The *Multiple Summaries Generation* stage uses a ranking distribution in order to provide to the editor several summary options of the same match

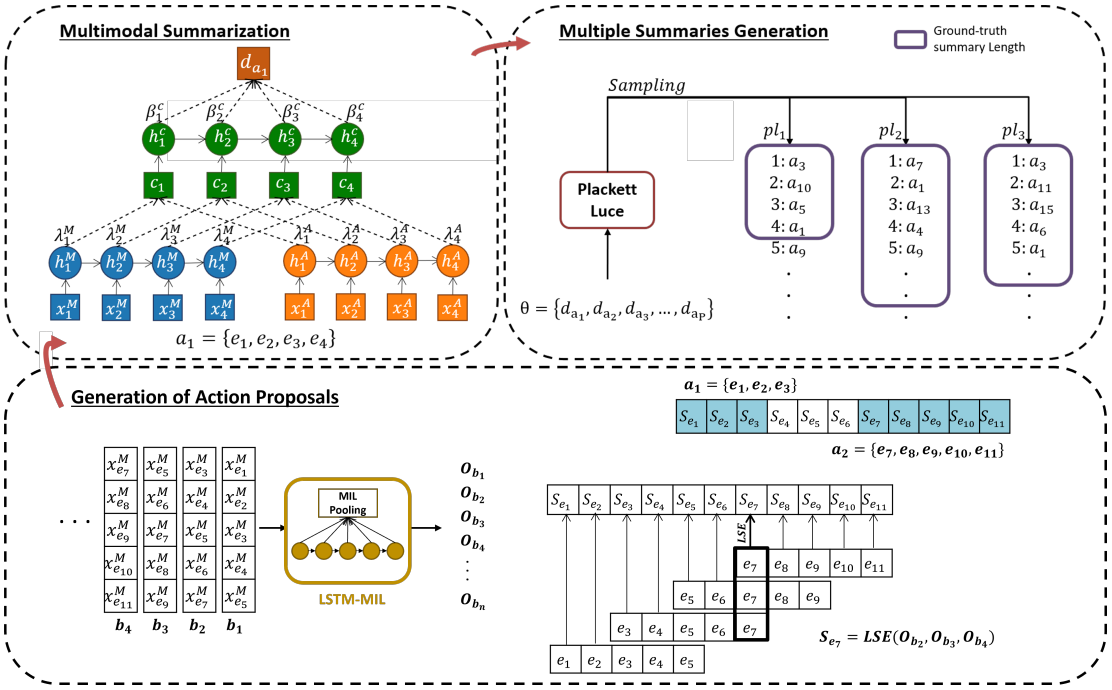


FIGURE 5.1: Our semi-automatic summaries generator. The *Generation of Action Proposals* stage takes as input events e_f grouped into bags b_n and outputs action proposals. Each bag b_n is processed by an LSTM Multiple Instance Learning (MIL) network, in test phase Log-Sum-Exp (LSE) function is applied on all the predicted values of each event to finally group the consecutive positive events into action proposals. The *Multimodal Summarization* stage takes as input the action proposals a_p and outputs the likelihood of each of the actions to be part of the summary. This stage has a hierarchical multimodal attention (HMA): bottom LSTM layer learns the importance of each modality at the event level and in the upper stage extracts the importance of each event inside each action a_p. The *Multiple Summaries Generation* stage takes as input the importance of each action θ_p as parameters for a Plackett- Luce distribution to generate multiple summaries of the same length as the ground-truth summary.

5.1 Multimodal Features

Similarly to our fully automatic video summarization system, we use the audio signal extracted from the broadcast videos of the dataset, it has a sampling rate of 48000. We use only the first channel of the audios. But unlike our first solution where only the audio energy was exploited, we also extract other features such as entropy, zero crossing rate, MFCC and some characteristics of the spectrum. Since each event e_f has a timestamp corresponding to a time in the match, we extract its corresponding video time $time_{e_f}$ in seconds. As our goal is to capture the sound reactions (from spectators, coaches or commentators) just after the event takes place, the audio features of the event e_f are extracted from the interval $[time_{e_f}, time_{e_f} + 2s]$. Inside this interval, we follow the frame-based feature extraction approach [137], the audio signal is first divided into short-term windows (frames) of 100 ms with 50% overlap, we then compute the audio features described in Table 5.1 for each frame to finally get the mean value across frames for each of the audio features.

The detailed description of the metadata extracted from the event data is presented in Table 5.2.

TABLE 5.1: Audio Features Description for our semi-automatic summaries generator.

Name	**Description**
Zero Crossing Rate	The rate of sign-changes of the signal during a particular frame
Energy	The sum of squares of the signal values, normalized by the respective frame length
Entropy of energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes
Spectral centroid	The center of gravity of the spectrum
Spectral spread	The second central moment of the spectrum
Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames
Spectral flux	The squared difference between the normalized magnitudes of the spectra of two successive frames
Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated
MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale

TABLE 5.2: Metadata Features Description for our semi-automatic summaries generator.

Name	**Description**
Start Location	(x, y) position of the field where the event started
End Location	(x, y) position of the field where the event ended
Time elapsed	Time difference between the current event and the previous one
Start Distance to the goal	Distance to the goal for the event's start location
End Distance to the goal	Distance to the goal for the event's end location
Start Angle to the goal	Angle to the goal for the event's start location
End Angle to the goal	Angle to the goal for the event's end location
Type	Type of event
Qualifier	Descriptor of each type of event
Outcome	Event success indicator

For the categorical features (i.e., type and qualifier) we use a target encoding representation instead of the one-hot encoding used in our fully automatic video summarization system. Target encoding creates a numerical representation of the categorical values. The numerical representation corresponds to the posterior probability of the target (class), conditioned by the value of the categorical attribute. In a binary classification, the probability is computed as the ratio between the number of samples with positive class and the total number of samples [138].

Another important modification related to the event data is the selection of event types and qualifiers. In our fully automatic video summarization system, we over described the events of the match; there were types and qualifiers that were present only few times in the entire dataset. Therefore, we decided to group some event types and choose the relevant qualifiers. For instance,

TABLE 5.3: Overview of the 19 event types used in our second approach.

Event Type	**Description**
Pass	Any pass from one player to another
Cross	A ball played from the offensive flanks aimed towards a teammate in the area in front of the opponent's goal
Throw-in	Throw-in
Free-kick	Direct or Indirect Free-kick
Corner	Corner kick
Take On	Attempted dribble past an opponent
Foul	Foul is committed resulting in a free kick
Tackle	Dispossess an opponent of the ball
Interception	Player intercepts any pass event between opposition players and prevents the ball reaching its target
Shot	An attempt towards the opposition's goal with the intention of scoring
Save	A shot on target faced by the goalkeeper, even if a goalkeeper does not touch the ball and the ball is going into the net or saved by a defender
Clearance	When the player, while having other option, to pass or to hold the ball, is instead clearing it, either with a long pass forward without a precise target or for a throw in/corner kick, playing safe
Dribble	An attempt to move past an opposing player whilst trying to maintain possession of the ball
Offside	Attempted pass made to a player who is in an offside position
Bad touch	A player makes a bad touch on the ball and loses possession
Referee decision	Video assistant referee (VAR) decisions
Goal	All goals
Start	Start of a match period
End	End of a match period

instead of having three different event types (*Miss*, *Post* and *Attempt Saved*) for a shot on target, we define a single event type *Save*. We also remove some qualifiers that were not very common neither belong to any summary. The description of the event types and qualifiers used in our semi-automatic summaries generator is available in Tables 5.3 and 5.4 respectively.

The metadata feature vector $x_{e_f}^M$ is then the concatenation of 10 numerical values (one value for each of the 10 features described in Table 5.2).

5.2 Multiple Instance Learning for Sequential Data

MIL paradigm assumes neither ordering nor dependency of instances within a bag. However, that does not apply in our problem since the selection of an action to be part of a summary is highly dependent on the sequence of its events. For instance, a penalty or a free-kick are always preceded by a foul. More importantly, the permutation of events could completely change the meaning or the interest of an action.

TABLE 5.4: Overview of the qualifiers used in our second approach. Event type column specifies the event type to which the qualifier is associated.

Event Type	**Qualifier**	**Description**
Pass	Key pass	The final pass or pass-cum-shot leading to the recipient of the ball having an attempt at goal without scoring
	Assist	The final touch (pass, pass-cum-shot or any other touch) leading to the recipient of the ball scoring a goal
	Blocked Pass	A player tries to cut out an opposition pass by any means
	Through ball	A pass played into the space behind the defensive line for a teammate to contest
	Fair Play	A clearance of the ball when a player needs medical treatment or the pass when the ball is being returned to the opponent team after being cleared out in the spirit of fair play
Free-kick	Direct	Direct free-kick
	Indirect	Indirect free-kick
Goal, Shot	Own Goal	A Goal scored by the player of a conceding team
Foul	Yellow Card	Player shown a yellow card
	Red Card	Player shown a red card
Shot	Opportunity	A clear chance of scoring a goal

For this reason, we argue that fully-connected layers as proposed by previous works are not completely suitable to capture this sequentiality. Recurrent neural networks are better suited to model such dependency. At the core of the LSTMs are memory cells which encode, at every time step, the knowledge of the inputs that have been observed up to that step. Therefore, unlike the fully automatic video summarization system described in the previous chapter, we propose our own MIL approach, an LSTM network followed by a MIL Pooling to get the bag representation. Our LSTM MIL Pooling method is like the one proposed by Janakiraman et al. [139] but instead of using the MIL aggregation at the prediction level, our method performs the aggregation at the feature level.

Like it was done in our fully automatic video summarization system, a bag is a sequence of event feature vectors. But now this sequence is the input of an LSTM with hidden state defined by:

$$h_t = LSTM(h_{t-1}, x^M_{e_f}) \tag{5.1}$$

where $x^M_{e_f}$ is the metadata feature vector of event e_f and $LSTM(h_{t-1}, x^M_{e_f})$ represents an LSTM function of hidden state h_{t-1} and input vector $x^M_{e_f}$. This recurrent network learns an embedding for each event preserving the sequential dependency between events.

Let $H^{b_n} = \{h_1, ..., h_k\}$ be the K embeddings of the K events from bag b_n. Each h_k embedding is of size L. Then the MIL Pooling step to learn the final bag representation z^{b_n} is

defined in Eq.(5.2).

$$\forall_{l=1,\ldots,L} : z_l^{b_n} = \max_{k=1,\ldots,K} h_{kl} \tag{5.2}$$

where z^{b_n} is a feature vector of size L, and is obtained from getting the maximum of each position l across all the K event embeddings of the bag (see Figure 5.2).

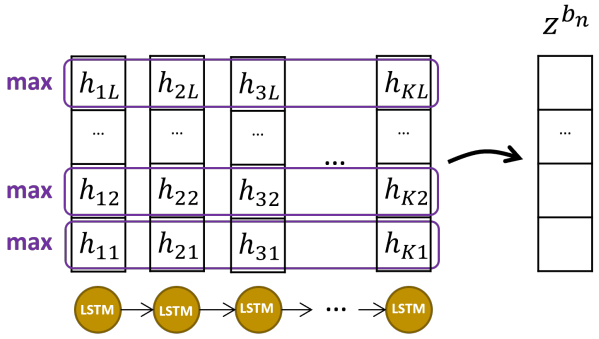


FIGURE 5.2: LSTM MIL Pooling schema for a bag b_n. h_{kL} is the hidden state of size L for event k.

This representation z^{b_n} is the input of a single sigmoid neuron which provides the score O_{b_n}, a value between 0 and 1, for the bag b_n to be an action proposal or not.

The Proposals stage of our fully automatic video summarization system presents a possible ambiguity to decide if a bag was positive or negative. We defined the condition: A bag is considered as positive if at least 3 of the 5 events belong to a ground truth action proposal. Figure 5.3 shows an example of how the bags are defined as positive using this condition. In this example there is an action proposal {*Pass, Pass, Pass, Goal-shot*}, a 5-events size sliding window and stride of 2. The bag with events {*Out, Pass, Pass, Pass, Pass*} is labeled as positive since 3 of the 5 events belong to the action proposal, however the 3 events are *Pass* type which are clearly not important events for a summary. This limitation was slightly overcome with the combination of the Log-Sum-Exp function, a small stride, and a small bag size.

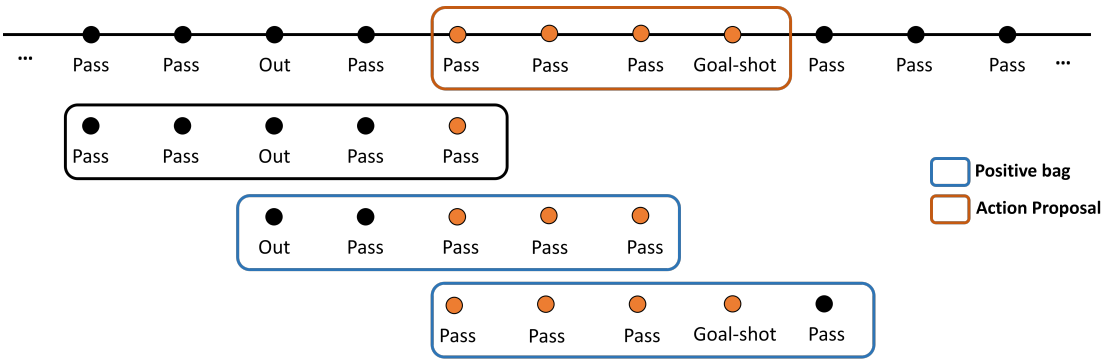


FIGURE 5.3: Example of the limitation of using fixed-size sliding window and a threshold condition to define if a bag is positive or negative.

One of the advantages of using an LSTM instead of fully-connected layers is that the input samples are not limited to a fixed size. Then instead of using a sliding window to create fixed-size bags, we can directly use the action proposals as bags. However, in testing phase there are no ground-truth action proposals. Therefore, in testing and validation phases we use the same process defined in our fully automatic video summarization system, a sliding window of 5-events size and stride 2 with the LSE function to obtain the score per event, as shown in the graphical representation of the Generation of Action Proposals block in Figure 5.1.

Finally, we find a threshold using the validation set and we consider as positive all the events with a score higher or equal than the threshold. Thus, an action proposal a_p is a set of positive consecutive events.

5.3 Hierarchical Multimodal Attention

In Section 5.2 we have described how we use LSTM MIL Pooling to obtain a score per event and how we have defined that an action proposal a_p is a set of positive consecutive events. Now the goal of the second stage of this approach is to define which of these proposals indeed belong to the summary.

As the experiments of previous chapter showed, the use of multiple modalities is relevant to decide which actions are important in a sport match. However, the naive concatenation previously used may not be the best approach since this type of fusion may limit the models' ability to dynamically determine the relevance of each modality to different parts of the action. The audio of an action can significantly vary not only from the type of the action but also from the events occurring inside the action. For instance, it is not the same kind of *goal* event, if the goal is preceded by several slow passes as if it is the result of an action starting by an interception or an error from the opponent team. For this reason, instead of learning the importance of each modality per action, we propose a hierarchical multimodal attention (HMA) mechanism that in the first stage learns the importance of each modality at the event level and in the second stage learns the importance of each event inside the action (see Figure 5.4).

Thus, in the first stage of the hierarchy, the multimodal representation vector per event is given by a weighted average:

$$c_i = \lambda_i^M h_i^M + \lambda_i^A h_i^A \tag{5.3}$$

where the weights of each modality $\{\lambda_i^M, \lambda_i^A\}$ are determined by an attention layer that shares the parameters W across time-steps:

$$\lambda_i^{\{M,A\}} = softmax(tanh(W(h_i^{\{M,A\}}))) \tag{5.4}$$

An action might contain several events that are not considered as important in a match, but they are relevant to provide a context to the fans. For instance, there are many fouls during the

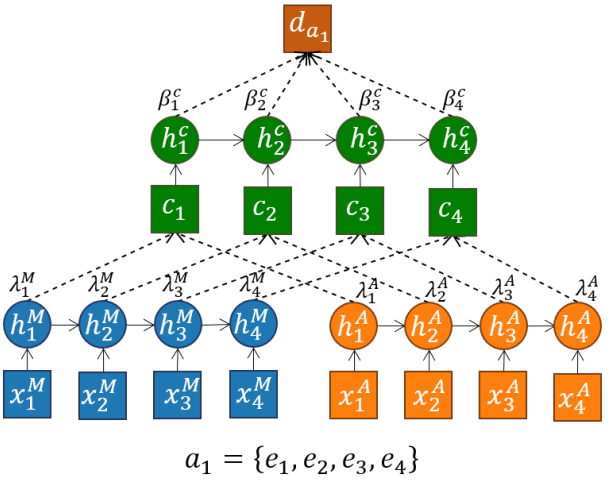


FIGURE 5.4: Definition of our hierarchical multimodal attention schema. Blue indicates metadatada, orange indicates audio and green indicate the multimodal representation of the events. λ, β and α are attention weights.

match, but if a card action belongs to the summary, it is important to show the *foul* and *pass* events that provoked this card. However, depending on the excitement of the crowd or the type of card, the importance of the events may vary. Therefore, after obtaining a multimodal representation per event c_i, we train an attention layer that learns the importance of each event inside the action, resulting in the weight β_i^c:

$$\beta_i^c = softmax(tanh(h_i^c)) \tag{5.5}$$

Thus, the representation vector per action proposal is given by a weighted average:

$$d_{a_p} = \sum_{i=1}^{L_p} \beta_i^c h_i^c \tag{5.6}$$

where L_p is the length (number of events) of action a_p. Finally, each of this d_{a_p} action representation is given to a sigmoid neuron which outputs a value between 0 and 1, that indicates the likelihood of the action a_p to be included in the summary.

5.4 Generation of Multiple Summaries

In Section 5.2 we have described how we use Multiple Instance Learning to detect action proposals to then in Section 5.3 exploit the metadata and audio features of the match to define which of these proposals indeed belong to the summary. However, one of the main challenges in sports summarization is the subjectivity since there is not a unique and perfect ground-truth summary for a match, it might depend on the platform where the video will be published, the league, the country, the length constraint, etc. As a solution to this subjectivity, we design a

strategy to propose candidate summaries giving the editors the chance to choose the best option according to their needs.

On the other hand, the wide use of social networks has changed the way new generations consume the content, users prefer shorter clips available in different online platforms. Then, during the creation of the summaries, the editors of broadcast companies are generally constrained by the length of the resulting video summary.

The natural way of creating a summary in a time-constraint context is to rank the actions by importance and then adding one by one until the summary's specified time-budget is filled. To give more freedom to the editor we would like to propose several rankings of the same match so they can generate several candidate summaries.

5.4.1 Ranking Distribution

One way to generate several rankings is to sample from a ranking distribution. A classic model for such distribution is the Plackett–Luce model, which was proposed by Plackett [140] to predict the ranks of horses in gambling.

The Plackett-Luce model is parameterized by a vector $\theta = (\theta_1, \theta_2, ..., \theta_P) \in \mathbb{R}_+^P$. Each θ_p can be interpreted as the *importance* of the option p, with higher values indicating that an item is more likely to be selected [141].

Consider a fixed set $A = \{a_1, ..., a_P\}$ of P action proposals. We identify a ranking over A with a permutation $\pi \in \mathbb{S}_P$, where $\mathbb{S}_P$ denotes the collection of permutations on $[P] = \{1, ..., P\}$. Thus, each π is a bijective mapping $[P] \to [P]$. The probability assigned by the Plackett-Luce model to a ranking represented by a permutation $\pi \in \mathbb{S}_P$ is given by

$$\mathbb{P}_\theta(\pi) = \prod_{i=1}^{P} \frac{\theta_{\pi^{-1}(i)}}{\theta_{\pi^{-1}(i)} + \theta_{\pi^{-1}(i+1)} + ... + \theta_{\pi^{-1}(P)}} \tag{5.7}$$

A ranking of P items can be viewed as a sequence of independent choices: first choosing the top-ranked item from all items, then choosing the second-ranked item from the remaining items and so on. In each step, the probability of an item to be chosen next is proportional to its *importance*. Consequently, items with a higher *importance* tend to occupy higher positions. In particular, the most probable ranking is simply obtained by sorting the items in decreasing order of their *importance* (like the previously explained natural way of creating a summary).

5.4.2 Ranking actions

The hierarchical multimodal approach explained in Section 5.3 gives as output a value between 0 and 1 that can be interpreted as the importance of each action. Therefore, the values of θ are determined by the output of the last layer of the multimodal summarization model.

To be more specific, if there are P actions detected in a match and therefore a θ_p value for each of these actions, each of the candidate summaries is built by sampling pl from the Plackett-Luce

distribution with parameter θ. A way to sample from the Plackett-Luce distribution is to sort Gumbel perturbed log-scores [142] as described in Equation 5.8.

$$\begin{aligned} pl &= argsort(\log\theta + g) \\ g &= Gumbel(0, \sigma) \end{aligned} \tag{5.8}$$

Let us consider a specific example. First, the Generation of Action Proposals stage provides P actions in a specific match, after the HMA outputs a score d_{a_p} for each of these actions. These scores are considered as the importance θ for the Plackett-Luce distribution. Then to generate a candidate summary, we sample a ranking from the distribution pl. The output of this sampling is a list of size P that indicates the position of each action, where the higher the ranking the higher the importance. The number of actions chosen to be part of the candidate summary is determined by the ground-truth summary length, the actions are chosen until the candidate summary has a duration that is very close to the ground-truth video summary duration. Finally, the candidate summary video contains the video clips of each chosen action, in the order they occur in the match.

For instance, in Figure 5.5 the first sampling pl_1 generates a ranking where the action a_3 is the most important followed by $a_{10}, a_5, a_1 a_9$ and so on. And the candidate summary generated from this ranking contains only the actions a_3, a_{10}, a_5 and a_1 since these 4 actions already comply the time constraint.

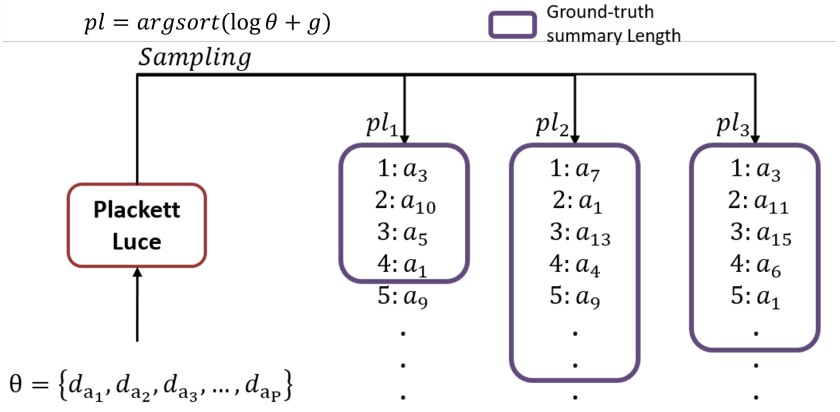


FIGURE 5.5: Schema of sampling from the Placket-Luce distribution. d_{a_p} are the scores predicted by the Multimodal Summarization stage. p_l represents a ranking sampled from the distribution. g are perturbed log-scores.

5.5 Experiments

In this section we introduce the new experimental setting we added for the semi-automatic summaries generator, describing two new datasets, the training and testing description for the Action proposals stage and the strategy to evaluate the generation of multiple summaries.

5.5.1 Summary Video Datasets

For our semi-automatic summaries generator, we created two datasets, one to train and evaluate our model, and the second to prove the generalizability of our approach. The first video dataset consists of 100 matches from the 2019-2020 season of the English Premier League. We take the exact videos broadcast on TV. The only ground-truth available for these matches are the 100 video summaries created by professional broadcasters. The duration of these video summaries varies from 110 to 270 seconds.

Pappalardo et al. [42] released a set of soccer-logs collected by Wyscout, containing all the spatio-temporal events that occur during all matches of an entire season of seven competitions. As a second video dataset, we choose the World Cup 2018 data since it is the competition for which we could find on Youtube the greatest number of matches and their corresponding summary video. We use 56 matches from the 64, since we do not consider the matches with extra times and the matches for which both match and summary videos are not available.

This second video dataset will be later used for evaluating transfer learning and generalization capability of our method.

5.5.2 Action Proposals

Training. In order to create the action proposals dataset, we follow the same steps described in Section 3.3. All the sequence of events matching with the action proposals are considered as positive bags. To obtain the negative (no-action) bags, we first randomly pool sequence of events from the parts of the match where there are not events labeled as action. The length of the pooled sequence of events varies from 4 to the maximum action proposal size in the training set. Finally, we randomly choose as many negative samples as the positive samples. We then train our sequential MIL model on all these sequences of events.

Testing. As previously mentioned, in a real-life scenario we do not have access to the ground truth intervals in the test phase, the bags are created in a class-agnostic way. We use a sliding window across the whole match events, with a stride such as there is an overlap between two consecutive windows, and we predict on these bags.

To obtain the action proposals for the Multimosal Summarization stage, we use LSE function and follow the steps previously described in section 4.2.

5.5.3 Evaluation of Summarization

In our fully automatic video summarization system we evaluated the generated summaries with the metrics used in summarization of general-purpose videos. In these metrics, Precision, Recall, and F-score are defined based on the overlapped time between the generated summary and the ground-truth summary (Equation 4.3). However, there are two main limitations of using that metric for our semi-automatic summaries generator. One is the use of events instead of frames and the other is the definition of action in sport videos.

Previously, a ground-truth action was an entire video clip, usually containing replay and reactions (from players, coach, or crowd). Now in our current solution, a ground-truth action contains only events, making them significantly shorter. Then comparing based on the overlapped time would not be fair since missing one event of a couple seconds length is not the same as missing the reply of an action. On the other hand, as we have mentioned several times in this dissertation and as it was depicted in Section 3.1.2, the start and end of an action is highly subjective in sports videos. Therefore, the location of the overlap between ground-truth and predictions is very relevant. It is not the same missing the last part of a *Goal* action where the goal-shot is located than missing some passes in the first part of the action.

Therefore, we decide to focus in a binary classification, whether the action is detected. In the rest of this dissertation, we still use Precision, Recall and F-scores to evaluate the summarization task but they are not based on the time overlap, they are based on the missing or detection of an action.

5.5.4 How to evaluate the Generation of Multiple Summaries

As mentioned before, there is not a unique and perfect ground-truth summary for a match due to subjectivity generally present in this task. A clear example is given by the goal-attempt actions, it is evident that the common actions between all the possible summaries are the goals of the match, but how can we assure that the goal-attempt of 4 minutes before the goal is more or less relevant in the summary than the one 6 minutes before? Sometimes the editorial reason to add some goal-attempts is just to show the persistence or the lead of a team. Then we can argue that the evaluation is not really fair when a non-detected goal-attempt is counted as false negative, and another very similar goal-attempt is counted as false positive.

Therefore, instead of evaluating the multiple generated summaries only in terms of time intervals, we consider that a ground-truth action is correctly detected if the predicted action meets two conditions:

1. The action type is the same.

2. It occurs in the time interval between the previous and the following ground-truth action.

In Figure 5.6 there is an example of a ground truth summary and two predictions. To check if the *Shot* action of the ground-truth summary is detected, we check if there is a predicted action

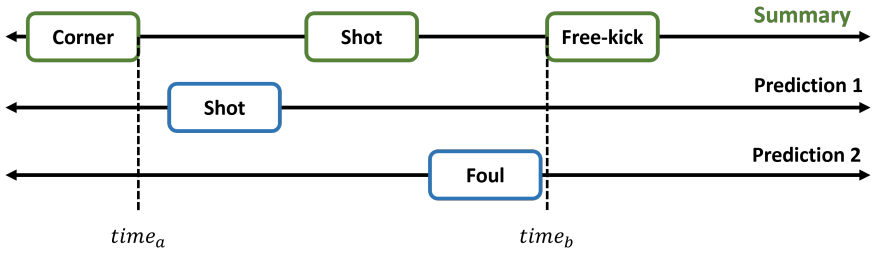


FIGURE 5.6: Example to illustrate how a ground truth action is considered as correctly detected. The arrows indicate the time thus, an action on the left occurs earlier in the match than an action on the right. The *Shot* action of the summary is detected in prediction 1 because there is a *Shot* action inside $[time_a, time_b]$ and it is not detected in prediction 2 because there is no *Shot* action inside this interval.

of the same type between the time interval $[time_a, time_b]$, where $time_a$ is the end time of the previous ground-truth action (i.e., *Corner*) and $time_b$ is the start time of the next ground-truth action (i.e., *Free-kick*). In the first prediction it is considered as predicted and even in the second the action is not detected since there is no *Shot* action inside the interval $[time_a, time_b]$.

We defined 10 types of actions T which cover all the possible actions of a summary: free-kick, corner, foul, shot, save, referee decision with video assistance (VAR), goal, end period, start period, other. An action is labeled with a certain type T if at least one of the events of the action is type T.

5.6 Results

For a fair comparison we use a 10-fold-cross-validation. Each fold has 80%, 10% and 10% of the matches for train, validation, and test set, respectively. For all the comparison experiments, we replicate the models from the source paper using Keras library and choose the parameters with the highest classification performance on our validation dataset

5.6.1 Generation of Action Proposals

In order to choose the best method to detect the actions, we compare with three different methods: SST [3], MI-Net [4] and MI-Net Attention [143].

As we previously mentioned, the idea of generating proposals has been tackled by several approaches. We chose SST [3], which was created to generate temporal action proposals for temporal action detection in untrimmed video sequences. This method is a RNN-based architecture that at each time step produces confidence scores of different action sizes ending at this time step. Instead of using video features as it was originally proposed, we use our event data features as input. We implement an LSTM network with 16 neurons, the proposal sizes are $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, it is trained with binary cross-entropy loss and Adam optimizer.

In training phase, an output interval is considered positive if at least 50% of it belongs to a positive action proposal.

In terms of Multiple Instance Learning, we chose two neural networks-based approaches. MI-Net is composed of fully connected layers to extract a representation per sample and then it gets a score per bag using a pooling layer with max operator over all the samples of the bag (see Figure 4.4). MI-Net Attention replaces the max-pooling layer by an attention mechanism that learns the importance of each sample of the bag.

The number of neurons for the fully connected layer of the MI-Net approaches are 32. The attention layer has 8 neurons. We use a stochastic gradient descent optimizer with a nestrov momentum 0.9, a weight decay 0.005 and initial learning rate 0.0005. The code is based on the authors implementation [144]. LSTM MIL Pooling network has 16 neurons. We use Adam optimizer with 0.9 and 0.999 as the exponential decay rate for the 1st and 2nd moment estimates, learning rate 0.001, binary cross-entropy as loss function and a batch size of 32 bags.

For the evaluation phase, we consider an action was correctly classified if at least 50% of the predicted action belongs to an action from the actions described in Section 3.3.

Since the objective of the *Generation of Action Proposals* stage is to detect all the possible actions of the match, we want to obtain the least false negatives rate possible. However, maximizing the recall might lead to predict all samples as positive. For this reason, we use the F2-score obtained in the validation set to choose the best threshold and best epoch per fold, since this score weights the recall twice as important without ignoring the precision. In order to use a sports terminology, from now on we will call *Missing Actions* to the false negatives rate, representing all the ground-truth actions that were not detected.

TABLE 5.5: Performance Comparison of Generation of Action Proposals methods.

Method	**Missing Actions**	**F2-score**
SST [3]	22.01	54.25
MI-Net [4]	14.47	71.68
MI-Net Attention [143]	19.01	70.87
LSTM MIL Pooling	8.82	73.73

Table 5.5 depicts the performance of the methods previously described. We will focus on the Missing Actions rate and F2-score. MIL methods are clearly better on detecting the different actions of the match since SST performs at least 16% worse than the rest of the methods. LSTM MIL Pooling outperforms all the other methods, it misses at least 5% less actions and gets a F2-score at least 2% higher compared with the second-best method (MI-Net).

The way of creating the Actions dataset proposed in Section 3.3, where we take as reference the training set which represents 80% of the matches, opens the question if it is really necessary a Generation of Action Proposals stage. One might argue that checking for the exact sequence of events of the training set on the testing set is enough to define the actions of the match.

TABLE 5.6: Detected actions before and after of the Generation of Action Proposals stage. Template Matching is assuming there is no learning to detect the proposals.

Method	Missing Actions	F2-score
Template Matching	10.87	45.55
LSTM MIL Pooling	8.82	73.73

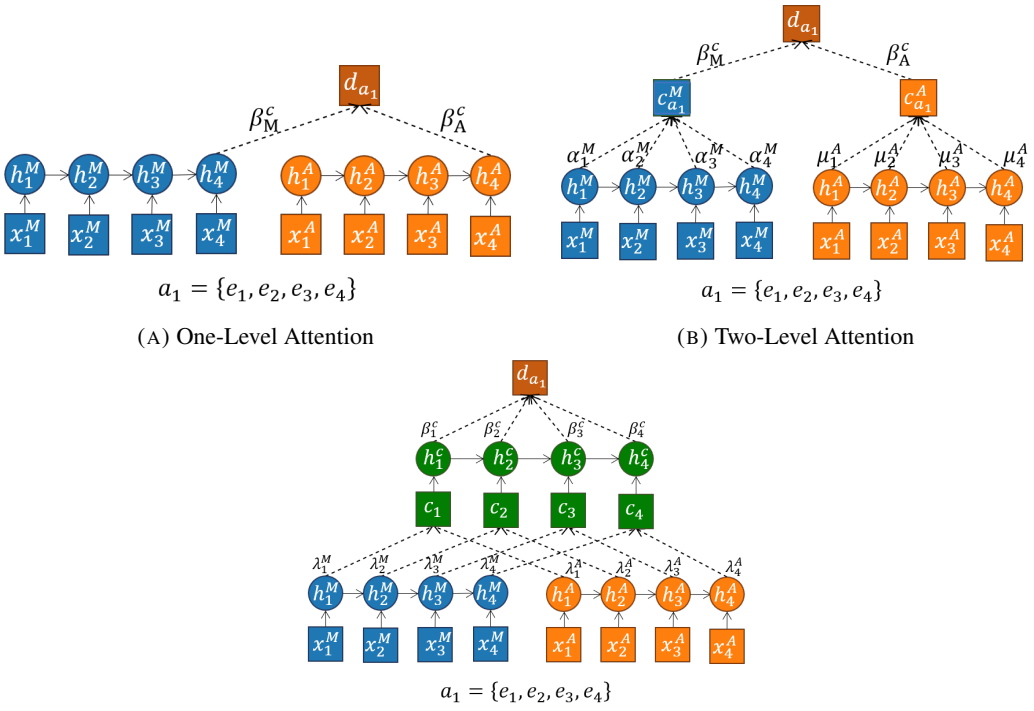
As ablation study of this stage, Table 5.6 compares the results between before and after the Generation of Action Proposals stage. The scores of the first row of the table (*Template Matching*) are obtained assuming there is no learning to detect the actions of the match, meaning that the actions are just the sequences of events that are identical to the reference actions (as described in Section 3.3).

After using LSTM-MIL Pooling we miss at least 2% less actions and get 28% more in F2-score. This can be explained analyzing the order of the events composing the actions. In the case of not having a Generation of Action Proposals stage, for instance if in the training set there are different actions formed from different combinations of the group of events *[interception, pass, pass, goal-shot]* but there is not a single action with the exact sequence of events {*pass, interception, pass, goal-shot*} found in the test set, which is the same set of events but in an order not found in any of the matches of the training set, then this action would be completely ignored by the Summarization stage. The same would happen with an action that contains the same group events but with some new event in the middle, e.g. {*interception, pass, pass, tackle, goal-shot*}. However, a method like LSTM-MIL Pooling might learn for instance that even if the same exact sequence of events is not present in the training set, a goal-shot event is always part of an action of the match.

5.6.2 Multimodal Summarization

We compare our Hierarchical Multimodal Attention (HMA) model with two of the most widely adopted structures for multimodal attention, we call them *One-Level Attention* and *Two-Level Attention*. *One-Level Attention* (see Figure 5.7a) computes a vector per modality and then uses an attention model to learn the importance of each modality [145, 146]. *Two-Level Attention* (see Figure 5.7b) uses an additional attention model for each modality independently [17, 147, 148]. *Multimodal H-RNN* is the multimodal Hierarchical LSTM proposed in our fully automatic video summarization system (Section 4.3), where we concatenate the audio and event features.

For the model parameters, we use 32 neurons for the LSTM of each modality, Adam optimizer with 0.9 and 0.999 as the exponential decay rate for the 1st and 2nd moment estimates, learning rate 0.001, binary cross-entropy as loss function and a batch size of 32 bags. Our model has 32 neurons in h^M and h^A, and 16 neurons in h^c.



(A) One-Level Attention

(B) Two-Level Attention

(C) Our Model

FIGURE 5.7: Comparison with other structures of multimodal attention. Blue indicates meta-datada, orange indicates audio and green indicate the multimodal representation of the events. λ, β and α are attention weights. (A) One-Level Attention learns a separate representation per modality and then attention layer learns the importance of each of them. This is usually called naive fusion. (B) Two-Level Attention implements an additional attention layer inside each modality model. (C) Our HMA model.

TABLE 5.7: Performance comparison of Multimodal Attention methods.

Method	Missing Actions	F-score
One-Level Attention	37.33	60.05
Two-Level Attention	32.45	65.75
Multimodal H-RNN	36.73	63.08
HMA	27.31	70.31

Table 5.7 shows the Missing Actions rate and F-score of the aforementioned methods. Our method misses at least 5% less actions and gets an increase of 7% in F-score, compared with the second-best method, Two-Level Attention.

We believe that our method outperforms *Two-Level Attention* because in this method the multimodal fusion is done at the action level. Indeed, their method has an attention layer at event level, but it is done separately per modality. Learning the importance of the event using only the audio features of a soccer match is a very difficult task. The left side of Figure 5.8 displays the attention learned in the audio part by the *Two-Level Attention* model in four different actions. It seems that the attention is just learning that the last events (i.e. the end of the actions) are more

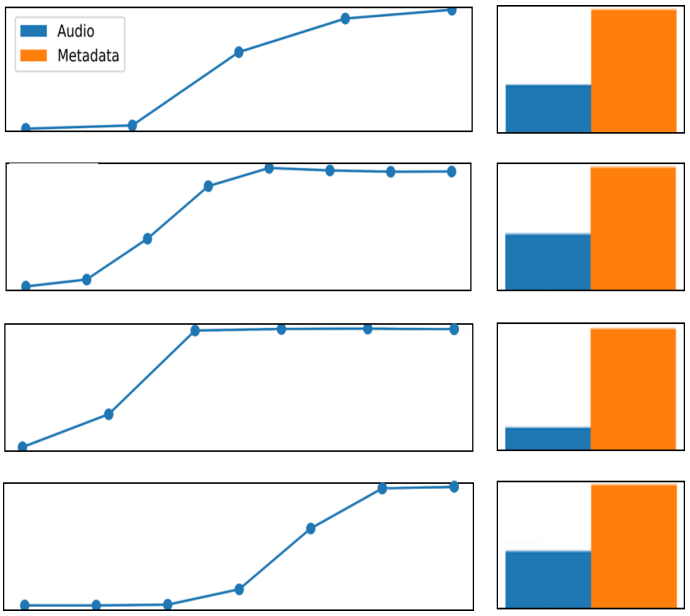


FIGURE 5.8: Examples of attention in different actions of Two-Level Attention model. On the left side, attention weights of audio part: The x axis is the sequence of events in the action and the y axis represents the weight values learned by the attention layer. On the right side, multimodal attention weights in the action level: the y axis is the weight values learned by the attention layer. Blue and orange represent the audio and the event data respectively.

important no matter the type of the events. The right side of this figure displays the importance learned by the attention for the audio and metadata modalities. This not only shows that for this model the metadata are often more important but also that the audio modality is most of the times neglected.

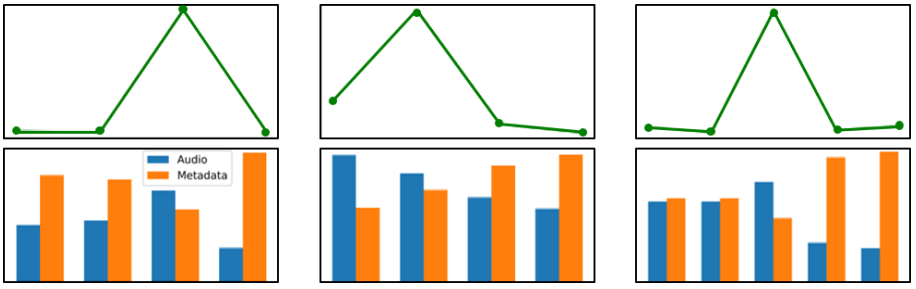


FIGURE 5.9: Examples of attention in different actions learned by our model. On the bottom, multimodal attention weights at the event level: The x axis is the sequence of events in the action and the y axis represents the weight values learned by the attention layer. On the top, attention weights learned from the multimodal representation of each event. Blue and orange represent the audio and the metadata respectively.

On the other hand, Figure 5.9 shows some qualitative results of our model. We can see that the multimodal attention does not follow a particular pattern, audio and metadata importance can be very different from one action to another. And the attention learned by the second stage of our model considers many important events where the audio was considered as more relevant from the previous stage.

TABLE 5.8: Performance of models for general-purpose videos, comparing the use of frames and events.

Method	Precision		Recall		F-score	
	Events	Frames	Events	Frames	Events	Frames
vsLSTM	72.08	24.24	63.44	32.88	67.49	27.91
H-RNN	54.33	54.73	50.00	41.10	52.07	46.94

Comparison with frame-based models. Similarly to the experiments we performed for our fully automatic video summarization system, we compared our semi-automatic summaries generator with vsLSTM [5] and H-RNN [6] methods.

For both methods trained with event features, the input is the same metadata feature vector $x_{e_f}^M$ used for our algorithm. In order to verify that the use of event data is more optimal than frames, Table 5.8 provides the precision, recall and F-score for the methods trained with frame features and event data. Both vsLSTM and H-RNN perform better using events. In addition, the training and inference time when using frames is at least 10 times higher.

TABLE 5.9: Performance comparison using only one modality.

Method	Missing Actions	F-score
Only Audio	24.91	61.83
Only Metadata	29.45	68.90
HMA	27.31	70.31

Multimodality. We also evaluate the performance of each modality separately. We train an LSTM with an attention layer for the audio and another one for the metadata, using the $x_{e_f}^A$ and $x_{e_f}^M$ vector features respectively. Table 5.9 shows that our method obtains the highest F-score compared with the models using only audio and only metadata features. Although using only audio features less actions are missing, the low F-score reveals the low precision of this method since it predicts a lot of false positives. This behavior is expected in sports videos since the crowd might produce a lot of noise even when an action is not important enough to be part of the summary. Comparing our results with the method using only metadata we can see that adding the audio features helps to reduce almost 2% of missing actions.

Soccer Baselines. Most of the state-of-the-art methods on sports summarization evaluate their performance based on the detection of the most common actions such as goals or shots on target. We propose three different baselines to do a fair comparison and to ensure that the summaries of our datasets do not follow a rule-specific pattern.

- *Random*: The prediction is a value taken from a continuous uniform distribution over the interval $[0, 1)$, where the samples with values below 0.5 are negatives and the ones greater or equal than 0.5 are positives.

TABLE 5.10: Performance comparison with Soccer Baselines.

Method	Precision	Recall	F-score
Only Goals	98.94	24.06	38.71
All Shots-on-Target	43.27	74.06	54.63
Random	41.37	43.49	42.40
HMA	68.08	72.69	70.31

- *Goals*: Since the easiest way to create a summary from a soccer video is to extract the goals of the match. This baseline considers as positive only the goal actions.
- *Shots-on-Target*: As the goals are not enough to create a soccer summary, this baseline broadens the type of actions considered as positive. All Shots on Target actions (i.e. goals, goalkeeper saving a shot on goal, any shot which goes wide or over the goal and whenever the ball hits the frame of the goal) are predicted as positive.

Table 5.10 compares the performance of these baselines and our method. Our F-score is clearly the highest, outperforming at least 15% the second best. The baseline *Goals* gets a precision score near to the maximum because it is very common that all the goals of the match belong to the summary, however its recall is the lowest since it misses many other type of actions. The recall of our approach is only outperformed by *All Shots-on-Target* since the type of actions considered in this baseline represent a big percentage of the actions generally included in summaries, yet its precision is at least 24% lower than ours, hence this baseline predicts more false positives than us. It can be interpreted as our algorithm manages to extract some knowledge to predict in a cleverer way which shot actions should be in the summary.

5.6.3 Multiple Summaries Generation

In order to generate multiple summaries from the same match, we sample ten times from the pl distribution shown in Equation 5.8, with $\sigma = 0.05$, where θ are the output values of HMA stage.

Each sample taken from pl generates a ranked list of actions per video, where the ranking represents the importance of the action. Each of these lists is used to create a candidate summary in the following way: The action with the highest rank is selected to be part of the candidate summary and it is removed from the ranked list, then the procedure is repeated until the candidate summary has a duration that is very close to the ground-truth video summary duration.

Let's explain more in detail the generation of multiple summaries from a specific example of our dataset. Table 5.11 shows the output of the 10 rankings generated by our approach for one of the Premier League matches. The Generation of Action Proposals stage predicted 19 action proposals for this match, which are described in the column *Proposals*. The table lists 21 because the two ground-truth actions (lines 26' and 66' of the table) that were not correctly classified are

TABLE 5.11: Ranking Example. Test ranking results of a match in the Premier League dataset. The rows are ordered by time in the match. The x symbol indicates there is a ground truth action inside that time interval which was not predicted by the Generation of Action Proposals stage. The symbol • in column *Proposals* shows if the was predicted as proposal. The numbers in the pl columns indicate the position of the action in each generated ranking. Blank space in the pl columns means that the action does not belong to the generated summary. Props column refers to Proposals

Time	**gt**	**Props.**	$\boldsymbol{pl_1}$	$\boldsymbol{pl_2}$	$\boldsymbol{pl_3}$	$\boldsymbol{pl_4}$	$\boldsymbol{pl_5}$	$\boldsymbol{pl_6}$	$\boldsymbol{pl_7}$	$\boldsymbol{pl_8}$	$\boldsymbol{pl_9}$	$\boldsymbol{pl_{10}}$
0’	start	•	1	3	1	1	2	3	2	2	3	3
19’	corner	•	13	12			12	12	10	13	12	13
22’		shot	10	9	10	10	10	11	13	10	10	11
23’	shot	•	4	4	5	4	4	6	4	4	4	4
26’	save	x	x	x	x	x	x	x	x	x	x	x
26’		save	6	5	4	5	5	5	6	5	5	5
29’		free-kick			11							10
30’	shot	•	9	8	9	9	9	9	8	9	9	9
39’		shot	12	11		12		13	12	11	13	
51’	save	•	5	6	6	6	6	5	5	6	6	6
51’		corner	11	10	12	11	11	10	11	12	11	12
57’	goal	•	2	2	3	2	3	1	1	3	1	2
66’	corner	x	x	x	x	x	x	x	x	x	x	x
67’		save	7	8	7	8	8	7	9	8	8	7
83’	save	•		7	8	7	7	8	7	7	7	8
85’		shot	8				13					
93’	end	•	3	1	2	3	1	2	3	1	2	1

also included for illustration purposes. The HMA network outputs a score for each of these 19 actions to then use it as θ for the Plackett-Luce distribution.

The generated summary using pl_6 ranking contains 13 actions, which means that if we sum-up the duration of the 13 most important actions according to this ranking, we will get a video summary with a duration very close to the ground-truth one. This ranking considers the goal action as the most important, followed by the end and start of the match, since they are ranked 1,2,3 respectively.

As a general analysis of this table, we can see that even though a save action (upper 26’ of the table) was not detected as an action in the first stage of our algorithm, there was another save action (lower 26’ of the table) in a near time which belongs to all the generated summaries. The summary generated from ranking pl_1 does not contain the last save action (line 85’ of the table) of the ground-truth summary but it has another save action (line 83’ of the table) from earlier in the match.

We define two baselines to compare with our method. The first one is based on the method

TABLE 5.12: Performance comparison of Multiple Summaries Generation.

Method	**Missing Actions**	**F-score**
Random Ranking	15.67	72.43
Collyda et al [149]	6.67	78.36
Ours	2.91	85.07

proposed by Collyda et al. [149], which ranks the output of a summarization method from higher to lower score. We rank the score generated by HMA stage. The second one is a *Random Ranking* where we shuffle the list of actions and the ranking is the position of the action in this shuffled list. The comparison results are depicted in Table 5.12. We choose the best ranking among 10 using the validation set and report the results of that ranking in the test set. Our method misses less than 3% of the actions and gets at least 6% higher F-score than the state-of-the art method.

An important advantage of our method is that it provides a possible solution to the always-present subjectivity in the summarization of sports videos. We provide multiple summaries to the final user that are close enough to not lose important information of the match but different enough to have options to choose. For instance, Table 5.11 shows that a free-kick action is in some of the generated summaries, adding some variability to the options without adding irrelevant actions.

5.6.4 Transfer Learning

As mentioned before, to the extent of our knowledge there is no available dataset for soccer summarization. However, Pappalardo et al. [42] released a set of soccer-logs collected by Wyscout, containing all the spatio-temporal events that occur during all matches of an entire season of seven competitions. We choose the World Cup 2018 data since it is the competition we could find online the most number of matches and summaries videos.

We follow the same steps described in Section 5.5 for the Premier League dataset. The code provided by Decroos et al. [43] was very useful to homogenize the two different sources of event data. Then we predict using the models trained with the Premier League data without doing any kind of fine-tuning. Both LSTM-MIL and HMA models were trained only with the Premier League dataset.

It is important to emphasize that there are many differences between the two competitions. In terms of audio, the Premier League commentators are French while in the World Cup videos are English, which is unofficially known to have a different level of excitement in the match coverage. Also, the crowd cheering is not the same in a league than in a world cup competition. For the metadata, Premier League matches are better detailed, so they have more events.

Table 5.13 shows the results obtained by the transfer learning of the HMA model. The Recall is high, but the Precision is very low which means that it outputs a lot of positive actions but

TABLE 5.13: Transfer Learning performance for the Summarization of the World Cup 2018 dataset.

Method	Precision	Recall	F-score
HMA	38.58	72.05	50.25
Our Generator of Multiple Summaries	93.79	87.03	90.28

many of them are not really part of the summary. This is an expected behavior since the video summaries of the World Cup dataset are clearly shorter (around 2 minutes) than the ones of the Premier League dataset (around 4 minutes). Usually when a summary is longer, it contains more actions like additional goal opportunities. Therefore, the model is trained to add more actions to the summary.

Even though this duration difference negatively affects the summarization scores, it is the perfect condition for our generation of multiple summaries. Those additional actions with high prediction score in the HMA model become good candidates to generate other summaries. The precision is improved by 55%, the Recall by 15% and the F-score by 40%. This confirms that our method can provide to the final user reliable summaries for the same match, even if the models were trained with a very different competition.

TABLE 5.14: Performance comparison of Multiple Summaries Generation with the World Cup 2018 dataset.

Method	Missing Actions	F-score
Random Ranking	11.81	55.54
Collyda et al [149]	12.39	81.39
Ours	12.98	90.28

In order to confirm our method also outperforms the baselines using the transfer learning on the World Cup dataset, Table 5.14 compares the results with the previously described baselines. We outperform the Collyda et al. method by almost 9%.

5.7 Discussion

To the extent of our knowledge, our method is the first to tackle the problem of subjectivity and time constraint by adding or removing actions from the final summary. Another method that we did not explore in this thesis is to modify the already existing actions to make them shorter or longer. We did not explore either the generation of summaries with different length for the same match. The main reason for not exploring these solutions is the lack of data, we do not have easily access to different summaries of the same match.

We could also evaluate how much our approach can be adjusted to one particular operator, since there are some differences between two operators summarizing the same game. To do so we will need enough matches summarized by the operator A, and enough matches summarized by the operator B, to see if we can learn the peculiarities and differences in their respective subjective representations.

As we previously explained, one of the main limitations of this thesis is the lack of public data to evaluate our approach. Even though Pappalardo et al. [33] made publicly available a dataset for event data, including the season 2017/2018 of five national soccer competitions in Europe and the World cup 2018, the videos of these matches with their respective summaries are not easy to find, mainly due to copyright restrictions. As a partial solution, we tried to consider SoccerNet [150] dataset as the ground-truth action proposals for the Generation of Action Proposals stage, since this dataset provides action spotting labels of 500 games. However, the matches in SoccerNet correspond to the seasons before 2017.

It is important to mention that the event data provided by Wildmoka Company was generated by OPTA and the event data provided by Pappalardo et al. [42] comes from WyScout. Therefore, we have defined a compatibility translation between the two datasets for the Transfer Learning experiments. In addition, OPTA data are more thorough than WyScout, therefore there are events missing in the Premier League dataset such as the VAR decisions, which we have manually included.

5.8 Conclusion

In this chapter, we proposed an algorithm composed of three consecutive stages: a Generation of Action Proposals stage where we describe a new Multiple Instance Learning for sequentially dependent instances, a Multimodal Summarization stage that exploits event features and audio features through a novel hierarchical attention at event level instead of action level, and finally a generator of multiple summaries, based on Plackett-Luce model, tackles subjectivity and time-budget constraints of sports video summarization. Experiments show that the three stages outperform state-of-the art methods and prove the generalizability of our model which can learn from one competition and transfer the knowledge to another competition acquired in different conditions and targeting different summary lengths.

There are several key contribution factors: a multiple instance learning model for sequential data, the multimodal attention per event instead of per action and fitting a ranking distribution on the data to generate multiple summaries per match.

Chapter 6

Additional Challenges on interpretability and missing event data

Although the main objective of this thesis is to detect the most important actions of a video, the path to achieve this goal involves many challenges. Such as, analyzing the interpretability of our model, learning the knowledge of the editors that led them to decide whether an action is relevant for a summary, removing the noise from the audio signal of the match, identifying keywords from the commentators' voice or detecting the players involved in an action. This chapter depicts some of the solutions we propose for these challenges.

6.1 Interpretability of our Sport Video Summarization system by profiling actions: An attention signal analysis

Broadcasting companies usually do not rely on automatic algorithms to provide their audience with the summary of a soccer game almost right after the end of the game, instead they mainly rely on human operators aided by algorithms. These operators use the video content to build up a summary but most of their work is based on event data. Indeed, processing the broadcast video would not be enough since some content is not shown on tv or possibly not under an optimal view angle, and watching while processing at the same time the content from all the cameras would not be tractable.

The number of events in a match is significantly smaller that the number of frames. Therefore, to build summaries, editors have designed handmade decision rules exploiting all the information hold by this event data, to produce the result as quickly as possible. These handmade rules are based on the type, the speed or the sequence of the events in order to determine different profiles of the actions.

How to make the choice between two *shots on target* (i.e. a shot that is very close to the goal frame but does not end in scoring a goal)? The operator looks at: How far from the goalmouth

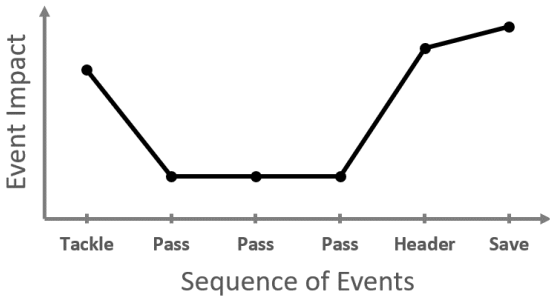


FIGURE 6.1: Example of an action profile that an operator has in mind when selecting this action for the summary: the first tackle to get the ball was amazing corresponding to a high impact of the event in the overall interest of the action. Then two unexciting passes led to a final long assist that reached the striker. Despite being in the middle of the defense, he successfully headed the ball, but it was unfortunately blocked by the goalkeeper.

the player shooting the ball was? From which angle? Who is the player shooting? At what time in the game the shoot happens? After which nice movement of the team? etc.

For instance in the example of Figure 6.1, the operator decides to choose this specific *shot on target* because before the goalkeeper caught the ball, the adversarial striker has made a very difficult header surrounded by adversaries after receiving the ball through a long pass from his winger, rewinding the action before this final assist there were two passes (without anything noticeable or emotional), but everything started with an amazing tackle to steal the ball. In this sequence of events, each event has a different weight representing the impact of the event's characteristics in the mind of the operator and so in the final choice of that action.

The question we intend to answer with this method is: Could we capture or learn in any way these mental representations leading an action to be relevant for the operator?

Previous works have shown that internal signals produced by neural networks during training or inference provide meaningful information to understand their decision making process and to interpret which input parts are involved in the final decision [151, 152].

In this method [153], we propose a way to analyze the model interpretability. We build an attentional recurrent neural network and show that its internal signals produce automatically, from event data, the profile of each action, providing the operator with a relevant action representation to support the final decision. We show also that our method:

- Generalizes from English Premier League to French Ligue 1 summaries using event data
- Generates meaningful profiles of the actions

6.1.1 Model

Our model consists of an LSTM network with an attention layer that captures the importance of each time step. Attentional models have been used before for translation, speech recognition and image caption generation [154, 155, 156]. Our attentional approach is like the image captioning methods [156] where the recurrent model learns to focus on the relevant parts of the

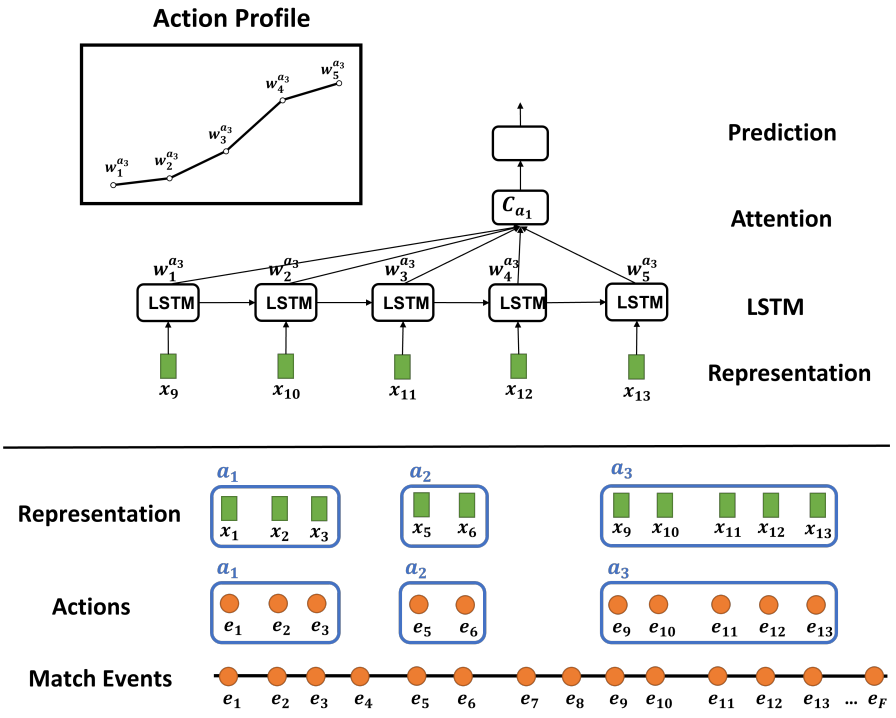


FIGURE 6.2: How the action profiles are generated. On the bottom, the description of events and actions used in this paper. On the top, the LSTM model with an attention layer, showing an example of our proposed graphical representation of an action profile.

image to better describe it. Our intuition is that this focus of attention is similar to the focus of attention of the operator.

Like in the previous sections, we describe a match as a sequence of events $e_1, e_2, ..., e_F$ and actions in the match by a_p, referring to a consecutive subset of events that does not overlap with others. For instance a goal action a_3 might be composed of five consecutive events $a_3 = \{e_9, e_{10}, e_{11}, e_{12}, e_{13}\} = \{pass, pass, tackle, pass, goal\}$. In this formulation x_{e_j} represents the feature vector of the j^{th} event e_j (see Figure 6.2 bottom).

In our approach, the LSTM takes as input the events of an action and predicts the likelihood of this action to be selected in the summary.

To be more precise let us take the action a_3 as an example (see Figure 6.2 top). This action has $L = 5$ events and the input of the LSTM is $\{x_9, x_{10}, x_{11}, x_{12}, x_{13}\}$. We denote $h_l^{a_3}$ as the hidden state of the LSTM unit at each time step l of action a_3. Then the attentional weights are defined as:

$$w^{a_3} = softmax(tanh(W(h_l^{a_3}))) \tag{6.1}$$

where W are the parameters of a single neuron.

Finally, the output of the final state of the LSTM is a weighted sum of all the hidden states of the action:

$$h_L^{a_3} = \sum_{l=1}^{L} w_l^{a_3} h_l^{a_3} \tag{6.2}$$

This final state is the input for a last sigmoid neuron that outputs a value between 0 and 1 that represents the likelihood of the action to be selected in the summary.

6.1.2 Graphical Action Profiles

Automatic summarization is important but sometimes the decision of whether an action is in the summary depends on different aspects like the style of editing, the enthusiasm of the fans or the target length of the resulting summary. Therefore, it is important to give additional information and different options to the operator.

Operators have usually designed hand-crafted rules to determine different profiles of the actions, these rules are based on the type, the speed, or the order of the events. These profiles represent different options of the same type of action. For instance, two possible profiles for a *Goal* action might be: the first one including several events before the actual goal because it was a very quick action, and a second one including only two events before the goal because it was preceded by a free-kick.

We have the intuition that the attention layer of our approach can implicitly learn a representation of the action profile. This new representation provides a new tool for the operator that might help her/him taking decisions.

We propose to extract the weights learned by the attention layer and plot them in a graph, where the x-axis represents the sequence of events and the y-axis is the weight value (see Figure 6.3). Hence, we create a graphical representation of the action profile.

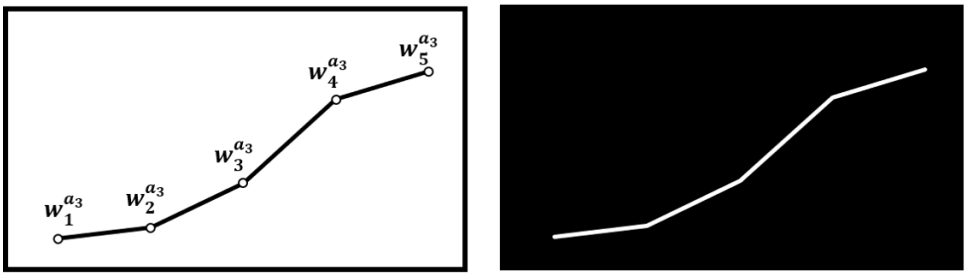


FIGURE 6.3: Graphical Action Profiles. This is an example for an action composed by five events. On the left, it is the curve generated from the weights learned by the attention layer, the x-axis represents the event order and the y-axis is the weight value. And on the right, it is the image representation used for the classification task.

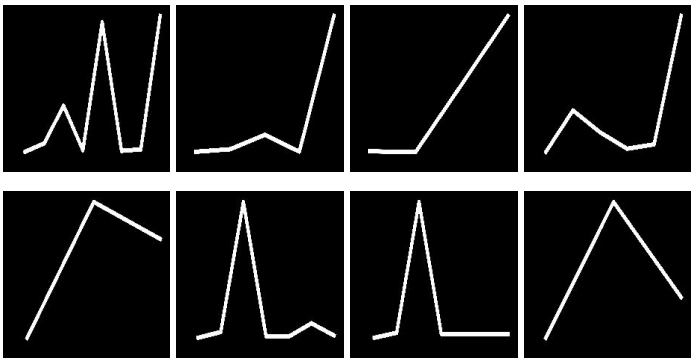


FIGURE 6.4: Examples of profiles for *Goal* actions (top) and *Miss* actions (bottom).

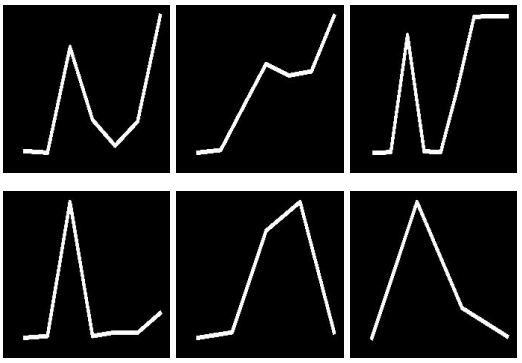


FIGURE 6.5: Profiles of positive (top) and negative (bottom) actions.

6.1.3 Experiments

Profile Features. The images of the profiles have black background and the curve representing the attention weights is white, as shown in the right side of Figure 6.3. We further extract GoogleNet features from these images to take the final decision: "should be in the summary or not?".

As aforementioned, our main goal in this method is to produce meaningful action profiles to help operators in selecting as quickly as possible the right sequences of events to put in the summary. In order to evaluate the quality of the profiles our attention LSTM has produced, without requesting the feedback from human operators, we train an SVM to determine whether the profile learned by the attentional layer indeed corresponds to an action that belongs to the summary. We generate an image profile for each of the actions of the dataset, then extract GoogleNet features from each of these images and train an SVM. The ground-truth is the same as the one used for the LSTM.

Some examples of the image profiles are shown in Figure 6.4. Four different *Goal* action profiles are depicted on the top of the Figure, where we clearly can see that the attention layer learned that the last event was very important, this last event is the actual *Goal* event. On the bottom of this figure, there are four examples of *Miss* action profiles. In Figure 6.5 we also can

TABLE 6.1: Classification results using graphical action profiles. *Only Goals* predicts all the goals as part of the summary. *All Shots-on-Target* predicts all the Shots on Target as part of the summary.

Method	**Precision**	**Recall**	**F1**
Ours	87.06	72.29	78.86
Only Goals	99.55	26.94	42.23
All Shots-on-Target	39.74	76.18	52.15
Random	20.16	47.58	28.21

TABLE 6.2: Generalization on classification results using graphical action profiles. All the results correspond to the classification scores for Ligue 1 matches. For *Ours*, the model was trained only with the Premier League matches. *Only Goals* predicts all the goals as part of the summary. *All Shots-on-Target* predicts all the Shots on Target as part of the summary.

Method	**Precision**	**Recall**	**F1**
Ours	79.88	68.01	72.73
Only Goals	100	31.93	47.14
All Shots-on-Target	37.72	73.84	49.44
Random	21.19	60.25	31.28

differentiate the profiles from positive and negative, i.e. actions that belong to the summary (top of the Figure) and actions that do not belong to the summary (bottom of the Figure).

The evaluation of most of the methods on video sports summarization are based on the detection of most important actions, then we compared with the three soccer baselines previously proposed:

- *Only Goals*: Only the goals of the match are predicted as positive. Since the easiest way to create a summary from a soccer video is to extract the goals of the match.
- *All Shots-on-Target*: All Shots on Target actions (i.e. goals, goalkeeper saving a shot on goal, any shot on goal which goes wide or over the goal and whenever the ball hits the frame of the goal) are predicted as positive.
- *Random*: The prediction is a random value between 0 and 1, where the samples with values below 0.5 are negatives and the ones greater or equal than 0.5 are positives.

Table 6.1 depicts the classification performance on our dataset. Our F1-score is clearly the highest. The Precision of our approach is only outperformed by *Only Goals*, considering it is very likely that all the goals of the match belong to the summary, however the Recall of this baseline is the lowest since it misses many other type of actions. The Recall of our approach is only outperformed by *All Shots-on-Target*, since the Shots on Target actions represent a big percentage of the actions included in summaries, yet the Precision of this baseline is at least 47% lower than ours. With these results we show the relevance of the graphical representation of the action profiles learned by our attention layer to help determining if an action is a good candidate

for the summary.

In order to prove the ability of our method to adapt properly to new unseen data, we also train our model using only the matches from the *Premier League* and analyze the results on 20 matches from *Ligue 1*. The performance results on Table 6.2 shows the relevance and generalizability of the representation our model learns from the data.

6.1.4 Conclusion

With this method, we have explored how current learning models could support human operators who produce handmade summaries which are broadcast right after each (soccer) match. Based on the idea that these operators use decision rules built on event data to keep actions or not in the summary, we have used an LSTM architecture model with an attention layer. The proposed approach generates images of action profiles from the weights learned by the attention layer. We prove the generalizability of our model which can learn from content provided by different broadcasters. We have also shown that the generated profiles contain meaningful information for the summarization task, since we can train an SVM to produce automatically a reasonably good summary from these profiles. Finally, neural networks produce internal signals (we could say intermediate signals), that are not only useful to understand better or to interpret the decision making process but these signals can also provide a new useful representation of the initial problem and lead to analyze it from a different perspective.

6.2 Missing event data: Keyword Spotting applied on Players Name Identification in Soccer Matches

As mentioned before, the event data collected by companies like Opta and Wyscout contain meaningful information providing a very detailed description, including the player involved in each of the events. However, these data present some constraints, they are not publicly accessible, they are not available for all the competitions and they still miss some information relevant for the storytelling of the match. For instance, there are events where players are not directly involved in the action, while actually they are very pertinent for the action. Those players are not registered in the event data because they did not touch the ball, but they are relevant either because they were blocking the opponent or because the action would have had a different outcome if they had actually touched the ball. Like a shot on goal which goes wide but the intention was to assist the main striker of the team who did not finally reach the ball.

In addition, the importance of an action during the match is not only determined by the type of the action but also by the players involved in it. This is mainly due to the way new generations are consuming the multimedia content and the huge amount of money lately invested for advertisement in this sport. Since soccer players have become *"super stars"*, their actions inside and outside the field became relevant information for the fans. Then the users share the best

moments of the match in social networks and react to the different events of the game. Sometimes a missed penalty of Mohamed Salah is more relevant than a goal scored by another less famous player of the Liverpool FC team. Therefore, identifying when the players are involved during the match becomes a relevant task.

On the other hand, the commentators play a very important role in a sports broadcasting, they are informed of all the facts of the teams and the players, they know the evolution of the game both on the field and on social media, and their main job is to tell the fans in real time what is happening during the match. For this reason, from the analysis of their speech we can extract relevant information, like the involvement of the players not visible in the event data previously mentioned. Furthermore, with an extensive processing of the commentators speech we could even obtain a good approximation of the event data collected by human observers from the field.

Keyword spotting refers to the task of detecting spoken words of interest in audio signals. Recently, many methods have been proposed in this area due to the increasing popularity of voice assistant systems. The task of detecting pre-defined keywords like “Ok Google”, “Alexa”, or “Hey Siri” in a stream audio is very similar to the detection of ”Kevin De Bruyne” in a broadcast soccer match. With this method, we want to detect every time a commentator mentions the name of a given player using the same technique that devices like Amazon Echo and Google Home implement to initiate a conversation with their users. To train our model, we exploit techniques such as in boosting, by iteratively focusing on the most difficult cases, to improve the detection results of keyword spotting.

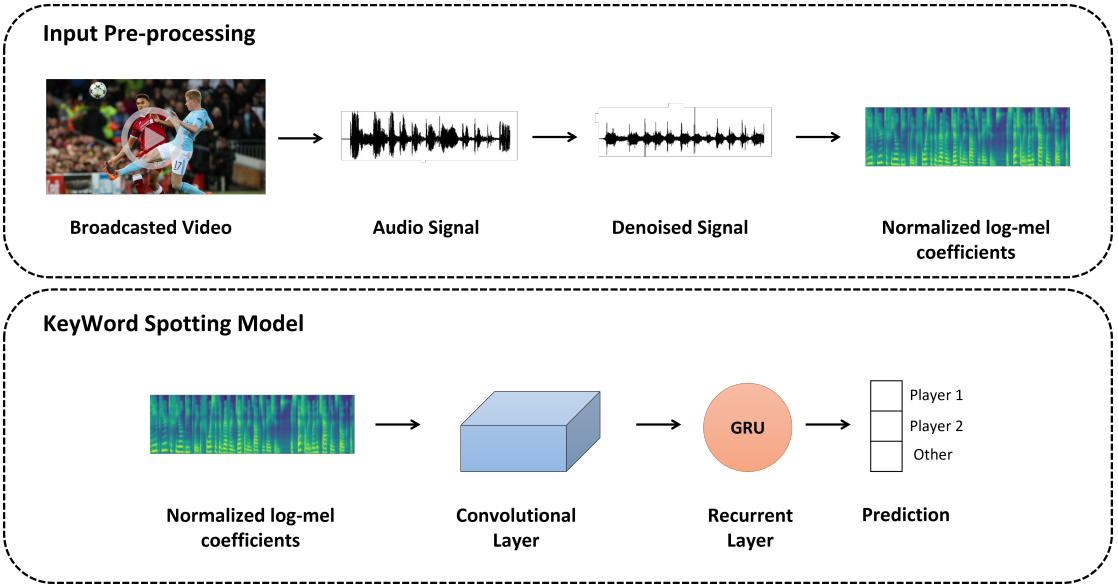


FIGURE 6.6: Our proposed approach for Player name Identification using KeyWord Spotting. The audio is extracted from the broadcasted video to then extract normalized log-mel coefficients from the denoised signal. The coefficients are gathered in an image-like representation to serve as input for a CRNN architecture.

Figure 6.6 shows all the steps of our approach. As a pre-processing step, we denoise the audio

extracted from the broadcast video, then we use the log-mel coefficients to create a representation per frame and finally we use Convolutional Recurrent Neural Network to predict if the keywords are mentioned in the input signal frame.

6.2.1 Related Works

Several approaches use the broadcast video to identify the player. Zhang et al. [157] propose a multi-camera framework for player segmentation and jersey number recognition, inspired on Mask R-CNN [158]. Lu et al. [159, 160] detect objects by searching over a histogram of gradient feature pyramid built from the image to automatically locate sport players. Theagarajan et al. [161] use YOLO9000 [162] to localize soccer players in a video and identify players controlling the ball. However, player identification using broadcast video is very challenging, players appear to be very small from the camera's perspective which causes a lot of occlusions, faces are blurry and in low resolution, making it impossible even for humans to identify players only from faces. For the methods based on jersey number recognition there are issues such as the body-shape variation leading to unrecognizable jersey number, and uniform similarity. Furthermore, in broadcast videos, owing to editorial choices not all actions are shown or only part of some actions is shown usually by following the ball. Then again, as for the event data, the "invisible involvement" of some players in key actions cannot be captured with such approaches.

To the extent of our knowledge, there is no method using the audio signal to identify players. Nevertheless there are several methods detecting keywords in audio signals [163, 164, 165], also known as keyword spotting or wakeword detection, related to the increasing popularity of voice assistant systems like Amazon Echo or Google Home. As these voice assistants usually run in small devices with limited memory and computation capabilities, most of keyword spotting methods focus on small-footprinting approaches. Tang et al. [166] explore the application of deep residual learning and dilated convolutions to discriminate one-second long utterances. Some other approaches use Recurrent Neural Networks (RNN) to capture the temporal behavior of the signal, Arik et al. [167] use a Convolutional RNN and Shan et al. [168] use an attention based RNN.

6.2.2 Denoising

One of the main challenges on processing the audio of soccer matches is the crowd noise of the signal. Separating the crowd noise from the commentator's voice using only signal processing techniques is not an easy task since both signals lay in almost the same range of frequencies.

Since in this work, we aim at exploring the potential of keyword spotting using Deep neural architectures in the field of sport content analysis, we used the NVIDIA RTX Voice plugin [169] to denoise the audio signal of the match and obtain a cleaner version where most of the crowd noise is removed. We evaluate the impact of denoising the signal on the players' name recognition task in the section Experiments, Tab. 6.3.

6.2.3 Keyword Spotting

After denoising the audio signal, we create input frames of size T. The raw time-domain inputs are converted to normalized log-mel coefficients. Then we split the resulting frequency-time vector over the time axis to obtain three different channels.

For the keyword spotting task, we focus on a well-known CRNN architecture, inspired by several speech processing systems [170, 167, 171, 166]. CRNN stands for Convolutional Recurrent Neural Networks, they take advantage of CNN for local feature extraction and RNN for temporal aggregation of the extracted features.

The three channels of the 2-D features $X \in \mathbb{R}^{FxL}$ are given as inputs to the convolutional layer, which employs 2-D filtering along both the time and frequency dimensions. After passing each feature map output through a rectifier linear unit (ReLU), each output becomes a tensor $V \in \mathbb{R}^{MxF'xL'}$, where M is the number of feature maps.

The tensor H is stacked over the frequency axis, resulting in $H \in \mathbb{R}^{L'x(MxF')}$ and then fed to a bidirectional gated recurrent unit (GRU).

Finally a single fully-connected layer with softmax activation reads the output of the recurrent layer and decides the name of the player pronounced in the frame input or it predicts the class "Other" if the audio does not correspond to any of the players to analyze.

In this architecture, the convolutional layer acts as a feature extractor and the GRU layer analyze the extracted features over time. The stack of convolutional, recurrent and fully-connected layers is trained end-to-end through backpropagation.

6.2.4 Boosting-like technique to enhance prediction

To create our ground-truth, we manually extract the pieces of audio containing only the searched keyword and store their beginning and end times, by benefiting from professional event data and their timestamps to coarsely localize in the timeline, the target player's name.

The commentators do not pronounce the keywords at the same speed or with the same emotion, according to the dynamic or the type of the action, the commentator pronounce the player's name in a different way, lengthening or shortening it. Therefore, we add some padding to the keyword snip in order to create samples of fixed size T. For the padding, we randomly choose 1000 samples of size T per match, where we evenly choose not-silence and silence samples. Not-silence samples might contain couch's shout, commentators' voice, etc.

We also create an "Other" class that represents any other piece of audio different from the keywords. For this class, we randomly choose three different pieces of audio of size T for each of the keyword samples.

Considering that the amount of time in a soccer match where the commentators do not mention the keywords is extremely high, the model has to be very robust identifying the "Other" class. To tackle this unbalanced problem, we decide to apply a technique used in boosting approaches.

The goal of boosting is to find the most ”representative” or ”informative” samples, by focusing on the difficult cases iteration after iteration.

After training our Keyword Spotting model with the previous dataset, where the samples belonging to the class “Other” are chosen randomly, we run the inference on the entire training matches. The parts of the match that belong to the class “Other” but were misclassified as keyword by our model are the *difficult* samples. These are the representative samples that help our model to better differentiate between keyword sample and the rest of the audio signal.

Hence, we create an extended version of our dataset, adding the difficult samples labeled as “Other” (see Figure 6.7).

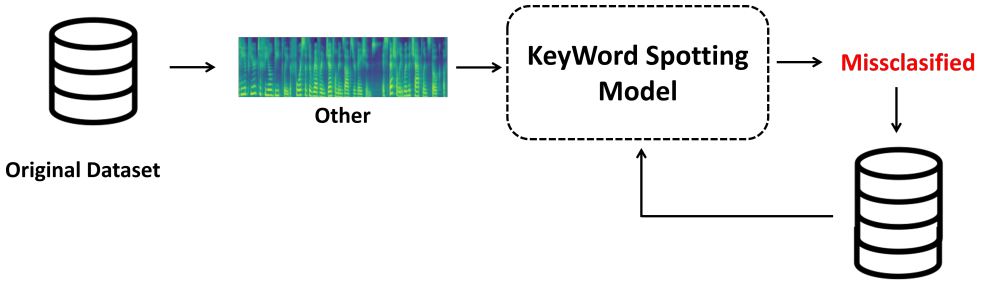


FIGURE 6.7: Boosting-like technique. “Other’ samples that were missclassifed during training are gathered to augment the training dataset.

6.2.5 Experiments

The experiments of this method are done using matches from the season 2019-2020 of the Premier League competition, broadcast in French language. We randomly chose 16 matches of the season where Manchester United and/or Liverpool were playing. The training samples are extracted from 12 of the matches and the validation and test samples from the remaining 4 matches.

Since Manchester City and Liverpool are the two teams leading the league, our dataset includes the detection of *Trent John Alexander Arnold*, the right-back player from Liverpool team and *Kevin De Bruyne*, the midfielder player from Manchester City team. We developed our Keyword spotting system for they keywords *Alexander Arnold* and *De Bruyne*, since those are the names used by the commentators in the broadcast matches.

For the player Kevin De Bruyne, the commentators might use also his full name instead of only his last name. For this reason, for this class we use two types of samples, commentators saying *Kevin De Bruyne* or *De Bruyne*.

Train and Validation sets. For these sets, we follow the process explained in Section 6.2.4. To create the fixed length samples, we choose a frame length $T = 1.5$ seconds, which is sufficiently long to capture a reasonable pronunciation of *Kevin De Bruyne* and *Alexander Arnold*.

Training samples are augmented by choosing different padding sample and locating the keyword in a random place inside the T seconds.

Test set. For the test set, we simulate a streaming scenario where inference is performed for overlapping frames of duration $T = 1.5$. The shift between the frames is 250ms (which is much longer than the spectrogram stride).

Input parameters. Using a sampling rate of 16kHz, each sample contains 24000 raw time-domain samples. We converted each of these samples into 40-dimensional log mel-filter-bank coefficients with delta and delta-deltas. They were computed with 20ms window, 10ms overlapped and normalized to have zero mean and unit variance [172, 167]. This configuration yields a 2-D vector dimensionality of 149x140, where the second dimension is split in three to create a 3-D vector of 149x40x3.

Network Specifications. The CNN layer has 32 filters, a kernel of size 5 and 3 as stride. The bidirectional GRU has layer 16 neurons in each direction. The weights were initialized with a Glorot normal distribution. To train our model we use Adam optimizer, binary cross-entropy as loss function and an early stopping with the validation accuracy.

6.2.5.1 Performance Results

TABLE 6.3: Comparison between using original audio signal and denoised audio signal.

Signal Type	Alexander Arnold	Kevin De Bruyne	Other
Original	0.741	0.176	1
Denoised	0.922	0.912	0.991

In the first part of our approach we propose to denoise the match audio signal as a pre-processing method. In Table 6.3, we show the accuracy per class using the original audio signal and the denoised audio signal on the validation set. The model trained with the denoised signal clearly outperforms the one trained with the original signal, obtaining more than 91% accuracy for all the classes.

TABLE 6.4: Accuracy per class before and after adding difficult samples to the training set.

Data Type	Alexander Arnold	Kevin De Bruyne	Other
Original dataset	0.922	0.912	0.991
Adding difficult samples	0.948	0.912	0.997

We also propose the use of a boosting approach to make the model more robust on identifying "Other" samples. The accuracy per class obtained on the validation set after adding the difficult

samples to the training set are provided in Table 6.4. The accuracy of the class “Alexander Arnold” improved at least 2% and 0.6% for the “Other” class.

Accuracy is a relevant metric, but we consider that it is also important to evaluate the precision and recall in the testing set. For this evaluation, we consider negative the “Other” class and positive the two player names. Table 6.5 shows that adding the difficult samples by boosting to the training set improves at least 27% the Precision, decreasing only by 4% the Recall. After adding these difficult cases to the training set, the model becomes more precise, it can differentiate better the names of the player from any other audio.

TABLE 6.5: Performance in the test set before and after adding difficult samples to the training set.

Data Type	**Precision**	**Recall**
Original dataset	0.395	0.86
Adding difficult samples	0.667	0.821

6.2.5.2 Challenging issues

To be fair on the comparison with the event data, we extract all the events from the event data where each player of the two teams (Manchester City and Liverpool) was involved, and then we select the player from each team with the highest number of events. These two players are Kevin De Bruyne and Alexander Arnold. However, as Kevin De Bruyne and Alexander Arnold are not the most famous players of their teams, in the 4 matches of the testing set, the amount of time representing the keywords is really small compared with the class “Other”, since the players are mentioned not more than 200 times in the testing set. Despite the clear predominance of the “Other” class, our approach has a specificity very close to 99.9%. This means that only very few samples where predicted as ”Alexander Arnold” or ”Kevin De Bruyne” when they were actually ”Other”.

The detection of players’ name mentioned by a commentator in a broadcast soccer rises several challenges. There is usually more than one commentator per match, the detector should be robust to different voices, speech overlapping, abrupt speech interruption. Also, the pronunciation can significantly vary according to the emotion of the game, the dynamic of the match or the type of action. The commentator then might pronounce the player’s name in a different way, lengthening or shortening it. One clear example of this challenge in our approach is that the class ”Kevin De Bruyne” is actually represented by two different ways to name the player, only his last name or his entire name. Regardless the variation inside the class, our method obtains more than 91% accuracy in “Kevin De Bruyne” class.

The audio signal of a soccer match contains many types of noise, crowd cheering, couch’s shouts, whistle, speakers in the stadium. This noise together with all the challenges previously mentioned makes even harder the keyword spotting detection task. As part of our on-going

work, we decide to use NVIDIA RTX Voice plugin since it removes a significantly amount of noise from the signal, however this tool targets general noise captured by a personal computer microphone like loud keyboard typing or ambient noise.

6.2.5.3 Comparison with event data

As we previously mentioned, another way to identify the player involved in each action is using the event data taken directly from the field. These event data record information like timestamp, action type, position, players involved, for all the events happening on the field. Table 6.6 shows the comparison between this event data and our approach on the test set. For the *Event Data* results, we take the event data of the test matches to then get the timestamp ts in seconds of all the events where the players Kevin De Bruyne and Alexander Arnold were involved. Then we manually check if the commentators mention the player's name in an interval $[ts - 2, ts + 2]$.

TABLE 6.6: Comparison with event Data. The ground-truth are all the times the players were mentioned by the commentators during the match.

Method	**False Positives**	**False Negatives**
Ours	0.189	0.176
Event Data	0.785	0.456

The False Negatives indicates the times where the player's name was mentioned by the commentators but according to the event data the player was not involved. The False Positives shows the times where the event data report the player was involved but the player's name was not mentioned by the commentators.

Based on the assumption that the commentators often mention the player's name when the player was involved directly or indirectly in the action, the high value on False Negatives shows that event data does not represent the real involvement of the player, since it misses many actions where the player is actually active (even if not touching the ball). On the other hand, our method has 0.28 less False Negatives.

Besides, the high value on False Positives for the event data indicates that even though there are many actions where the player touches the ball, most of them are not relevant for the story of the match. And our method has 0.6 less False Positives.

6.2.6 Conclusion

In this work, we identify players' name from the audio signal of broadcast matches. We use a keyword spotting approach to detect when the commentators mention the players' name. We also show how the robustness of the model is improved by iteratively focusing on the most difficult cases. Experiments performed with two different players in the English Premier League show the importance of denoising the audio signal of the match, the capability of our approach to identify

when the players are mentioned with a high accuracy and with both low false positive and false negative rates, and how the event data indeed miss the players who are indirectly involved in the action.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation we present our comprehensive processing of different modalities to select the most important information of soccer video matches. We contribute to summarization of sport videos field exploiting techniques applied for general-purpose videos but detecting, analyzing, and solving sport-specific issues. We split this work into six main contributions.

The first contribution is the use of event data instead of video as main source to detect match actions and generate summary videos.

Chapter 3 explained the main challenges of using soccer videos in terms of the amount of information to process and in terms of content. Event data on the other hand, provides more concrete information in less space, reducing significantly the subjectivity added by editors and focusing only in sports-related events. We analyzed the results of frames-based models in Chapters 4 and 5, and showed that the models using event data perform better and take less time.

The second contribution is to deal with inter-categorical similarity and intra-categorical diversity using Multiple Instance Learning on top of an LSTM. Thus, processing time dependent instances.

In Chapter 4 we presented some examples of inter-categorical similarity where events of the match labeled as part of the summary can be found also in some section of the match where nothing relevant is happening. We also showed how the intra-categorical diversity is present in soccer matches where actions of the same type are usually formed by different sequences of events. Then in Chapter 5 we went a step further and proposed to add a MIL Pooling layer on top of an LSTM, leveraging the capacity of LSTM to deal with sequential data. This method outperforms state of the art methods and detects as action proposals sequence of events that were not present in the training set.

The third contribution is a Hierarchical Multimodal attention that learns the importance of each modality (audio and event data) at the event level to then find the importance of each event in the action.

In Chapter 4 we started exploring the combination of different modalities, showing that the simple concatenation of audio features and event data outperforms the models using only one of the modalities. In Chapter 5 we proposed a more sophisticated model that takes into account sports-related characteristics. State of the art methods usually create a single feature-vector per modality to then learn their importance, however in sport actions the importance of the modality can vary at each event inside the action. Experiments showed that compared with state-of-the-art multimodal attention methods, our method misses less actions and does not neglect the audio modality and compared with soccer baselines it outperforms at least by 15% in terms of F-score.

The fourth contribution is a semi-automatic method to provide the editors multiple summaries of the same match. In Chapter 5 we described the last stage of our approach where we try to solve two challenges in the creation of sport summaries, subjectivity and length constraint. The score obtained from the multimodal hierarchical attention model is used as parameter of a ranking distribution to then extract different samples that correspond to sorted lists of actions. To evaluate this method considering the subjectivity we defined a way to determine if an action was correctly classified that allows to choose an action of the same type located in a relatively close time. The results showed that our ranking strategy outperforms the state of the art and a random baseline. In addition, the improvement of 55% in Precision, 15% in Recall and 40% in F-score proved how our method is able to learn from one competition and transfer the knowledge to a new dataset that is very different to the one it was originally trained with.

Chapter 6 brought two main contributions:

For the fifth contribution we analyze interpretability by defining a graphical representation of action profiles using the weights learned by an attention layer. Based on the assumption that editors in broadcasting companies have created rules from their knowledge and experience in order to decide what actions are more relevant, we showed that we can capture these mental representations (that we called *action profiles*) from the event data. We trained an attention layer an use its weights to build a graph per action and we then extract GoogleNet features from these images. The method outperformed soccer baselines in two competitions on the task of deciding whether the action profile corresponds to an action summary.

The sixth contribution of the thesis is to use a keyword spotting method to detect every time the commentators mention a player's name. One of the features provided by the event data is the player. However, only the player who touches the ball is reported which is not a direct indicator of the relevance of the player in the action. In a sample of two players in 16 matches, we showed that at least 45% of the times the player's name was mentioned by the commentators but according to the event data the player was not involved. Also, at least 78% of the times where the event data reports the player was involved but the player's name was not mentioned by the commentators. The keyword spotting method on the denoised audio signal with the addition of difficult samples by boosting, obtained an accuracy higher than 90% in the identification of two players in entire soccer matches. And the model was able to classify the pieces of audio where the commentators do not mention the two players' name almost perfectly.

7.2 Future Works

In this section we present the future steps that could be taken to continue towards the goal of detecting the most important actions of a match, taking the most from all the available sources of information. We will first discuss about new strategies to create video summarization datasets. We then present possible extensions of our method for other sports and different features to use as additional modalities.

7.2.1 Video Similarity

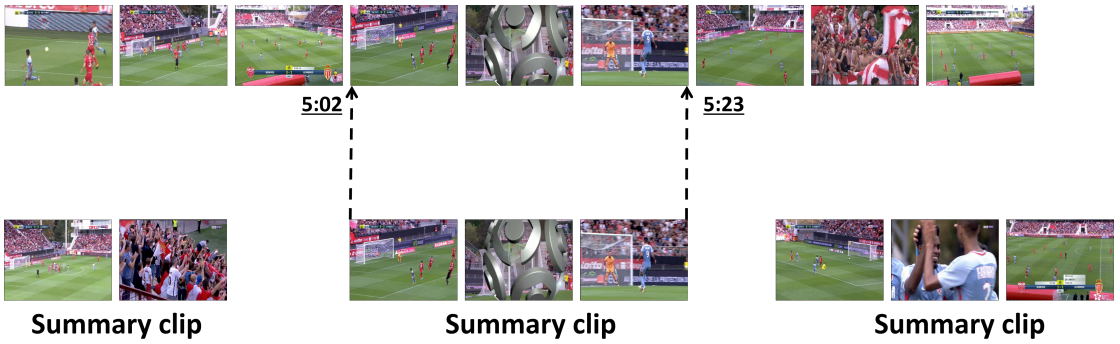


FIGURE 7.1: Example of the location of a summary clip in the original video. The frames in the top represent the original video match. In the bottom each group of frames represent a clip of the summary, i.e., an action. The arrows indicate the corresponding place of the summary clip in the original match video.

Despite the huge amount of sports video content available online, there are not public datasets for summarization. A dataset for video summarization is usually composed of a set of videos with their corresponding ground-truth, which are the parts of the video chosen to be part of the summary. The ground truth can be represented by the start and end times of each of the summary clips or the identification of the summary frames. Figure 7.1 shows a video and its corresponding summary clips, in this specific example the summary clip corresponds to the piece of match from minute 5:02 to 5:23.

General-purpose summary datasets such as Summe[111] and TVSum[112] are built asking several subjects to watch the entire video and select the most relevant parts. This task is very time consuming and expensive. One advantage of the use of a popular sport like soccer is that the videos of games and summaries are available on many platforms. However, one limitation besides copyright restrictions is to find the time intervals of the summary clips in their corresponding video match.

A solution to this problem is to split the original video into consecutive overlapping segments and perform an extensive comparison between these segments and each summary clip. Then the task of creating our own dataset becomes a task of finding the similarity between two videos. Then the ground-truth is defined as the segments that are very similar to the summary clips.

Considering that the summary videos are extracted directly from the original video, the naive approach is to compare the frames pixel by pixel. But even if the two videos have the same resolution, the frames are never the same since they are compressed during the creation of the video.

Assuming that the frames are not exactly the same, but they are very similar, we could use Euclidean distance or structural similarity [173]. However, in soccer the images of the field during the entire match are very much alike, which leads to several false positives. It seems more appropriate to compare groups of frames instead of single frames. For instance, the first frame (from left to right) of the second summary clip in Figure 7.1 is very similar to the second frame of the original match but the following frames are significantly different.

Due to the increased volume of video content available on the Web, there are several methods for video retrieval systems which is one of the applications of video similarity. A straightforward approach is to aggregate frame-level features into a single video-level representation to then compute a similarity measure. Some examples of these video-level representations are hash tables [174, 175], global vectors [176, 177] and bag of words [178, 179]. However, these methods do not take into account the spatial and temporal structure of the video. As future work, we could use a method like ViSiL [180], a video similarity learning network that considers both the spatial (intra-frame) and temporal (inter-frame) structure of the visual similarity.

It is important to remember that a video is composed not only by frames but also by audio. Therefore, similarity between videos can be found by relying only in audio features (since it is more difficult to find inter-clip similarity in audio) or by combining the two modalities (e.g., using audio to roughly find the location of the clip and then the video to define more precise segment's borders). As future work, we could use an audio similarity method like AuSiL [181].

Throughout this doctorate, we supervised students who tried to solve this issue during their Master internships. Mamadou Diop, Souraya Idrissi and Oumar Dieng explored the use of structural similarity between raw frames, CNN features and raw audio signal. Ruiqing Chang used Shazam[182] as inspiration to compare the descriptors extracted from audios of the summary and the original videos.

7.2.2 Other sports

While the study of this thesis has been focused to soccer our method is not restricted to this sport, we do not make any sport-based assumption and all the aforementioned companies (such as Wyscout, OPTA and STATS) acquiring event metadata are already providing similar information for several sports with large audiences (basketball, hockey, tennis, rugby, cricket, among others).

There are still some opened questions regarding other sports. It is not clear how the performance would be for sports where their number of actions is not as sparse as in soccer, like basketball where the number of points are in general higher than 50. Or for sports where the movement of the players is more limited like tennis. However, it would be interesting to explore

the challenges to face and how easily our method can be transferred to sports with similar characteristics like handball.

As an alternative to the commercial event data, there is also play-by-play information available for different sports. Even though play-by-play data only provides few features of the event such as player, time and event type, future work could analyze the performance of our method using such limited information for the events.

7.2.3 Other modalities

This thesis explores three different modalities: audio, event-data and video. For the audio and video, we extract general features which are not directly related to sports. For audio we extract features such as energy, entropy and MFCC and for video we obtain CNN features. These techniques are very powerful, but we could also detect whistle, silence (i.e when the commentators are not talking), competition's logo (which indicates a replay), replay, changes of camera, type of view, among others. Even though the extraction of all this information is time consuming, they might help to decide which actions are more important.

Another modality that is not highly explored in the state of the art which can be valuable in our context is the data from social networks. The number of views, comments or reactions is a good indicator of the importance or interestingness of an action. There might be some limitations on the time of acquisition since not all the matches have quick reactions from fans, but it can be a solution to decide between similar actions.

One aspect that Wildmoka is recently exploring is the adaptation of frames to the size and rotation in mobile screens. The image needs to be cropped or zoomed in to fit in the viewport of a small device. The naive approach is to detect the player and crop a window around her/him. However, the challenge is not only to decide which part of the image is more relevant in the current frame but also being able to forecast the following movement of the player in order to avoid abrupt changes or showing the user irrelevant parts of the frame. Pose estimation [183] might be a good start point to predict the movement of the players and decide what is the most relevant part of the frame to show to the fans.

7.2.4 Denoising

Section 6.2 shows that for the extraction of keywords mentioned by the commentators, the results are clearly better when we use the denoised audio signal. However, separating the crowd noise from the commentators' voice using only signal processing techniques is not an easy task since both signals are in the same range of frequencies, and using a private library as NVIDIA RTX Voice plugin [169] is not a very efficient solution. Youssef Benjelloun and Anass Nazih explored this problem under our supervision in their Master internship.

To train a denoiser algorithm, we need the original noisy audio and the denoised audio signal. The noisy audio corresponds to almost any soccer match found online, where the commentators'

voice is mixed with all the background noise of the stadium. It is relatively easy to find soccer videos with only crowd noise, but it is very difficult to have access to the audio signals that include only the commentators' voice. For this reason, the students used audio files from the console game Pro Evolution Soccer (PES). These set of files are different-size audios containing the recorded voice of professional commentators around the world.

Using these two sets of audios, one containing only commentators' voice (taken from PES) and another containing only crowd noise (taken from soccer videos available online), we created a synthetic dataset by randomly concatenating pieces of both sets with an additional random noise. As future work, we could train state-of-the-art methods such as, Conv-TasNet[184] and PIT-DNN [185], with this synthetic dataset to then use it as denoiser for our soccer dataset.

Appendix A

Publications

Submission: *Melissa Sanabria, Frédéric Precioso, Pierre-Alexandre Mattei, and Thomas Menguy. "A Multi-stage deep architecture for summary generation of soccer videos." IEEE transactions on neural networks and learning systems.*

Melissa Sanabria, Frédéric Precioso, and Thomas Menguy. "Hierarchical Multimodal Attention for Deep Video Summarization." 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021

Melissa Sanabria, Frédéric Precioso, and Thomas Menguy. "Profiling Actions for Sport Video Summarization: An attention signal analysis." 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP). IEEE, 2020.

Melissa Sanabria, Sherly, Frédéric Precioso, and Thomas Menguy. "A deep architecture for multimodal summarization of soccer games." Proceedings of the 2nd International Workshop ACM on Multimedia Content Analysis in Sports. 2019.

Bibliography

[1] T. Giannakopoulos and A. Pikrakis, *Introduction to audio analysis: a MATLAB® approach*. Academic Press, 2014.

[2] J. Van Haaren, P. Robberechts, T. Decroos, L. Bransen, and J. Davis, "Analyzing performance and playing style using ball event data," 2019.

[3] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles, "Sst: Single-stream temporal action proposals," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2911–2920.

[4] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognition*, vol. 74, pp. 15–24, 2018.

[5] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *European conference on computer vision*. Springer, 2016, pp. 766–782.

[6] B. Zhao, X. Li, and X. Lu, "Hierarchical recurrent neural network for video summarization," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 863–871.

[7] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image processing*, vol. 12, no. 7, pp. 796–807, 2003.

[8] M. Tavassolipour, M. Karimian, and S. Kasaei, "Event detection and summarization in soccer videos using bayesian network and copula," *IEEE Transactions on circuits and systems for video technology*, vol. 24, no. 2, pp. 291–304, 2014.

[9] T. Liu, Y. Lu, X. Lei, L. Zhang, H. Wang, W. Huang, and Z. Wang, "Soccer video event detection using 3d convolutional networks and shot boundary detection via deep feature distance," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 440–449.

[10] M. Merler, D. Joshi, Q.-B. Nguyen, S. Hammer, J. Kent, J. R. Smith, and R. S. Feris, "Automatic curation of golf highlights using multimodal excitement features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2017, pp. 57–65.

[11] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and system for segmentation and structure analysis in soccer video." in *ICME*, vol. 1. Citeseer, 2001, pp. 928–931.

[12] M.-H. Sigari, H. Soltanian-Zadeh, and H.-R. Pourreza, "Fast highlight detection and scoring for broadcast soccer video summarization using on-demand feature extraction and fuzzy inference," *International Journal of Computer Graphics*, vol. 6, no. 1, pp. 13–36, 2015.

[13] W. Zhao, Y. Lu, H. Jiang, and W. Huang, "Event detection in soccer videos using shot focus identification," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015, pp. 341–345.

[14] D. S. Pandya and M. A. Zaveri, "Frame based approach for automatic event boundary detection of soccer video using optical flow," in *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE, 2017, pp. 402–406.

[15] H. Jiang, Y. Lu, and J. Xue, "Automatic soccer video event detection based on a deep neural network combined cnn and rnn," in *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*. IEEE, 2016, pp. 490–494.

[16] M. Merler, K.-N. C. Mac, D. Joshi, Q.-B. Nguyen, S. Hammer, J. Kent, J. Xiong, M. N. Do, J. R. Smith, and R. S. Feris, "Automatic curation of sports highlights using multimodal excitement features," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1147–1160, 2018.

[17] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based multimodal fusion for video description," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4193–4202.

[18] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," *Pattern Recognition Letters*, vol. 25, no. 7, pp. 767–775, 2004.

[19] C.-L. Huang, H.-C. Shih, and C.-Y. Chao, "Semantic analysis of soccer video using dynamic bayesian network," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 749–760, 2006.

[20] H. Liu, "Highlight extraction in soccer videos by using multimodal analysis," in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2017, pp. 2169–2173.

[21] S. Hu, L. Sun, C. Xiao, and C. Gui, "Semantic-aware adaptation scheme for soccer video over mpeg-dash," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 493–498.

[22] Y.-Q. Yang, Y.-D. Lu, and W. Chen, "A framework for automatic detection of soccer goal event based on cinematic template," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 6. IEEE, 2004, pp. 3759–3764.

[23] T. I. F. A. Board, "Laws of the game 2019/20," *IFAB*, 2019.

[24] R. Leonardi, P. Migliorati, and M. Prandini, "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled markov chains," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 634–643, 2004.

[25] L. Ballan, A. Bazzica, M. Bertini, A. Del Bimbo, and G. Serra, "Deep networks for audio event classification in soccer videos," in *2009 IEEE International Conference on Multimedia and Expo*. IEEE, 2009, pp. 474–477.

[26] M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian, "Creating audio keywords for event detection in soccer video," in *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, vol. 2. IEEE, 2003, pp. II–281.

[27] E. Antonioni, V. Suriani, F. Solimando, D. Nardi, and D. Bloisi, "Learning from the crowd: Improving the decision making process in robot soccer using the audience noise," 07 2021.

[28] A. Hanjalic, "Adaptive extraction of highlights from a sport video based on excitement modeling," *IEEE transactions on Multimedia*, vol. 7, no. 6, pp. 1114–1122, 2005.

[29] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics*, vol. 158, p. 107020, 2020.

[30] "Wyscout," Jun 2021. [Online]. Available: https://wyscout.com/

[31] "Opta stats," Jun 2021. [Online]. Available: https://www.statsperform.com/opta/

[32] "Statsbomb," Jun 2021. [Online]. Available: https://statsbomb.com/

[33] L. Pappalardo, P. Cintia, A. Rossi, E. Massucco, P. Ferragina, D. Pedreschi, and F. Giannotti, "A public data set of spatio-temporal match events in soccer competitions," *Sci Data*, vol. 6, no. 1, pp. 1–15, 2019.

[34] H. Liu, W. Hopkins, A. M. Gómez, and S. J. Molinuevo, "Inter-operator reliability of live football match statistics from opta sportsdata," *International Journal of Performance Analysis in Sport*, vol. 13, no. 3, pp. 803–821, 2013.

[35] P. Lucey, D. Oliver, P. Carr, J. Roth, and I. Matthews, "Assessing team strategy using spatiotemporal data," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1366–1374.

[36] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, "Identifying team style in soccer using formations learned from spatiotemporal tracking data," in *IEEE international conference on data mining workshop*, 2014, pp. 9–14.

[37] J. Van Haaren, V. Dzyuba, S. Hannosset, and J. Davis, "Automatically discovering offensive patterns in soccer match data," in *International Symposium on Intelligent Data Analysis*. Springer, 2015, pp. 286–297.

[38] L. Gyarmati and X. Anguera, "Automatic extraction of the passing strategies of soccer teams," *arXiv preprint arXiv:1508.02171*, 2015.

[39] V. Vercruyssen, L. De Raedt, and J. Davis, "Qualitative spatial reasoning for soccer pass prediction," in *CEUR Workshop Proceedings*, vol. 1842, 2016.

[40] T. Decroos, J. Van Haaren, and J. Davis, "Automatic discovery of tactics in spatio-temporal soccer match data," in *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*. ACM, 2018, pp. 223–232.

[41] G. Liu and O. Schulte, "Deep reinforcement learning in ice hockey for context-aware player evaluation," *arXiv preprint arXiv:1805.11088*, 2018.

[42] L. Pappalardo, P. Cintia, P. Ferragina, E. Massucco, D. Pedreschi, and F. Giannotti, "Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 5, pp. 1–27, 2019.

[43] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis, "Actions speak louder than goals: Valuing player actions in soccer," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1851–1861.

[44] L. Bransen and J. Van Haaren, "Measuring football players' on-the-ball contributions from passes during games," in *International Workshop on Machine Learning and Data Mining for Sports Analytics*. Springer, 2018, pp. 3–15.

[45] A. Chacoma, N. Almeira, J. Perotti, and O. Billoni, "Modeling ball possession dynamics in the game of football," *Physical Review E*, vol. 102, no. 4, p. 042120, 2020.

[46] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199*, 2014.

[47] R. Christoph and F. A. Pinz, "Spatiotemporal residual networks for video action recognition," *Advances in neural information processing systems*, pp. 3468–3476, 2016.

[48] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[49] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[50] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*. Springer, 2016, pp. 20–36.

[51] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1510–1517, 2017.

[52] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[53] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[54] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1491–1498.

[55] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2914–2923.

[56] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1914–1923.

[57] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7094–7103.

[58] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 768–784.

[59] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *IEEE transactions on pattern analysis and machine intelligence*, 2015, pp. 91–99.

[60] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen, "Temporal context network for activity localization in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5793–5802.

[61] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5783–5792.

[62] Z. Yang, J. Gao, and R. Nevatia, "Spatio-temporal action detection with cascade proposal and location anticipation," *arXiv preprint arXiv:1708.00042*, 2017.

[63] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," in *Procedings of the British Machine Vision Conference*, 2017.

[64] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 988–996.

[65] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[66] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[67] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2843–2851.

[68] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 189–203, 2016.

[69] S. Rahman, S. Khan, and N. Barnes, "Deep0tag: Deep multiple instance learning for zero-shot image tagging," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 242–255, 2019.

[70] C. Mercan, S. Aksoy, E. Mercan, L. G. Shapiro, D. L. Weaver, and J. G. Elmore, "Multi-instance multi-label learning for multi-class classification of whole slide breast histopathology images," *IEEE transactions on medical imaging*, vol. 37, no. 1, pp. 316–325, 2017.

[71] P. Sudharshan, C. Petitjean, F. Spanhol, L. E. Oliveira, L. Heutte, and P. Honeine, "Multiple instance learning for histopathological breast cancer image classification," *Expert Systems with Applications*, vol. 117, pp. 103–111, 2019.

[72] G. Quellec, G. Cazuguel, B. Cochener, and M. Lamard, "Multiple-instance learning for medical image and video analysis," *IEEE reviews in biomedical engineering*, vol. 10, pp. 213–234, 2017.

[73] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," *IEEE multimedia*, vol. 9, no. 3, pp. 42–55, 2002.

[74] M. Otani, Y. Nakashima, E. Rahtu, and J. Heikkila, "Rethinking the evaluation of video summaries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7596–7604.

[75] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Video summarization using deep neural networks: A survey," *arXiv preprint arXiv:2101.06072*, 2021.

[76] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1059–1067.

[77] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3584–3592.

[78] R. Panda and A. K. Roy-Chowdhury, "Collaborative summarization of topic-related videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[79] M. Rochan, L. Ye, and Y. Wang, "Video summarization using fully convolutional sequence networks," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 347–363.

[80] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3090–3098.

[81] X. Li, B. Zhao, and X. Lu, "A general framework for edited video and raw video summarization," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3652–3664, 2017.

[82] M. Rochan and Y. Wang, "Video summarization by learning from unpaired data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 7902–7911.

[83] X. He, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, "Unsupervised video summarization with attentive conditional generative adversarial networks,"

in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2296–2304.

[84] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Unsupervised video summarization via attention-driven adversarial learning," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 492–504.

[85] C. Huang and H. Wang, "A novel key-frames selection framework for comprehensive video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 577–589, 2019.

[86] Y. Yuan, H. Li, and Q. Wang, "Spatiotemporal modeling for video summarization using convolutional recurrent neural network," *IEEE Access*, vol. 7, pp. 64 676–64 685, 2019.

[87] J. Wang, W. Wang, Z. Wang, L. Wang, D. Feng, and T. Tan, "Stacked memory network for video summarization," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 836–844.

[88] M. Elfeki and A. Borji, "Video summarization via actionness ranking," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 754–763.

[89] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[90] B. Zhao, X. Li, and X. Lu, "Property-constrained dual learning for video summarization," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 10, pp. 3989–4000, 2019.

[91] M. Y. Eldib, B. S. Abou Zaid, H. M. Zawbaa, M. El-Zahar, and M. El-Saban, "Soccer video summarization using enhanced logo detection," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 4345–4348.

[92] N. Nguyen and A. Yoshitaka, "Soccer video summarization based on cinematography and motion analysis," in *2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2014, pp. 1–6.

[93] J. Yu, A. Lei, and Y. Hu, "Soccer video event detection based on deep learning," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 377–389.

[94] A. Javed, A. Irtaza, Y. Khaliq, H. Malik, and M. T. Mahmood, "Replay and key-events detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine," *Applied Intelligence*, pp. 1–19, 2019.

[95] R. Agyeman, R. Muhammad, and G. S. Choi, "Soccer video summarization using deep learning," in *IEEE MIPR*, 2019, pp. 270–273.

[96] D. Corney, C. Martin, and A. Göker, "Two sides to every story: Subjective event summarization of sports events using twitter." in *SoMuS@ ICMR*. Citeseer, 2014.

[97] A. Tang and S. Boring, "# epicplay: Crowd-sourcing sports video highlights," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2012, pp. 1569–1572.

[98] Y. Huang, C. Shen, and T. Li, "Event summarization for sports games using twitter streams," *WWW*, vol. 21, no. 3, pp. 609–627, 2018.

[99] K. Tang, Y. Bao, Z. Zhao, L. Zhu, Y. Lin, and Y. Peng, "Autohighlight: Automatic highlights detection and segmentation in soccer matches," in *IEEE International Conference on Big Data*, 2018, pp. 4619–4624.

[100] E. Mendi, H. B. Clemente, and C. Bayrak, "Sports video summarization based on motion analysis," *Computers & Electrical Engineering*, vol. 39, no. 3, pp. 790–796, 2013.

[101] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in *Proceedings of the eighth ACM international conference on Multimedia*, 2000, pp. 105–115.

[102] A. Baijal, J. Cho, W. Lee, and B.-S. Ko, "Sports highlights generation based on acoustic events detection: A rugby case study," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, 2015, pp. 20–23.

[103] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework," in *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, vol. 3. IEEE, 2003, pp. III–401.

[104] V. Bettadapura, C. Pantofaru, and I. Essa, "Leveraging contextual cues for generating basketball highlights," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 908–917.

[105] A. Raventos, R. Quijada, L. Torres, and F. Tarrés, "Automatic summarization of soccer highlights using audio-visual descriptors," *SpringerPlus*, vol. 4, no. 1, p. 301, 2015.

[106] P. Shukla, H. Sadana, A. Bansal, D. Verma, C. Elmadjian, B. Raman, and M. Turk, "Automatic cricket highlight generation using event-driven and excitement-based features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1800–1808.

[107] A. A. Khan, J. Shao, W. Ali, and S. Tumrani, "Content-aware summarization of broadcast sports videos: An audio–visual feature extraction approach," *Neural Processing Letters*, vol. 52, no. 3, pp. 1945–1968, 2020.

[108] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 961–970.

[109] Y. G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2014.

[110] H. Zhao, A. Torralba, L. Torresani, and Z. Yan, "Hacs: Human action clips and segments dataset for recognition and temporal localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8668–8678.

[111] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 505–520.

[112] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5179–5187.

[113] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, and A. de Albuquerque Araújo, "Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[114] O. 2011. (2011) Open video project. [Online]. Available: https://open-video.org/

[115] K.-H. Zeng, T.-H. Chen, J. C. Niebles, and M. Sun, "Generation for user generated videos," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 609–625.

[116] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "Soccernet: A scalable dataset for action spotting in soccer videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1711–1721.

[117] C. L. M. D. F. René and V. A. R. G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[118] W. Chen, C. Xiong, R. Xu, and J. J. Corso, "Actionness ranking with lattice conditional ordinal random fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 748–755.

[119] G. A. Sigurdsson, O. Russakovsky, and A. Gupta, "What actions are needed for understanding human actions in videos?" in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2137–2146.

[120] B. Scharf, "Critical bands," *Foundation of modern auditory theory*, vol. 1, pp. 159–202, 1970.

[121] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[122] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[123] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 845–853.

[124] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8971–8980.

[125] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly supervised region proposal network and object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 352–368.

[126] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3628–3636.

[127] D. Guo, W. Li, and X. Fang, "Fully convolutional network for multiscale temporal action proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3428–3438, 2018.

[128] K. Zhang, K. Grauman, and F. Sha, "Retrospective encoders for video summarization," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 383–399.

[129] B. Zhao, X. Li, and X. Lu, "Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7405–7414.

[130] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 202–211.

[131] G. Fião, T. Romão, N. Correia, P. Centieiro, and A. E. Dias, "Automatic generation of sport video highlights based on fan's emotions and content," in *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*, 2016, p. 29.

[132] D. Chakrabarti and K. Punera, "Event summarization using tweets," in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[133] J. Huang, Y. Dong, J. Liu, C. Dong, and H. Wang, "Sports audio segmentation and classification," in *2009 IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 2009, pp. 379–383.

[134] H.-G. Kim, S. Roeber, A. Samour, and T. Sikora, "Detection of goal events in soccer videos," in *Storage and Retrieval Methods and Applications for Multimedia 2005*, vol. 5682. International Society for Optics and Photonics, 2005, pp. 317–325.

[135] M. Raghuram, N. R. Chavan, S. G. Koolagudi, and P. B. Ramteke, "Efficient audio segmentation in soccer videos," in *2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2016, pp. 1–4.

[136] R. Gade, M. Abou-Zleikha, M. Graesboll Christensen, and T. B. Moeslund, "Audio-visual classification of sports types," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 51–56.

[137] S. Chandrakala and S. Jayalakshmi, "Environmental audio scene and sound event recognition for autonomous surveillance: A survey and comparative studies," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.

[138] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *ACM SIGKDD Explorations Newsletter*, vol. 3, no. 1, pp. 27–32, 2001.

[139] V. M. Janakiraman, "Explaining aviation safety incidents using deep temporal multiple instance learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 406–415.

[140] R. L. Plackett, "The analysis of permutations," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 24, no. 2, pp. 193–202, 1975.

[141] A. El Mesaoudi-Paul, E. Hüllermeier, and R. Busa-Fekete, "Ranking distributions based on noisy sorting," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3472–3480.

[142] A. Grover, E. Wang, A. Zweig, and S. Ermon, "Stochastic optimization of sorting networks via continuous relaxations," *arXiv preprint arXiv:1903.08850*, 2019.

[143] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," *arXiv preprint arXiv:1802.04712*, 2018.

[144] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Minns," https://github.com/yanyongluan/MINNs, 2017.

[145] J. Xu, T. Yao, Y. Zhang, and T. Mei, "Learning multimodal attention lstm networks for video captioning," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 537–545.

[146] O. Caglayan, L. Barrault, and F. Bougares, "Multimodal attention for neural machine translation," *arXiv preprint arXiv:1609.03976*, 2016.

[147] H. Li, P. Yuan, S. Xu, Y. Wu, X. He, and B. Zhou, "Aspect-aware multimodal summarization for chinese e-commerce products," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8188–8195.

[148] J. Zhu, H. Li, T. Liu, Y. Zhou, J. Zhang, and C. Zong, "Msmo: Multimodal summarization with multimodal output," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 4154–4164.

[149] C. Collyda, K. Apostolidis, E. Apostolidis, E. Adamantidou, A. I. Metsai, and V. Mezaris, "A web service for video summarization," in *ACM International Conference on Interactive Media Experiences*, 2020, pp. 148–153.

[150] A. Deliege, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, and M. Van Droogenbroeck, "Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4508–4519.

[151] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[152] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[153] M. Sanabria, F. Precioso, and T. Menguy, "Profiling actions for sport video summarization: An attention signal analysis," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[154] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[155] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2227–2231.

[156] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[157] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, "Multi-camera multi-player tracking with deep player identification in sports video," *Pattern Recognition*, vol. 102, p. 107260, 2020.

[158] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[159] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little, "Identifying players in broadcast sports videos using conditional random fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, 2011, pp. 3249–3256.

[160] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1704–1716, 2013.

[161] R. Theagarajan, F. Pala, X. Zhang, and B. Bhanu, "Soccer: Who has the ball? generating visual analytics and player statistics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1749–1757.

[162] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[163] A. Coucke, M. Chlieh, T. Gisselbrecht, D. Leroy, M. Poumeyrol, and T. Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6351–6355.

[164] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B. Hoffmeister, and S. Vitaladevuni, "Multi-task learning and weighted cross-entropy for dnn-based keyword spotting." in *Interspeech*, vol. 9, 2016, pp. 760–764.

[165] A. H. Michaely, X. Zhang, G. Simko, C. Parada, and P. Aleksic, "Keyword spotting for google assistant using contextual speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 272–278.

[166] K. Tan and D. Wang, "A convolutional recurrent neural network for real-time speech enhancement." in *Interspeech*, 2018, pp. 3229–3233.

[167] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," *arXiv preprint arXiv:1703.05390*, 2017.

[168] C. Shan, J. Zhang, Y. Wang, and L. Xie, "Attention-based end-to-end models for small-footprint keyword spotting," *arXiv preprint arXiv:1803.10916*, 2018.

[169] G. D. Cabrera, "Nvidia rtx voice: Setup guide," Apr 2020. [Online]. Available: https://www.nvidia.com/en-us/geforce/guides/nvidia-rtx-voice-setup-guide/

[170] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[171] H. Zhao, S. Zarar, I. Tashev, and C.-H. Lee, "Convolutional-recurrent neural networks for speech enhancement," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2401–2405.

[172] J. Bae and D.-S. Kim, "End-to-end speech command recognition with capsule network." in *INTERSPEECH*, 2018, pp. 776–780.

[173] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[174] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep video hashing," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1209–1219, 2016.

[175] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 423–432.

[176] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and Y. Kompatsiaris, "Near-duplicate video retrieval with deep metric learning," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 347–356.

[177] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 218–227.

[178] K. Liao, H. Lei, Y. Zheng, G. Lin, C. Cao, M. Zhang, and J. Ding, "Ir feature embedded bof indexing method for near-duplicate video retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 12, pp. 3743–3753, 2018.

[179] Y. Cai, L. Yang, W. Ping, F. Wang, T. Mei, X.-S. Hua, and S. Li, "Million-scale near-duplicate video retrieval system," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 837–838.

[180] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, "Visil: Fine-grained spatio-temporal video similarity learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6351–6360.

[181] P. Avgoustinakis, G. Kordopatis-Zilos, S. Papadopoulos, A. L. Symeonidis, and I. Kompatsiaris, "Audio-based near-duplicate video retrieval with audio similarity learning," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5828–5835.

[182] A. Wang *et al.*, "An industrial strength audio search algorithm." in *Ismir*, vol. 2003. Citeseer, 2003, pp. 7–13.

[183] M. Einfalt, C. Dampeyrou, D. Zecha, and R. Lienhart, "Frame-level event detection in athletics videos with pose-based convolutional sequence networks," in *Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports*, 2019, pp. 42–50.

[184] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[185] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.