# TOWARDS AUTOMATIC REFEREEING SYSTEMS THROUGH DEEP EVENT DETECTION IN SOCCER GAME VIDEOS VERSION 2

by

Chunbo Song

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer and Information Science

2021 Fall

**TOWARDS AUTOMATIC REFEREEING SYSTEMS THROUGH DEEP EVENT DETECTION IN SOCCER GAME VIDEOS VERSION 2**

by

Chunbo Song

Approved: _____

Kathleen F. McCoy, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: _____

Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____

Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Christopher E. Rasmussen, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chandra Kambhamettu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Li Liao, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Paul Huang, Ph.D.
Member of dissertation committee

## ACKNOWLEDGEMENTS

My achievements wouldn't have been possible without the support systems provided by UD throughout the years. I would like to thank my advisor, Professor Christopher Rasmussen, for giving me the exciting research opportunities during the past few years. He has always been inspiring in guiding me through various academic spheres. He put enormous efforts into giving constructive suggestions that betters my work and me as a researcher.

Thanks to Professor Chandra Kambhamettu, Professor Li Liao, and Professor Paul Huang for serving on my dissertation committee and giving much appreciated and valuable comments.

I also want to thank all my friends, Jiefu Li, Gongbo Zhang, Jiayi Zhao, Pengyuan Li, Sanhu Li, Yuqi Kong, Jianwei Wu, Zhuo Li and Fanchao Meng, who enlighten me academically and personally during our memorable time together in UD. Last but not least, thank my family's company and support I feel lucky to always have.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## ABSTRACT

In this decade, emerging technologies such as deep learning have become crucial in video analysis to understanding the action and event caused by human interactions. Rapidly and precisely detecting/recognizing events and participants is an important and challenging problem in various areas, among which sports – accurate and timely judgements are expected by all games . This dissertation aims at developing new systems of action and event detection in soccer games and making progress towards automatic refereeing systems.

Firstly, we propose an approach for detecting events in untrimmed soccer game videos. The game videos are captured by multiple fixed cameras and do not contain shot boundaries. To obtain more precise results, we propose a network built upon inflated 3D (I3D) ConvNets for video action recognition to detect and differentiate these events, and two novel grouping methods for localizing the boundaries of events. Comprehensive evaluations indicate that our approach achieves fairly good performance.

Secondly, based on the annotated foul participants on the static frames at the foul moment, we show our detection experiments for identifying foul subjects and objects. The detection experiments compare the popular object detector (Faster R-CNN) with training from scratch with a state-of-the-art pedestrian detector (Pedestron) fine-tuned on a pedestrian dataset. An investigation is launched to demonstrate that the predictions can be affected by different non-maximum suppression approaches (NMS and soft-NMS) for post-processing. These detection experiments' results show satisfactory performance of detecting foul subjects and objects.

Furthermore, we detect foul participants and identify foul subjects and objects on video clips in a cluttered visual environment. Our system can differentiate foul participants from bystanders with high accuracy and localize them in a wide range

of game situations. We also report reasonable accuracy for distinguishing the player committing the foul, or subject, from the object of the infraction. We also experiment camera calibration and clustering approaches on filtered on-the-field persons' torsos to differentiate them by colors. Quantitative analysis showed that the clustering approach achieves good performance.

Lastly, we build a neural network for sports spotting over the entire game video in an end-to-end manner by combining a state-of-the-art action recognition network (SlowFast) into the temporal action spots detection network. Instead of extracting and storing all video features for the temporal detection, we do end-to-end training and inference. This architecture would be more efficient for practical applications as it reduces the otherwise multiple steps of training and inference. By enhancing and organizing spatial and temporal contextual information extracted by action recognition parts on short video clips, future work of modifications on the neural network architecture can be done to improve the detection performance. We also would like to investigate attention-based methods for learning from the temporal distribution of action semantics to find significant features for action spotting in soccer game videos.

# Chapter 1

# INTRODUCTION

Computer vision techniques have been widely applied to people's daily life. As parts of the main topics in computer vision, event detection, multiple-object tracking, and recognizing semantic objects of certain classes have been applied in many areas, including video analysis, video summary, person detection, etc. Sport is a great source of human actions and more and more people pay their attention to this area. From 2009, more and more large video dataset have been published, such as Hollywood2, HMDB51, UCF101, Sports-1M, ActivityNet, Charades, Kinetics, AVA, MovieNet and SoccerNet [97, 73, 124, 69, 11, 121, 70, 52, 65, 44]. Thanks to these dataset, deep neural networks are able to be applied in various areas and make understanding video's semantic context possible. And since 2017, deep network based models for videos have sprang up just like mushrooms. With the improvements and the wide deployments of computer vision techniques in videos, the application in sport-related videos have been hot topics that are not only for general video analysis purpose but also in more specific areas like sports game video assistant referee.

In literature, many solutions have been proposed to recognize events in general videos. An event represents a state that is a collection of actions performed among different agents [61]. Different types of information including visual, audio, text and other information from sensors have been widely utilized. In videos, deep learning based approaches have been the main stream solutions to exploring human interaction in complex scenes, such as CNN 3D based methods [68, 14], two-stream based methods [122, 145], attention-based methods [87, 86], etc.

The term event is interchangeably used with action in many scenarios, especially for trimmed videos. Typical human action recognition and detection have been in the

spotlight for more than ten years. After deep neural networks are studied in almost all places in this area, there are two main streams regarding their approaches: 1). Either CNN 2D or CNN 3D based is applied on frames (images) and temporal optical flow images. The classic approach for recognizing person's action [145] combined two CNNs for extracting appearance from static frames (images) and optical flow images, respectively. Based on this, some researchers proposed many methods to improve the performance, such as combining Fisher Vector encoding of Dense Trajectory Descriptors [144], extracting short snippets over a long video sequence [143], incorporating human/object detection results into leverage semantic cues [145], fusing spatial and temporal networks at convolutional layers [35], etc. 3D convolutional neural networks extract features from both the spatial and the temporal dimensions by performing 3D convolutions, then capture the motion information encoded in multiple adjacent frames [68]. After that, [28] proposed a bilinear model to pool together the outputs of the last convolutional layers of the pre-trained networks. 2). Some researchers combined RNN and CNN for directly classifying sequences without any segmentation. [5] trained RNN to classify each sequence considering the temporal evolution of the learned features for each timestep. [51] demonstrated that LSTM's performance remains robust even as experimental conditions deteriorate. [30] introduced an end-to-end hierarchical RNN for skeleton based action recognition.

With the improvement of analysis in general videos, more and more sports benefit from technologies in video analysis in not only training but also gaming. In training, it can help coaches and athletes to share the same view together for improving performance. In game, referees can double-check the judgement, coaches can arrange tactics accordingly. TV directors also benefit from it since richer contents can be easily rendered to audiences. The literature of sport-related video analysis is also vast. Its applications include broadcast enhancement; real-time, in-depth player and team performance measurement; and automatic summarization of key events. Behind these analysis tasks, several common visual skills are widely utilized, such as ball tracking; player segmentation, recognition, and pose estimation; and recognition of formations,

players, and situations. As one of the earliest models for detecting highlights or events, [4] presents a system for automatic annotation of the principal highlights in soccer videos.

To automatically detect different states in soccer games, Xie *et al.* present statistical techniques [153] to identify two mutually exclusive states *play* and *break*. [83] proposes a framework combining temporal action localization and *play* and *break* rules for soccer video event detection. Beyond the *play* and *break*, more semantic visual events attract many studies to recognize and localize across the game video. For the nature of broadcasting videos, more kinds of semantic events are explored in [44, 26, 19, 45] With the huge demands for annotated video data for object segmentation, tracking and interactions, [76] in 2009 proposes a semi-automatic system that generates an initial ground truth estimation. Efforts have also been made for ball/players *detection* and *tracking*. [132] proposes a ball detection and tracking approach in real soccer game. They apply an indirect non-ball elimination strategy in view of difficulties of direction detection. [89] makes use of cascaded Convolutional Neural Network (CNN) to detect player locations from whole images.

To map the broadcasting frame to the world coordinate, [8] proposes an approach to extract the frame line mask and tune the initial homography. [116] investigates the benefit of employing self-attention on the spatio-temporal embeddings extracted from ball and players trajectories as well as bounding boxes around the players to detect group activity in soccer games. Besides the visual information, some studies [104] combine the body-sensors for feature extraction, object tracking and background subtraction.

Practically, as Fig. 1.2 and Fig. 1.3 depicted, in professional soccer games, novel computer vision techniques have been accepted and used for helping referees to make decisions. Video Assistant Referees (VAR) [36] has been officially applied since 2018 FIFA World Cup and promoted to many popular top-level professional leagues. There are three main incidents (plus one administrative) being identified as game-changing: goals, penalty decisions, direct red card incidents and mistaken identity. A VAR review

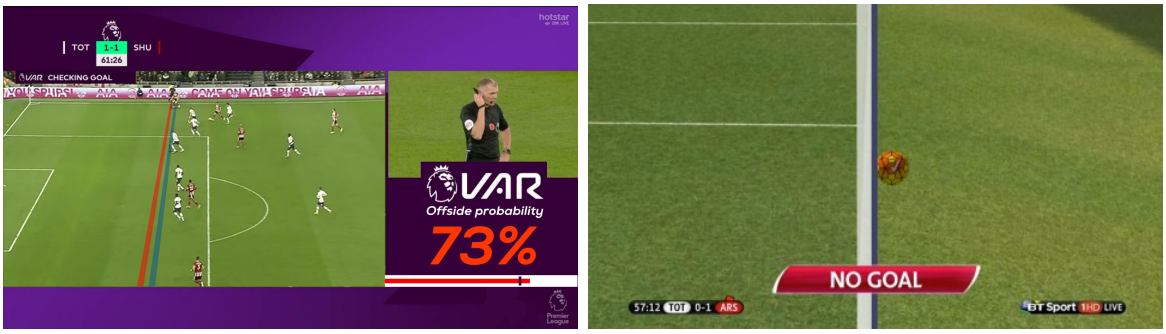


**Figure 1.1:** Left: example of applying VAR in Premier League for checking possible offside. When there's a possible offside, typically, 3 best frames are manually selected for representing the point at which the attacking player made first contact with the ball. Right: Goal-line technology assists referees and assistant referees in making crucial decisions where there is doubt as to whether or not the ball has crossed the line.

allows the referee on the field to get advice from one or more referees who can analyse and replay video images of the situation at question.

Indeed, VAR is still a more manually labored system, rather than an autonomous and automatic one. It needs assistant referees in the video operation room to watch the live action on the field while the VAR is undertaking a "check" or a "review", to keep a record of reviewable incidents, and to communicate the outcome of a review to broadcasters, as showed in the left of Fig. 1.1. The goal-line technology (as showed in the right of Fig. 1.1) uses electronic devices to determine if a goal has been scored. Despite of making progress on assisting referees, more efficient and accurate systems need to be proposed.

In this dissertation, we start from detecting *play/break* and *break* types in the multi-camera scenario. We manually annotate the event types of *play/break* and all *breaks* across the 2 complete soccer games from 3 fixed cameras. In this multi-camera scenario, we propose a combination of two grouping methods using the confidence score output by the 3D deep network. Next, we move to detecting players who are involved in a foul. At this task, we firstly explore its possibility in single static images. Since


**Figure 1.2:** An on-field review (OFR) can only be conducted on the recommendation of the VAR. Slow motion replays are only used to establish point of contact for physical offences and handball, while full-speed replays are shown to determine the intensity of an offence or whether a handball occurred in the first place. During an OFR, the VAR can transmit several video replays from different camera angles and positions to allow the referee to make their decision.

temporal dimension have more information than static images, we make an assumption that foul participants and bystanders can be more easily differentiated given foul moments. Furthermore, the foul subject and object can be also identified. We utilize a multi-object tracking method on the given video clips to generate players' tracks. Then the post-processed tracks are the consecutive sequences as the input of the deep network. The occlusion among players is still an obstacle of identifying the foul subject and object. We use camera calibration to generate top-view of the field for estimating distance among players and their positions on the field. We also extract players' torsos


**Figure 1.3:** A view of the video assistant refereeing operation room at the 2018 Fifa World Cup international broadcast centre in Moscow. Assistant video assistant referee (AVAR) is appointed to assist the VAR in the video operation room. The responsibilities of the AVAR include watching the live action on the field while the VAR is undertaking a 'check' or a 'review', to keep a record of reviewable incidents, and to communicate the outcome of a review to broadcasters. Photo: AFP

to get the torsos for differentiating their teams based on their jersey colors. We also experiment action spotting for localizing different actions/events in entire game videos. An overview of the topics covered in this dissertation can be found at Fig. 1.4.

## 1.1 Statement

This dissertation focuses on exploring soccer game videos analysis based on state of the art deep neural networks, especially for detecting the foul event across the game video and identifying participants of fouls on either fixed and multi-camera videos or broadcasting videos dataset, progressing towards automatic video refereeing system. We also detect foul subjects and objects on static images at an instant moment.

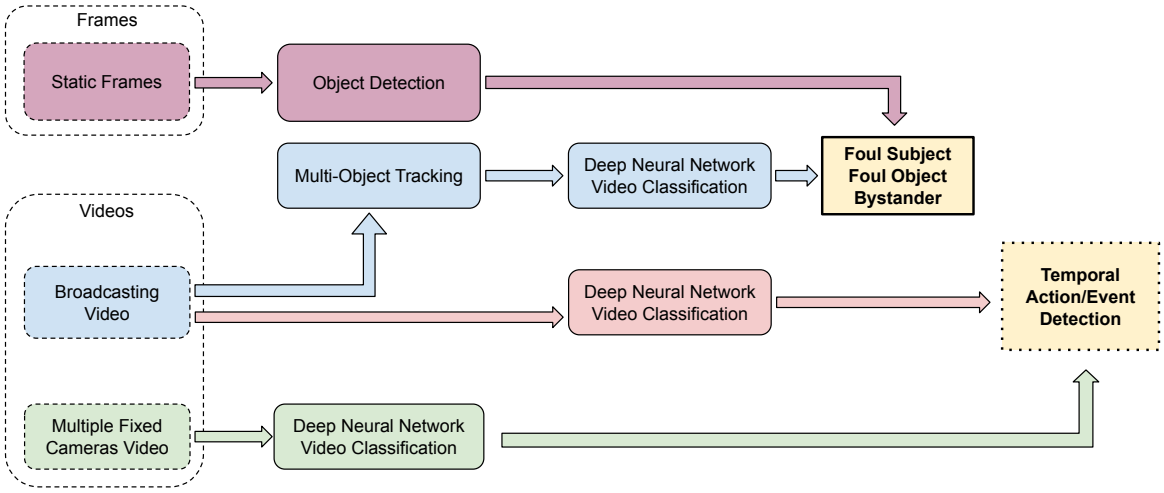


**Figure 1.4:** Overview of topics covered in this dissertation. Our work shows the development towards event detection and recognition. Each different colored block with solid lines shows the research work of a chapter in this dissertation.

## 1.2 Contributions

**Event Detection in Multi-Camera Scenario** We firstly annotate 'play', 'break' and different 'break' types on multi-camera soccer game videos. Then, we present a fusion model for detecting events in multi-camera scenario. The model combines three towers of deep 3D networks with consecutive frames fed from three fixed-position cameras. Towers of deep networks in this model share weights with each other.

**Novel temporal grouping methods for play/break event detection** We have presented two separate grouping methods for refining the bounding of *play/break* and *break* types in the temporal dimension. These two can also be combined together to further improve mean Average Precision (mAP). The proposed two methods take the confidence scores output from the deep network model and adjust the temporal boundary in an effective way.

**Annotation of Foul Participants at Foul Moments** We extract and annotate fouls in 442 SoccerNet games that have text transcripts of audio commentary on game events which are timestamped by half game clock with one-second precision. The fouls

are roughly located by searching all transcripts. Furthermore, we annotate tracks for foul subjects, objects and other bystanders in complete 50-frame of parts of the foul clips in the 442 games.

**Detection of Foul Subjects, Objects and Bystanders** We present a system of differentiation of foul participants from bystanders with high accuracy and localization of them over a wide range of game situations. We also report reasonable accuracy for distinguishing the player who committed the foul, or subject, from the object of the infraction, despite very low-resolution images.

**Differentiation of Players by Clustering** We firstly utilize players' skeletons to generate their torsos' masks from every single frame. And we apply an unsupervised learning method to differentiate them by their corresponding color histograms. The differentiation is important for further figuring out who is the foul subject and who is the foul object.

**End-to-End Temporal Action Spots Detection on Broadcasting Soccer Game Videos** We design an end-to-end model for detecting action spots over the entire game video. Despite of unsatisfactory performance, we strongly believe its value cannot be ignored since it provides thoughts to handle the problem of action detection in long duration videos. We also provide possible solutions to achieve desirable performance.

### 1.3 Blueprint of the Dissertation

The rest of this dissertation is organized as follows.

Chapter 2 discusses a temporal grouping approach for *play/break* and *play* and *break types* detection in fixed multi-camera soccer game videos, as showed in Fig. 2.1. The proposed approach is based on an efficient grouping mechanism on the confidence scores output by a trained deep neural network. We also make our annotations on the Soccer Video and Player Position Dataset (SVPP) for abstract categories.

In Chapter 3, based on SoccerNet V1 dataset we establish our benchmark on detecting foul subject and object on static images that are captured from videos at the foul moment, as showed in Fig. 1.5. We experiment with state-of-the-art object

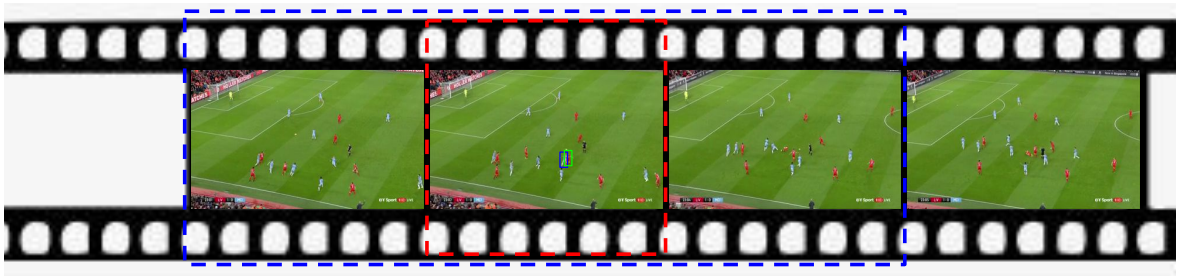

**Figure 1.5:** The view of SoccerNet dataset. In Chapter 3, we focus on detecting foul subjects (solid blue box) and objects (solid green box) on static frames at the foul moment (dash red box). In Chapter 4, we move to detecting foul subjects and objects on video clips that are anchored at the foul moment frames (dash blue box).

detectors on our task. And we compare the post-processing methods by measuring their performance in our cluttered environments.

In Chapter 4, we move from 2D to the 3D domain for detecting players involved in fouls in broadcasting soccer game videos, as showed in Fig. 1.5. And we measure the performance of multi-object tracking on trimmed videos. Furthermore, who made the foul and who is fouled in the video is also investigated. In this task, we also create a novel benchmark based on SoccerNet V1 dataset for tracking players and identifying their actions in fouls. Furthermore, we propose to differentiate players by jersey colors. To get rid of the influence from persons detected out of the field, we apply a camera calibration method to estimate camera positions and overlay the frame onto the field template for extracting the region of the field. Then we do clustering on the detected torsos for the differentiation. To evaluate the clustering performance, we annotate torso colors of the detected persons from a few subsets of video clips.

In Chapter 5, thanks to the success in the action recognition model and temporal action spots detection model, we propose to construct an end-to-end model for detecting action spots over the soccer game video. Despite of far below expectation results, we strongly believe that we are on the track of achieving good performance for the temporal action spots detection. In this chapter, we will discuss the task and make

analysis.

Chapter 6 draws a conclusion to this dissertation and points out a few potential future directions.

# Chapter 2

# MULTI-CAMERA TEMPORAL GROUPING FOR PLAY/BREAK EVENT DETECTION IN SOCCER GAMES

In this chapter, we build upon inflated 3D (I3D) ConvNets for video action recognition to detect and differentiate six abstract categories of events in untrimmed videos of soccer games from multiple fixed cameras: normal play, plus breaks in play due to kick-offs, free kicks, throw-ins, and goal and corner kicks. Raw video unit classifications by variants of the basic I3D network are post-processed by two novel and efficient grouping methods for localizing the boundaries of events. Our experiments show that the proposed methods can achieve 84.2% weighted precision for event categories at the level of video units, and boost event temporal localization mean average precision at 0.5 tIoU (mAP@0.5) to 62.0%.

## 2.1 Introduction

Computer vision is fast becoming a powerful tool for sports video analysis. All kinds of vision-based tasks traditionally performed by the players themselves, spectators, referees, camera operators, and expert commentators can potentially be automated or enhanced for a myriad of applications. These include training and coaching feedback, enhanced rule enforcement accuracy, replay annotation and explanation for broadcasters, measuring detailed player and team statistics, and even serving as perception modules for robotic sports participants. While the exact purpose of the analysis may vary, as well as the sensors employed, there are certain visual skills such as ball tracking [132, 96], player segmentation [13, 89, 66], recognition [43], and pose estimation [71], and recognition of formations, plays, and situations [4, 135, 137, 44] that many sports vision systems have in common.

**Figure 2.1:** The views from fixed multi-cameras

One of the most basic forms of sports video understanding, at a high level, is *play/break* categories classification [153, 32, 125]. That is, can one infer whether a particular video sequence depicting part of a game is showing actual game *play*, or is there a *break* in the action? We follow the event definition introduced by Giancola, *et al.* [44] to represent *play/break*s in videos of soccer games, who defined an event as an action that is anchored in a single time instance, defined within a specific context respecting a specific set of rules. Distinguishing between these two game states is not trivial, because during breaks the players (as well as the ball) may still be visible, and still moving. Events like shots, passes, and fouls that occur in the course of play are understandably popular subjects of study for game analysis [125, 13, 135]. However, here we investigate break events, which may be due to a timeout, a foul, halftime, an injury, a ball out of bounds, or any number of sports-specific events. In particular, we study soccer *break* event categories as defined in the FIFA rule book [37]: (1) kick-offs (to start each half or after a goal), (2) free kicks (after a foul), (3) penalty kicks, (4) throw-ins (touch line out of bounds), (5) goal kicks (end line out of bounds caused by offensive team), (6) corner kicks (end line out of bounds caused by defensive team), and (7) dropped balls (all other situations), detecting these *break* event segments in the soccer game video is a difficult task due to the sparsity within a video, but also they have different duration.

Rather than recognizing events or actions in the long untrimmed video either from one camera or from a broadcast feed (in this case, a video contains camera panning

and zooming, shot boundaries and subjects and scenes selected of the action), in this paper we aim to differentiate and localize *play* and *break* events using the Soccer Video and Player Position Dataset (SVPP) [104] which has two complete soccer games from three fixed cameras, like Fig. 2.2 shows. This dataset doesn't have event categories, we manually annotate them in a frame level. Therefore, the event segment can be extracted.

We first consider the Two-Stream Inflated 3D ConvNet (I3D) [14] trained on three cameras be the one worthy for the comparison since it is one of state-of-the-art architectures. The I3D, which takes several seconds of video context or a sequence of frames in a fixed length (which we call video unit for differentiating with the event segment), is able to recognize play and different break categories fairly reliably. Because of multi-camera, an assistant neural network (AN) is then utilized to combine all I3D's predictions on synchronous units from all cameras. We also extend I3D to our C-I3Ds by integrating observations from multiple cameras, even those not directly viewing the action, are able to boost performance non-trivially. In C-I3Ds, each camera corresponds to one I3D with two-stream (RGBs and Optical Flows). The integration of these I3Ds takes synchronous video units from all cameras as inputs. Outputs are combined to generate predictions.

Here is an assumption: if a classifier performs well with unit inputs, boundaries will be localized easily and efficiently. Unlike recent methods [38, 39, 171] feeding by trained deep features for localizing actions or generating action proposals in untrimmed videos, we propose two efficient methods to group adjacent video units for the event localization: probability-based grouping (PBG) and class-based grouping (CBG). Both grouping methods build upon predicted probabilities and classes by our I3D-based model. They and their combination achieve promising performance on our testing.

In summary, our contribution are three-fold: (1) We extend the I3D network to be suitable for the multi-camera case to classify video units. (2) We propose probability-based and class-based grouping methods to facilitate C-I3Ds for event localization. (3) The combination of both grouping methods boosts performance on both

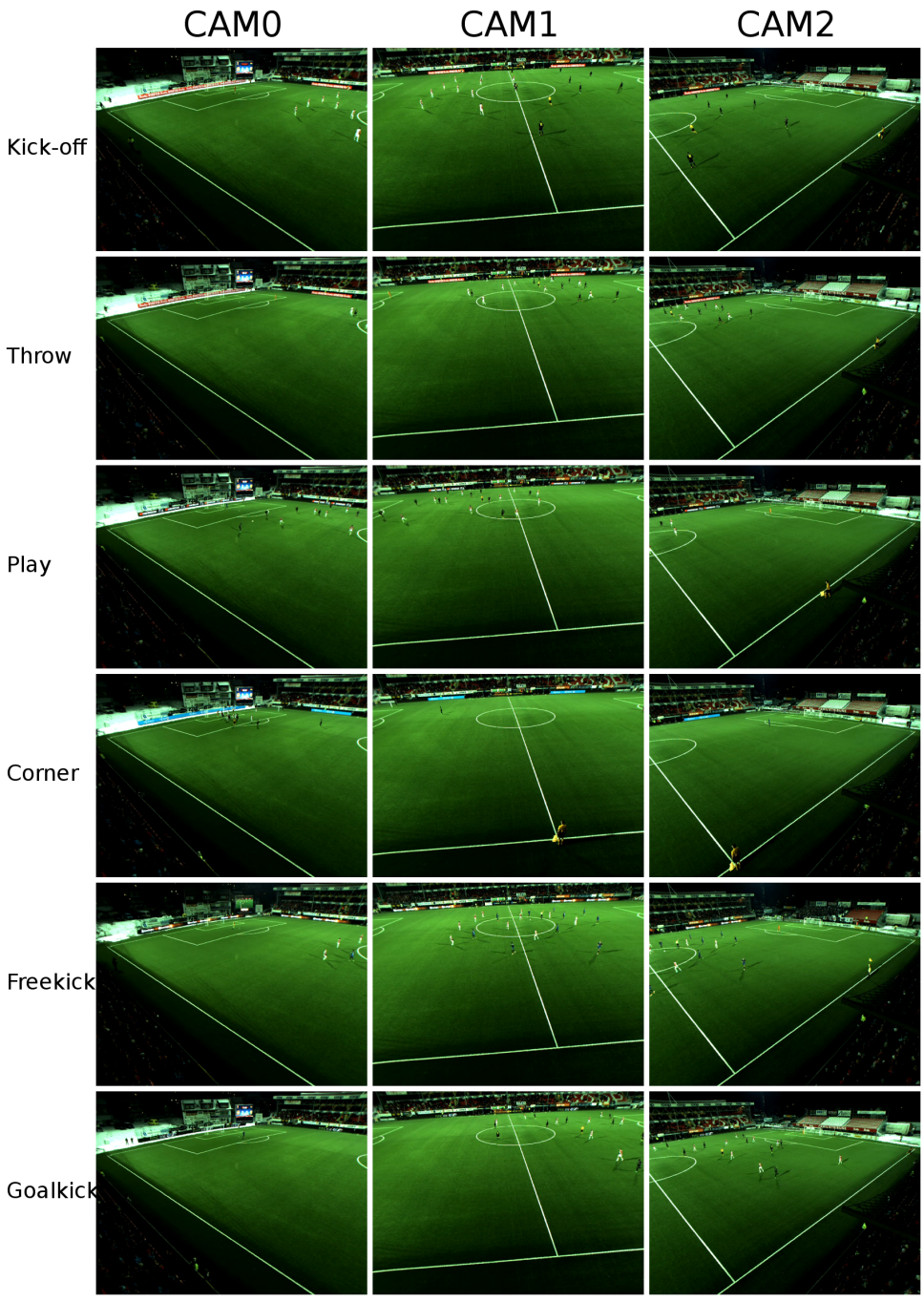


**Figure 2.2:** Example frames of all events defined in the rule of soccer games. In the SVPP dataset, three fixed cameras capture different regions of the field.

classification and localization during testing.

## 2.2 Related Work

Deep learning architectures for video classification and action recognition in videos have also shown great promise recently [69, 138, 141], including LSTM networks for human action classification [51] and recognizing pass, shoot, dribble actions from multi-camera video with player and ball trajectories [135]. Strategies for fusing optical flow with spatial information have also achieved considerable success [122, 14, 143, 145], as well as 3D convolutional neural networks which extract features from the spatial and the temporal domains jointly by performing 3D convolutions to capture the motion information encoded in multiple adjacent frames [68]. Based on an Inception module [127], I3D expands 2D filters and pooling kernels to 3D to make it possible to learn seamless spatio-temporal features from video and applied on two-stream (RGB and Optical Flow). The optical flow input may provide some sense of recurrence [14]. The I3D network trained on optical flows carries optimized, smooth flow information. Experimentally it is valuable to classify actions. After pre-training on the Kinetics dataset [24], I3D models have reached 80.9% on HMDB-51 and 98.0% on UCF-101 [14, 124, 73] which is the most state-of-the-art method to our best knowledge.

Despite these advances, localizing action boundaries in a long, untrimmed video is still a difficult problem. Applying temporal sliding window is a typical scheme after classification [101, 142]. The feature extracted from deep neural networks is globally pooled within each window for generating SVM inputs. Yuan *et al.* [162] proposed an approach to address the uncertainty of action occurrence and utilization of information from different scales. Although these works have shown promising performance in their task, the efficiency is still unresolved. Many recent methods have examined this problem as analogous to object detection but in the temporal dimension, they utilize features from deep neural networks to localize action boundaries, including temporal action proposals [39, 38, 15].

In the work which is similar with ours, Giancola, *et al.*[44] try to "spot" three soccer event categories: (*goal, card*, and *substitution*). However, they didn't try to identify the boundaries of an action within a video, but simply the anchor time that identifies an event with one-minute resolution.

Some other soccer datasets include ISSIA [76], which contains player, referee, and ball positions as seen from multiple fixed cameras; and SoccerNet [44]. But, ISSIA is very short – only 2-minute sequences, and while SoccerNet is huge (764 hours of video), it only contains very sparse yellow/red card, goal, and substitution events at essentially 1-minute label resolution. AZADI [8] has play/break labels and Soccer 152-A [83] has a number of actions, including those of referees, coaches, and spectators, but neither of these could be obtained for this work.

### 2.3 Dataset and Annotation

The Soccer Video and Player Position Dataset [104] (SVPP) is used in our work. The portion of the dataset that we use consists of two complete soccer game videos captured at 30 fps by three fixed cameras whose overlapping fields of view each roughly cover one-third of the length of the field. These two games are TromsoIL vs. Anzhi ($TvA$) and TromsoIL vs. Stromsgodset ($TvS$). The original resolution of each frame in the video is $1280 \times 960$. The video of the games are untrimmed, and no broadcast content. 324,284 frames of each camera are annotated with *play* occupying about 65.9% and *break* 34.1%. There are no instances of penalty kicks or drop balls in the videos, so we remove these two break categories. Of the break frames, 0.4% are kick-off (only at the beginning of the game or after the half, as there are no goals), 32.4% free kick, 24.4% throw-in, 14.7% corner kick, and 28.1% goal kick. And different event categories have various time duration. Examples of annotated frames are illustrated at Fig. 2.2.

## 2.4 Methods

### 2.4.1 Classification

#### 2.4.1.1 I3D, Assistant Neural Network (AN) and C-I3Ds

Because deep neural networks have displayed good ability of generalization [167, 100], we firstly train one I3D network on units from all cameras. And we assign one trained I3D model to a related camera during testing. It implies that different I3D networks share weights with each other. Thus, synchronous units from different cameras are sent to their corresponding I3D networks. Their outputs (confidence scores or logits) are concatenated to feed into AN, which is, in our work, a fully-connected network for outputting event classification results by combining confidence scores from different cameras' related I3D models.

However, in the multi-camera case, both machine and human may be error-prone on pointing out the event when some cameras are unavailable. Training one I3D network on units from all cameras may result in bad recognition. Like the right frame at the third row showed in Fig. 2.2, people cannot tell the exact event, because most players are in the penalty area in the other side for a corner. Therefore, deploying several I3D networks for different cameras on training is an alternative way. Unlike the previous way we used, these I3D networks don't share weights with each other. Fig. 2.3 shows its architecture. Each pair of two-stream I3D networks corresponds to a camera. And the output of these separate I3D networks are combined lately, without applying AN. We call this C-I3Ds. Because synchronous video units have the same categories, we train these separate I3D networks jointly and average their predictions at both training and testing time.

### 2.4.2 Event Boundaries Localization

#### 2.4.2.1 Probability-based grouping (PBG)

A temporal sequence of predicted probabilities may indicate the transition from one state to another. Ideally, such transition would be smooth and precise. But, in videos, classifiers may not always achieve perfect results due to several reasons such

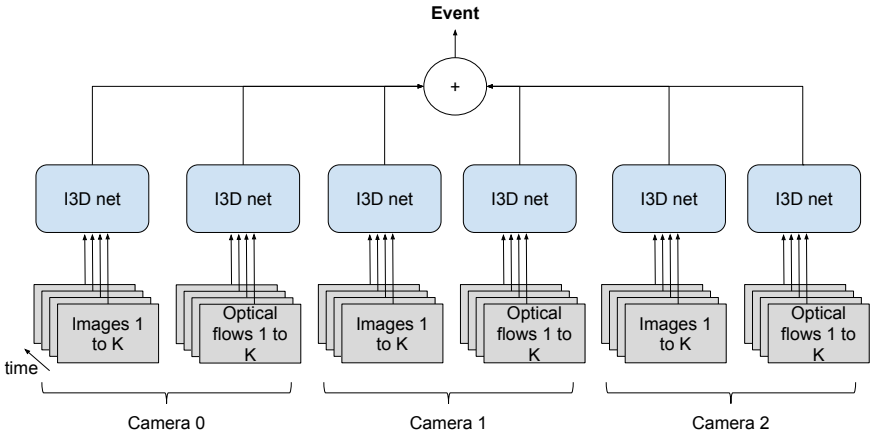


**Figure 2.3:** C-I3Ds: the combination for multiple two-stream I3D framework.

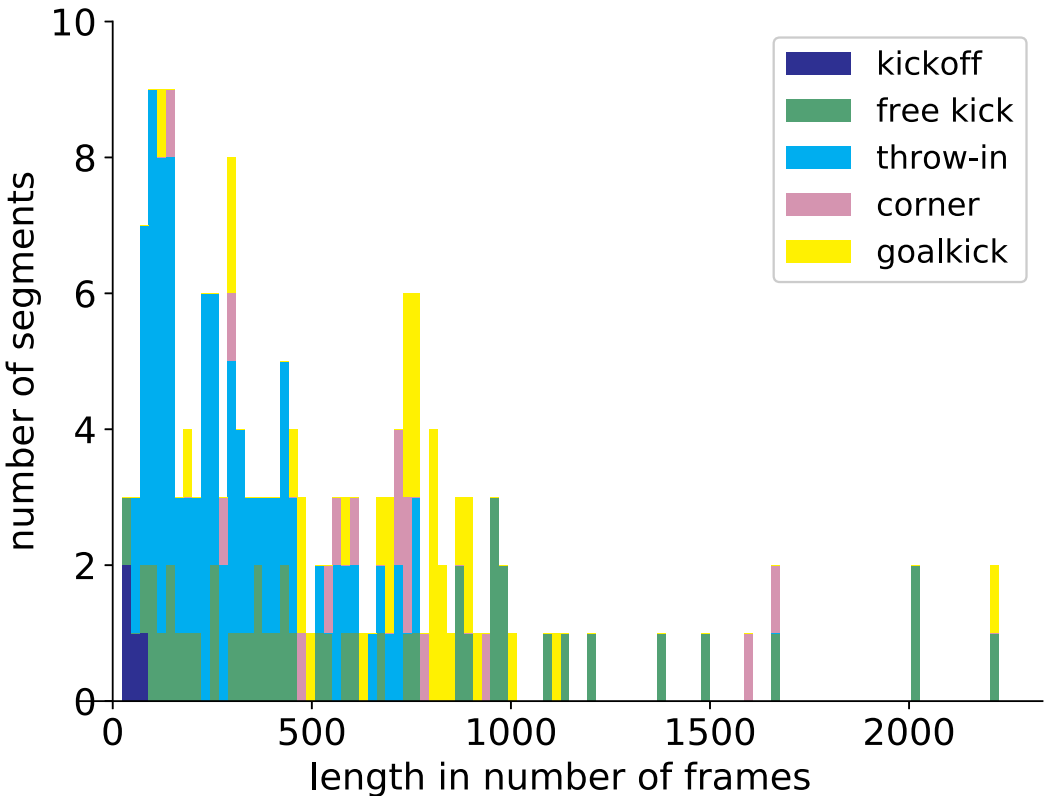


**Figure 2.4:** The distribution of different *break* categories in length on our training set.

as subjective labelling, restrictions of the classifier, limited data and etc. Using good classifiers, false classification on frames/units still commonly exists, and thus makes localization difficult. To address this problem, we apply a sliding window manner on predicted probabilities from deep neural networks to not only filter out some errors, but also can group adjacent segments together. The probability-based grouping has two steps: actionness scores grouping and break categories assigning. Fig. 2.5 illustrates the full pipeline of PBG.

We extend the definition of actionness scores in [171] and use it to describe the probability of a given video unit is a break event. For an unit $k$, we get its actionness score by $prob_{k,a} = 1 - prob_{k,p}$, where $prob_{k,a}$ is the actionness score and $prob_{k,p}$ is the probability of 'play' of $k$. If $\forall k \in [i, j], prob_{k,a} \geq t_a$, we will be able to get a class-agnostic segment $S_{i,j}$, $t_a$ is a threshold of differentiating 'break' with 'play', and $i$, $j$ are the boundary of a segment. Based on observations, the beginning of any break event is usually very similar with 'play' and the beginning of any play is also very similar with its previous neighbor 'break'. Therefore, for each break segment $S_{i,j}$, we utilize $l_{aw} - 1$ units before $i$. And apply a mean window $W_a$ with size $l_{aw}$ and stride $st$ from $i - l_{aw} + 1$ to $j - l_{aw} + 1$. It will adjust $S_{i,j}$ to $S_{p,j}$ given $t_a$. After that, an adjusted segment may overlap or be too close with its neighbor segments. We collect these averaged actionness scores from $m$ to $n$ and apply another mean window $W_{sep}$ to determine if separate or group them. We define $l_{sep}$ be the size of $W_{sep}$. $l_{sep}$ also implies the minimum length (i.e. if the distance between two adjacent segments is less than $l_{sep}$, we think they are too close). We densely slide $W_{sep}$ across $[m, n]$ with stride 1, and compare every scores with a threshold $t_{ma}$. If the number of consecutive steps for $W_{sep}$ is more than $l_{sep}$ and the mean scores are less than $t_{ma}$, then we separate them. It is worth noticing that we shrink the size of $W_{sep}$ if $n - t + 1 < l_{sep}$ for outputting $n - m + 1$ mean scores, where $t$ is the index of the current unit.

For assigning break categories, we average probabilities of all categories for all units within the refined segment, denoted by $P_{i',j'}$. Because, for each category $c$, the shortest and the longest lengths $G_{c,short}$ and $G_{c,long}$ can be obtained from the training

set, we iteratively check and assign the most possible category to the segment based on the its length $l_{i',j'}$, if $l_{i',j'} \geq G_{c,short}$ and $l_{i',j'} \leq G_{c,long}$. If no category can satisfy this segment, 'play' will be assigned to it.

#### 2.4.2.2 Class-based Grouping (CBG)

The drawback of PBG is, $l_{aw}$ and $l_{sep}$ might not be very large because large window size will eliminate some short but true predictions. Therefore, many false positives are retained. Based on the rule of professional soccer games, we observed facts that any break category must start at the end of 'play', rather than other break categories, except 'kickoff'. And, any break category will usually not takes too short, similar as what we mentioned in assigning break categories in PBG. So based on these facts, we utilize predicted classes to further adjust both boundaries and categories. For each input segment $S_{i,j}$ (including 'play') with length is $l_{i,j}$, its two neighbor segments $S_{x,i-1}$ and $S_{j+1,y}$ are extracted if $l_{i,j} < t_{len}$, where $t_{len}$ is a threshold for indicating small segments. Then, group $S_{j+1,y}$ with $S_{i,j}$ and assign its category to $S_{i,j}$ if $l_{x,i-1} < l_{j+1,y}$. Otherwise, combine $S_{x,i-1}$ to it. This step is processed iteratively until all lengths are greater than $t_{len}$. After that, if any adjacent segment all belong to any 'break' category (except 'kickoff'), we merge the short segment with the adjacent longer one and assign the category to it.

## 2.5 Experiment and Analysis

### 2.5.1 Data Preparation

We randomly extract synchronous video units from three halves' videos and all cameras to generate the training set. The three halves are: the 2nd half of *TvA* and the 1st and 2nd half of *TvS*. In our work, each video unit has 64 frames with 1 frame of unit's stride. Fig. 2.4 displays the distribution of length of event segments in different break categories in our training set. We assign the label of the last frame in an unit to be the category of this video unit. Due to highly imbalanced number of categories in our dataset, we over-sample video units which are break categories. Thus, for

each category (include play), 9,000 synchronous video units from 3 cameras are in the training set. Data augmentation is necessary to improve the ability of generalization of models because of limited instances of some categories. For each frame, we randomly crop with size $1160 \times 921$. Frames in the same video unit are cropped at the same place, as well as the corresponding optical flows images. These frames are re-sized to $224 \times 224$ for feeding into I3D and C-I3Ds. We also apply random right-left flipping, frames and corresponding optical flows images in the same video unit do have the same flipping direction. For the test set, we use the 1st half of $TvA$ with unit's stride 1 frame as well for both the event classification and boundary detection. There are 81,471 units in our test set.

### 2.5.2 Implementation Details of Training

We train the I3D network in an end-to-end manner, with units of video frames as the input. The optical flows are calculated by Dual $TV\_L^1$ method [163]. The I3D network is trained on randomly selected units from all cameras. For both I3D and C-I3Ds, we use SGD to learn parameters. The learning rates are set to 0.01. And dropout of both I3D and C-I3Ds is 0.5 during training. We make AN have 2 layers of 20 hidden nodes. We deploy the same I3D models to predict confidence scores on different camera units. The input of AN is the confidence score from I3D models on synchronous units. The optimization of AN is launched by Adam optimizer with learning rate 0.0001. The training iterations of both I3D and C-I3Ds are 240K, and they are all trained from scratch. The batch size is 4 because of the memory issue. AN is trained for 20K iterations with batch size 64.

### 2.5.3 Evaluation Metrics

For event classification, we calculate Precision for different event categories (include 'play'). The Weighted Precision is calculated as well for indicating the overall classification performance. For event localization, we report mean Average Precision

(mAP) and Average Recall (AR) using temporal Intersection over Union (tIoU) threshold of 0.5. Because none of the 'kickoff' units is recognized, it is not included in the analysis of the results.

### 2.5.4 Classification

Tab. 2.1 displays the precision of the different model on testing. I3D network trained on units from all cameras doesn't perform really well, even AN is applied. The C-I3Ds perform better than the I3D with AN on almost all categories, except 'free kick'. Without any grouping method, its weighted precision achieves 78.7%. Units can obtain labels after applying grouping methods with the C-I3Ds. If both $l_{aw}$ and $l_{sep}$ are 46, the weighted precision reaches 83.5%. If the CBG is applied with $t_{len}$ is 125, the weighted precision (80.9%) is lower than using PBG, but still higher than C-I3Ds'. We also combine PBG and CBG to adjust predicted categories of units and it achieves relatively good weighted precision performance (84.2%).

The classification result indicates that different levels of difficulty of these event categories. This may be caused by limited number of events in our training set, even though we over-sample frames with this category to make the training set balance. Moreover, owing to diversities, the 'free kick' is also hard to be differentiated from other categories.

**Table 2.1:** Per-unit (stride 1) classification precision (%)

| method | play | free kick | throw-in | corner | goalkick | weighted precision |
|---|---|---|---|---|---|---|
| I3D | 88.4 | 15.8 | 18.9 | 51.1 | 37.6 | 73.3 |
| I3D+AN | 85.4 | 29.4 | 31.8 | 63.4 | 44.4 | 74.0 |
| C-I3Ds without grouping | 88.7 | 16.5 | 50.5 | 66.8 | 61.2 | 78.7 |
| C-I3Ds+PBG($l_{aw}, l_{sep} = 33$) | 90.6 | 17.8 | 65.5 | 68.5 | 72.1 | 82.2 |
| C-I3Ds+PBG($l_{aw}, l_{sep} = 46$) | 91.3 | 20.3 | 71.2 | 72.6 | **73.8** | 83.5 |
| C-I3Ds+CBG($t_{len} = 65$) | 89.5 | **29.9** | 77.5 | **73.3** | **73.8** | 83.2 |
| C-I3Ds+CBG($t_{len} = 125$) | 88.7 | 18.3 | 77.3 | 58.5 | 72.6 | 80.9 |
| C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 65$) | 91.0 | 19.2 | 76.2 | 65.0 | 72.6 | 82.9 |
| C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 125$) | 91.2 | 21.3 | 76.6 | 65.0 | 73.1 | 83.3 |
| C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 65$) | 91.3 | 22.0 | **79.6** | 72.6 | 72.3 | 84.0 |
| C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 125$) | **91.6** | 27.7 | 75.7 | 72.6 | 71.6 | **84.2** |

### 2.5.5 Event Localization

We use C-I3Ds as the baseline to evaluate performance on the localization by making input video units be in the chronological order and localizing boundaries because of its decent classification performance. Tab. 2.2 and Tab. 2.3 display the precision and the recall on the event localization.

While the C-I3Ds achieves a decent performance on the classification, the result of event localization is bad. Given tIoU threshold as 0.5, the mAP is less than 1%, and AR is 14.0%. After applying PBG after C-I3Ds with $l_{aw}$ and $l_{sep}$ are 33, the mAP@0.5 and AP@0.5 have reached 33.4% and 41.9%, respectively. If $l_{aw}$ and $l_{sep}$ are all 46, the mAP@0.5 is 39.3% and the AR@0.5 is 46.5%. Because some segments are pretty short in the training set, it appears both window sizes $l_{aw}$ and $l_{sep}$ are small to maintain these correct segments as many as possible.

C-I3Ds with CBG performs well, which achieves 41.3% mAP@0.5 when setting $t_{len}$ to be around the unit size (i.e. 65). Assigning it a larger value for $t_{len}$, some short but true segments will be merged into their neighbors. When $t_{len}$ is much larger (e.g. 125), both mAP and AR will be low (30.2% and 25.6%) due to the incorrect merging. C-I3Ds with PBG achieves higher recalls than CBP (46.5% vs. 34.9%). The PBG will still leave too many short segments because of its short window sizes. In these segments, the number of false positives is far more than true positives'. And, CBG with relatively larger $t_{len}$ can be applied for eliminating them. Thus, we test the combination of these two grouping methods after C-I3Ds. The combination boosts mAP@0.5 up to 62.0% without sacrificing AR much as Tab. 2.2 and Tab. 2.3 show. Fig. 2.6 shows qualitative examples on testing. The four frames display some correct and incorrect recognition. Besides 'kickoff', 'free kick' is the most difficult category for recognition, like the first frame with the number 3237. The corresponding segment of the third frame with the number 49216 is eliminated by the grouping since C-I3Ds only predicts a few short segments. 'goalkick' is the easiest category to be detected in the testing, as the rightmost frame shows. From the Fig. 2.6, although some short segments in the ground truth are hardly detected by C-I3Ds, the predicted boundary

can be adjusted accurately by applying our grouping methods. Both PBG and CBG are efficient. Running both after C-I3Ds only spends less than 1 second on testing.

**Table 2.2:** Results for event localization in precision(%) and mAP(%)@0.5 tIoU

| method | free kick | throw-in | corner | goalkick | mAP |
|---|---|---|---|---|---|
| C-I3Ds without grouping | 0.3 | 0.0 | 0.0 | 1.0 | 0.3 |
| C-I3Ds+PBG($l_{aw}, l_{sep} = 33$) | 5.3 | 38.5 | 50.0 | 44.4 | 33.4 |
| C-I3Ds+PBG($l_{aw}, l_{sep} = 46$) | 9.4 | 41.7 | 60.0 | 56.3 | 39.3 |
| C-I3Ds+CBG($t_{len} = 65$) | 22.2 | 50.0 | 50.0 | 44.4 | 41.3 |
| C-I3Ds+CBG($t_{len} = 125$) | 33.3 | 0.0 | **66.7** | **70.0** | 30.2 |
| C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 65$) | 10.0 | 55.6 | 50.0 | 47.1 | 41.9 |
| C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 125$) | 20.0 | **83.3** | 50.0 | 53.3 | 56.7 |
| C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 65$) | 13.6 | 62.5 | 60.0 | 56.3 | 48.9 |
| C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 125$) | **37.5** | **83.3** | 60.0 | 56.3 | **62.0** |

**Table 2.3:** Results for event localization in recall(%) and AR(%)@0.5 tIoU

| method | free kick | throw-in | corner | goalkick | AR |
|---|---|---|---|---|---|
| C-I3Ds without grouping | 12.5 | 0.0 | 0.0 | 55.6 | 14.0 |
| C-I3Ds+PBG($l_{aw}, l_{sep} = 33$) | 25.0 | **27.8** | **50.0** | 88.9 | 41.9 |
| C-I3Ds+PBG($l_{aw}, l_{sep} = 46$) | **37.5** | **27.8** | **50.0** | **100.0** | **46.5** |
| C-I3Ds+CBG($t_{len} = 65$) | 25.0 | 11.1 | **50.0** | 88.9 | 34.9 |
| C-I3Ds+CBG($t_{len} = 125$) | 25.0 | 0.0 | 33.3 | 77.8 | 25.6 |
| C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 65$) | 25.0 | **27.8** | 33.3 | 88.9 | 39.5 |
| C-I3Ds+both($l_{aw}, l_{sep} = 33, t_{len} = 125$) | 25.0 | **27.8** | 33.3 | 88.9 | 39.5 |
| C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 65$) | **37.5** | **27.8** | **50.0** | **100.0** | **46.5** |
| C-I3Ds+both($l_{aw}, l_{sep} = 46, t_{len} = 125$) | **37.5** | **27.8** | **50.0** | **100.0** | **46.5** |

## 2.6 Conclusion and Future Work

In this paper, we firstly introduce our construction upon the I3D network to make it be suitable with multi-camera in the soccer game and apply it to classify soccer game event rather than actions from individuals. We also propose PBG and CBG to localize/adjust event boundaries in the video of the soccer game. The performance demonstrates the combination of these two grouping methods can achieve a promising result. In the future, we will test our methods on the event classification and localization in more general scenarios. And, due to our grouping methods are not

in a learning manner, we are still interested in inferring event boundaries by machine learning approaches.

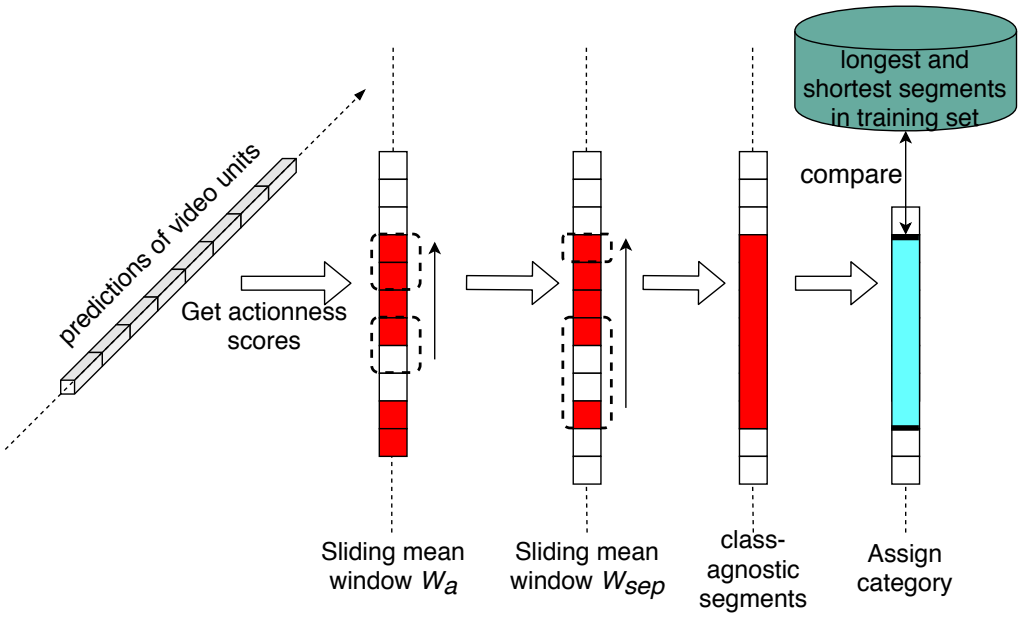


**Figure 2.5:** Probability-based grouping

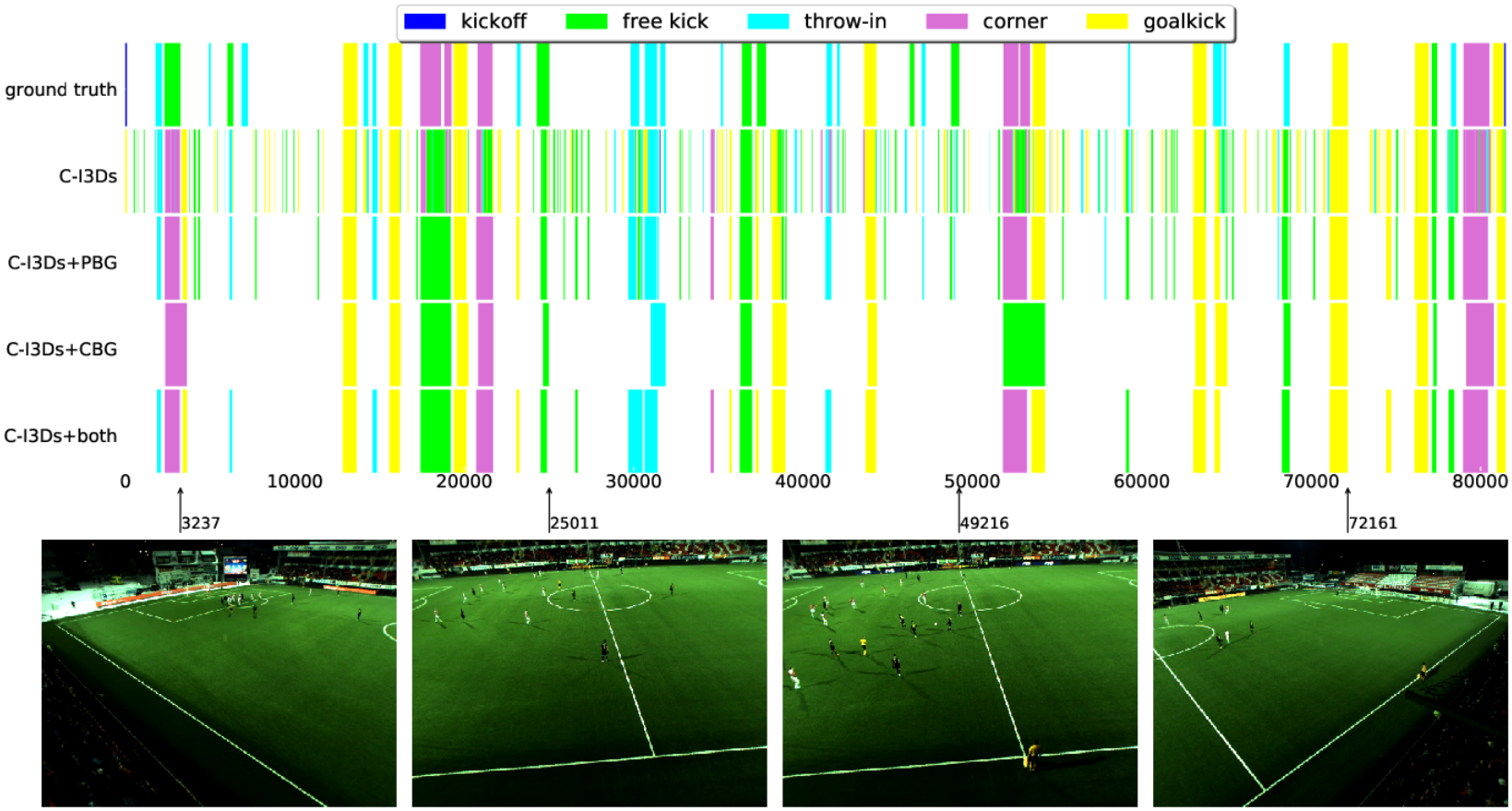


**Figure 2.6:** This is the demonstration of our methods' prediction on the test set. Applying both PBG and CBG after C-I3Ds, precise event boundaries are localized. We sampled four images that are the last frames of their corresponding segments in the ground truth to illustrate some categories are hard to be recognized.

# Chapter 3

# IDENTIFYING FOUL SUBJECTS AND OBJECTS ON STATIC IMAGES

In this chapter, we move to static frames extracted from TV broadcast soccer game videos. We aim at detecting foul subjects and objects on static frames at the foul moment, to make progress towards systems of automatic refereeing in soccer games or assistance in game analysis. Despite recent achievements in both object detection and pedestrian detection, to our best knowledge, we are the first to propose the detection of the foul subject and object for soccer game videos.

In soccer games, competing is very common for winning but it may also result in a foul. From TV broadcast video perspectives, competing usually means players on the field are more likely in cluttered environments. Detecting objects in this environment is a challenging problem due to the arbitrary shapes of objects.

In this task, we first manually annotate and extract frames at the foul moment from game videos provided by SoccerNet dataset. We experiment popular object detector and pedestrian detector respectively, to detect foul subjects and objects on the annotated frames and make comparison. We also investigate the predicted bounding boxes for person on the field for basically estimating the areas that the fouls occur. And we compare the effects of applying non-maximum suppression and soft non-maximum suppression to remove redundancy.

The contributions in this task are as follows – First, based on SoccerNet V1 [44] (In this Chapter, we use SoccerNet V1 and SoccerNet interchangeably), we construct a benchmark by both extracting the moments of visible fouls in video clips and annotating the positions of foul participants at the exactly corresponding frame. Second,

we show the comparison of the detection performance between a common object detector and a person detector. We also show the comparison between non-maximum suppression and soft non-maximum suppression on foul subject and object predictions from two detectors.

## 3.1 Introduction

Nowadays, computer vision techniques have been applied into a massive number of applications. Such applications facilitate human's daily life and make many works much easier than before, like beautification of the selfie, traffic signs recognition, medical treatment, autonomous driving, and images/videos analysis.

Most of the improvements of computer vision techniques mainly focus on describing concrete actions from an extremely person-centric perspective. Some of them in different scenarios may be phrased by abstract descriptions which may be more closely related to common human understanding. Most current approaches for human action recognition usually learn very concrete subject-verb or subject-verb-object concepts. For example, a 'foul' in the sport of association football is a kind of such abstract descriptions that consists of actions/behaviors of one or two players.

A foul is an unfair act by a player, deemed by the referee to contravene the game's laws and interferes with the active play of the game. Besides the abstract descriptions, occlusion condition will also increase the difficulty of separating individuals in crowd. We can easily find such situations in competing games, such as soccer games, basketball games and so on. According to the FIFA rules, many different actions and behaviors can be treated as fouls. In this task, we define that fouls in 3 kinds: the regular foul, the handball foul and the offside foul. In our definition, the regular foul has two participants, the player who committed a foul is called the foul subject, and the player who is fouled which is called a foul object. For the 'handball' and the 'offside', there is only one player – foul subject, who participates in the 'foul' event. The main target in this task is to figure out where does the 'foul' occur and which player is the

foul subject, which player is the foul object in static images given an oracle frame at the foul moment.

As in a supervised learning manner, we firstly annotate data at the frame level based on the SoccerNet dataset [44] in V1. SoccerNet dataset consists of 500 games in 6 famous European leagues from 2014 to 2017. Fig. 3.1 and Fig. 3.4 show some examples of frames about the annotation we made upon the SoccerNet dataset. Depending on some keywords related to the 'foul' event in the commentary provided by the SoccerNet dataset [44], but unlike the annotation in the SoccerNet which is based on timestamps in seconds, we locate the exact time of foul events at the frame level and extract corresponding frames. We call the exact time of foul events (single frame for each foul video clip) the foul moment. Since 442 games out of 500 games have the commentary provided in the SoccerNet V1 dataset, the annotation is made on these games. Then we manually annotate the foul subject and object positions on these extracted frames for our following detection task.

For detection, we look into two detectors, comparing the detection of players who participate into fouls. We also observe that fouls are more likely occurred in crowd, especially for the fouls having two participants. Further investigation of the predicted bounding boxes on static frames is launched by comparing non-maximum suppression (NMS) with soft non-maximum suppression (Soft-NMS) [7].

Alternatively, bypassing detecting participants in cluttered scene, we experiment with the detection on the foul area that is the union of the foul subject and the foul object (if available), as the supplement of the foul subject and object detection. And the experimental result shows a promising performance on this detection.

In this task, the main contributions are as follows:

1. Firstly establish a benchmark based on SoccerNet V1 with new annotations about fouls in the soccer game based on the SoccerNet dataset. In the annotation, three kinds of fouls are annotated: 'handball', 'offside' and 'regular foul'. Besides, the person positions are annotated at the bounding box level at the foul moment
2. Compare performance of different detectors on several tasks

3. Explicitly show the effects of applying post-processing methods on NMS and Soft-NMS

### 3.2 Related Work

Object recognition and detection have achieved huge success based on deep neural networks. The deep learning based object detection can be mainly divided into two types – single-stage and two-stage detectors. Two-stage detector [48, 47, 112, 57, 12, 22] usually has high localization and object recognition accuracy. For instance, in the first stage of Faster R-CNN is called Region Proposal Network (RPN). RPN proposes candidate bounding boxes and put them into RoI Pooling operation for the following classification and bounding box regression tasks. Single-stage object detectors [110, 111, 84, 81, 6] usually has high inference speed. They propose predicted bounding boxes directly from input images without region proposals. Both types of detectors establish a number of convolution and pooling layers to learn features of given object. Powerful backbone network such as ResNet[58], Inception[127, 128], VGG[123] and etc. is necessary for extracting rich features.

Thanks to the success of object detector in general purpose tasks, specific task like pedestrian detection has attracted a lot of attentions, and several datasets for this task have been created for either surveillance application [23, 151] or autonomous driving [42, 170, 29]. To tackle the problem of person detection in heavy occlusion and highly crowed group, many works utilize body parts [174, 98, 172], encode high-level semantic information [169] or improve non-maximum suppression (NMS) [62, 93] to prevent from removing trues in crowd environments. Some other works [56, 9, 117, 170] look into the performance of a pedestrian detector pre-trained on a large-scale dataset and illustrate that a general object detector has better ability in generalization to unseen domains.

Action recognition in static images is also of great practical interest for some purposes like photo collections, based on human pose, facial expression or other activities. Despite lacking of temporal information, many efforts are made by using of


**Figure 3.1:** This work focuses on three kinds of fouls: 'handball', 'offside' and 'foul'. The 'handball' and 'offside' only has the foul subject, but the 'foul' category consists of the foul subject and the foul object. Both foul subjects and foul objects are manually annotated based on the FIFA rule book and the corresponding commentaries. The 'foul area' is simply defined as the union of the foul subject and the foul object or just the foul subject if there is no one being fouled.

cues such as human body or body parts [130, 95, 160, 46], human-object interactions [27, 107], or learning motions by watching video clips and predicting on static images [41].

### 3.3 Foul Subject and Object Recognition on Static Images Benchmark

Existing datasets for person/pedestrian detection such as USC [151] and INRIA [23] are collected for surveillance application. None of them concentrate on detecting person in sports. Also large datasets are typically videos and they only provide very coarse annotations on temporal semantic level, rather than focusing on the individuals'

activities.

To fill in the blank and facilitate to evaluate object detection on foul subjects and objects on static images and also provide a fair technical benchmark, we introduce a new dataset which is built upon SoccerNet V1. The new dataset is called Foul Participants Recognition (FPR). We plan to make the dataset public for research purposes. This dataset is constructed from different perspectives: 1. all fouls must be visible, 2. three kinds of fouls(regular foul, handball and offside), 3. the frame is extracted at the exact foul moment, 4. all foul moments are identified based on commentaries. Our dataset is annotated upon SoccerNet V1 dataset [44]. The dataset is composed of 500 complete soccer games from six main European leagues, covering three seasons from 2014 to 2017 and a total duration of 764 hours. 442 games have commentaries which indicate the events in the game at the second level. Typically, it indicates the event at that moment. For example, the commentary:

> **29:47:** ***"Nathaniel Clyne (Liverpool) robs an opponent of the ball and explodes in anger when Michael Oliver blows for a foul."***

It indicates that the player of Liverpool made a foul at 29:47. And we keep commentaries based on a list of selected keywords so that those irrelevant commentaries can be initially filtered out. The list of keywords contains: "foul", "violate", "trip", "bad challenge", "rough challenge", "handball", "blows his whistle", "blows the whistle" and "offside". Because the commentary cannot always provide the exact names of players who made a foul, the foul subject is therefore used for the player who made a foul, and the foul object is used for the player who was fouled.

Fouls typically happened in an instant, frames in one second could have a lot of differences for individuals. Unlike the frame index being able to retrieve the exact frame for localizing the temporal position of an event, timestamps are usually not so precise that the exact frame of the foul can be pinpointed. But commentaries only provide rough timestamps in games, we have to manually adjust the time location in frame level in order to extract accurate frames, instead of using the timestamp directly. We firstly seek the position at given timestamps based on keywords appearing in commentaries

we mentioned above. If a foul event is visible, we label the frame when there is an obvious 'contact' (regular fouls that have two participants) or a clear action of violating rules (handballs and offsides, both have only one participant). And such frames are named frames at foul moment in our task.

We do this annotation on low-resolution version videos of the SoccerNet V1 dataset. Many videos have been trimmed to 45 minutes long (no extra-time) but corresponding commentaries may contain the information longer than 45 minutes. We ignore the part of the commentary if there is no related videos to match. In this annotation, there are 6494 video segments related to the foul event in which there are 123 'handballs', 1507 'offsides' and 4864 'regular fouls' extracted from 442 soccer games that have corresponding commentaries. Fig. 3.2 shows the distribution of the three kinds of fouls.

Even though the SoccerNet dataset also provides high-resolution version videos, our annotation is done only on the low-resolution version due to the storage space issue in this task. Most of videos in the low-resolution version have the same size that is $398 \times 224$. And the size of the frame in the high-resolution version is usually $1280 \times 720$ or even higher. The example of both resolutions are showed in Fig. 3.3.

### 3.3.1 Foul Subject and Foul Object

Annotation of the foul subject and the foul object is on the frame at the foul moment. Based on the players' gestures and the information from commentaries, we can annotate foul subjects and objects in the most cases. Others may be more difficult to be directly differentiated by who is the foul subject and object on a single frame at the foul moment. Accompanied with the commentary, we play the sequence of frames before and after to figure out the foul subject and object on the frame at the foul moment. Sometimes, the bounding boxes of both participants are highly overlapped at the foul moment. We can only estimate their positions by our human inference. At the second row of the Fig 3.1, green boxes represent foul subjects and 'blue boxes'

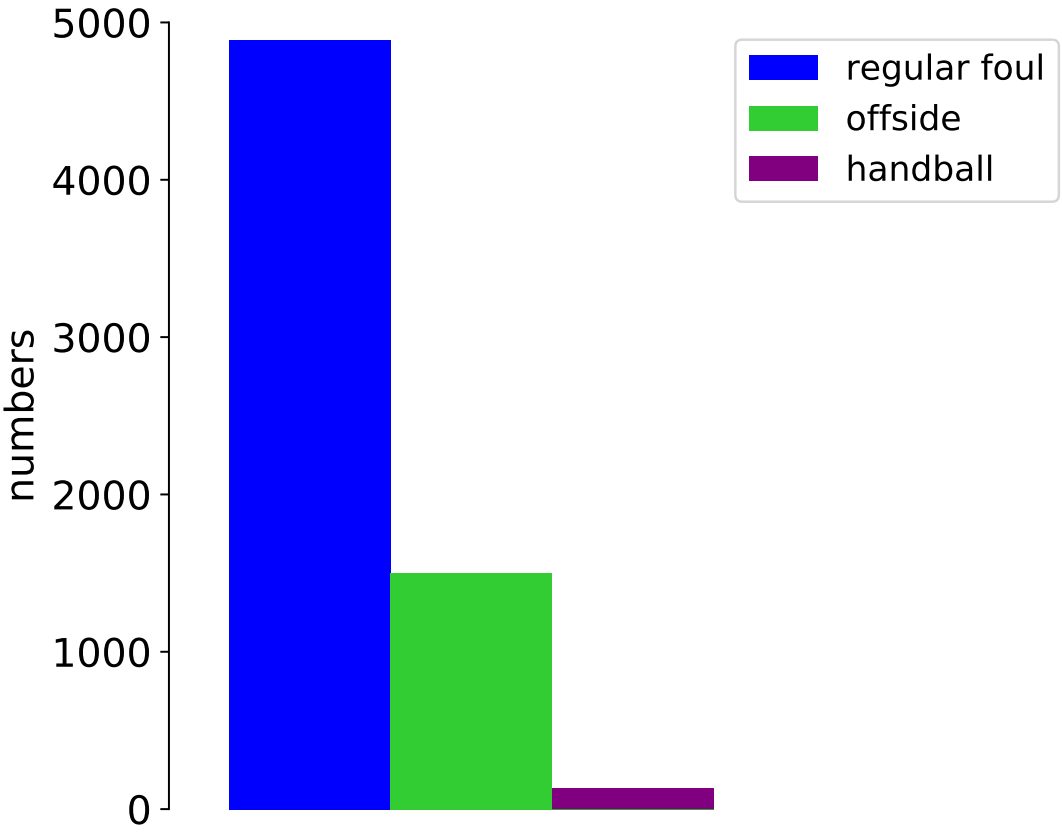


**Figure 3.2:** The distribution of three kinds of fouls annotated in the 442 games.


**Figure 3.3:** The frame in high resolution game videos has different sizes, most of them are $1280 \times 720$ (bottom). Frequencies in high resolution videos are also different. Frames (top) in low resolution videos are resized to $398 \times 224$ from high resolution videos and make the frequencies be the same (25 fps).

represent foul objects. The highly overlapped persons are determined and inferred based on previous and after frames of the current one.

### 3.3.2 Foul Area

To alleviate the impact of the overlap between the foul subject on our following detection and the foul object, and make the foul subject and object detection much easier. Thus we simply define an abstract definition: foul areas. In the frame at the foul moment, the foul area is a region that is the union of both the foul subject and the foul object. If there is no foul object (i.e. offside and handball), the foul area is just the foul subject. Examples of the foul area are illustrated at the third row of the Fig 3.1.

### 3.3.3 All Persons

Persons at the foul frames may have different pose or perform different actions. Because the foul subject and the foul object are typically annotated in crowd at the foul moment frame, annotating all persons on the frame would be helpful for identifying the foul subject and the foul object, or even for other purposes. We select a subset of all foul frames to annotate all persons especially for the persons on the field. Due to the poor resolution, most audiences are not annotated, as well as most staffs who are out of the field. Some examples are showed in Fig. 3.4.

The number of foul frames having all persons annotated is 235, consisting of 3318 persons. The selected foul frames are randomly extracted from all games.

### 3.3.4 Far-Scales and Close-Scales

As broadcast video director may select either zoom-in or zoom-out scales for rendering during the game playing, different scales and perspectives contain different information. For example, far views are able to be utilized for the camera calibration and close views often have more clear quality of human activities. Considering camera positions are often not the same in different games and such position information is

**Figure 3.4:** Ground truth for foul subjects (green boxes), foul objects (red boxes) and 'others' (blue boxes). The foul subjects and objects are full annotated in all frames at the foul moment. In foul subject/object detection, other persons are ignored. All persons are annotated in a subset of all frames at the foul moment. The dataset will not consider if they are foul participants or bystanders but treat them as persons.

inaccessible, we roughly make two kinds of views at all foul moment frames: the far view and the close view. Fig. 3.5 shows some examples of these two views.

### 3.4 Detector

Faster R-CNN [112], as the most popular object detection framework, is composed of a Region Proposal Network (RPN) and a backbone CNNs. Faster R-CNN exhibits good trade-off between speed and accuracy since the RPN and the CNN are combined together into a single network with a large number of shared layers. On a single GPU, it runs at a near real-time speed. In this task, we use Faster R-CNN as the baseline for the detection.

However, as our target is to detect foul subjects and objects which are kinds of fine-grained classes of 'person', it cannot be directly applied to task of learning to do the detection on static images. Therefore, based on the Faster R-CNN network, We choose ResNext-101 [154] with FPN [80] as the backbone networks as it is deeper and shows better accuracy on object detection benchmarks compared to the original network (VGG16 or ZF) used in the task.

Besides learning by Faster R-CNN, we apply another detector – cascade R-CNN using HRNet as its backbone network, since many methods are the extension of Faster/Mask R-CNN [112, 57] and achieve excellent performance on pedestrian detection, and it achieves perfect performance on detecting person in crowd [56, 140, 12]. Cascade R-CNN is also a direct extension of Faster/Mask R-CNN family, containing multiple detection heads in a sequence, which progressively try to filter out harder and harder false positives.

### 3.5 Post-Processing

Typically, as one of most applied post-processing approaches, non maximum suppression (NMS) is applied to get rid of overlapping bounding boxes. It takes a bunch of proposal boxes from RPN with corresponding confidence scores and overlap threshold as inputs. However, such setting is tricky, especially for the scenario where


1 2

3 4

**Figure 3.5:** At all visible foul moment, the frames are basically represented in two different views: 1. far-scale view, in which a larger part of ground is captured, like 1 and 4, larger part of ground and more players are captured by the camera; 2. close-scale view will more focus on individual players for their activities, gestures and facial expressions. The second and third frames show the close view since it has more details about players' activities.

two objects are side by side or have a large overlapping area in which true proposals are eliminated. Heavy occlusion is common for the foul subject and object at the foul moment. We calculate Euclidean distance between every pair of two persons in ground truth, see Fig. 3.9.

Soft-NMS [7] reduces the confidence of the proposals proportional to the IoU value, rather than completely removing the proposals with high IoU and high confidence. It obtains consistent improvements on standard dataset like PASCAL VOC 2007 and MS-COCO.

We also experiment a neural network that consists of two towers – each tower takes one predicted bounding box as its input, to predict if the pair is a pair of a foul subject and a foul object, with or without their corresponding coordinates in the image. However, due to lacks of enough spatial relationship, the two-tower network doesn't have sufficient ability in differentiating pairs.

## 3.6 Experiments

### 3.6.1 Data Preparation

From all annotated frames in 442 games, we make our dataset for training and testing detection models. Training frames are randomly selected such that it consists of 4545 images, the validation set has 975 images and the testing set has 974 images.

In the foul subject and object detection task, the training set consists of 4545 foul subjects and 3412 foul objects. The validation set consists of 975 foul subjects and 749 foul objects. And 974 foul subjects and 737 foul objects are in the test set. Each static image has only one foul area at the foul moment. So the total number of foul area equals the number of images of the corresponding dataset.

As we also investigate the influence of different scales, In the training set, the far-scale has 4291 images and 254 images are the close-scale. In the validation set, there are 920 far-scale images and 55 close-scale images. In the test set, there are 913 far-scale images and 61 close-scale images.

The all persons subset randomly extracts frames from the training, validation and test set. In its training set, 148 images contain 2042 annotated persons. In its validation set, 41 images contains 606 persons. And in the test set, 670 annotated persons are from 46 images.

### 3.6.2 Training

We use on-the-fly data augmentation during training. The augmentation contains random flipping, random cropping, and color distortion. The ratio of random flipping is 0.5. The color distortion contains changing brightness, contrast, saturation and hue. The data augmentation is applied on both detectors in this task.

In Faster R-CNN, we set that the anchor strides are [4, 8, 16, 32, 64] with ratios [0.5, 1.0, 2.0]. We follow other settings as specified in [56] and train the model from scratch because of no available pre-trained model. The loss function for predicting classes is Cross Entropy, Smooth L1 loss is used for bounding box regression. Training Faster R-CNN takes 100 epochs with stochastic gradient descent(SGD) optimizer. The initial learning rate is 0.02, the momentum is 0.9 and the weight decay is 0.0001.

In cascade R-CNN detector, except for the anchor stride in the RPN head is modified to [4, 8, 16, 32, 64] with ratios [0.5, 1.0, 2.0], we follow other settings as specified in [56] and fine-tune the model that pre-trained on CityPerson dataset [170]. The loss function for predicting classes is Cross Entropy, Smooth L1 loss is used for bounding box regression. We train the model 100 epochs with SGD optimizer. The initial learning rate is 0.02 with momentum set to 0.9 and weight decay 0.0001.

### 3.6.3 Detection of Foul Subjects and Objects

Firstly we use these two detectors to detect the foul subject and the foul object directly with applying NMS as the only post-processing method. These kinds of detection will ignore the short spatial distance between the foul subject and object if there is a foul having two participants involved.

In Tab. 3.1, we compare Faster R-CNN and Cascade R-CNN with transitional NMS and Soft-NMS on our test set. For both post-processing methods, we set the overlap threshold $N_t$ to 0.5 and the score threshold to 0.05. The maximum number of detection is set to 100. Our Faster R-CNN model achieves performance on 27.1% average precision (AP) and average recall (AR) 48.7% for $AR_{100}$ with NMS and 27.7% AP and 53.2% $AR_{100}$ with Soft-NMS, some results are shown in Fig. 3.6.

Although foul subjects and objects stem from persons, cascade R-CNN reaches worse performance on AP that is 20.7% $AR_{100}$ is 44.1% with NMS and 22.5% AP , 53.6% $AR_{100}$ with Soft-NMS, some results are shown in Fig. 3.6. Generally, post-processing by Soft-NMS can help to achieve better performance since fouls typically occur in crowd environments. But NMS tend to falsely eliminate many predicted bounding boxes in the same environment for reducing redundancy.

**Table 3.1:** Average Precision (%) of the Foul Subject and the Foul Object Detection with NMS and Soft-NMS

| Model | $AP_{0.5:0.95}$ | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ | $AR_{10}$ | $AR_{100}$ |
|---|---|---|---|---|---|---|---|---|
| Faster R-CNN + NMS | 27.1 | 56.5 | 22.6 | 27.5 | 25.7 | 24.8 | 48.7 | 48.7 |
| Faster R-CNN + Soft-NMS | 27.7 | 56.3 | 24.0 | 28.0 | 26.8 | 26.1 | 53.2 | 53.2 |
| Cascade R-CNN + NMS | 20.7 | 43.3 | 16.6 | 22.3 | 12.6 | 1.0 | 43.9 | 44.1 |
| Cascade R-CNN + Soft-NMS | 22.5 | 43.3 | 19.3 | 24.1 | 13.4 | 1.0 | 52.7 | 53.6 |

### 3.6.4 Foul Area Detection

On foul area detection, the Faster R-CNN model surprisingly shows much better results than the Cascade R-CNN person detector model. Even though our Faster R-CNN model is trained from scratch, the learned weights are more suitable with the foul area detection. The Cascade R-CNN is fine-tuned on CityPerson dataset that is more specific dataset for pedestrians. [56] suggests that detectors pre-trained on more general large-scale dataset have better ability in generalization in cross-dataset evaluation.

Thus, the Faster R-CNN model shows much better generalizing ability compared to our pedestrian-specific model, as Tab. 3.2 shows. The AP is 42.9% which


**Figure 3.6:** Prediction of foul subjects (red) and objects (green) by Faster R-CNN. The threshold of prediction score is 0.3. Our Faster R-CNN has good performance on detecting foul subjects/objects (1-4). But it is incapable of understanding the rule of soccer games (5), and it is unable to establish the relationship between the foul subject and object (6).


**Figure 3.7:** Prediction of foul subjects and objects by Cascade R-CNN. The threshold of prediction score is 0.3 for rendering. Most predicted foul subjects are eliminated since either their scores are less than the threshold or their scores are less than the predicted foul objects that have highly overlapped regions.

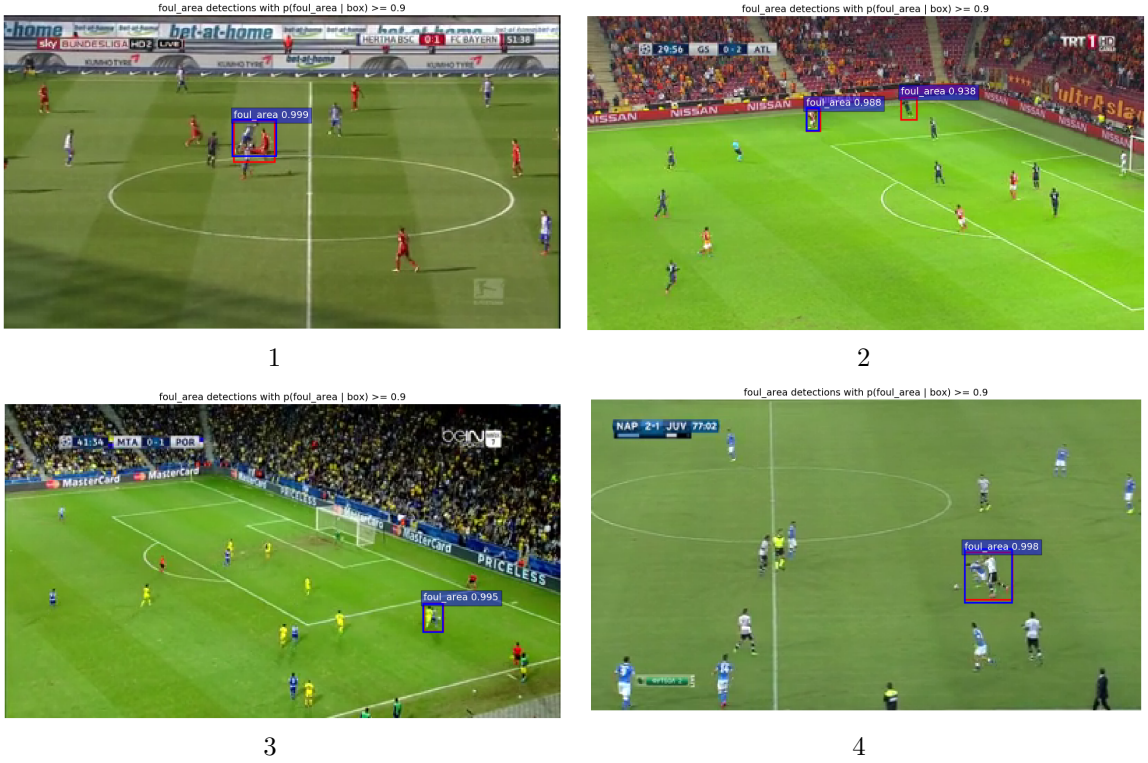


**Figure 3.8:** For foul area detection, it is easier than directly detecting the foul subject and object. Boxes in red are predictions. Blue boxes are ground truth.

outperforms 38.7% AP performed by Cascade R-CNN. But the average recall of Cascade R-CNN is better on testing (in $AR_{100}$, 63.5% vs. 55.3% ). Fig. 3.8 depicts some of the detection results. It suggests that, even though Cascade R-CNN has better performance on pedestrian detection and the foul subject/object detection without post-processing, it is weak in more general scenarios.

We apply NMS as the post-processing method because there is only one foul area in one image at the foul moment.

**Table 3.2:** Average Precision (%) of the Foul Area Detection

| Model | $AP_{0.5:0.95}$ | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ | $AR_{10}$ | $AR_{100}$ |
|---|---|---|---|---|---|---|---|---|
| Faster R-CNN | 42.9 | 78.6 | 44.1 | 41.5 | 45.3 | 56.5 | 55.3 | 55.3 |
| Cascade R-CNN | 38.7 | 60.9 | 45.0 | 38.6 | 50.8 | 34.1 | 63.2 | 63.5 |

### 3.6.5 All Persons Detection

We also experiment the detection on all persons by cascade R-CNN on the corresponding dataset. We argue that different scales have influence on the performance, as depicted in Tab. 3.3. Training 60 epochs on all scales will achieve 44.4% AP and 58.0% $AR_{100}$ in testing. Only training on close-scale static images, the AP is 45.4% and the $AR_{100}$ is 61.2%. Only training on far-scale static images, the AP is 50.9%, the $AR_{100}$ is 62.8%. If we use the model trained on all scales but test on close-scale images, the AP is 47.3%, the $AR_{100}$ is 62.6%. And the AP is 50.9% and the $AR_{100}$ is 57.6% if testing on far-scale images. Although the performance of training and testing just on close-scales images is not better than training on all scales and testing on close-scales images, the margin is not large. And we think it may be caused by limited number of close-scales images in the training set. Training and testing on far-scales images beats training on all scales and testing on far-scales images.

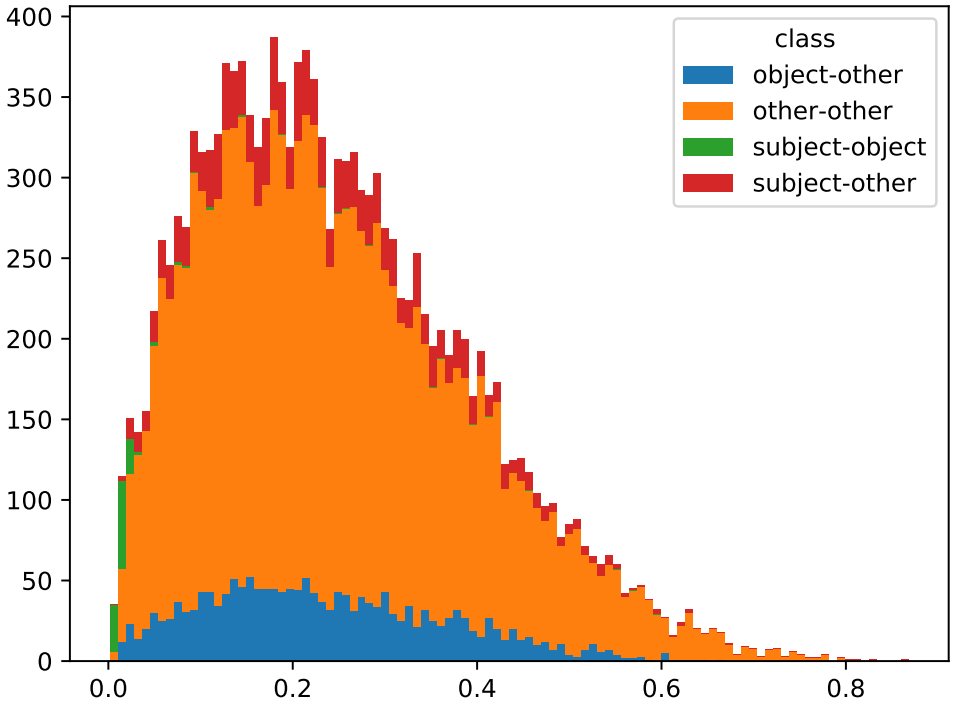


**Figure 3.9:** Euclidean distances are calculated between pairs of two centroids of bounding boxes at the 'foul moment' frame.

**Table 3.3:** Average Precision (%) of the Person Detection (no large scale bounding boxes detected for all far images)

| Model | $AP_{0.5:0.95}$ | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ | $AR_{10}$ | $AR_{100}$ |
|---|---|---|---|---|---|---|---|---|
| all scales for all images | 44.4 | 80.2 | 44.3 | 46.4 | 42.5 | 40.9 | 40.9 | 58.0 |
| all scales for close images | 47.3 | 87.5 | 44.2 | 50.6 | 48.7 | 46.8 | 57.7 | 62.6 |
| all scales for far images | 46.4 | 80.7 | 50.9 | 47.0 | 60.4 | NULL | 37.9 | 57.6 |
| close images | 45.4 | 86.6 | 43.0 | 49.9 | 46.3 | 45.8 | 56.8 | 61.2 |
| far images | 50.9 | 91.6 | 51.3 | 50.8 | 64.9 | NULL | 37.0 | 62.8 |

### 3.6.6 Classification on Two Scales

Since different scales produce influence on the person detection task in our testing, we make a supplementary experiment for classifying close-scales and far-scales. We deploy ResNet-101 [58] to make such binary classification. All input images are resized to 224x224 to feed into the network. The loss function is Cross Entropy, and We use Adam optimizer with the initial learning rate set to 0.001 and the momentum 0.9.

We reach almost perfect performance on testing. The best accuracy is 99.79% on the test set (including 914 distant images and 61 close images). The ground truths of the 2 misclassified images are all close image.

## 3.7 Conclusion and Future Work

In this chapter we have presented the proposed foul subject and object dataset as well as the subset for detecting all persons in the static frames extracted from broadcast soccer game videos.

We also experiment the detection on foul subjects and objects based on both Faster R-CNN detector and Cascade R-CNN detector. Deep investigation is launched for analyzing the different performance of both detectors. The experimental result surprisingly shows that Faster R-CNN outperforms Cascade R-CNN on foul subjects/objects detection, even the Cascade R-CNN model is fine-tuned on pedestrian detection dataset and our Faster R-CNN is just trained from scratch. And the influence of applying NMS and Soft-NMS are compared for detecting foul subjects and objects. We also propose an abstract definition – foul area, to bypass the difficult task

of differentiating subjects and objects. Different scales also impact the detection on all persons.

However, it is worth further investigating the person detection in crowded environments especially for establishing relationships between persons who are very close since post-processing method like NMS and Soft-NMS just try to reduce redundancy and don't establish relationships. We attempted to propose a detection framework that proposes a pair of bounding boxes for dealing with this problem, but the experiment shows it doesn't work at all since there is no enough structural spatial relationship. And it will be still limited to detect two persons in pairs if it works. Recent researches on post-processing may be another direction for dealing with this problem of failing to establish the relationships.

# Chapter 4

# IDENTIFYING FOUL SUBJECTS AND OBJECTS IN BROADCAST SOCCER VIDEOS

Identifying foul subject and object on static images will lose information of temporal dimension. It results in difficulties of recognizing each individual's activities. In this chapter, we move to identifying foul subjects and objects in broadcast videos from still frames at the foul moment. The task of identifying which players are involved in a foul at a given moment is one of spatiotemporal action recognition problems in a cluttered visual environment. We describe how to employ multi-object tracking to generate a base set of candidate image sequences which are post-processed to mitigate common mistracking scenarios and then classified according to several two-person interaction types. For this work we create a large soccer foul dataset with a significant video component for training relevant networks. Our system can differentiate foul participants from bystanders with high accuracy and localize them in a wide range of game situations. We also report reasonable accuracy for distinguishing the player who committed the foul, or subject, from the object of the infraction, despite very low-resolution images.

Besides, we apply unsupervised learning approaches to differentiate players in teams based on their detected torsos. Frames are filtered for having the fields by camera calibration so that we only need to focus on players and the referee who are on the field. Experiment results show that the unsupervised learning way will be capable of differentiating players in teams if torsos are detected precisely and the warping field is accurately laid on the template.

## 4.1 Introduction

Computer vision is becoming ubiquitous for sports video analysis, with applications that include broadcast enhancement; real-time, in-depth player and team performance measurement; and automatic summarization of key events. Across analysis tasks there are several common visual skills such as ball tracking [132, 96]; player segmentation [13, 89, 66], recognition [43], and pose estimation [71]; and recognition of formations, plays, and situations [4, 135, 137, 44]. To deeply understand sports videos, some efforts are put into localizing fields from moving cameras [108, 55]. Some work is aimed at tracking and identifying players on the field [90, 91]. Other work, like [133, 126, 152, 168, 53, 1, 31, 77, 155, 33, 10, 139, 165, 164] try to estimate players' poses to further infer their actions. Or, [44, 26, 19] detect and recognize high-level actions for further video analysis, and retrieve replays across the broadcast videos. Tracking and identifying players in sports videos captured by moving cameras are difficult problems because of blurry facial appearance and almost invisible jersey numbers. [90] introduces a system that aims at tracking and identifying players in sports videos. It adopts a tracking by detection manner to associate detected persons into tracklets. [91] proposes a identification system that consists of a tracking system, a person identification system and a conditional random field (CRF) model to infer identities of players.

Video-based assistance with officiating, in particular, is proliferating. The metric accuracy of high-speed, multi-camera ball tracking systems (e.g., [88]) is relied upon in many sports including tennis and volleyball for line calls, baseball for balls and strikes, and soccer for so-called "goal line technology".

In soccer particularly, the Video Assistant Referee (VAR) [36] is commonly used for close and controversial decisions surrounding goals, major fouls, and player expulsions in this decade. Referees have benefited from these various new technologies. Despite the appearance of high technology, it is really nothing more than an off-field human who flags situations that deserve further review by the head referee via video replays in slow motion from multiple angles.

Even though the experiment result shows the performance of our method is competitive on static images that is discussed in Chapter 3, the motion track of players contains much more information in videos. Thereby the interference from occlusions might be reduced. Tracking sequences of players provide such information to make machine able to recognize their actions during that time period.

Static image analysis has a certain utility for this problem based on player poses and formations, but highly overlapping among them will lead to the action recognition being difficult on static images. Moreover, subtle actions and poses are unclear but people are able to infer them based on their sequential movements which are usually captured by the camera in sequence. Thus, we assert that player movement patterns can be exploited to identify and differentiate foul participants. Here we describe an approach to recognizing telltale motions associated with soccer fouls such as slide tackles, pushing and gesturing, and falling to the ground via a three-stage pipeline. First, players are detected and tracked by a state-of-the-art multiple object tracking (MOT) method which we train to perform well on broadcast soccer videos. Second, raw tracks are cleaned and augmented to account for common tracking errors that could result in crucial players not being covered by a complete track. Finally, processed tracks are fed to two video activity recognition networks to classify whether each person is (a) doing "normal" soccer things vs. exhibiting signs of being involved in a foul, and (b) if they do seem to be involved in a foul, to attempt to discriminate between the person committing the foul and the object of the foul. Fig. 4.2 shows this three-stage pipeline. The results demonstrate that our method can achieve promising performance.

To our best knowledge, we are the first to keep track and identify players who made foul and who was fouled from broadcasting videos in soccer games. These would attract more interests from both academia and industries to design 'pure' automatic refereeing systems.

After that, we also propose a method for identifying players on the field. In the task, SlowFast model only does classify a given sequence of patches of a player without considering the spatial contextual relationship with other detected persons, or

taking into account some low level features that can really help to identify the one who made a foul. For example, a foul is performed at a player who is in the opposite team. And, a foul is seldom performed on the person who are not on the field (it is likely to happen but very rare). These observations are based on: 1. since 1890 in England, it is required that no two teams could wear the same color so as to avoid confusion during the game; 2. a match is played by two teams, each with a maximum of 11 players (one must be the goalkeeper), and a referee is also on the field during the game.

Then, we meet another problem that after identifying players who made foul and who was fouled from broadcasting videos in soccer games, the input sequence of the action classification model doesn't provide extra information like teams, roles (player, goalkeeper, referee and others). Some persons should not be considered since they are out of the field.

We handle this problem by utilizing person skeleton to generate the region of their torsos. Based on the observation that two teams could wear different colors, we construct histogram on these colors and apply clustering method to differentiate them. To evaluate clustering performance on these torsos, we manually label mask images from 7 games. According to the appearance of torsos, we make 6 categories as clusters – 1, players (on the field) from one team; 2, players (on the field) from another team; 3, the referee; 4, a goalkeeper from one team; 5, a goalkeeper from another team; 6, others. Our empirical evaluation shows that our method is able to differentiate players by jerseys' colors detected from the skeleton.

We also launch a generative adversarial network model [16] to calibrate cameras on the game video and produce top-on views to filter out persons who are out of the game field. It improves the performance of clustering after filtering, hence this method will help to differentiate players over the game video.

In summary, our contribution of this task can be summarized into the following aspects:

1. We are the first to annotate multiple players' tracks in broadcasting soccer game videos

2. We annotate the foul subject and object across trimmed video clips

3. Our pipeline on tracking players and detecting foul subjects and objects with proper post-processing methods shows promising performance
4. Differentiating players by clustering after applying person pose estimation and camera pose estimation, evaluation shows that our approach achieves good performance

## 4.2 Related Work

Person detection is one of the main topics in the area of the object detection. It typically applies similar network architectures as standard object detection models like Faster R-CNN [112] and Mask R-CNN [57] with some specific modifications for improving localization [56, 170].

Thanks to the advantages of deep neural networks, great improvements have been made in action recognition, action detection, human-object interaction (HOI), and multi-object tracking [103]. Action recognition could either apply 2D convolutions on per-frame input followed by another 1D module for aggregating the features [69, 122] or apply stacked 3D convolutions to model temporal and spatial features [134, 14]. [34] uses two different pathways to operate on different frame rates for capturing both spatial semantics and temporal motions. Recently there has been more focus on *interactions* [49, 94, 173] with the goal of identifying {*human, verb, object*} triplets in static images and videos. Detecting and recognizing the relationship is crucial for determining the global interpretation of an event. [21] proposes a method to handle high diversity of each class in detecting relationships. [49] proposes a method that consists of three branches of object detection, action recognition and interaction, in order to detect {human, verb, object} triplets on static images. [173] introduces a model that consists of a relation ranking module and a triple-stream classifier for relation prediction. [109] introduces a model that incorporates structural knowledge to address the task of detecting and recognizing human-object interactions. [63] constructs their model based on deep reinforcement learning to refine the low-level features and high-level relations of group activities. In multi-object tracking (MOT) task, tracking by detection is the main way [103, 149, 131] because of the progress in object detection.

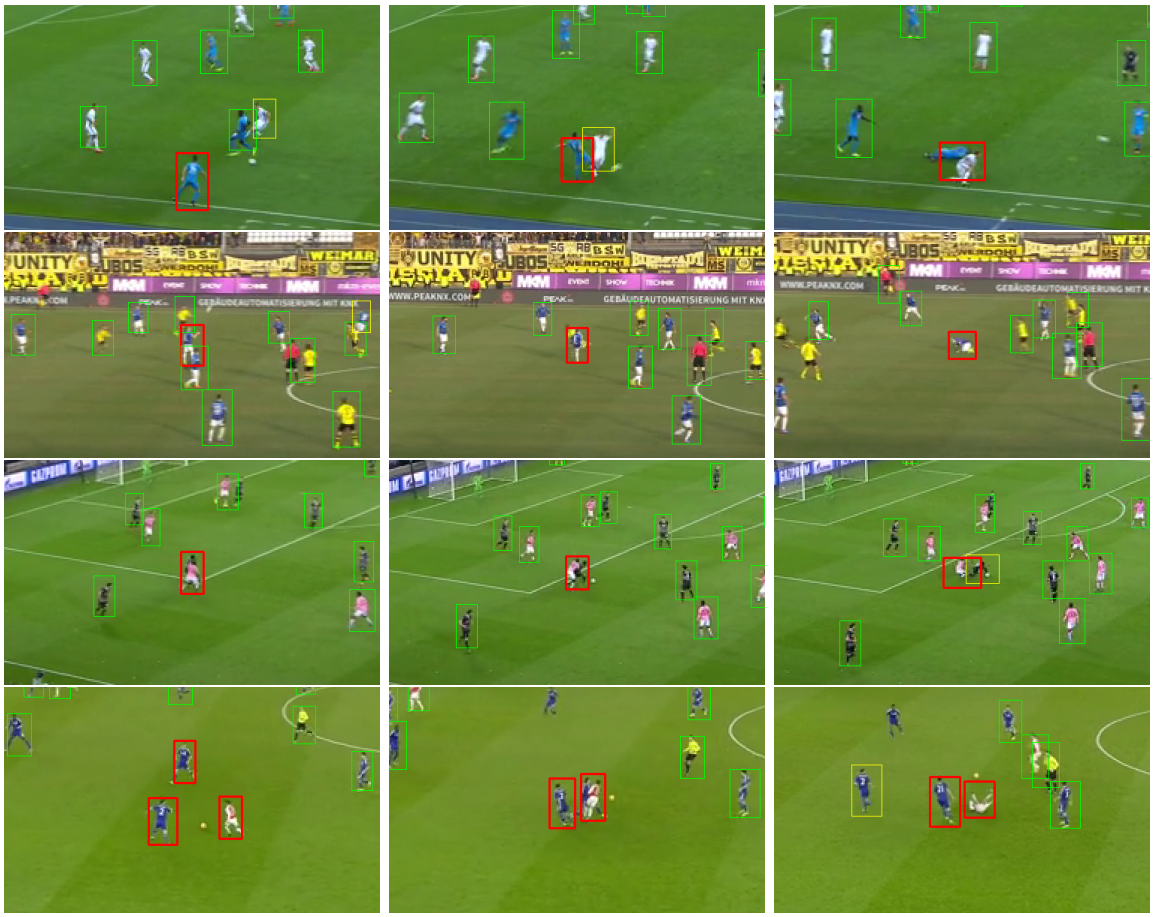

**Figure 4.1:** An image sequence for each tracked person, and their activity is classified as foul-related or not. Samples of foul participant detections are shown here with maximum likelihood candidates in red, over threshold in yellow, and non-participants in green (each row spans 2 seconds and the images are cropped to highlight the detections)

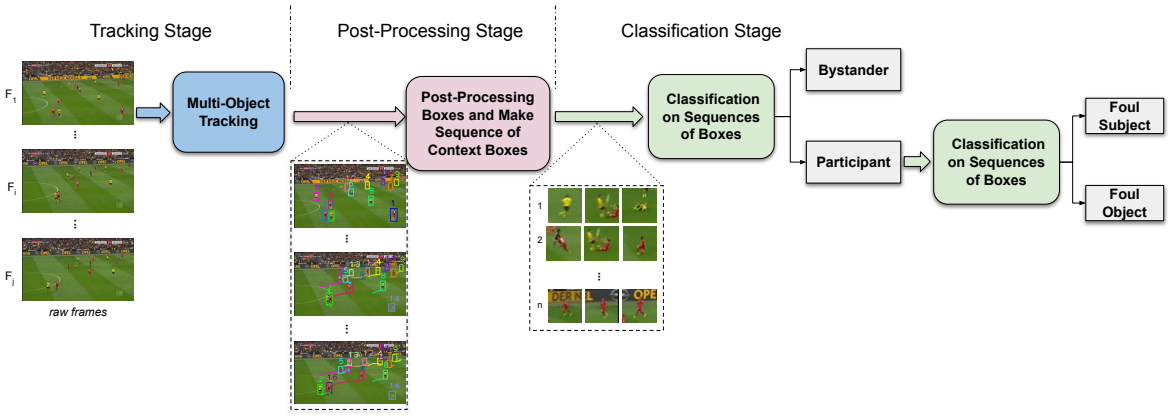


**Figure 4.2:** The three-stage pipeline of our identification work. At the first stage, we input the sequence of raw frames to Multi-Object Tracker to get players' tracks. Then, each sequence of patches are extracted with post-processing instead of resizing to get context ROIs. At the last stage, the sequences go into the 3D classifier for identifying bystanders, foul subjects and objects.

[85] demonstrates high-quality spatial-temporal activity detection in surveillance video scenarios, and more and more state-of-the-art methods have been utilized in the area of sports. [136] introduces a multi-tower temporal 1D convolutional network to detect events in ice hockey game and soccer game videos. [63] constructs their model based on deep reinforcement learning that shows only part of people's activities have impacts on the entire group and tests their model on volleyball videos. [116] uses self-attention models to learn and extract relevant information from a group of soccer players for activity detection from both trajectory and video data. [44] tries to "spot" three soccer event categories: *goal, card*, and *substitution*.

For estimating human pose, early methods use morphological operations to extract skeletons [120, 114, 146]. Recently, deep neural networks approaches [133, 126, 152, 168, 53, 1, 31, 77, 155] are widely utilized to estimate human pose in different perspectives. [133] presents a cascade of deep neural network regressors to human pose estimation. Using a backbone network named HRNet [140], [126] keeps the high-resolution representations throughout feature extraction. After gathering dense


**Figure 4.3:** Our MOT dataset on SoccerNet V1. Each column represents a sequence of frames from $t1$ to $t5$. In each sequence, different bounding box colors means different person ID.


**Figure 4.4:** A sample two-person foul with 2-second temporal context around the *foul moment* at $t = 0$. The *foul subject* is denoted with a green bounding box (track 8 in the last column) and the *foul object* is marked with a blue box (track 10).

correspondences for human appearing in the COCO dataset, [53] delivers dense correspondence by regressing body surface coordinates at any pixels.

Pose estimation is extremely useful in sports [33, 10, 139, 165, 164]. From multi-view video, [10] does multi-person 3D pose estimation and tracking. It associates poses between views in greedy manner. The associated poses can be used to generate 3D skeletons. Based on the tracking trajectory, a pose estimation model proposed by [139] helps to create an AI coach for assisting athletic training.

As camera calibration is inevitable and important for autonomous broadcasting and sports analysis, many studies [108, 55] usually treat the playing surface as flat to make camera calibration the same as estimating the homography in the image. These works require user interactions to provide reference images. [166] proposes an automatic camera calibration system that eliminates distortion and calibrates the camera in each

two consecutive frames. To localize the field, [59] proposes a method that bypasses the reliance on humans annotating key-frames for each new game or installing fixed cameras around the arena, but obtaining semantic segmentation of the field as the evidence for localization. By learning deep features via a siamese network, [16] uses two-GAN (generative adversarial network) model to detect field markings in images.

In this task, we utilize AlphaPose model and two-GAN model to generate masks for localizing players' torsos on the field.

### 4.3 Datasets and Annotation

Our foul dataset is built upon SoccerNet V1[44], which comprises of 500 complete soccer games from six European professional leagues, covering three seasons from 2014 to 2017, encoded mostly at 25 fps with a total duration of 764 hours. The footage is from broadcasts, so it includes camera pans and zooms, cuts between cameras, graphics overlays, and replays. Both high-definition and lower-resolution (224p) versions are available; here we use the low-resolution version for all learning, evaluation, and paper figures.

442 SoccerNet games have text transcripts of audio commentary on game events which are timestamped by half game clock with one-second precision. A sample foul is shown in Fig. 4.4 (and in more detail in Fig. 4.5) which corresponds to the following comment: *1 - 15:33: This yellow card was deserved. The tackle by Aranguiz (Bayer Leverkusen) was quite harsh and Christian Dingert didn't hesitate to show him a yellow card.* We roughly locate fouls by searching all transcripts for relevant words and phrases such as: "foul", "violate", "trip", "bad challenge", "rough challenge", "handball", "blows [...] whistle", and "offside." Video frames in the temporal neighborhood of each candidate's timestamp are then manually examined to determine a precise *foul moment*. Clues from the commentary about which player committed the foul are used to resolve any visual ambiguities about the placement of the foul subject and object bounding boxes (green and yellow, respectively, in Fig. 4.4).

In all, 6545 foul events are labeled, of which 4862 are two-player fouls, as well as 1507 offside offenses and 123 handball offenses. Almost all of these events occur in "far" camera views such as shown in Fig. 4.4, but some are in close-ups or "near" views.

Existing multi-object tracking benchmarks such as MOT dataset [99] for persons only provide collections mainly coming from surveillance applications. Positions of the cameras recording those video sequences are mostly fixed. To provide a more realistic dataset from broadcasting videos in soccer games, we sample 17 from 6545 foul event video clips to make our **MOT subset**. Sometimes, working on single frames to determine which person participated a foul lacks of corresponding temporal information. The trajectory of persons will provide additional information and help detect and recognize the foul subject and the foul object. We use the pre-trained person detector to localize all persons at the first frame of the video clip and run the CSRT tracker [92] on every single detected persons. We manually adjust and modify the position and size of the bounding box if it is apart from the target person or the size is either too large or too small. If the target person actually went out of the image boundary, we manually stop keeping the tracking for avoiding wrong annotations. If an actual person went into the frame during the video but not at the first frame, we also manually initialize the CSRT tracker on the corresponding box. Obviously, different persons have different IDs, rather than having the same IDs. Our **MOT subset** 85-100 frame bounding box sequences (*tracks*) for all people (n = 309) present in 17 randomly-selected person detection frames (16 far, 1 near) annotated over 4-second temporal windows ([-1, +3] s) surrounding the foul moment. Tracks are manually trimmed at any shot boundaries (e.g., near/far transitions). Fig. 4.3 shows some examples about our MOT dataset. Each column represents a sequence of sampled frames by stride 10 in a 25 fps video clip. Players on the field are all annotated in bounding boxes with their corresponding person-id to be tracked.

**Action recognition subset** Complete 50-frame tracks for the foul subject and object

are annotated over 2-second temporal windows ([-1, +1] s) surrounding 833 randomly-selected two-person foul moments (all far views with no shot boundaries). Furthermore, 50-frame tracks for people ($n = 5006$) not involved in the foul, whom we call *bystanders* (e.g. other players, coaches, and referees) are obtained from CTracker [103] tracks that span the entire clip and do not overlap the ground truth subject or object bounding boxes.

**Jersey Color Clustering Subset** Due to the low resolution version of SoccerNet dataset is not suitable for human pose estimation, we use the high resolution version instead. The high resolution videos come from online providers, in a variety of encoding (MPEG, H264), containers (MKV, MP4, and TS), frame rates (25 to 50 fps) [44].

In the task of clustering, we still aim at video clips that are at the foul moment. And to keep all video clips consistent, we sample frames of the extracted video clips such that all frame rates are 25 fps – i.e. 75 frames per video clip in 3 seconds. The resolution are kept as HQ version.

Based on the mask images and the corresponding raw images, We initially list 6 categories as ground truth (1. players (on the field) of team A; 2. players (on the field) of team B; 3. referee; 4. goalkeeper of team A; 5. goalkeeper of team B; 6. others). We manually label mask images output from the AlphaPose model on 5 games, as showed in Tab. 4.1. The category 'others' represent the region of the corresponding AlphaPose mask that is definitely out of the field or is not a part of the 'torso' of a person on the field.

**Table 4.1:** Annotated torsos detected by AlphaPose model based on their roles and positions

| Game | Team A | Team B | Referee | GoalKeeper A | GoalKeeper B | Others |
|---|---|---|---|---|---|---|
| Real Madrid vs. Las Palmas | 589 | 796 | 92 | 0 | 0 | 184 |
| Schalke vs. Dortmund | 641 | 631 | 111 | 12 | 0 | 944 |
| Bayer Leverkusen vs. Dortmund | 780 | 881 | 76 | 22 | 6 | 402 |
| Besiktas vs. Napoli | 1035 | 898 | 142 | 0 | 0 | 643 |
| Dortmund vs. Galatasaray | 1519 | 1215 | 301 | 48 | 6 | 1763 |

We also count the number of 'others' based on their location in these games used for evaluating the clustering performance before and after filtering out the stands.

Tab. 4.2 shows that a huge number of 'torsos' detected by AlphaPose come from out of the field.

**Table 4.2:** The number of different locations of 'torsos' of Others

| Game | On the field | Out of the field |
|---|---|---|
| Real Madrid vs. Las Palmas | 10 | 174 |
| Schalke vs. Dortmund | 90 | 854 |
| Bayer Leverkusen vs. Dortmund | 9 | 393 |
| Besiktas vs. Napoli | 9 | 634 |
| Dortmund vs. Galatasaray | 22 | 1741 |

## 4.4 Methods

### 4.4.1 Multi-Object Tracking and Inference

#### 4.4.1.1 Multi-Object Tracking

For identifying the player actions "committing a foul" and "being fouled," we adopt the SlowFast network [34] for video recognition. To adapt this network for our spatiotemporal task, we stabilize the video around each candidate player by assembling image sequences from tracker bounding boxes derived from an MOT tracker's output. Here we use Chained-Tracker (CTracker) [103], which combines object detection, feature extraction, and data association in a single end-to-end model that chains paired bounding box regression results estimated from overlapping nodes, of which each node covers two adjacent frames. CTracker achieves fast tracking speed (30+ Hz) and a Multiple Object Tracking Accuracy (MOTA) on MOT17 online of 66.6, which is highly competitive with other state-of-the-art algorithms.

As an example, the foul subject and object in Fig. 4.4 (indicated by the green and yellow bounding boxes, respectively, at $t = 0$) are followed in tracks 8 and 10, respectively, produced by CTracker. Synopses of the sequences resulting from this *tight* tracking box, cropped and scaled to SlowFast's $224 \times 224$ input, are shown in the top two rows of Fig. 4.5.

Tight tracker ROI sequence for subject (top), object (bottom) in Fig. 4.4

Medium context ROIs on same subject and object

**Figure 4.5:** Sample *tight* and *context* ROI sequences derived from tracker output as input to the action recognition network

Raw tracker output can be noisy, exhibiting sudden shifts and scale changes that present challenges for video recognition, especially when the source ROIs are on the order of $\sim 15 \times 30$ pixels. Moreover, the entire player might not be shown, losing valuable information about leg and hand motion, and certainly any depiction of *interactions* with nearby players is lost. Therefore, we expand the spatial context around each tracked bounding box on the hypothesis that it will aid the video recognition task. We define *context* ROIs as squares with sidelength proportional to the median max dimension of all tracker bounding boxes over an entire clip ($1.5\times$ scaling for *medium*). Samples are shown in Fig. 4.5.

#### 4.4.1.2 Track post-processing

Tracks may be incomplete. In order to supply the video recognition network with sequences that span the full temporal context $T$ and to mitigate mistracking and track merging and splitting (see Fig. 4.6 for an example), we transform CTracker's output to create a modified set of *candidate* tracks. First, tracks with small "gaps" of up to 5 or 6 frames are **patched** with linear interpolation between adjacent bounding boxes. In a second pass, tracks which end near another viable track are **joined** to them in order to extend them. Also in this pass, **branches** may be created between continuing tracks and new tracks that start nearby, increasing the overall number of tracks. In clips with high player densities, this may result in enlarged sets of candidate tracks with subsets in common.

#### 4.4.1.3 Inference

SlowFast network involves two pathways – a slow pathway for capturing spatial semantics by operating at low frame rate, and a fast pathway for capturing temporal motion by operating at high frame rate. SlowFast achieves good performance (top-1 accuracy) on Kinetics-400 [70] as 79.8%, which outperforms previous state-of-the-art methods.

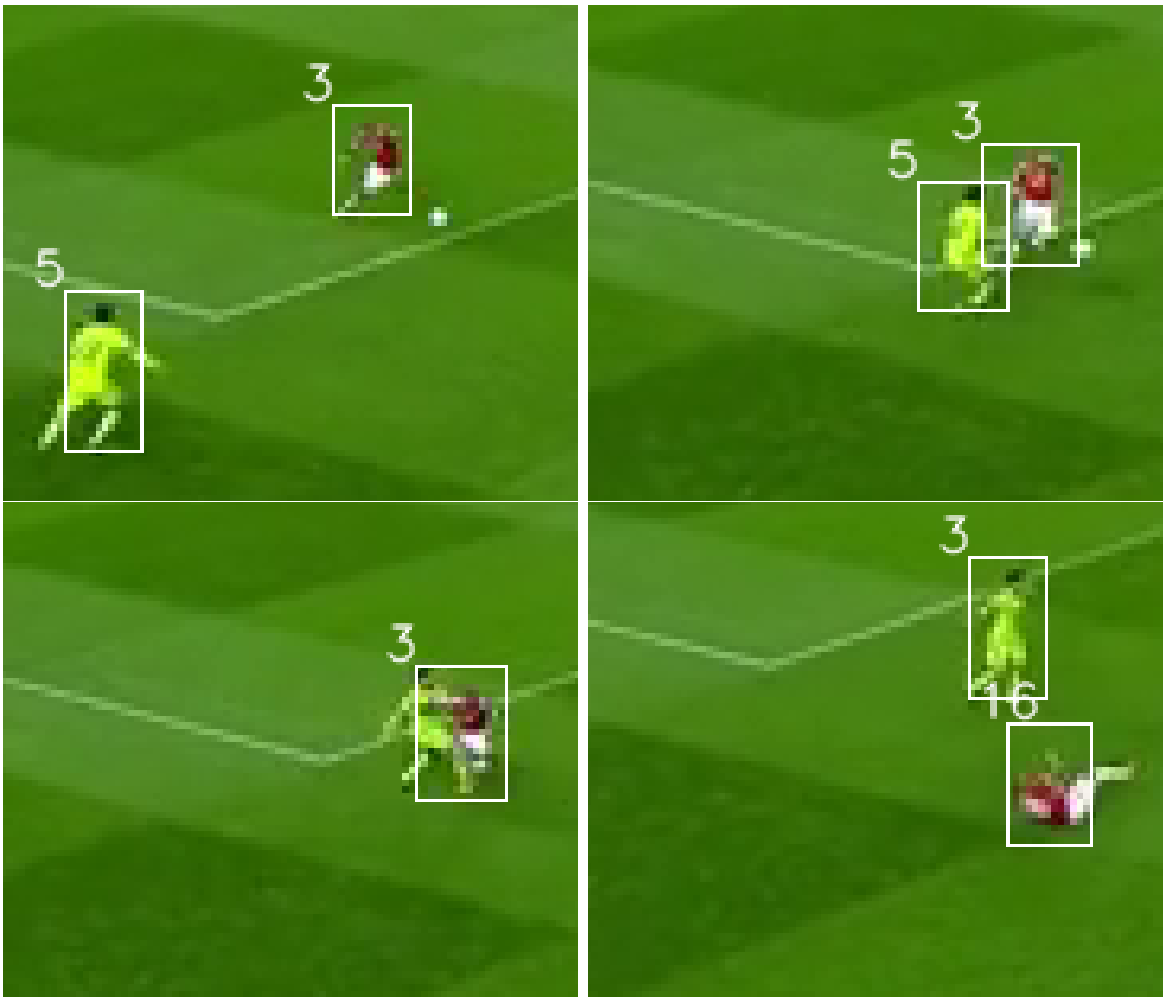


**Figure 4.6:** Example of CTracker mistracking: Track 5 disappears when the two players come together, and when they separate, track 3 follows the wrong player. Our post-processing corrects this: One *candidate* track is created via a `join` of the truncated 5 and the "wrong" ending of 3, and another track is made via a `branch` from the middle of 3 to the new track 16. The complete, erroneous track 3 also remains as a candidate.

Several SlowFast networks [34] are used for different purposes. `SF_PvB` classifies each candidate track video as either a foul *participant* (without regard to subject or object) or *bystander*, and `SF_SvO` classifies each candidate track video as a foul *subject* or a foul *object*. Because of the oracle assumption, we know that there is exactly one subject and one object per clip, transforming detection into a maximum likelihood problem. However, as seen in Fig. 4.6, there is not necessarily a one-to-one correspondence between tracks and people – we must always allow for the possibility that two players are being tracked by one box.

Participant detections are the bounding boxes at the foul moment from those tracks with the highest likelihood according to `SF_PvB`. There may be a tie due to floating point precision and the network output saturating; these are broken first by voting in the case that multiple maximum likelihood tracks share the same foul moment bounding box, and second randomly. Subject and object detections are maximum likelihood classifications according to `SF_SvO`, but they are only considered if already recognized as participants.

We also use just one SlowFast network to identify bystanders, the foul subject and the foul object (**SF_BvSvO**). Same as the oracle assumption we have above that there is one subject and one object per video clip but we make predictions on 3 categories directly here. Multi-label classification is also experimented for measuring how much a foul subject has a foul object and vice versa.

### 4.4.2 Differentiation of Players

#### 4.4.2.1 Person Torsos Detection

Players on the field may be taking various actions rapidly and their poses are volatile. These changes in videos usually results in extreme difficulties of capturing information about players' identities. In soccer games, players in one team rarely commit fouls to their teammates, thus the possibility of overlapping by players in the same team may be reduced. This observation allows us to hypothesize that, if knowing

players' wearing, we may be able to at least remove the wrong tracks just like the track 3 showed in Fig. 4.6. And this will require to detect players' torsos during their motion.

Human pose estimation can help to deal with this problem by detecting unique keypoints on the human (player) body. In MS-COCO [82], 17 keypoints are defined for the person class. In Halpe, 26 and 136 keypoints are defined for more detailed annotation of human. AlphaPose [31, 77, 155] is a system for multi-person pose estimation, which is the first open-source system that achieves 70+ mAP (75 mAP) on MS-COCO dataset and 80+ mAP (82.1 mAP) on MPII dataset.

In hope of detecting the torso part of players, we utilize AlphaPose to first estimate players' poses (see middle figures in Fig. 4.7). Because the keypoints are well-defined for representing human parts, thus, for representing the torso, we make a polygon by connecting four keypoints – 'left-shoulder', 'right-shoulder', 'left-hip' and 'right-hip', or constructing a triangle if one of these four keypoints is missing. The polygon/triangle is represented as a part of the torso of players. Therefore, no matter which directions players are looking at and no matter what kinds of actions they are taking, person torsos are able to be extracted by masking the polygon to the image, as Fig. 4.7 depicts. In experiment, we remove extremely small 'torsos' to reduce the possibility of errors. Therefore, we can make clustering based on the extracted torsos for the following.

#### 4.4.2.2 Sports Camera Calibration

In Two-GAN (Two Generative Adversarial Networks) [16], unlike some methods like [17] in which the location of the camera is exactly known in training/testing, they design an approach to detect field markings only assuming the camera location is roughly known. [16] model involves a camera pose engine, a deep feature extractor (a siamese network), a two-GAN model for field marking detection and a camera pose refinement process. Most sports cameras are pan-tilt-zoom (PTZ) cameras [129, 18, 16]. In the camera pose engine, the camera position is supposed to be above and along the center line for soccer games. To transfer world coordinates to pixel coordinates, the

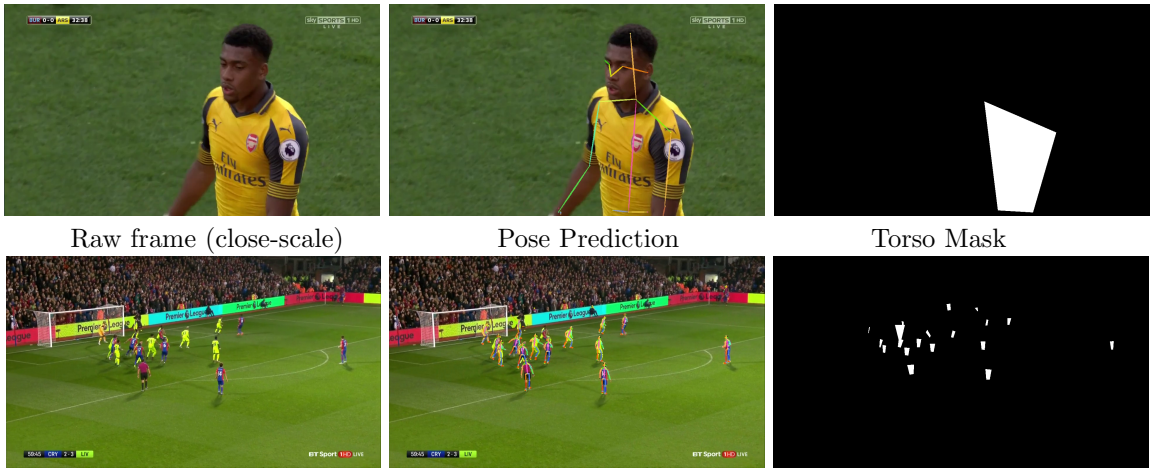


**Figure 4.7:** A frame may contain multiple players. Masks are generated by the human pose predicted by AlphaPose. The first row is close-scale, and the second row is far-scale. Far-scale frames usually have more persons detected, and the detail on player's torso is unavailable except for the jersey color. The masks are generated by making polygons or triangles from 3 or 4 keypoints.

camera pose engine in soccer games only needs to consider 3 parameters – the focal length $f$, the camera tilts by $\phi$ and the camera pans by $\theta$. These 3 parameters' ranges are known from the World Cup dataset. These 3 parameters create a homography matrix to transform the field template to the edge image. Hence, camera poses and paired edge images are generated by uniformly sample these 3 parameters within the ranges.

The camera pose engine is used for synthesizing a set of edge images which are embedded into a low dimensional feature space for obtaining a feature-pose database. By inputting a pair of edge images, a siamese network learns if two images in the pair is similar. The siamese network is a two-branch network. Each branch is a convolutional neural network $f_w(\cdot)$ and constructs a feature-pose dataset. We also use the one branch of the siamese network to extract features from detected field marking images in testing.

Then, a two chained conditional GAN [16] is used for detecting field markings. In [16], the first GAN segments foreground areas from the raw frame and outputs

a mask image. The second GAN detects field markings from the foreground image. Both GANs predict if their corresponding input images (mask image and field marking image) are real or fake, respectively. Instead of detecting hard boundaries that will cause artifacts, [16] interpolates values between foreground and background within the range of [30,50] pixels. We input the detected field marking image into the siamese network and extract the feature. The feature will be used for finding the nearest neighbor from the feature-pose database. The camera pose is refined by computing the distance image on features of the detected field marking image and the generated edge images by the camera pose engine. The pipeline is illustrated in Fig. 4.8.

The calibration accuracy is measured by the intersection over union (IoU) score. The IoU is computed by warping the projected model to the top-on view. In particular, $IoU_{whole}$ [60, 16] measures the IoU on the whole area of the model and $IoU_{part}$ [118, 16] measures the visible area in the image. The measurement shows that Two-GAN [16] achieves accurate results on the World Cup dataset [60] (89.4 in mean $IoU_{whole}$) and on the Volleyball dataset [67](94.5 in mean $IoU_{part}$).

We first detect field markings using the two-GAN model on the resized images ($256 \times 256$) of our annotated video clips. Then, we query an initial camera pose from the feature-pose database using the deep feature of the detected edge image. After that, we refine the camera pose by the distance image and the Lucas-Kanade algorithm. In this task, Enhanced Correlation Coefficient (ECC) is used for measuring the confidence in the homography. If it is not over 0.9, the homography is probably not so great.

As we hope to remove the influence of persons who are out of the field, we utilize Two-GAN model to filter out the stands. Then, top-on views are generated and keep the persons who are on the field, as depicted in Fig. 4.12. We set threshold $m_t$ on the mask of the field image to specify the field boundary, and it is set to 32 in our experiments.

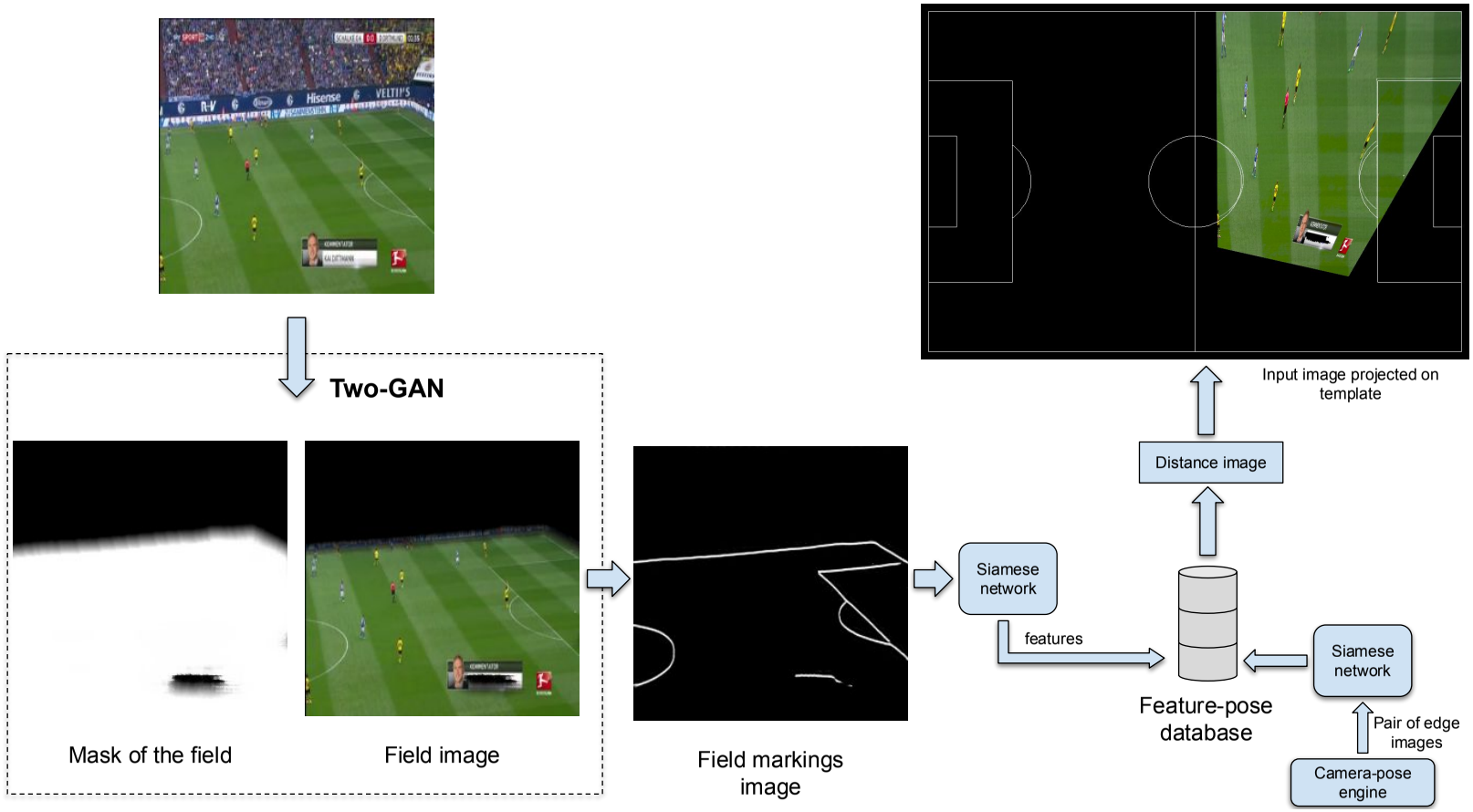


**Figure 4.8:** In [16], the input frame is resized to $256 \times 256$ to Two-GAN model. Then the model segments the field on the image and applies the mask to the original image to generate field image. From the field image (foreground), Two-GAN model will detect field markings. Using the trained siamese network to extract features from the detected field markings image and find the near neighbor in the feature-pose database that created by the camera-pose engine and the siamese network. Then, apply LK algorithm to estimate homography matrix.

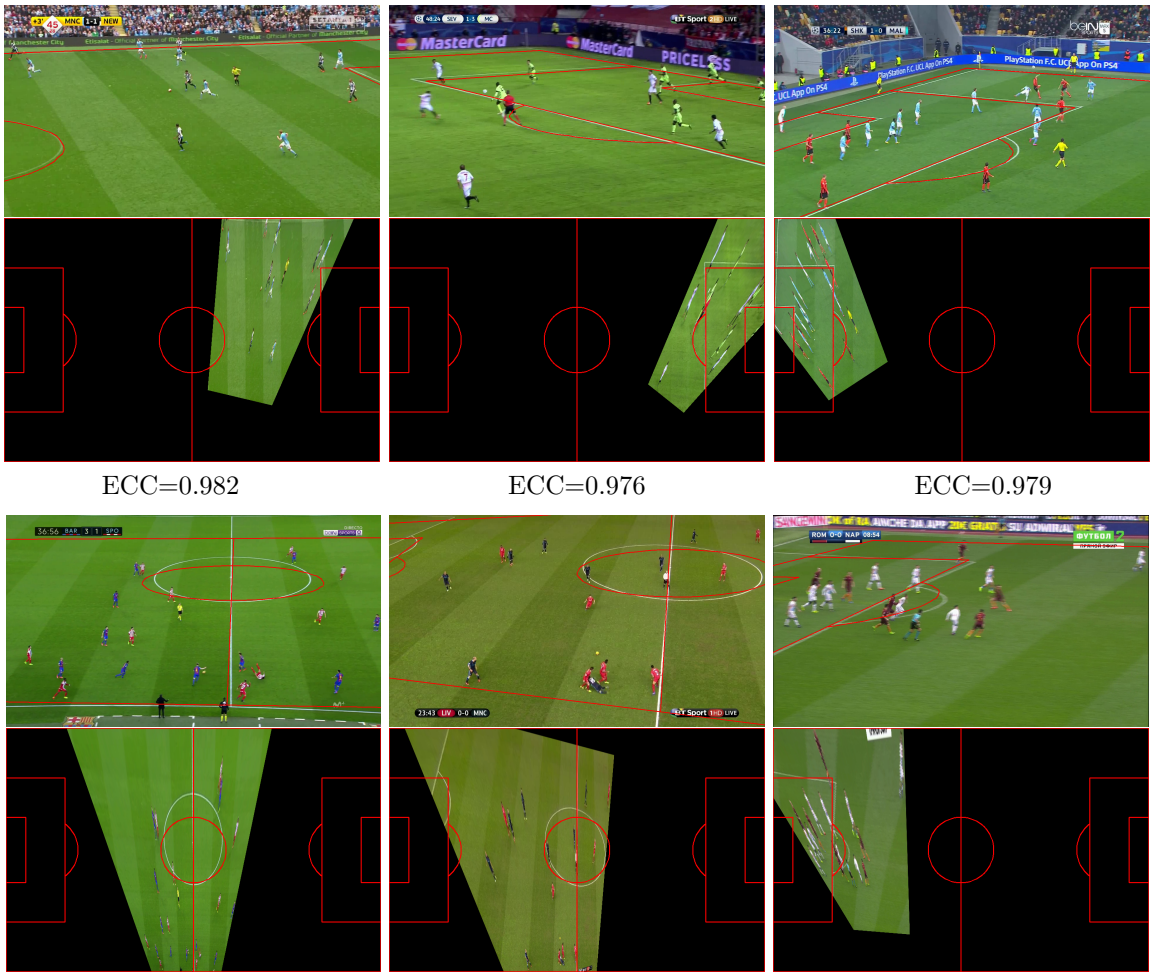


**Figure 4.9:** Examples of our camera calibration results by using [16]. The first two rows show 3 great homography matrix and their ECCs are over 0.95. The rest two rows show bad results since the field template mismatches the field edges after warping. In this case, their ECCs are usually less than 0.9. The bad result may be caused by the influence of the scoreboard, billboards and watermarks.

#### 4.4.2.3 Clustering for Differentiating Players

We would like to differentiate players in teams by clustering approaches. Jersey colors may be various in different games even for the same team. Each team may have at least 2 different jersey colors (usually 3 jerseys per season for one team) to avoid confusion during the game in every game season. People who are not fans typically don't know which jersey the team would wear in advance, especially for the away team. Moreover, the goalkeeper and the referee may also wear various colors in different games. It results in difficulties in building a general ground truth of jersey colors for identifying teams. In light of this, we adopt unsupervised learning approaches to differentiate players without knowing their wearing colors in advance. Typically, jerseys contain multi-colors and may represent differently from different perspectives. We work on normalized histograms of colors of detected torsos in hopes of differentiating them by colors. In this task, we mainly test it by Agglomerative clustering [74, 147] and K-Means [148, 75, 3] algorithms. Fig. 4.10 shows how we do clustering on torsos from raw frames.

The next problem is, for clustering like K-Means and Agglomerative clustering algorithms, how to choose the number of clusters before feeding data to fit the model. We initially make 6 categories in our clustering subset for comprising all possible categories on the field, but some categories may not have any of its elements existing in some frames or video clips when the cluster model is being fit. For example, like the top-left figure in Fig. 4.7 shows, there are no goalkeepers for both teams. And sometimes the referee is invisible. Because we fit the clustering model by one foul video clip per game, in our experiments, we mainly set the number of clusters to be 2 or 3.

Besides, we also try to test DBSCAN (Density-Based Spatial Clustering of Applications with Noise) in this task since we don't need to set the number of clusters prior to clustering, but we find its sensitive parameter setting may lead to bad performance in clustering.

## 4.5 Experiments

### 4.5.1 Training Details

**CTracker** A CTracker network with a ResNet-101 backbone pre-trained on the MOT dataset [99, 103] is fine-tuned on 10 4-second clips (9 far, 1 near) from our dataset in which all player tracks are manually annotated, with standard data augmentation mentioned in [103].

**SlowFast** We use the ResNet-50 $8 \times 8$ variant of the network, pre-trained on the Kinetics dataset, for both of our video action classifiers. 666 48-frame, 2-second clips (with ground truth for 666 subjects and objects and 7996 bystanders) are randomly selected from our foul action recognition subset and `SF_PvB` and `SF_SvO` are fine-tuned for 10 and 40 epochs, respectively.

### 4.5.2 Results

#### 4.5.2.1 CTracker Evaluation

We follow the MOT evaluation metrics [99] to measure the performance of CTracker model. The CTracker's basic tracking performance is evaluated on our data, which consists of 6 test video clips (all of them are far-scales), resulting in a MOTA of 89.2%. The Tab. 4.3 exhibits our multi-object tracking performance based on MOT evaluation metrics. We also simply post-process the result by seeking the closest boxes for dealing with cross-frame association problem. Despite its performance being a little better (MOTA is 90.9%), we still keep on using the outputs without post-processing due to worse MOTP.

**Table 4.3:** Tracking Performance of CTracker, trained on 10 video clips with 100 epochs

| | Rcll | Prcn | GT | MT | ML | FP | FN | IDs | FM | MOTA | MOTP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CTracker | 98.0% | 93.2% | 44 | 38 | 1 | 584 | 161 | 134 | 52 | 89.2% | 0.820 |
| CTracker/p | 96.8% | 95.3% | 44 | 37 | 1 | 400 | 266 | 101 | 45 | 90.9% | 0.779 |

#### 4.5.2.2 SlowFast Evaluation

The classification performance of `SF_PvB` and `SF_SvO` are measured on a test set of 167 clips (with ground truth ROI sequences for 167 subjects and objects and 1008 bystanders). Precision-recall curves for each network trained on *tight* tracker ROIs vs. the looser *context* ROIs discussed in Sec. 4.4 are plotted in the first row of Fig. 4.11. For both training regimens, `SF_PvB` is nearly perfect, with an average precision (AP) of 0.997 for tight ROIs and 0.999 for context ROIs after 10 epochs. The subject vs. object task seems harder, as blame is hard to be attributed to either one of the two tussling players. And while foul objects often wind up sprawled on the ground, so do the foul subjects whether intentionally or not. This assessment is borne out of `SF_SvO`'s lower performance after 40 epochs of training, with an AP = 0.749 for tight ROIs and 0.861 for context ROIs.

The context variant of `SF_PvB` successfully detect 64.24% of foul participants @ 0.5 IoU threshold at the foul moment over a test set of 167 clips (vs. 52.51% for the tight variant with the same tracks). Fig. 4.1 shows three examples of such detections. The second row demonstrates the detector's ability to pick out one anomalous motion in a crowd (in this case the foul object sinking to the ground). Subjects and objects are detected at the same IoU threshold with 30.15% and 45.21% accuracy, respectively (16.39% and 30.06% for tight). The detection accuracy is considerably higher at lower IoU thresholds (e.g. 84.34% @ 0.1 IoU), indicating that this approach locates the rough foul area quite robustly.

The classification performance of `SF_BvSvO` on 3 categories is also measured on the same test set as the experiment of `SF_PvB` and `SF_SvO`. We also compare tight ROIs with context ROIs. The second row in Fig. 4.11 plots the precision-recall curve of testing results on *tight* tracker ROIs vs. the looser *context* ROIs. Similar as the cascade method, the AP of bystander is almost perfect. Moreover, the performance on the context ROIs is better than the tight ROIs (90.4% vs. 89.4%). As the similar average precision on bystanders, the context ROIs improves the classification especially for the foul subject and object.

Video quality also impacts the performance, as the left figure in the last row of Fig. 4.11 shows . As SoccerNet provides the original high quality videos, the corresponding foul clips in HQ are also extracted. We do the same experiment on context ROIs in HQ version. Because of the variety of frame rates in HQ videos of SoccerNet, we sample frames of the extracted video clips such that all frame rates are 25 fps. And we apply CTracker model on these HQ frames to extract sequence of player patches. For the 3 categories task (bystanders vs. foul subjects vs. foul objects), the mAP increases to 91.8%.

Considering the fact that significant overlap between foul subject and object context bounding boxes for a lot of frames of a lot of fouls, this means that a video of an object being fouled often includes, very close by, the subject who is fouling him (and vice versa for the subject video). We can tell which is which by who is closest to the center of the bbox, but sometimes these videos are almost identical. So saying that one of them is 100% subject and 0% object doesn't seem right – and makes the training task harder. So the multi-label is proposed to say subject videos with 'how much' object is in them and vice versa. In our experiment, the AP is 98.0% on the multi-label classification, as the right figure in the last row of Fig. 4.11 depicts.

### 4.5.3 Field Localization

We evaluate the camera calibration on our annotated foul clips. We evaluate this task by computing ECC between the distance image computed from edge Two-GAN image and another distance image computed from synthetic camera generated by the camera pose engine.

We follow the specification in Two-GAN [16] to calibrate cameras. On 211 foul video clips (10117 frames in total), the average ECC is 93.0%. If we look into the evaluation by games (foul clips in two halves), the worst ECC is about 74.1% and the best one is about 98.4%.

There are still some drawbacks for the camera calibration. The scoreboard and watermarks in broadcast videos may heavily affect camera calibration if they are not

filtered out by the field mask. Figures in last two rows in Fig. 4.9 depict that the scoreboard and watermark on the 'grass' region in the broadcast video will easily cause failure.

### 4.5.4 Clustering Evaluation

Fig. 4.10 shows that the input of clustering algorithms is the normalized histogram of colors on torsos. We evaluate the clustering performance based on the annotated torsos from the subset.

We firstly evaluate the performance of the two clustering algorithms – K-Means and Agglomerative clustering approaches on foul clips from the 5 games. The K-Means is to fit on all 'torsos' from the first foul clip in the corresponding game with different K values, and test it on 'torsos' from rest foul clips. The agglomerative clustering is to fit and predict on 'torsos' from all foul clips in each game. In this experiment, we set the number of clusters K to be 2 or 3 for both approaches.

We threshold the area of the torsos for filtering out wrong detections during fitting and predicting. We don't put them into clustering if they are too large or too small by setting two thresholds $max_p$ and $min_p$. $max_p$ is the maximum number of pixels of a mask region and $min_p$ represents the minimum number of pixels. In our task the $max_p$ is set to the 6 times the median bounding box area in a frame, and the $min_p$ is 80.

Tab. 4.4 shows that the comparison of these two clustering algorithms in different number of clusters (K is 2 or 3, the number of bins is 5 for each channel). Both algorithms have the similar performance when setting the K value to be 3. Please note that the Agglomerative algorithm runs clustering on all 'torso' regions from all foul clips, but K-Means is fit by the 'torso' regions from the first foul clip.

If masks output from AlphaPose contain a lot of 'noise' that mainly come from other persons out of the field, it will be sensitive and error-prone if K is 2, unless ignoring the influence from the referee and both goalkeepers (they wear different colors) and restricting persons who are on the field, hence it would be difficult to get correct

clusters. Tab. 4.4 shows that, when K is 2, the clustering for clips in the game – *'Schalke vs. Dortmund'* will enroll a lot of irrelevant 'torsos'. See the top-right figure in Fig. 4.12, a lot of false 'torsos' are from the billboard at the far-side. These false 'torsos' coming from out of the field make the clustering difficult. Moreover, some clips only contain players from two teams without goalkeepers. The clustering accuracy will decrease if the K is more than 2.

**Table 4.4:** Comparison of Agglomerative and K-Means Clustering (K is the number of clusters), number of histogram bins is 5 (per channel)

| Game | Aggl. Clustering (K=2) | Aggl. Clustering (K=3) | K-Means (K=2) | K-Means (K=3) |
|---|---|---|---|---|
| Real Madrid vs. Las Palmas | 85.2 | 94.4 | 85.2 | 94.3 |
| Schalke vs. Dortmund | 71.7 | 81.5 | 70.1 | 69.2 |
| Bayer Leverkusen vs. Dortmund | 86.5 | 71.5 | 80.3 | 84.2 |
| Besiktas vs. Napoli | 79.6 | 85.5 | 79.6 | 85.3 |
| Dortmund vs. Galatasaray | 89.0 | 85.6 | 60.0 | 86.2 |

To get rid of the influence of persons who are out of the field, we firstly apply the mask on the broadcast frame and use the estimated homography matrix to project the broadcast frame on a field template.The field edge can clearly draw on the top-on view after applying homography warping. We estimate detected persons' foot positions by the coordinates of their bounding boxes, and transform the foot positions to top-on view for estimating their field positions. Due to the warped image may not perfectly match the field template, we add 15 pixels for each side (top, bottom, left and right) for tolerating persons who are out of the field template but within the range that is not further from the boundary than 15 pixels, see Fig. 4.13. We keep persons who are on the field (also in the range of tolerance) and use their torsos to do clustering.

Fig. 4.12 renders that a lot of out of field persons and false positives can be eliminated if the camera calibration result is good. In our experiment, about 85% - 95% regions out of the field are filtered out with camera calibration.

Tab. 4.5 lists our results after filtering by the field template. The number of bins have minor effects on the clustering performance, Tab. 4.6 illustrates clustering results on different number of bins. We set K to 3 for this experiment because most frames don't contain goalkeepers. These experiments show that the clustering method enables

the differentiation in jersey colors without the prior knowledge. Fig. 4.14 shows some examples of our clustering results. The example frames are from 6 different video clips that are from 5 different games. At the far-side, players are usually very small such that the torso color distribution may vary drastically based on their poses. And the clustering result also depends on the performance of the Two-GAN model for camera calibration and the AlphaPose model.

**Table 4.5:** K-Means Clustering Accuracy (%) after filtering by camera calibration ( $bins = 5$)

| Game | $K = 2$ | $K = 3$ | $K = 4$ | $K = 5$ | $K = 6$ |
|---|---|---|---|---|---|
| Real Madrid vs. Las Palmas | 89.4 | 94.2 | 79.5 | 78.9 | 79.0 |
| Schalke vs. Dortmund | 83.4 | 74.4 | 64.3 | 63.8 | 60.8 |
| Bayer Leverkusen vs. Dortmund | 81.0 | 84.4 | 85.4 | 81.5 | 76.4 |
| Besiktas vs. Napoli | 81.8 | 88.4 | 85.9 | 76.2 | 52.1 |
| Dortmund vs. Galatasaray | 86.6 | 92.2 | 77.6 | 61.8 | 53.0 |

**Table 4.6:** K-Means Clustering Accuracy (%) after filtering by camera calibration ( $K = 3$)

| Game | $bins = 3$ | $bins = 4$ | $bins = 5$ | $bins = 6$ | $bins = 7$ | $bins = 10$ | $bins = 15$ | $bins = 30$ |
|---|---|---|---|---|---|---|---|---|
| Real Madrid vs. Las Palmas | 94.3 | 94.2 | 94.2 | 93.7 | 93.7 | 93.7 | 93.7 | 93.7 |
| Schalke vs. Dortmund | 76.8 | 74.2 | 74.4 | 73.8 | 73.5 | 73.5 | 73.0 | 73.0 |
| Bayer Leverkusen vs. Dortmund | 83.0 | 83.9 | 84.4 | 84.9 | 85.4 | 85.6 | 85.6 | 85.3 |
| Besiktas vs. Napoli | 88.0 | 87.9 | 88.4 | 88.4 | 88.4 | 88.4 | 88.4 | 88.4 |
| Dortmund vs. Galatasaray | 92.4 | 92.4 | 92.2 | 92.6 | 92.1 | 92.6 | 92.0 | 92.0 |

## 4.6 Conclusion and Future Work

We report strong performance on a sports spatiotemporal video activity recognition task. There are a number of directions to take before removing the foul oracle assumption and working on the scale of entire games, including extending the system to near-view clips with more training examples, dealing with shot boundaries automatically, and incorporating foul-relevant information outside of foul subject and object bounding boxes. We also show the experiment of the differentiation among players by clustering approach with assistance by using human pose estimation and camera

pose estimation and parsing of field line features [20]. The experiment shows promising results for further usage in making sure candidate pairs of foul subject and object are on opposite teams so that it could boost performance. Other attempts may also be feasible in differentiating players but require a per-game learning of jersey colors and patterns using, for example, deep image clustering [78]. Using high-res versions of the game videos would enable further analysis such as ball tracking and reading player names/jersey numbers to correlate with roster data and/or commentary.

In the future, we also need to consider how to utilize the clustering way into the multi-object tracking, especially for the player association/re-identification in highly overlapped scenarios cross frames, since all persons on the field would wear definite colors in soccer games.

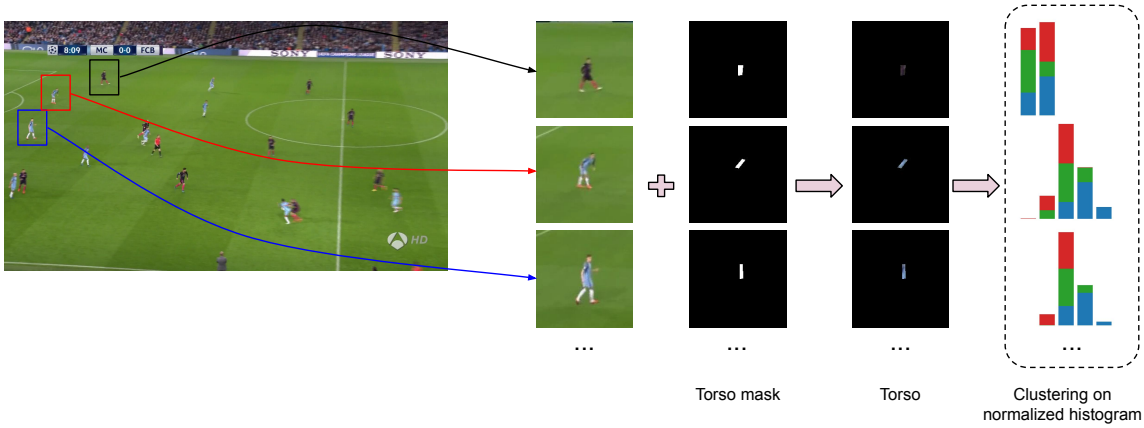


**Figure 4.10:** We estimate poses for each players' tracks. We generate torsos by AlphaPose and extract torso colors to do clustering.

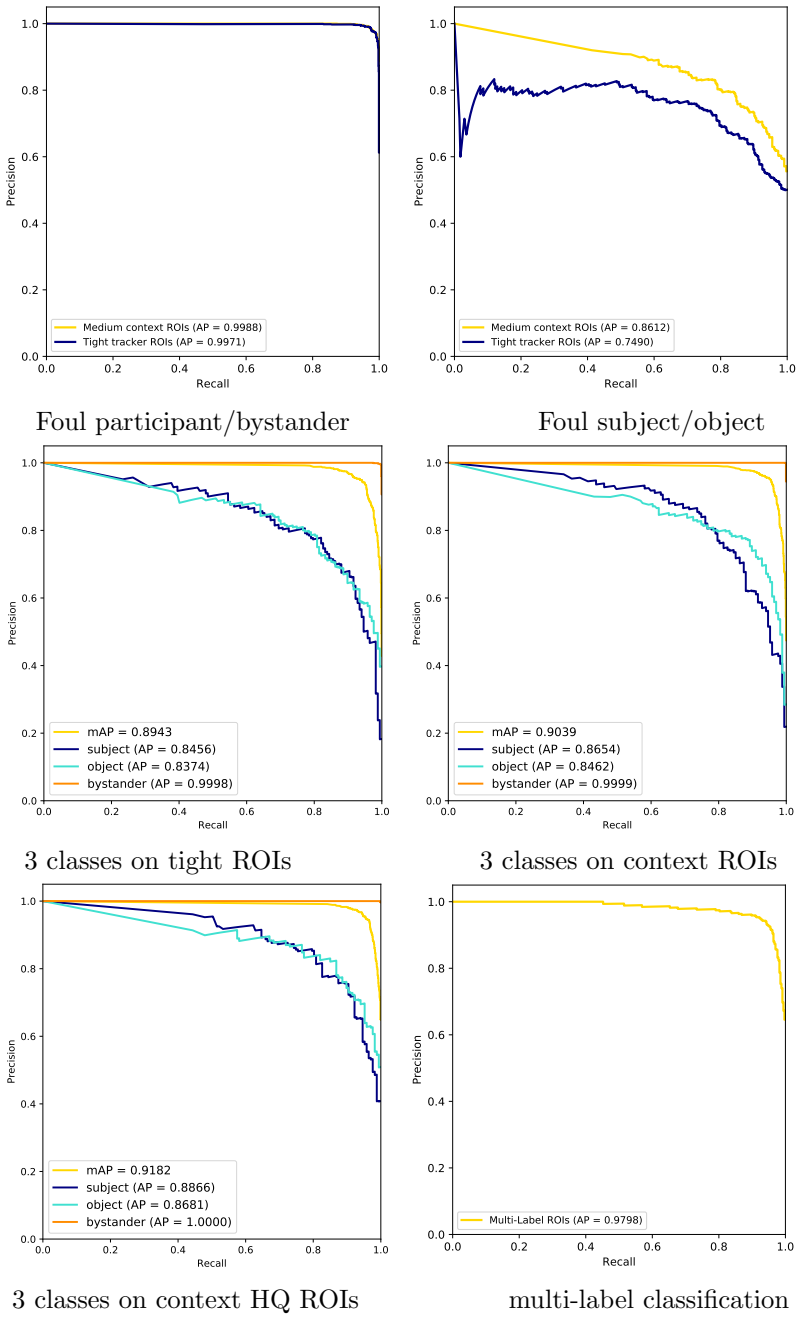


**Figure 4.11:** Precision-recall curves for SlowFast action classifiers. Two curves at the first row are for foul participant/bystand and foul subject/object. In the middle row, we can find that the context ROIs (right) is better than the tight ROIs on 3 classes. Resolution also has influence on the classification performance (bottom-left). And we also do multi-label classification, the PR curve is at bottom-right.

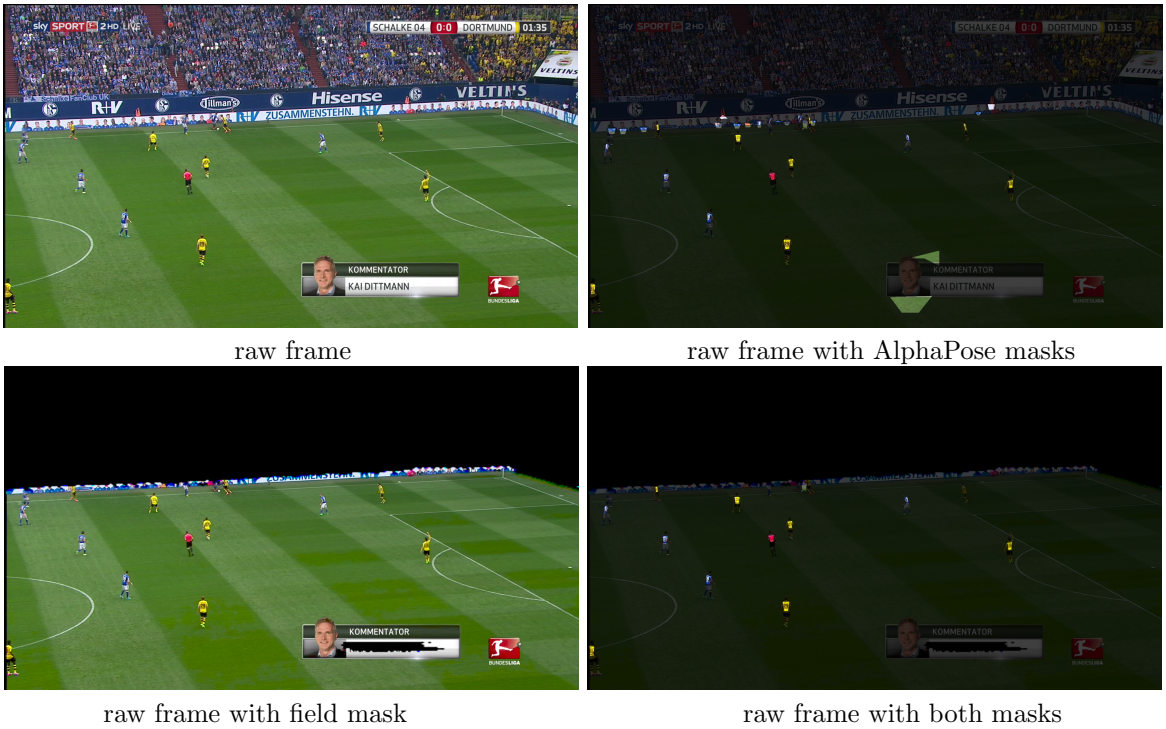


**Figure 4.12:** Example frame in the game '01-04-2017 Schalke vs. Dortmund'. AlphaPose model detects person over the entire frame without limiting on the region of the field (as top-right shows). The predictions contain a lot of false positives. By Two-GAN model, we estimate the boundary of the field to eliminate the effects from detected persons out of the field(For observing the field boundary, we darken the field region but highlight detected 'torsos' instead of blacking the entire field, as bottom-right shows.)

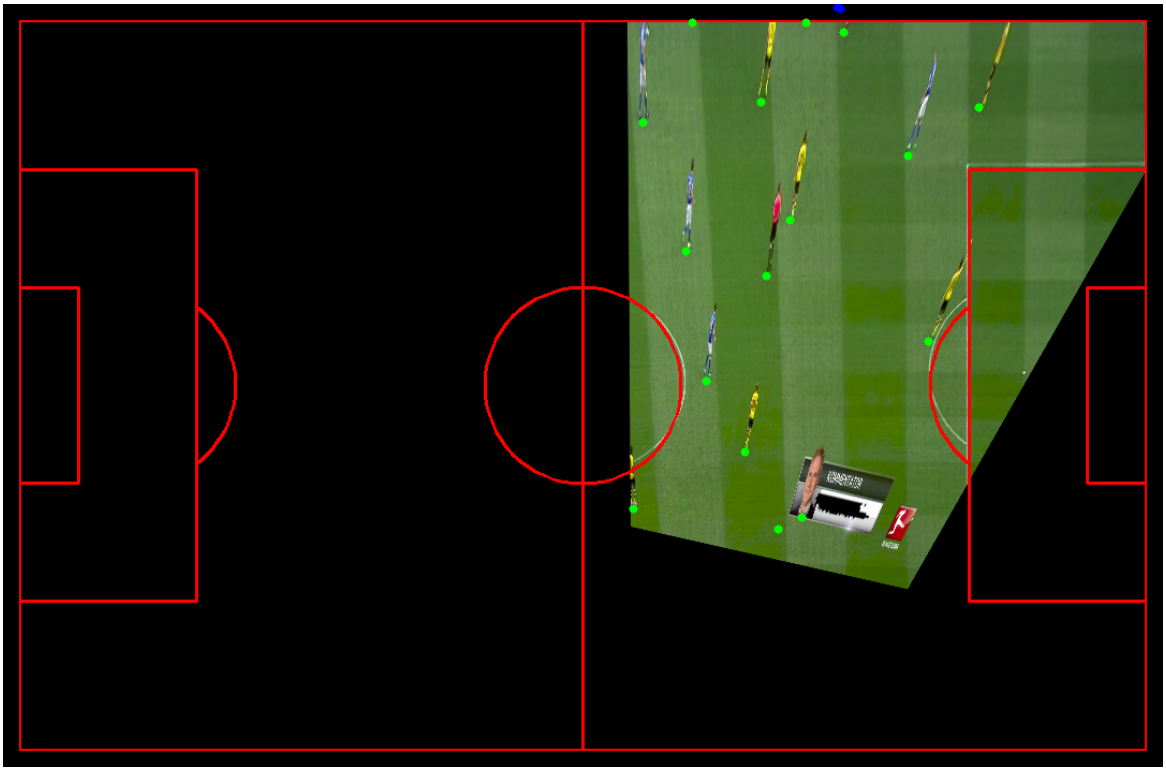

**Figure 4.13:** Field template overlaid on the image using the estimated camera poses. Using the predicted bounding boxes, we estimate the players' positions at the top-on view. Persons who are out of the field can be easily filtered after the project. Persons on the field (in green and blue spot) are kept for differentiating their jerseys' color.


**Figure 4.14:** Examples of the K-Means clustering result after applying the field mask ($K = 3$, $min_p = 80$) from 5 games. We use bounding boxes detected by AlphaPose (YOLO detector) with different colors to represent they are in different clusters. Torsos without bounding boxes represent they are ignored. In **1**, some players wearing blue are mis-clustered. It is error-prone if detected players are usually small (especially for their positions are at far-side). In **2,5,6**, all players on the field are correctly clustered by their jersey colors. **3** has 3 players not detected by the detector. **4** shows players near the side line (far-side) may be filtered out by the GAN mask.

# Chapter 5

# TOWARDS GENERAL ACTION SPOTTING IN SOCCER GAMES

In this chapter, depending on SoccerNet V2 dataset, we design and experiment a new architecture to detect action spotting across the entire game video by combining a state-of-the-art video classification network and an action spotting network. Firstly, the action clips are extracted in 3-second duration from raw videos in 25 fps. Due to the highly imbalanced number of clips in different categories, the classification accuracy is only 41% on 17 action categories and 1 'background' trained on randomly picked 20 games (40 game halves). Considering longer duration can provide more temporal contextual information, we adopt the manner used in SoccerNet by making multi-label classification on long duration video chunks. But it requires pre-trained features at frame level. To efficiently train and make inference on actions from raw videos, we also try to use the video classification network to extract features and feed to the temporal action detection network. Unfortunately, during the training, the temporal action spotting loss doesn't explicitly decrease, the loss in multi-label classification decreases though.

Even though our architecture doesn't work as expected, we still believe that the value of this experiment cannot be ignored. We will give our explanation and analysis for the attempts in the following. Based on the current architecture, possible solutions are also provided for the future work.

## 5.1 Introduction

Nowadays, computer vision techniques have been applied into a massive number of applications. Such applications facilitate human's daily life and make many works

much easier than before, like beautifying the selfie, traffic signs recognition, medical treatment, autonomous driving, and images/videos analysis.

These new techniques are also brought into sporting games. For sports video analysis, computer vision is becoming ubiquitous with applications that include broadcast enhancement; real-time, in-depth player and team performance measurement; and automatic summarization of key events. Across analysis tasks there are several common visual skills such as ball tracking [132, 96]; player segmentation [13, 89, 66], recognition [43], and pose estimation [71]; and recognition of formations, plays, and situations [4, 135, 137, 44]. All of these make progresses towards automatic refereeing systems.

The automatic refereeing system will recognize and detect players actions during the game. In human action recognition and detection, early methods [72, 106, 102] mainly make use of handcrafted features from videos and linear classifiers. Deep learning based methods [50, 39, 38, 157, 34] aim at localizing and identifying human actions over videos, either being developed on or incorporating 2-D object detection frameworks.

The event as an often used but seldom well defined term, in computer science and applied fields, is mainly used as a 'specific occurrence involving participants' [161] by adding human or agent factor. This definition is similar as the human action but may be more abstract or in high semantic level. Deep neural network has been widely applied into detecting events in videos as well. In early deep neural network applied in this area, the Fisher vector and Vector of Locally Aggregated Descriptors (VLAD) are still used in encoding CNN features [158, 115, 2]. Most work build their event detection model upon the network of detecting human actions with some modifications like pooling, encoding etc., to aggregate features.

In soccer games, different actions are usually not about the concrete actions but some specific events[1] since the actions are described as some occurrences that have participants. For example, a foul typically consists of two participants, a ball goes

[1] In this chapter, we use the term event and action interchangeably

out of the field means the ball is the 'participant'. Once an event occurred, players or other people may have different reactions. Exploring how to detect these events or actions in soccer game videos is still a challenging problem because it requires a deeper understanding of these game actions. In soccer videos analysis, [44] created a benchmark for the temporal action spotting. In their first version, 3 different actions are annotated in 500 complete soccer games – Goal, Substitution and Card. The second version [26] expands to 17 different action annotations in the same 500 games. Fig. 5.4 shows some visible examples about different actions in game videos.

In [44, 19, 26], the actions are anchored with a single timestamp. Unlike common action localization, the duration of the action spotting is uniformly 1 second. This will ignore the boundary of an action as well as its duration. And, broadcast producers often switch cameras or replay some occurrences to either capture or render more meaningful context. This will result in that some actions are invisible at the moment they occurred. In SoccerNet, each timestamp of an action spotting has a binary visibility tag that states whether the associated action is shown in the broadcasting video or unshown, in which case the action must be inferred by the viewer.

In SoccerNet [44, 26, 19], they use temporal convolutions and pooling upon features from a pre-trained ResNet-152 model on 3 categories (goals, cards and substitutions) to detect actions. They identify each action spotting by one frame. To train their network, the raw video is first sub-sampled to 2 fps from 25 fps in their low-resolution videos. They also apply PCA to reduce the feature from 2048 to 512 for each frame of the sub-sampled videos. They train their temporal action spotting network on batches of 2 minutes long video chunks. Each chunk is around a ground truth action. In their observation, there are no chunks of 2 minutes containing more than 5 actions. Thus the network will give 5 predictions on each chunk.

Although SoccerNet's results [26, 19] are great, their sophisticated processing is still a huge obstacle to extend to more games. In this chapter, we will describe our network architecture for end-to-end training and prediction. As depicted in Fig. 5.1, raw videos are the input and the network predicts the actions occurred in a given chunk

and localize these actions. This new network is built by combining SlowFast network and the temporal action spotting network. Since deeper ResNet can produce better performance on other public dataset and the backbone of the temporal action spotting network is ResNet-152 in SoccerNet [44, 26, 19], we choose the deepest ResNet that has the pre-trained model as the backbone of SlowFast network. However, the experiments show that the new network doesn't work as we expected. We will give our analysis and possible solutions to handle this problem.

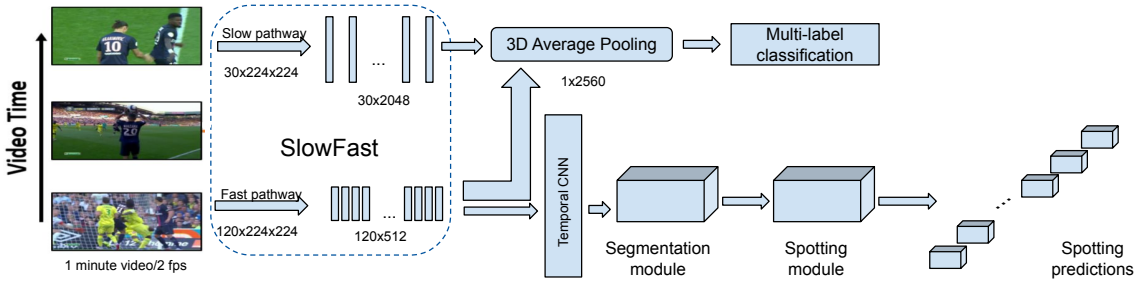


**Figure 5.1:** We designed the architecture for detecting action spots over the game video. It combines SlowFast network and the temporal action detection network of SoccerNet V2 [26]. We follow the temporal action spotting network [19] which consists of a segmentation module and a spotting module. It takes the feature vectors from SlowFast's fast pathway as inputs to the segmentation module and the spotting module. The feature vectors are also fused with other feature vectors from the slow pathway for multi-label classification Even though the loss in multi-label classification decreases during the training, the temporal action spotting doesn't work.

## 5.2 Related Work

Action recognition could either apply 2D convolutions on per-frame input followed by another 1D module for aggregating the features[69, 122] or apply stacked 3D convolutions to model temporal and spatial features [134, 14]. In single stream manner, [34] uses two different pathways to operate on different frame rates for capturing both spatial semantics and temporal motions.

Action detection mainly discusses how to localize actions across an untrimmed video (include recognize the action). Many studies in this year applied some approaches which are similar to some methods of object detection. For temporal action detection, some studies build their approaches upon 3D features, Convolutional-De-Convolutional (CDC) network [119] places CDC filters on top of 3D ConvNets for abstracting action semantics but reduce the temporal length of the input data. Cascaded Boundary Regression model [40] is proposed to deal with a problem that traditional sliding windows may not cover the entire action instance. To address an issue about common "detection by classification" method which the boundaries of action instance proposals have been fixed, a single-shot Action Detector (SSAD) makes use of 1D temporal convolutional layers to skip the proposal generation step via directly detecting action instances in untrimmed video[79]. Region Convolutional 3D Network is introduced to encode the video streams and generate candidate temporal regions containing activities, and classify selected regions into specific activities[156].

Relationship also implies actions and positions of participants. Recently, learning the relation between objects has attracted many researchers to put efforts on detecting and recognizing the relation between objects and the interaction between the human and the object [54, 49, 64, 94, 159, 21, 173, 150]. Describing the relationship between objects is crucial to determine the global interpretation of the scene. Human actions, especially for the interaction between the human and the object, are more specific and an individual person can perform multiple actions simultaneously.

[85] demonstrates high-quality spatial-temporal activity detection in a surveillance video scenarios, and more and more state-of-the-art methods have been utilized in the area of sports. [136] introduces a multi-tower temporal 1D convolutional network to detect events in ice hockey game and soccer game videos. [63] constructs their model based on deep reinforcement learning that shows only part of people's activities have impacts on the entire group and tests their model on volleyball videos. [116] uses self-attention models to learn and extract relevant information from a group of soccer players for activity detection from both trajectory and video data.

It is necessary to have large-scale datasets for training deep video understanding models. While many early works utilize custom datasets which are usually small and private, some available dataset such as MLB-Youtube[105], SoccerNet[44], Sports-1M[69], UCF Sports[113] and etc. are still worth mentioning. [44] annotates action spotting in three categories: *goal, card*, and *substitution*, in 500 popular games from 2014 to 2017. As the extension, [26] annotates 17 actions in the same 500 games and they establish a benchmark of detecting these actions over the entire game video. They utilize features provided with the [44] and propose a context-aware loss function (CALF) for detecting actions in the videos. The loss function heavily penalize the frames far-distant from the action and decrease the penalty for those gradually closer. Without penalizing the frames just before the action to prevent misleading information, CALF [19] heavily penalize those just after as the action has occurred. In this task, we utilize the CALF to compute the loss in detecting action spots.

Inspired by the action recognition and the detection, in this chapter, our work is mainly built upon these models for efficient training and inference.

### 5.3 Actions in SoccerNet V2 Dataset

Based on SoccerNet V1 dataset, the v2 version consists of 300k timestamped annotations temporally anchored within SoccerNet's 764 hours of video of 500 games. It significantly extends the actions of SoccerNet V1 with 16x more timestamps and 14 extra classes[26]. In total 110,458 actions are annotated in this version, on average 221 actions per game. Regardless the background (we also call it 'common playing' in this task), totally 17 types of actions from the most important in soccer are identified. Each action of the 500 games is annotated with a single timestamp in SoccerNet V2. Fig. 5.2 shows the number of actions annotated in SoccerNet V2. Action 'ball out of play' and 'throw-in' account for more than 45%, 'red card' and 'yellow-red card' only have about 50 instances. This results in heavy data imbalance.

Visibility is another issue of affecting the recognition and detection, some actions are actually performed but they are invisible at the moment. Fig. 5.3 shows the

distribution of visibility across all actions in SoccerNet V2. Some actions like 'kick-off', 'clearance' and 'indirect free-kick' account for around 50% invisible actions. Recognizing unshown actions is challenging since it needs a better understanding on the context of the game.

SoccerNet V2 also annotates camera change timestamps comprehensively in a subset of 200 games, the others contain the replay shots in the remaining games. In the fully annotated games, each game has 583 camera transitions on average. Unlike the previous tasks we did, the shot transition introduces more challenging difficulties. Different types of transitions from one shot to its next occur even in the same game video. The statistics provided in SoccerNet V2 says that the camera transition can be abrupt changes between two cameras (71.4%), fading transitions between the frames (14.2%), or logo transitions (14.2%) [26].

## 5.4 Methods

### 5.4.1 Classification on 3 Seconds Video Clips

We utilize SlowFast network to make classification on video clips because it achieves strong performance for action classification in videos. SlowFast network operates at two different framerates but works as a single stream architecture [34]. The slow pathway operates at low frame rate by a large temporal stride $\tau$ on input frames. Only one out of $\tau$ frames is processed at this pathway, so that, it is designed for mainly focusing on the spatial domain and semantics. Unlike the slow pathway, the fast pathway operates at high frame rate. And it has a ratio $\beta(\beta < 1)$ channels of the slow pathway. The fast pathway is lightweight for capturing temporal contextual information.

### 5.4.2 Temporal Action spotting

We combine SlowFast network [34] to the temporal action spotting network proposed in SoccerNet V2 [26]. In Fig. 5.1, the slow pathway take sub-sampled frames of the video chunk to focus on the spatial domain and semantics. The fast pathway will not perform any temporal pooling to maintain temporal fidelity before the prediction

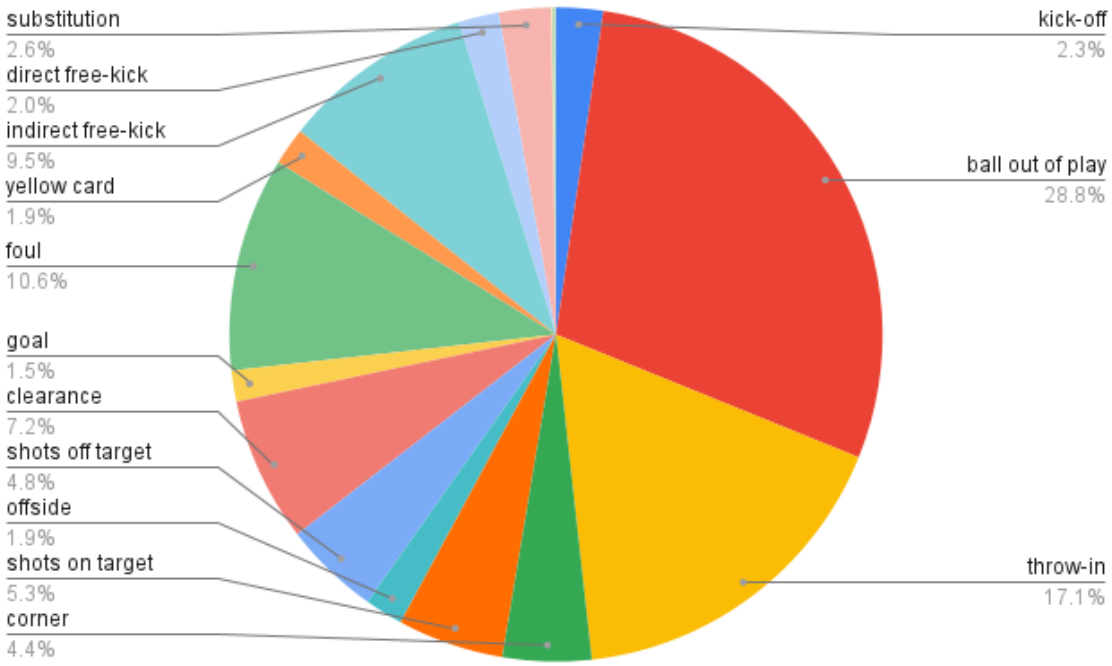


**Figure 5.2:** Distribution of actions annotated in SoccerNet V2. The major actions such as 'Throw-in', 'Ball out of play' and 'Foul' have more than 10000 instances. But 'Red card', 'Yellow− >red card' only have around 50 instances.

head. Same as the setting in SlowFast network, these two pathways are fused by lateral connections.

In our architecture design, SlowFast network part has two heads – one head takes outputs from its slow pathway and fast pathway to do multi-label classification, another head doesn't perform average pooling on the temporal dimension to keep the number of input frames in the chunk and it takes the features from the fast pathway to the temporal action detection network. The duration of a video chunk is typically 30 seconds to 2 minutes long (in our experiment, the duration is 1 minute). So a chunk can contain more than one actions annotated in SoccerNet. And the actions can be in the same category. Regardless of the number of occurrence of actions in a chunk, we only consider if it occurs (1) or not (0) in the multi-label classification branch. We

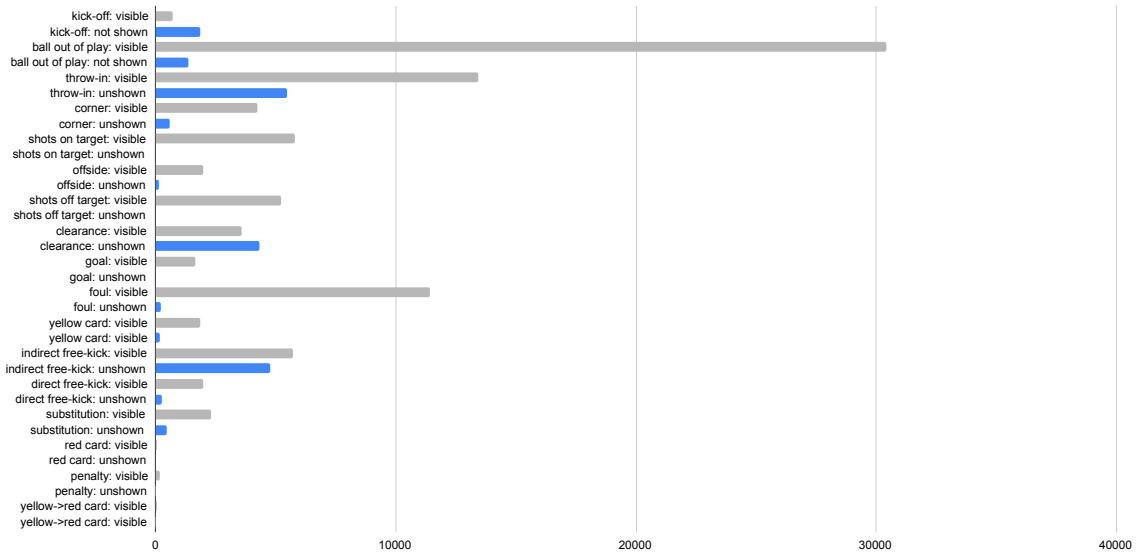


**Figure 5.3:** Visibility distribution of actions annotated in SoccerNet V2.

basically follow the setting for both pathways in the original SlowFast network [34], except for the feature size after the last ResNet stage in the fast pathway, it is expanded to 512 from 256 in order to contain more spatial context. Please note that 512 is also the feature size in [26, 19] after PCA reduction. And we don't make pooling on the temporal dimension at the fast pathway for keeping awareness of spatial contextual information per frame.

We follow the temporal action spotting specified in [19, 26] for our detection. The network is made of a frame feature extractor and a temporal CNN outputting $C$ class feature vectors per frame, and two modules – a segmentation module and a spotting module. The segmentation module produces a segmentation score per class for each frame. It transforms feature vectors from the temporal CNN into an output of dimension $N_F \times C$. The segmentation scores output by this module is assessed through the segmentation loss specified in [19]. The spotting module takes as input feature vectors from the temporal CNN and the segmentation scores, and outputs the


**Figure 5.4:** Examples of visible actions annotated in SoccerNet V2 dataset, consist of 'kickoff', 'goal', 'substitution', 'offside', 'shots on target', 'shots off target', 'clearance', 'ball out of play', 'throw-in', 'foul', 'indirect free-kick', 'direct free-kick', 'corner', 'yellow card', 'red card'. The category – yellow to red card is rare, it is not in our 40 games for both training and testing

spotting predictions of the network.

In the temporal action spotting network, we follow the time-shift encoding for temporal segmentation [19]. The segments regroup frames by if they are far before, just before, just after, far after an action, or in transition zones between these segments. In [19], for each category $c$, the temporal segments are delimited by specific slicing parameters and are materialized through time-shift encoding. We also follow the temporal action spotting network proposed by SoccerNet, which consists of a segmentation module and a spotting module.

## 5.5 Experiments and Analysis

### 5.5.1 Data Preparation

SoccerNet splits 500 games into 3 sets – 300 games in the training set, 100 games in the validation set and 100 games in the test set. Due to the huge amount of actions in the original SoccerNet dataset, we randomly select 20 games from SoccerNet's training set to train our network and randomly select 20 games from SoccerNet's test set to do testing. There are 4761 and 4816 actions in our training set and test set, respectively. As depicted in Fig. 5.6, the shapes of their distribution in different action categories are almost the same. We generate 'background' actions in the average number of frames in different action categories. We also randomly flip frames by videos to alleviate the imbalanced data.

### 5.5.2 Training

We extract chunks from raw videos and sub-sample them to 2 fps to make our inputs. Due to the memory limit, we adopt the chunk size as 1 minute – 120 frames per video chunk. The feature extraction at the fast pathway is after the average pooling, such that $7 \times 7$ at each feature depth is pooled to 1.

The duration of each chunk is 1 minute long. Based on the observation [19], there are no more than 5 action spotting in each 2-minute video chunks. The chunk is anchored at an event along with corresponding shifts. The input chunk is sub-sampled

to 2 FPS and frames are resized to 224x224. Two pathways of SlowFast network have different settings for different purposes. Fast pathway mainly captures temporal contextual information and the slow pathway focuses on spatial contextual information. As the original setting in SlowFast, the slow pathway further down-samples the input video to 0.5 fps but keeps more spatial features. The fast pathway keeps the sample temporal dimension but down-samples the output feature to 512 (in original SlowFast version, the feature size is 256). Outputs from both pathways are combined for the multi-label classification. The loss function is binary cross entropy for the multi-label classification. The fast pathway also provides inputs for the action prediction. The action prediction part follows the network used in SoccerNet v2 benchmark. The features from SlowFast network are input to a spatio-temporal pyramid to produce 120 features for each frame.

And we use loss function (CALF) proposed by [19] to calculate the segmentation loss and spotting loss. Including the multi-label classification loss, all losses are added together for the stochastic gradient descent (SGD) optimization with an initial learning rate $lr = 10^{-3}$.

Due to the memory limit and saving time of video decoding, the training processing takes one game half by one game half at each training epoch. It means that we extract all chunks over the game half video. We also extract 'background' chunks that don't contain any actions. The total number of extracted 'background' chunks is the average number of frames in different categories. Extracted chunks are evenly sampled by the category to prevent the data imbalance. We train the network 100 epochs which takes about 4 days.

Considering the temporal action spotting network takes features from ResNet-152 as the input, and SlowFast network only provides pre-trained models on ResNet-50 and ResNet-101. We use the ResNet-101 as the backbone of SlowFast network. And the model is trained on AVA dataset [25].

### 5.5.3 SlowFast Network for Classification on 3-second Video Clips

We experiment action classification on SlowFast network but the performance is far from satisfactory. The classification model is trained on the same 20 games as the games used in training temporal action spotting model. In this experiment, we anchor the action at the exact time in game seconds provided by SoccerNet V2 annotation. And we randomly extract 'background' clips from the game half video. The number of 'background' is the average number of frames in different action categories. The same way of extracting 'background' is also used in training the action spotting model.

To make each clip contain more contextual information, we expand each clip before and after by 1 second. Each 3 seconds video clip is 25 fps. To make it able to evenly sub-sample frames over the video clip for the slow pathway in SlowFast network, we extract and append 1 extra frame after the corresponding clip. All frames in the clip are resized to $224 \times 224$ before feeding into the network. We fine-tune SlowFast network pre-trained on Kinetics dataset. The applied backbone network of SlowFast is ResNet-50. We run the training 100 epochs and follow other parameter settings as SlowFast [25]. The testing shows that the classification accuracy is about 41%. Fig. 5.1 depicts the confusion matrix on the testing of the classification result.

We believe the reason of unsatisfactory performance is the lack of sufficient contextual information. As mentioned in [44], training on smaller windows will result in a drop in performance.

### 5.5.4 Temporal Action spotting and Analysis

We follow the measurement proposed by SoccerNet [44]. An action spot is defined as positive if its temporal offset from its closest ground truth is less than a given tolerance. The training of our network meets a problem that the training loss in the action spotting just fluctuates, but the loss in multi-label classification decreases as we expected. This results in our testing result is bad since the Average-mAP is only around 4%.

**Table 5.1:** Confusion matrix on 3 seconds video clips classification

| | Penalty | Kick-off | Goal | Substitution | Offside | Shots on target | Shots off target | Clearance | Ball out of play | Throw-in | Foul | Indirect free-kick | Direct free-kick | Corner | Yellow card | Red card | Yellow->red card | background |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Penalty | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| Kick-off | 0 | 60 | 0 | 11 | 0 | 0 | 1 | 13 | 1 | 8 | 0 | 7 | 1 | 0 | 2 | 0 | 0 | 1 |
| Goal | 0 | 1 | 21 | 0 | 0 | 2 | 0 | 7 | 29 | 0 | 3 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| Substitution | 0 | 5 | 0 | 49 | 3 | 0 | 0 | 10 | 4 | 11 | 0 | 26 | 0 | 0 | 3 | 0 | 0 | 0 |
| Offside | 0 | 0 | 4 | 0 | 13 | 5 | 5 | 5 | 33 | 1 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 0 |
| Shots on target | 0 | 1 | 56 | 1 | 3 | 22 | 28 | 18 | 87 | 1 | 7 | 1 | 3 | 3 | 0 | 0 | 0 | 0 |
| Shots off target | 0 | 1 | 52 | 0 | 8 | 19 | 34 | 8 | 79 | 1 | 0 | 2 | 3 | 4 | 0 | 0 | 0 | 0 |
| Clearance | 0 | 39 | 0 | 39 | 4 | 0 | 3 | 103 | 12 | 33 | 7 | 78 | 5 | 4 | 2 | 0 | 0 | 0 |
| Ball out of play | 0 | 29 | 112 | 21 | 108 | 6 | 11 | 57 | 769 | 97 | 21 | 51 | 1 | 8 | 12 | 0 | 0 | 35 |
| Throw-in | 0 | 31 | 0 | 61 | 26 | 0 | 1 | 34 | 26 | 473 | 2 | 103 | 21 | 6 | 7 | 0 | 0 | 13 |
| Foul | 0 | 7 | 13 | 5 | 110 | 11 | 15 | 20 | 92 | 29 | 153 | 16 | 3 | 0 | 1 | 0 | 0 | 21 |
| Indirect free-kick | 0 | 63 | 6 | 43 | 35 | 0 | 1 | 49 | 12 | 52 | 2 | 122 | 26 | 1 | 6 | 0 | 0 | 15 |
| Direct free-kick | 0 | 2 | 6 | 1 | 0 | 5 | 5 | 5 | 11 | 2 | 0 | 12 | 36 | 7 | 0 | 0 | 0 | 2 |
| Corner | 0 | 5 | 25 | 7 | 0 | 2 | 0 | 19 | 8 | 20 | 0 | 21 | 2 | 78 | 0 | 0 | 0 | 1 |
| Yellow card | 0 | 0 | 0 | 5 | 3 | 0 | 1 | 9 | 3 | 7 | 1 | 49 | 0 | 1 | 13 | 1 | 0 | 2 |
| Red card | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Yellow->red card | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| background | 0 | 11 | 3 | 15 | 49 | 1 | 12 | 24 | 34 | 8 | 6 | 29 | 5 | 2 | 4 | 0 | 0 | 11 |

We think it is because the feature from SlowFast focuses on one timestamp rather than all timestamps in the given video chunk. The original head of SlowFast does 3D average pooling to make the temporal dimension to be 1. We add another head that doesn't sample features on the temporal dimension in order to keep features of all frames. It represents that the shape of features is [1, 2048, 30, 7, 7] for the slow pathway and [1, 512, 120, 7, 7] for the fast pathway, respectively. Here, 30 and 120 are the number of input frames (sub-sampled and not sub-sampled), 1 is the batch size – we cannot increase it due to GPU memory limits, 2048 and 512 are the depths of features of both slow and fast pathways. The original head will decrease the above dimension to [1, 1, 2048] for the slow pathway and [1, 1, 512] for the fast pathway by an average pooling. Then we apply a linear classifier to do the prediction for multi-labels on the video chunk. The head we add is used for feeding features to the temporal action spotting network. Since the slow pathway lacks of temporal context, we only take the output from the fast pathway to the temporal action spotting network.

Because the backbone network of SlowFast is the ResNet, we assume the ResNet features in SlowFast contain more contextual information for each individual frame. As SoccerNet [19] did, the raw video is sub-sampled to 2 fps and features are extracted by ResNet-152 and reduced by PCA to 512 features for each frames of the sub-sampled videos. Despite the temporal dimension is kept in the head we add before the temporal action spotting network, it looks like the contextual information kept in each frame at the fast pathway doesn't support the detection of action spots.

To verify detection of action spots is infeasible, we only train the fast pathway in SlowFast network and feed the output to the temporal action spotting network. But the loss also doesn't decrease during the training. We guess that the fast pathway in SlowFast lacks of enough ability of representing spatial semantics, even though we expand the feature depth to 512.

We didn't try the slow pathway in SlowFast to detect action spots due to its low frequency in frames. Despite the rich ability of representing spatial semantics, 0.5 fps in the slow pathway will ignore a lot of temporal contexts in soccer game videos.

Increasing its frequency will lead to more frames in the video chunk being needed since the fast pathway will require higher frequency. Due to the memory limit, this attempt is prohibitive.

Besides, other issues may also have influence on this bad result. First, the data input manner may be one reason, we randomly select a half video from the training game videos. And we get all frame indices of video chunks on the selected video. Due to highly imbalanced classes distribution and limited RAM capacity, we randomly extract 50 chunks (in training, the input is one by one, rather than sending 50 chunks together to the network). The 50 chunks are selected randomly – we first randomly select classes present in the video to make sure small amount of classes can be extracted, then randomly extract video chunks on the selected class (of course, repeated selection will happen). It means, chunks in some classes are highly over-sampled. And the chunks are extracted from one-by-one half videos in order to save decoding time – 50 chunks from one half are sent to the network, then 50 chunks from another half video are the next. Such input manner may be not good for training. And, in SoccerNet [19], they apply Adam as the optimizer in training the temporal action spotting network.

## 5.6 Conclusion and Future Work

In this chapter, we attempt to provide an end-to-end model for detecting action spots over the entire game video. We build this work upon two successful models – SlowFast network and temporal action spotting network. Unfortunately, our attempt doesn't work as we expected. The training loss in detection part doesn't decrease while the loss in multi-label classification part does. And because of the huge amount of actions annotated in all 300 games for training, the training time is prohibitive (typically it will take more than 15 days on training 40 epochs on these 300 games). All of these block our progress to achieve desired results.

In the future, we still plan to make changes on our current architecture. As depicted in Fig. 5.7, the modification is built upon our current architecture. Because of only one SlowFast network is deployed on video chunks and features in each frames

cannot provide enough temporal contextual information, we plan to split a chunk into several parts (e.g. 5 parts), each part has one SlowFast network to do two things: 1. spotting action classification, 2. assuming each part is 12 seconds long, each SlowFast network outputs features of its corresponding part at the dimension [1, 24, 512] for the fast pathway and [1, 6, 2048] for the slow pathway. In a sliding window way of deploying SlowFast models over the entire video chunk and fusing their output, this attempt can make each model focus on a relatively short video segment without losing temporal contextual information. And weights sharing of SlowFast models is able to make the training feasible.

Moreover, some actions annotated in SoccerNet V2 may take longer than 1 seconds to perform. For example, the action – 'Yellow− >red card'. This will require a better encoding way on ground truth.


**Figure 5.5:** Examples of not shown actions. The unshown actions may be caused by either the camera focuses on other players without noticing the one who is taking the action (left) or the camera looking at other persons (right) or replays.

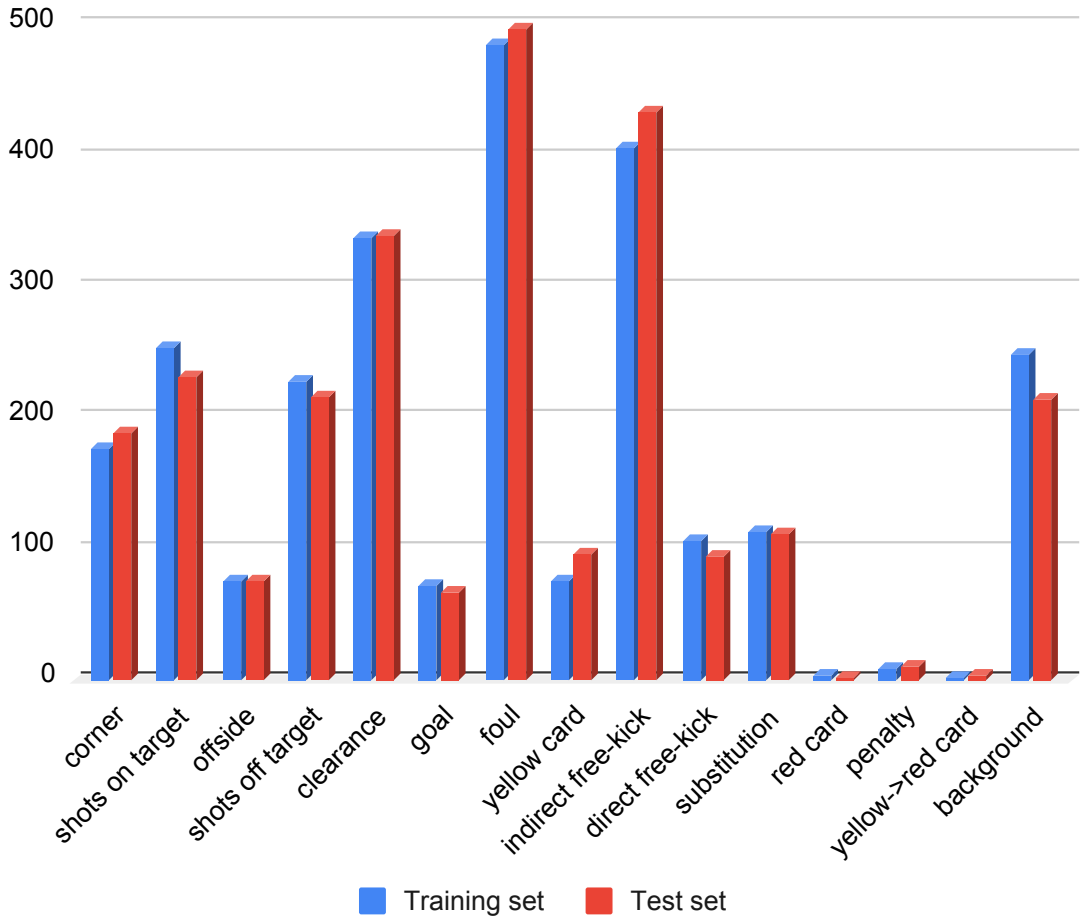


**Figure 5.6:** We randomly extract 20 entire games for our training and test set, respectively. The shapes of their distribution in different action categories are almost the same.

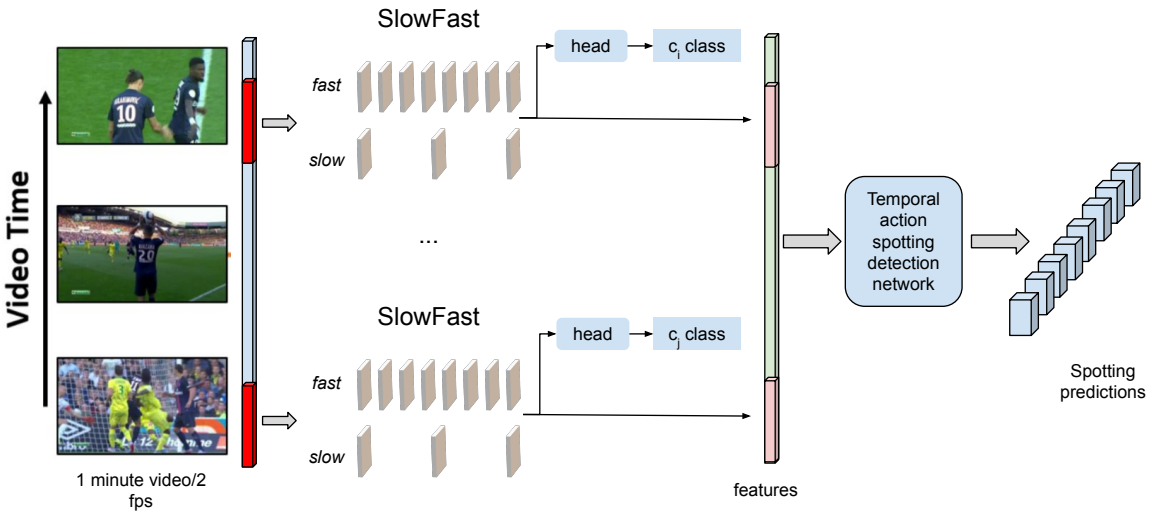


**Figure 5.7:** An attempt to combine SlowFast network with the temporal action spotting network.

# Chapter 6

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

In this dissertation, our target is to make progress towards automatic refereeing systems in soccer game videos. Firstly, we work on multi-camera scenario to detect 'break' and 'play' events in soccer game videos. The event reflects the state of the proceeding game. The 'break' event is also split into 6 different 'break' types. We annotate these events on SVPP dataset which has two complete soccer games from three fixed cameras. These events have various length in the long untrimmed game video. We investigate a deep neural network used for video classification for recognizing fixed-size video clips from a single camera. In light of that a single camera only captures parts of the field and there is no way to recognize the event from only one camera, we extend the neural network for fitting with multi-camera scenario. In a sliding window manner, we output the confidence scores of the predicted events over the entire game video and adopt two novel grouping methods for refining the event boundary. Our work enables the event detection with various length in the untrimmed video captured by multi-camera. Next, thanks to game videos provided by SoccerNet, we aim to detect and recognize the foul subject and object in the static frame at the foul moment. We annotate and extract frames that are at the foul moment according to the public commentary. We further annotate the ground truth for the foul subject and object. We begin with the object/person detector on the low resolution frames of broadcast videos. To utilize the richness of temporal contextual information, we move to detecting the foul subject and object in the video clips that are at the foul moment. Experimental results show strong performance on this activity recognition

task. We also experiment clustering algorithms for differentiating players by colors before and after camera calibration. Finally, according to SoccerNet V2 dataset, we test an approach combining the feature extraction and temporal action detection to directly predict action spots on broadcast game videos.

Our research on the action/event detection and recognition problem can be viewed as an incremental development from fixed multi-camera videos (Chapter 2) to static frames from broadcasting game videos (Chapter 3), to short video clips from broadcasting game videos (Chapter 4), to detect and differentiate action spots over the entire broadcasting game videos (Chapter 5). We briefly summarize each chapter as follows:

In Chapter 2, we annotate events on two entire game videos based on SVPP dataset. And we introduce our construction upon the I3D network to make it suitable with multi-camera in the soccer game and apply it to classify soccer game events rather than actions from individuals. We further propose two grouping methods to localize/refine event boundaries in the video of the soccer game. Our proposed approach demonstrates a promising result on testing.

In Chapter 3, via popular object/person detector, we detect foul subjects and objects on static frames at the foul moment from broadcasting game videos. To achieve this target, based on SoccerNet V1, we first manually annotate and extract frames that are at the foul moment according to the public commentary, and establish a benchmark for the detection of foul subjects and objects on static images. Besides, we experiment Faster R-CNN and Cascade R-CNN detectors. The Faster R-CNN model is trained from scratch but the Cascade R-CNN model is fine-tuned on a pre-trained model. We surprisingly find that our Faster R-CNN model outperforms the Cascade R-CNN model in the detection task. We experiment post-processing methods for achieving better performance and specialize them for detecting the foul subject and object. Comprehensive experimental results show that, despite lacking of temporal contextual information, it still achieves very competitive detection accuracy.

In Chapter 4, we extend the work in Chapter 3 on static images to videos

because of richness of temporal contextual information. We make a subset of video clips at the foul moment for the task of multi-object tracking. We employ multi-object tracking to generate a base set of candidate image sequences which are post-processed to mitigate common mistracking scenarios and then classified according to several two-person interaction types. In very low-resolution image, the proposed system is enabled to differentiate foul participants from bystanders with high accuracy and localize them over a wide range of game situations.

In Chapter 5, in order to remove the foul oracle mentioned in Chapter 4, we move to detecting action spots in broadcasting game videos. This task will generalize the detection to the video that consists of shot boundaries, replays, etc. We build our network architecture upon two successful neural network architectures to enable end-to-end inference over the entire raw broadcasting game video. Despite of unsatisfactory results in the experiment of this task, we still believe that modifications on this architecture can reach our target.

## 6.2 Future Work

Action and event detection on videos are one of the most important yet most challenging problems in computer vision. Despite tremendous success in the action detection on videos with the power of deep learning in this decade, there are still much more studies and works in different areas to be done to compete with human performance. In the previous chapters, we point out a few directions for future research, and briefly summarize parts of them as follows:

**Robustness** As mentioned in Chapter 2, the videos are captured by fixed cameras and there are no shot boundaries in the video. And the videos mentioned in Chapter 4, the input videos are trimmed to 2-second temporal windows. These input data needs more devices and pre-computation to be achieved. In broadcasting videos, more robust model is necessary to deal with the situations such as shot boundaries, replays, logo, moving cameras, etc. In the future, we would like to work on detecting shot boundaries and replays for further improving the performance of the action/event detection.

**End-to-end Temporal Action Spotting** As mentioned in Chapter 5, the experiment doesn't achieve a satisfactory result on the current neural network architecture. We still plan to make modifications on it to reach our goal. Fig. 5.7 gives one possible network architecture. Because of only one SlowFast network is deployed on video chunks and features in each frames cannot provide enough temporal contextual information, we plan to launch SlowFast model on the video chunk in a sliding window manner with some specific strides. It means that a chunk is split into several parts. On each part, SlowFast network extracts features and gives the prediction of actions. We concatenate features extracted by SlowFast over all parts in the video chunk and feed them into the temporal action detection network. Considering that the slow pathway in SlowFast provides more spatial contextual information, we would like to incorporate the features coming from the slow pathway into the temporal action detection network, instead of only using features from the fast pathway. Therefore, the model may leverage the temporal context and the spatial context and push towards the state-of-the-art action spots detection.

**Spatio-Temporal Attention** Most action recognition models in 3D CNNs treat all input video frames equally, it results in the temporal and spatial differences being ignored. Exploring these differences could be helpful for better performance on detection actions and events over the soccer game video. As the attention mechanism has been widely used in various fields, we would like to investigate the attention mechanism in the temporal dimension in order to localize temporal 'ROIs' across the video. Also, learning spatio-temporal information could potentially reduce the complexity for detecting and recognizing actions over the entire video, without running forward predictions on every frame.

**Computational Complexity** Chapter 2, 3, 4, 5 all face the problem of computational efficiency. For example, it takes more than 20 seconds to launch AlphaPose inference on a 3-second video in HQ version. And SlowFast will take about 10 seconds to make predictions on all sequences of person images in a 2-second video clip. For the purpose of practical automatic applications, it will be more desirable to have real-time or

lightweight methods without compromising the performance. We also interest in applying deep learning models to embedded devices to make progress towards automatic refereeing systems.

**Other Scenarios** Detecting and analyzing actions in soccer game videos is just one of our purposes. We think most of the work presented in this dissertation can be directly extended to other scenarios. We would like to investigate some more general cases, like other sports, movies, home security systems, senior caring systems, etc., and extend our work to these areas. All of these require deep understanding of videos. Detecting salient objects/persons or actions in different scenarios in the video would help to reach the deeper video understanding and also narrow down the temporal interval for the further analysis.

## REFERENCES


[1] Andriluka, M., Iqbal, U., Insafutdinov, E., Pishchulin, L., Milan, A., Gall, J., Schiele, B.: Posetrack: A benchmark for human pose estimation and tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5167–5176 (2018)

[2] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)

[3] Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. Tech. rep., Stanford (2006)

[4] Assfalg, J., Bertini, M., Colombo, C., Bimbo, A.D., Nunziati, W.: Semantic annotation of soccer videos: automatic highlights detection. Computer Vision and Image Understanding **92**(2), 285–305 (2003)

[5] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: International Workshop on Human Behavior Understanding. pp. 29–39. Springer (2011)

[6] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)

[7] Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. pp. 5561–5569 (2017)

[8] Bozorgpour, A., Fotouhi, M., Kasaei, S.: Robust homography optimization in soccer scenes. In: Iranian Conference on Electrical Engineering (2015)

[9] Braun, M., Krebs, S., Flohr, F., Gavrila, D.M.: Eurocity persons: A novel benchmark for person detection in traffic scenes. IEEE transactions on pattern analysis and machine intelligence **41**(8), 1844–1861 (2019)

[10] Bridgeman, L., Volino, M., Guillemaut, J.Y., Hilton, A.: Multi-person 3d pose estimation and tracking in sports. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

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

[12] Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)

[13] Canales, F.: Automated Semantic Annotation of Football Games from TV Broadcast. Ph.D. thesis, Department of Informatics, TUM Munich (2013)

[14] Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the Kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)

[15] Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1130–1139 (2018)

[16] Chen, J., Little, J.J.: Sports camera calibration via synthetic data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

[17] Chen, J., Zhu, F., Little, J.J.: A two-point method for ptz camera calibration in sports. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 287–295. IEEE (2018)

[18] Chen, J., Zhu, F., Little, J.J.: A two-point method for ptz camera calibration in sports. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 287–295. IEEE (2018)

[19] Cioppa, A., Deliege, A., Giancola, S., Ghanem, B., Droogenbroeck, M.V., Gade, R., Moeslund, T.B.: A context-aware loss function for action spotting in soccer videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13126–13136 (2020)

[20] Cuevas, C., Quilón, D., García, N.: Automatic soccer field of play registration. Pattern Recognition **103** (2020)

[21] Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: Proceedings of the IEEE conference on computer vision and Pattern recognition. pp. 3076–3086 (2017)

[22] Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. pp. 379–387 (2016)

[23] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 886–893. Ieee (2005)

[24] DeepMind: Convolutional neural network model for video classification trained on the Kinetics dataset (2017), https://github.com/deepmind/kinetics-i3d

[25] DeepMind: Pyslowfast: video understanding codebase from fair for reproducing state-of-the-art video models (2020), https://github.com/facebookresearch/SlowFast

[26] Deliège, A., Cioppa, A., Giancola, S., Seikavandi, M.J., Dueholm, J.V., Nasrollahi, K., Ghanem, B., Moeslund, T.B., Droogenbroeck, M.V.: Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos (2021)

[27] Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for static human-object interactions. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops. pp. 9–16. IEEE (2010)

[28] Diba, A., Sharma, V., Van Gool, L.: Deep temporal linear encoding networks. arXiv preprint arXiv:1611.06678 (2016)

[29] Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE transactions on pattern analysis and machine intelligence **34**(4), 743–761 (2011)

[30] Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1110–1118 (2015)

[31] Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: RMPE: Regional multi-person pose estimation. In: ICCV (2017)

[32] Fani, M., Yazdi, M., Clausi, D., Wong, A.: Soccer video structure analysis by parallel feature fusion network and hidden-to-observable transferring markov model. IEEE Access **5**, 27322–27336 (2017)

[33] Fastovets, M., Guillemaut, J.Y., Hilton, A.: Athlete pose estimation from monocular tv sports footage. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1048–1054 (2013)

[34] Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 6202–6211 (2019)

[35] Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1933–1941 (2016)

[36] FIFA.com: Video assistant referees (VAR) (2019), https://football-technology.fifa.com/ en/media-tiles/ video-assistant-referee-var

[37] Fédération Internationale de Football Association (FIFA): Laws of the game (2015), https://img.fifa.com/image/upload/datdz0pms85gbnqy4j3k.pdf

[38] Gao, J., Chen, K., Nevatia, R.: Ctap: Complementary temporal action proposal generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 68–83 (2018)

[39] Gao, J., Yang, Z., Chen, K., Sun, C., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3628–3636 (2017)

[40] Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. arXiv preprint arXiv:1705.01180 (2017)

[41] Gao, R., Xiong, B., Grauman, K.: Im2flow: Motion hallucination from static images for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5937–5947 (2018)

[42] Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)

[43] Gerke, S., Muller, K., Schafer, R.: Soccer jersey number recognition using convolutional neural networks. In: IEEE International Conference on Computer Vision Workshop (2015)

[44] Giancola, S., Amine, M., Dghaily, T., Ghanem, B.: Soccernet: A scalable dataset for action spotting in soccer videos. In: CVPR Workshop on Computer Vision in Sports (2018)

[45] Giancola, S., Ghanem, B.: Temporally-aware feature pooling for action spotting in soccer broadcasts (2021)

[46] Girish, D., Singh, V., Ralescu, A.: Understanding action recognition in still images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 370–371 (2020)

[47] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)

[48] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence **38**(1), 142–158 (2015)

[49] Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object interactions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8359–8367 (2018)

[50] Gkioxari, G., Malik, J.: Finding action tubes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 759–768 (2015)

[51] Grushin, A., Monner, D.D., Reggia, J.A., Mishra, A.: Robust human action recognition via long short-term memory. In: Neural Networks (IJCNN), The 2013 International Joint Conference on. pp. 1–8. IEEE (2013)

[52] Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions (2018)

[53] Güler, R.A., Neverova, N., Kokkinos, I.: Densepose: Dense human pose estimation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7297–7306 (2018)

[54] Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: Using spatial and functional compatibility for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(10), 1775–1789 (2009)

[55] Gupta, A., Little, J.J., Woodham, R.J.: Using line and ellipse features for rectification of broadcast hockey video. In: 2011 Canadian Conference on Computer and Robot Vision. pp. 32–39. IEEE (2011)

[56] Hasan, I., Liao, S., Li, J., Akram, S.U., Shao, L.: Generalizable pedestrian detection: The elephant in the room (2020)

[57] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)

[58] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[59] Homayounfar, N., Fidler, S., Urtasun, R.: Sports field localization via deep structured models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5212–5220 (2017)

[60] Homayounfar, N., Fidler, S., Urtasun, R.: Sports field localization via deep structured models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5212–5220 (2017)

[61] Hongeng, S., Nevatia, R., Bremond, F.: Video-based event recognition: activity representation and probabilistic recognition methods. Computer Vision and Image Understanding **96**(2), 129–162 (2004)

[62] Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4507–4515 (2017)

[63] Hu, G., Cui, B., He, Y., Yu, S.: Progressive relation learning for group activity recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 980–989 (2020)

[64] Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3588–3597 (2018)

[65] Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: Movienet: A holistic dataset for movie understanding (2020)

[66] Huda, N., Jensen, K., Gade, R., Moeslund, T.: Estimating the number of soccer players using simulation-based occlusion handling. In: CVPR Workshop on Computer Vision in Sports (2018)

[67] Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1971–1980 (2016)

[68] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence **35**(1), 221–231 (2013)

[69] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)

[70] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset (2017)

[71] Kazemi, V., Sullivan, J.: Using richer models for articulated pose estimation of footballers. In: British Machine Vision Conference (2012)

[72] Kläser, A., Marszałek, M., Schmid, C., Zisserman, A.: Human focused action localization in video. In: European Conference on Computer Vision. pp. 219–233. Springer (2010)

[73] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: IEEE International Conference on Computer Vision (2011)

[74] Learn, S.: Hierarchical clustering. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

[75] Learn, S.: Kmeans. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[76] Leo, M., Mosca, N., Spagnolo, P., Mazzeo, P., et al.: A semi-automatic system for ground truth generation of soccer video sequences. In: Advanced Video and Signal Based Surveillance (2009)

[77] Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.S., Lu, C.: Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. arXiv preprint arXiv:1812.00324 (2018)

[78] Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J., Peng, X.: Contrastive clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)

[79] Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. arXiv preprint arXiv:1710.06236 (2017)

[80] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)

[81] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

[82] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)

[83] Liu, T., Lu, Y., Lei, X., Zhang, L., Wang, H., Huang, W., Wang, Z.: Soccer video event detection using 3D convolutional networks and shot boundary detection via deep feature distance. In: International Conference on Neural Information Processing (2017)

[84] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)

[85] Liu, W., Kang, G., Huang, P.Y., Chang, X., Qian, Y., Liang, J., Gui, L., Wen, J., Chen, P.: Argus: Efficient activity detection system for extended video analysis. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops. pp. 126–133 (2020)

[86] Long, X., Gan, C., De Melo, G., Liu, X., Li, Y., Li, F., Wen, S.: Multimodal keyless attention fusion for video classification. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

[87] Long, X., Gan, C., De Melo, G., Wu, J., Liu, X., Wen, S.: Attention clusters: Purely attention based local feature integration for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7834–7843 (2018)

[88] Ltd., H.E.I.: Products: Ball tracking (2020), https://www.hawkeyeinnovations.com/ products/ ball-tracking

[89] Lu, K., Chen, J., Little, J.J., He, H.: Light cascaded convolutional neural networks for accurate player detection. In: British Machine Vision Conference (2017)

[90] Lu, W.L., Ting, J.A., Little, J.J., Murphy, K.P.: Learning to track and identify players from broadcast sports videos. IEEE transactions on pattern analysis and machine intelligence **35**(7), 1704–1716 (2013)

[91] Lu, W.L., Ting, J.A., Murphy, K.P., Little, J.J.: Identifying players in broadcast sports videos using conditional random fields. In: CVPR 2011. pp. 3249–3256. IEEE (2011)

[92] Lukezic, A., Vojir, T., ˇCehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6309–6318 (2017)

[93] Luo, Z., Fang, Z., Zheng, S., Wang, Y., Fu, Y.: Nms-loss: Learning with non-maximum suppression for crowded pedestrian detection. arXiv preprint arXiv:2106.02426 (2021)

[94] Ma, C.Y., Kadav, A., Melvin, I., Kira, Z., AlRegib, G., Peter Graf, H.: Attend and interact: Higher-order object interactions for video understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6790–6800 (2018)

[95] Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed representation of pose and appearance. In: CVPR 2011. pp. 3177–3184. IEEE (2011)

[96] Maksai, A., Wang, X., Fua, P.: What players do with the ball: A physically constrained interaction modeling. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)

[97] Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2929–2936. IEEE (2009)

[98] Mathias, M., Benenson, R., Timofte, R., Van Gool, L.: Handling occlusions with franken-classifiers. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1505–1512 (2013)

[99] Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)

[100] Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: Advances in Neural Information Processing Systems. pp. 5947–5956 (2017)

[101] Ni, B., Yang, X., Gao, S.: Progressively parsing interactional objects for fine grained action detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1020–1028 (2016)

[102] Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with fisher vectors on a compact feature set. In: Proceedings of the IEEE international conference on computer vision. pp. 1817–1824 (2013)

[103] Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In: European Conference on Computer Vision. pp. 145–161. Springer (2020)

[104] Pettersen, S.A., Johansen, D., Johansen, H., Berg-Johansen, V., Gaddam, V.R., Mortensen, A., Langseth, R., Griwodz, C., Stensland, H.K., Halvorsen, P.: Soccer video and player position dataset. In: ACM Multimedia Systems Conference (2014)

[105] Piergiovanni, A., Ryoo, M.S.: Fine-grained activity recognition in baseball videos. In: Proceedings of the ieee conference on computer vision and pattern recognition workshops. pp. 1740–1748 (2018)

[106] Prest, A., Ferrari, V., Schmid, C.: Explicit modeling of human-object interactions in realistic videos. IEEE transactions on pattern analysis and machine intelligence **35**(4), 835–848 (2012)

[107] Prest, A., Schmid, C., Ferrari, V.: Weakly supervised learning of interactions between humans and objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(3), 601–614 (2011)

[108] Puwein, J., Ziegler, R., Vogel, J., Pollefeys, M.: Robust multi-view camera calibration for wide-baseline camera networks. In: 2011 IEEE Workshop on Applications of Computer Vision (WACV). pp. 321–328. IEEE (2011)

[109] Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 401–417 (2018)

[110] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)

[111] Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)

[112] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

[113] Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2008)

[114] Saha, P.K., Borgefors, G., di Baja, G.S.: A survey on skeletonization algorithms and their applications. Pattern recognition letters **76**, 3–12 (2016)

[115] Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International journal of computer vision **105**(3), 222–245 (2013)

[116] Sanford, R., Gorji, S., Hafemann, L.G., Pourbabaee, B., Javan, M.: Group activity detection from trajectory and video data in soccer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2020)

[117] Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)

[118] Sharma, R.A., Bhat, B., Gandhi, V., Jawahar, C.: Automated top view registration of broadcast football videos. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 305–313. IEEE (2018)

[119] Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. arXiv preprint arXiv:1703.01515 (2017)

[120] Sie Ho Lee, T., Fidler, S., Dickinson, S.: Detecting curved symmetric parts using a deformable disc model. In: Proceedings of the IEEE international conference on computer vision. pp. 1753–1760 (2013)

[121] Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: European Conference on Computer Vision. pp. 510–526. Springer (2016)

[122] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)

[123] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[124] Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. Tech. Rep. CRCV-TR-12-01, University of Central Florida (2012)

[125] Sozykin, K., Khan, A.M., Protasov, S., Hussain, R.: Multi-label class-imbalanced action recognition in hockey videos via 3D convolutional neural networks. In: IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (2018)

[126] Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5693–5703 (2019)

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

[128] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)

[129] Thomas, G.: Real-time camera tracking using sports pitch markings. Journal of Real-Time Image Processing **2**(2), 117–132 (2007)

[130] Thurau, C., Hlavác, V.: Pose primitive based human action recognition in videos or still images. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)

[131] Tian, W., Lauer, M., Chen, L.: Online multi-object tracking using joint domain information in traffic scenarios. IEEE Transactions on Intelligent Transportation Systems **21**(1), 374–384 (2019)

[132] Tong, X., Lu, H., Liu, Q.: An effective and fast soccer ball detection and tracking method. In: International Conference on Pattern Recognition (2004)

[133] Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1653–1660 (2014)

[134] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 4489–4497 (2015)

[135] Tsunoda, T., Komori, Y., Matsugu, M., Harada, T.: Football action recognition using hierarchical lstm. In: CVPR Workshop on Computer Vision in Sports (2017)

[136] Vats, K., Fani, M., Walters, P., Clausi, D.A., Zelek, J.: Event detection in coarsely annotated sports videos via parallel multi-receptive field 1d convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 882–883 (2020)

[137] Wagenaar, M., Okafor, E., Frencken, W., Wiering, M.: Using deep convolutional neural networks to predict goal-scoring opportunities in soccer. In: International Conference on Pattern Recognition Applications and Methods (2017)

[138] Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)

[139] Wang, J., Qiu, K., Peng, H., Fu, J., Zhu, J.: Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 374–382 (2019)

[140] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. TPAMI (2019)

[141] Wang, L., Li, W., Li, W., Van Gool, L.: Appearance-and-relation networks for video classification. arXiv preprint arXiv:1711.09125 (2017)

[142] Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 4325–4334 (2017)

[143] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision. pp. 20–36. Springer (2016)

[144] Wang, Y., Hoai, M.: Improving human action recognition by non-action classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2698–2707 (2016)

[145] Wang, Y., Song, J., Wang, L., Van Gool, L., Hilliges, O.: Two-stream sr-cnns for action recognition in videos. In: BMVC (2016)

[146] Widynski, N., Moevus, A., Mignotte, M.: Local symmetry detection in natural images using a particle filtering approach. IEEE Transactions on Image Processing **23**(12), 5309–5322 (2014)

[147] Wikipedia: Hierarchical clustering — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Hierarchical%20clustering&oldid=1049874216 (2021), [Online; accessed 20-October-2021]

[148] Wikipedia: K-means clustering — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=K-means%20clustering&oldid=1049807614 (2021), [Online; accessed 20-October-2021]

[149] Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)

[150] Woo, S., Kim, D., Cho, D., Kweon, I.S.: Linknet: Relational embedding for scene graph. In: Advances in Neural Information Processing Systems. pp. 560–570 (2018)

[151] Wu, B., Nevatia, R.: Cluster boosted tree classifier for multi-view, multi-pose object detection. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8. IEEE (2007)

[152] Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV). pp. 466–481 (2018)

[153] Xie, L., Xu, P., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with domain knowledge and hidden markov models. Pattern Recognition Letters **25**(7), 767–775 (2004)

[154] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)

[155] Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose Flow: Efficient online pose tracking. In: BMVC (2018)

[156] Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. arXiv preprint arXiv:1703.07814 (2017)

[157] Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10156–10165 (2020)

[158] Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative cnn video representation for event detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1798–1807 (2015)

[159] Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 17–24. IEEE (2010)

[160] Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: 2011 International conference on computer vision. pp. 1331–1338. IEEE (2011)

[161] Yu, M., Bambacus, M., Cervone, G., Clarke, K., Duffy, D., Huang, Q., Li, J., Li, W., Li, Z., Liu, Q., et al.: Spatiotemporal event detection: a review. International Journal of Digital Earth **13**(12), 1339–1365 (2020)

[162] Yuan, J., Ni, B., Yang, X., Kassim, A.A.: Temporal action localization with pyramid of score distribution features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3093–3102 (2016)

[163] Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint pattern recognition symposium. pp. 214–223. Springer (2007)

[164] Zecha, D., Einfalt, M., Eggert, C., Lienhart, R.: Kinematic pose rectification for performance analysis and retrieval in sports. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1791–1799 (2018)

[165] Zecha, D., Einfalt, M., Lienhart, R.: Refining joint locations for human pose tracking in sports videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

[166] Zeng, R., Lakemond, R., Denman, S., Sridharan, S., Fookes, C., Morgan, S.: Calibrating cameras in poor-conditioned pitch-based sports games. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1902–1906. IEEE (2018)

[167] Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530 (2016)

[168] Zhang, F., Zhu, X., Ye, M.: Fast human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3517–3526 (2019)

[169] Zhang, J., Lin, L., Zhu, J., Li, Y., Chen, Y.c., Hu, Y., Hoi, C.S.: Attribute-aware pedestrian detection in a crowd. IEEE Transactions on Multimedia (2020)

[170] Zhang, S., Benenson, R., Schiele, B.: Citypersons: A diverse dataset for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3221 (2017)

[171] Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2914–2923 (2017)

[172] Zhou, C., Yuan, J.: Multi-label learning of part detectors for heavily occluded pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3486–3495 (2017)

[173] Zhou, T., Wang, W., Qi, S., Ling, H., Shen, J.: Cascaded human-object interaction recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4263–4272 (2020)

[174] Zhu, J., Yuan, Z., Zhang, C., Chi, W., Ling, Y., et al.: Crowded human detection via an anchor-pair network. In: The IEEE Winter Conference on Applications of Computer Vision. pp. 1391–1399 (2020)

# Appendix A

## TITLE OF APPENDIX

This is the information for the first appendix, Appendix A. Copy the base file, appA.tex, for each additional appendix needed such as appB.tex, appC.tex, etc. Modify the main base file to include each additional appendix file.

If there is only one appendix, then modify the main file to only use app.tex instead of appA.tex.