Master Thesis

Top-Down vs Bottom-Up Human Pose Estimation in Football Action Classification in a Monocular Camera System

by

Ivo van Miert

(2634569)

First supervisor:	Vincent François-Lavet
Daily supervisor:	Niels Nederlof
Second reader:	Charlotte Gerritsen

September 30, 2024

Submitted in partial fulfillment of the requirements for the VU degree of Master of Science in Business Analytics

Top-Down vs Bottom-Up Human Pose Estimation in Football Action Classification in a Monocular Camera System

Ivo van Miert
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
i.a.j.van.miert@student.vu.nl

ABSTRACT

This research project examined the utilization of top-down and bottom-up human pose estimation models in the classification of football actions captured by a single-view, monocular camera. The study developed a pipeline that integrated, in addition to the human pose estimation models, multiple other algorithms, including field registration, player detection and tracking, ball detection and tracking, and team classification. These components were incorporated into Long Short-Term Memory (LSTM) models, which were designed to categorize short video clips into five primary action classes and their respective sub-classes. The primary objective was to evaluate the impact of the distinct methodologies of top-down and bottom-up pose estimation on classification performance within this machine learning system. The results demonstrated that integrating these HPE models into the pipeline enhanced the classification models' performance. Both pose estimation models exhibited distinct advantages. The top-down model achieved a marginally superior score on the hierarchically higher level of primary events. However, at the attribute level, the bottom-up model demonstrated superior performance. Future research could investigate the influence of both HPE methods on classification in the case of multi-camera systems.

1 INTRODUCTION

The utilisation of data and statistics has been of significant benefit in the domain of sport over the past decade. Various organizations within the sporting industry employ performance data and analytical tools to gain insight into tactics, player performance, and other pertinent aspects, with the objective of attaining a competitive advantage over their rivals. In response to this heightened interest from the sports sector, a greater investment of time and resources has been dedicated to the investigation of diverse methodologies for the collection and processing of this information into actionable insights for individuals within the sports industry.

Predictive analytics has been studied and applied in the context of several sports over the past decades [74], including basketball, tennis and baseball. Although the statistical analysis of data has produced impressive results in various sports, football has been a late adopter of this data collection. Several factors have influenced this late arrival.

Firstly, collecting data from football matches requires a lot of manual labour and sometimes expensive equipment. The privilege of obtaining such insights is limited to high-profile sporting events and prestigious football clubs that can afford such costs [40].

Secondly, due to its outdoor and highly dynamic nature, football is played under far less controllable conditions than other sports, with a larger pitch, a large number of players involved, a low number of player substitutions and longer uninterrupted game sequences than other sports. Only recently have major breakthroughs been made in deep learning, providing techniques that can handle such high-dimensional data sets [58].

In addition, the credibility of decision-making depended primarily on human specialists such as managers, retired players and scouts, all of whom had track records and experience in professional football. [35] highlighted a cultural hesitancy to integrate data science into football and an over-reliance on gut instinct, noting that 'until very recently, football had escaped the Enlightenment'.

The combination of these reasons has meant that football analytics companies have only relatively recently started to collect big data (e.g. high-resolution video, annotated event streams, player tracking and pose information). The player tracking and pose information is done using deep learning techniques, but for specific actions, such as passing, shooting, etc., and their outcomes or attributes, such as accuracy or the leg used in the action, manual annotation must be done. These annotations are only done for specific high-level games, and the companies sell this information at high prices, making it unattractive for individuals to get it done for their own games or training sessions. Therefore, this research takes a first step towards specific action classification, in an attempt to make it possible to automate more than just player and ball tracking, and to take a first step towards replacing the need for manual annotation of in-game actions.

The main contributions of this work are the following:

- A unified video processing pipeline, comprising a combination of existing and self-developed methods, is employed for the extraction of information from football matches. The existing methods employed in this research are used for the purposes of field localizaton, object detection and tracking, and human pose estimation. The self-developed method has been developed with the objective of team classification and has been designed to align with the data used in this research. Furthermore, the potential for the host organisation to adapt this team classification model for implementation with new data is also discussed.
- The shortcomings of the current methodologies are discussed.
- This study compares and contrasts two methods of human pose estimation: top-down and bottom-up. The aim is to highlight the differences in their use and performance.
- This study presents a novel hierarchical classification pipeline for the detailed analysis of short football clips. The approach utilises computer vision to extract comprehensive performance data, bridging the gap between human perception and automated analysis.

In conclusion, this thesis presents a pipeline that can classify a short (0-20 second) video clip representing a specific action in a football match, captured using a monocular view. The classification is done on different levels/attributes of certain actions. The monocular view is a camera placed on the long side of the football field, which is often zoomed out and captures a large part of the football field. No additional hardware or sensors are required. The proposed data processing pipeline is illustrated in Figure 1

The remainder of the paper describes the overall research process of this project and is structured as follows: Section 2 provides an introduction to the host organization and department, presents the problem statement and discusses the relevance of the problem to the host organization. Section3 describes the current use of computer vision in sport and discusses the literature that currently forms the basis for the different computer vision methods used in this thesis. This is followed by Section4 which describes the collection, exploration and initial pre-processing of the data. Chapters 5 and 6 describe the methods used to extract information from the video clips and are evaluated on a subset of the data to show how they perform in different situations. Section7 describes the development of the classification model, which is followed by Section8 which reviews the results obtained. Section9 then summarizes the findings and discusses the implications of the research, followed by Section10 which makes both practical recommendations and suggestions for future research/implementation for the host organization.

2 PROBLEM DESCRIPTION

This section introduces the host organisation and department where the research was conducted in Section 2.1. Section 2.2 presents the problem statement, which is then followed by a discussion of the scientific foundation and an analysis of how this research differs from existing research in the field. Finally, Section 2.3 discusses the relevance of the problem to the host organization.

2.1 Host organization

Ajax was founded in Amsterdam on 18 March 1900. The club developed into a listed football club with international appeal and recognition. Since the introduction of professional football in the Netherlands (1956), Ajax has played continuously at the highest level, the current Eredivisie. Ajax is currently run by a four-man board, consisting of an interim general manager, a financial director, a technical director and a director of football, all of whom are approved by the RvC. AFC Ajax (NV) has been listed on the stock exchange since 1998, making it the only Dutch football club with a stock market presence. It has a complex shareholder structure, with 73 per cent of the shares controlled by the AFC Ajax Association. Ajax's financial year runs from 1 July to 30 June. The income stream can be divided into three different sections: Football income, sponsorship and TV rights (approximately 50 per cent, 25 per cent and 15 per cent respectively).

AFC Ajax's slogan is 'For the Future', which reflects the club's commitment to nurturing young players and developing them into successful footballers. This philosophy is important to AFC Ajax as it provides a vital source of income, closing the financial gap between Ajax and clubs in bigger football leagues (such as the Premier League), fuelled by lucrative TV rights deals. To illustrate the significant financial difference, AFC Ajax received the largest share of Dutch TV rights revenues, totalling 9.5 million euros in 2023, compared to the lowest earning English Premier League team, Sunderland, which received 128.2 million euros in TV rights in the same period [14]. This significant difference underlines the financial disparity Ajax faces in competing with bigger league clubs and highlights the need for alternative revenue streams, such as talent scouting and development, to remain competitive.

This internship took place within the International Youth Scouting Department of AFC Ajax. The process of developing young talent into quality footballers begins with identifying potential from an early age. AFC Ajax scouts talent from as young as 6 years old and players must meet certain performance criteria to be considered. With around 10 professional scouts and around 140 volunteer scouts, Ajax's scouting department carefully evaluates players on Dutch football pitches to determine their suitability for the club. However, the recruitment of international players follows different protocols and faces different challenges [45]. International players are allowed to join Ajax from the age of sixteen, with a limit of two new international players per year. Unlike the scouting of Dutch footballers, the scouting of international prospects is logistically challenging. The initial challenge that the international scouting department must overcome is the identification of talented individuals. Ajax has established connections with a number of scouts in various countries, who possess knowledge and insights regarding talent within their respective regions. However, this network does not encompass all countries. In order to gain insights into the youth market in countries with no connections, the international scouting department relies on the utilisation of data that is accessible on the aforementioned scouting platforms. The data available on these scouting platforms is limited to basic information on player performances unless an expensive subscription is purchased, which allows for more comprehensive data analysis on those players. The various data packages are discussed in more detail in Section 4.1. The more comprehensive the data available on player performances, the more effectively the international scouting department can identify and assess talent within a region or competition, and the more efficient the search for talent will be. Furthermore, the scouting platforms provide access to match footage. These videos can be viewed online by designated "video scouts" on a weekly basis. Match footage from over 600 competitions worldwide is uploaded to the scouting platforms utilized by Ajax. However, the number of video scouts available to watch these games and write reports on the players is limited. Therefore, it is essential to select the games to be watched by the video scouts with careful consideration. With more sophisticated data on player performances, more informed decisions can be made regarding which games require direct observation.

2.2 Problem Statement

This research represents an initial attempt to develop a comprehensive advanced data collection system based on video footage of in-game football events. Multiple Long Short Term Memory (LSTM) classification models are trained to categorize short video clips into 5 distinct main classes, and multiple sub classes. This research and the resulting models can then be employed as a foundation for a

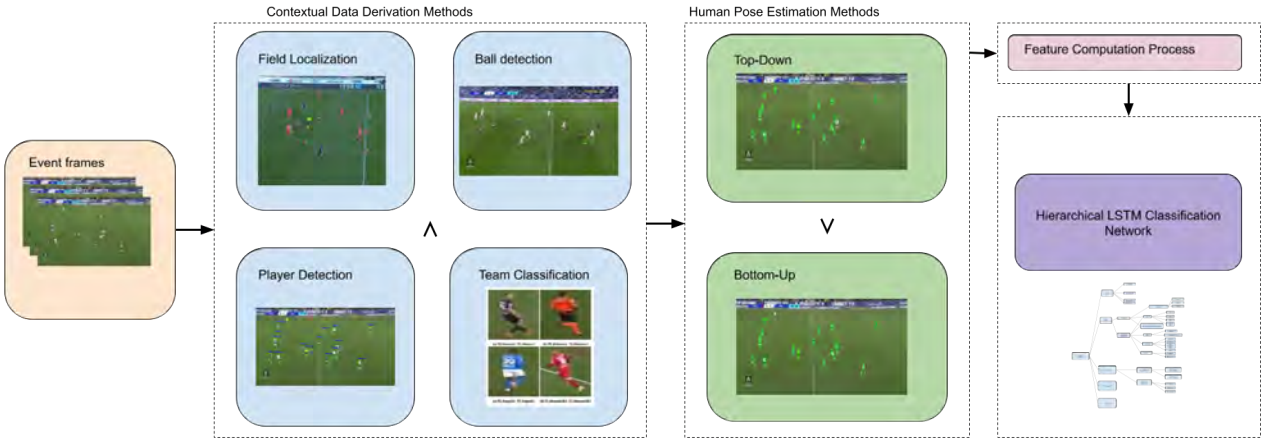


Figure 1: Visualization of Project Pipeline

more comprehensive system for the collection of data from the entirety of a match. This research will involve the creation of a pipeline comprising a variety of existing and self-designed methods for the gathering of features from the input clips, which are subsequently used in the classification process. The aforementioned methods comprise a field registration algorithm, a player detection and tracking algorithm, a ball detection and tracking algorithm, a team classification algorithm and two distinct human pose estimation algorithms.

In this research, the algorithms are evaluated in a variety of scenarios, with a focus on their strengths and limitations. In particular, the performance of both human pose estimation models will be examined in relation to the following research question: "How will the utilization of distinct 2D human pose estimation algorithms, encompassing top-down and bottom-up methodologies, coupled with models for field registration, player and ball detection and tracking, and team classification integrated in a machine learning system, impact the capacity to classify actions in football using footage from a monocular, action-tracking camera?"

2.2.1 Scientific Foundation. In Section 3.3 the current and newest studies on football action recognition are mentioned. The majority of these studies addressed a challenge proposed by Soccernet [63], which focused on the recognition of sparse football actions (goals, red cards, substitutions). These actions can most of the time easily be distinguished by discernible changes in camera view. The objective was to automatically summarise an entire match by recognising the most important moments. This involved the recognition of these sparse actions. Such methods may then be employed to automatically generate highlights from matches. A limited number of studies have addressed the classification of frequently occurring actions that are more challenging to distinguish within the context of a game. This is due to the unavailability of annotated data on this subject. The data used in this study, which is described in Section 4.1.1, is drawn from a paid service that is accessible to a limited number of individuals. One limitation of this annotated data is that it was not originally designed for action recognition. Instead, it was created for the purpose of tracking statistics, and thus it is not fully suitable for the task of action recognition. Furthermore, a review of the literature for this thesis revealed no studies that have compared top-down and bottom-up human pose estimation models for action classification in sports from a monocular perspective. However, the recent release of a new challenge on action recognition by SoccerNet [64], has prompted the annotation of a greater number of in-game actions, as previously described. This development suggests that significant advancements in this field may be expected in the near future.

2.3 Relevance of the problem to the organization

This research is the first step into creating an automated advanced data collection system based on video footage. The implementation of such an automated advanced data collection system can confer a number of advantages to the host organization. Firstly, the scouting process can be optimised by processing the vast quantity of games with the system, thereby reducing the number that require in-depth analysis within the limited time afforded by the available personnel.

Moreover, the implementation of such a system has the potential to enhance the operations of other departments within the organization as well.

A team of analysts analyse the matches played by the first team. During the course of the matches, the analysts analyse the games in order to provide the trainer with the opportunity to, for example, during half-time, demonstrate to the players a variety of scenarios in order to explain certain aspects of the game. A system that transforms video footage into data and is capable of identifying specific scenarios from the collected data can automatically provide the trainer with footage of these scenarios, thus allowing match analysts to dedicate their attention to other, more complex tasks. Moreover, the monitoring of player performance in each match enables the organization to gain insight into the status and progression of the player in question. At present, the analysis of matches involving youth teams is frequently conducted manually by trainers

[45], a process that is both time-consuming and incomplete, as it does not encompass the full range of statistics pertaining to the players in question. The implementation of an automated system that could record these statistics and present a comprehensive view of the performance of youth players over a longer period of time would therefore provide trainers with invaluable insight into the areas requiring training.

In addition to providing enhanced value to the coaching and playing staff of AFC Ajax, the successful implementation of such a system is also an attractive proposition for players from other clubs. The implementation of this digital innovation affords AFC Ajax the opportunity to establish a competitive advantage over other clubs, thereby enhancing its attractiveness to top-talented individuals.

3 RELATED WORK

In this section, existing literature on various computer vision methods designed for applications in sports and football, and of high value in this research, are discussed.

3.1 Field Localization

Field localization in football refers to the process of accurately estimating the correspondence between the playing field seen by the camera and the metric model of the field. In field localization for football, it is assumed that the playing surface is flat, so the process of field localization involves determining a homography matrix H for the transformation of an image from the camera into two-dimensional sports field coordinates [71]. A homography transformation can be estimated given a set of feature matches between two images, or in this case between an image and a 2D planar coordinate system [25]. Four or more point correspondences provide enough constraints to obtain the homography using the DLT algorithm [17].

A variety of techniques have been developed for the registration of sports fields using monocular cameras. These techniques can be grouped into the following categories: those that rely on horizontal and vertical lines, those that utilize existing dictionaries of camera views, and those that directly predict camera parameters from an image.

The method of [30] is part of this first category. [30] leverages the segmentation of horizontal and vertical lines to derive a set of plausible field poses from the vanishing points, and selects the best field after a branch-and-bound optimization. The method needs at least two of both the horizontal and vertical lines, which makes it in practice often unusable, since for broadcast views, not all frames contain at least two of those.

The works of [59] and [9] can be categorized under those that utilize existing dictionaries of camera views. [59] uses a conditional generative adversarial network (CGAN) [49] to directly generate edge images from an RGB image. Edge images are represented by features that are much more efficient than raw edge images in search. Besides that, [59] used Chamfer transformation and HOG to represent edge images for soccer games. Based on this and on the research on color-based kernel [27] and line and ellipse detection [51] to distinguish field-marking pixels from other pixels, Chen and Little [9] propose a novel sports camera pose engine with only three significant free parameters, with an effective feature extraction method for edge images and an end-to-end two-GAN model to detect field markings. As mentioned before, the methods of [59]. and [9] Little make use of a database of edge images generated with known homographies to extract the pose. The bottleneck of these two methods is the necessity of a database, which hinders their scalablitiy.

[15] can be categorized as technique that directly predicts camera parameters from an image. [15] proposed a strategy that involved segment detection to discard most unwanted edge data, classification by using a probabilistic decision tree that identifies the most probable classification for the set of all detected lines, a combination of a region growing algorithm and a Least Square Fitting algorithm to efficiently model the center circle of the field of play, and a three-step validation stage to determine whether the registration is correct.

3.2 Detection and Tracking

Object detection, as one of the most fundamental and challenging problems in computer vision, has received great attention in recent years [88]. It is a task that deals with detecting instances of visual objects of a certain class in digital images. Besides prediction of location, object detection also implies the prediction of the class the object belongs to. Object detection is commonly used in many applications of computer vision, such as image retrieval, security and surveillance, autonomous car driving, and many industrial applications but a single best approach to face that problem does not exist: the choice of the right object detection method depends on the problem that needs to be solved and on the set-up of the experiment [5]. For action recognition in team sports, such as football, object detection should be as accurate as possible with reliable detection of relevant players, the ball, and of other objects of interest [65]. Object detection and tracking in sports videos brings some extra challenges in comparison with other object detection and tracking appliances. It should be able to deal witch challenging conditions like the variable number of objects with a wide range of possible sizes ranging from players that can cover most of the image to the objects that are far away from the observer, that are occluded or those that can be as small as few pixels yet carry a lot of information, such as the ball.

[44] proposed a novel approach called DeepPlayer-Track to track the players and referees, by representing the dep features to reatin the tracking identity. The proposed methodology consists of two parts: the You Only Look Once (YOLOv4) for detection and a modified deep feature association with a simple online real-time (SORT) tracking model which connects nodes from frame to frame to correlate the coefficient of the player identities. The limitation of this method is that, when a player with the same jersey color is occluded, the ID of the player is switched. [57] explored the usage of a Histogram of Oriented Gradients (HOG) trained on pedestrian detection in combination with a color based detector. [77] propose a Player Tracking algorithm that combines a CNN-based multibox detector from [69] and a Kanade Lucas Tracking (KLT) tracker inspired by the implementation of [79], which is an approach to feature extraction and is dealing with the problem that traditional image registration techniques are generally computational expensive. [79] proposed a novel deep learning approach for 2D ball detection and

tracking (DLBT). 2-stage buffer median filtering background modeling is used for moving objects blob detection in combination with a deep learning approach for classification of an image patch into the classes ball, player and background. For robust ball tracking a probabilistic bounding box overlapping technique is proposed and novel full and boundary grid concepts are implemented to resume tracking in so called ball-track-lost and ball-out-of-frame situations. It achieved an accuracy of 87.45 but according to the review of [43], it could not detect when the ball moved out of play in the fields, in the stands region, or from partial occlusion by players, or when ball color matched the player's jersey. [2] Proposed a particle-filter-based approach in which they introduced the notion of shared particles densely sampled at fixed positions on the model field. They globallly evaluate targets' likelihood of being on the model field particles using a combined appearance and motion model. Their proposed tracking algorithm is embedded in a real-life soccer player tracking system called Sentioscope.

3.3 Human Action Recognition in Football

Human Action Recognition (HAR) is a challenging task used in sports to detect players and recognize their actions and teams' activities during training, matches, warm-ups or competitions [31]. HAR aims to detect the person performing the action on an unknown video sequence, determine the action's duration, and identify the action type.

HAR in football is a typical supervised learning task on sports data. It begins with collecting and annotating data for a task of interest, preprocessing such as removing digital noise, and extracting features. Feature extraction in traditional machine learning (ML) techniques was manual, but with the advent of deep learning, it was automated [50]. Feature extraction can be described as a pre-processing part to remove the redundant part from the data. Features are divided into low-level and high-level features. The key points for low-level features are corners, edges, or contours. High-level features in action recognition represent an action by detecting high-level concepts and often build upon local features. The main idea is to preserve structural information of actions. These high-level features can be a spatio-temporal volume (STV) generated by 2D contours, 3D shapes induced by silhouettes, motion descriptor based on smoothed and aggregated optical flow, kinematic features and so on [65]. The feature extraction step in HAR is used for description, but does not explain the action. [52] uses Support Vector Machine in the classification process, [8] use the K-nearest neighbour, [24] the K-means algorithm. Besides machine learning, as mentioned before, actions in HAR can also be classified using Deep Learning based methods, which include automatic feature extraction, description and classification. The most popular models being used in DL-based implementation of HAR in sports are the Convolutional Neural Networks (CNN) and the Long Short-Term Memory (LSTM) [55]. LSTM's are part of the family of Recurrent Neural Networks, which feeds activations from an input in a previous time step back into the network to affect the output for the current input. This property makes the RNNs suitable for modeling sequences, such as video frames in action recognition.

[18] introduced a pose-projected action recognition hourglass network which performed action recognition on player-level. It includes an embedded pose projection component that regularizes the player's pose vector's range and incorporates the temporal information. A parallel structure is obtained for extracting projected pose vectors from all frames of an input sequence and using LSTM layers to integrate the pose vectors across the input frames. [1] leveraged the spatiotemporal learning capability of three-dimensional CNN and and LSTM for summarizing long soccer videos. To summarize a soccer match video, they modeled the video input as a sequential concatenation of video segments whose inclusion in a summary video production is based on its validated relevance. They recognized five actions: centerline, corner-kick, free-kick, goal action and throw-in to create a highlight recognition framework, and were assessed using Mean Opinion Scores from 48 soccer enthusiasts and received a 4 of 5 MOS. [56] considered the problem of explicityl modeling interactions between players and ball. For that, they proposed self-attention models to learn and extract relevant infrmation from a group of soccer players for activity detection from both trajectory and video data. Their results show that most events can be detected using either vision or trajectory based approaches with a temporal resolution of less than 0.5 seconds, and that each approach has unique challenges. [75] used the extraction of histogram of oreitned gradient (HOG) features for feature extraction and a Support Vector Machine for classification to classify soccer videos of 5-9 seconds into one of six events: Goal, Head goal, Penalty save, Penalty goal, Red Card and Substitute. Following the action spotting challenge of Soccernet [63], which was based on spotting temporally sparse actions within a complete soccer game, such as goals, player substitutions, and card scenes, [41] implemented a Transformer model, which allows capturing important features before and after action scenes. [72] proposed a lightweight and modulare RMS-Net for the same challenge, and [39] proposed a dilated recurrent neural network with LSTM units, grounded on Two-stream CNN features to model long-range and mid-range dependencies, for that same challenge also.

3.4 Human Pose Estimation Models

Human Pose Estimation (HPE) is a significant issue that has been taken into consideration in the computer vision network for recent decades. It is a vital advance toward understanding individuals in videos and still images [16]. HPE aims to locate the human body parts and build human body representations from input data such as images and videos.

HPE models can be categorized between 2D or 3D HPE. 2D HPE is used to estimate the 2D position or spatial location of human body keypoints. 3D HPE on the other hand is used to predict the locations of body joints in the 3D space. Motion capture systems can collect 3D pose annotations in controlled lab environments; however, they have limitations for in-the-wild envirionments. For 3D HPE from monocular RGB images and videos, the main challenge is depth ambiguities [87]. In this research, 2D HPE will be used.

Traditional 2D HPE methods adopt different hand-crafted feature extraction techniques for body parts, and these early works describe the human body as a stick figure to obtain global pose structures. Recently, deep learning-based approaches have improved results

on HPE significantly. HPE can be done in single-person and multi-person scenarios.

For single-person pipelines, there are two categories that employ deep learning techniques: regression methods and heatmap-based methods. Regression methods apply an end-to-end framework to learn a mapping from the input image of the positions of body joints or parameters of human body models [70]. Heatmap-based methods predict approximate locations of body parts and joints [10] [46] which are supervised by heatmaps representation.

Due to the impressive performance of the model 'DeepPose' presented in [73], which is based on the regression framework, the research paradigm of HPE began to shift from classic approaches to the use of CNNs. Instead of using the traditional joint-based representation,[67]introduced a structure-aware regression method which adopts a bone-based representation that contains human body information and pose structure. [36] first designed a transformer-based cascade network in which the spatial correlation of joints and appearance is captured by self-attention mechanisms. A popular strategy to learn better feature representation, which is critical for regression-based methods, is sharing representations between related tasks so that the model can generalize better on the original task. Following this, [37] proposed a heterogeneous multi-task framework that consists of two tasks: predicting joint coordinates from full images by a regressor and detecting body parts from image patches.

Instead of directly predicting the 2D coordinates of human joints, heatmap-based methods for HPE focus on estimating 2D heatmaps. These heatmaps are created by placing 2D Gaussian kernels at each joint's location, resulting in a set of K heatmaps $\{H_1, H_2, \ldots, H_K\}$ corresponding to K keypoints. Each pixel value $H_i(x, y)$ in a heatmap represents the probability of the keypoint being at the position (x, y). The ground-truth heatmap for each keypoint is a 2D Gaussian centered at the joint's true location. Pose estimation networks are trained to minimize the difference, often measured using Mean Squared Error (MSE), between the predicted and ground-truth heatmaps. This approach retains spatial location information and typically results in a smoother training process compared to directly predicting joint coordinates. Therefore, the interest in leveraging heatmaps to represent the joint locations recently grew. [80] introduced a sequential framework based on convolutional networks called Convolutional Pose Machines, which predicts the locations of keypoints using a multi-stage process. [46] proposed an encoder-decoder network known as the "stacked hourglass." In this model, the encoder compresses features through a bottleneck, and the decoder expands them for further processing. Complex variations of the stacked hourglass architecture have been developed, such as the Hourglass Residual Units by [13], which extend the original design by capturing features at multiple scales.

[66] presented the High-Resolution Net (HRNet), a novel architecture that learns reliable high-resolution representations by connecting multi-resolution subnetworks in parallel and performing repeated multi-scale fusions. This approach leads to more accurate keypoint heatmap predictions. Inspired by HRNet, [85] introduced Lite-HRNet, a lightweight version that uses conditional channel weighting blocks to facilitate information exchange between channels and resolutions. Due to their superior performance, HRNet [66] and its variations [85], [12] [86] have been widely adopted in HPE and other pose-related tasks.

In addition to these efforts in designing effective networks for HPE, body structure information has also been explored to provide better supervision for building these models. [84] developed an end-to-end CNN framework that improves HPE by incorporating spatial and appearance consistency among human body parts, allowing it to identify challenging examples more effectively.

Multi-person pipelines, which are used in this research, are more difficult and challenging because it is necessary to work out how many people are present in the image and how to group the key points for these different people. Multi-person HPE methods can be divided into top-down and bottom-up methods. This distinction is also the difference between the two HPE models used in this research. Top-down methods use person detectors to obtain a set of bounding boxes, with one bounding box for each individual person, from the input image, and then apply single-person pose estimators to each bounding box to estimate individual human poses, and then concatenate the results to generate multi-person poses. Bottom-up methods, on the other hand, first locate all body joints in an image and then group them into individual subjects. Unlike the top-down pipeline, where the number of people in the input image directly affects the computation time, this does not happen with the bottom-up method, so the computation speed for bottom-up methods is usually faster than for top-down methods [42].

The bottom-up pipeline has two main steps, namely body joint detection (extracting local features and predicting body joint candidates) and the assembly of these body joint detections into individual bodies. One of the first two-step bottom-up approaches, Deepcut [53], first detects all candidate body parts, then labels each part, and assembles these parts into a human pose using integer linear programming. [6] developed an algorithm using Convolutional Pose Machines [80] to predict keypoint coordinates using heatmaps and Part Affinity Fields, a set of 2D vector fields with vector maps that encode the position and orientation of limbs. This algorithm provides a significant speed improvement over Deepcut. Many existing bottom-up HPE methods are based on OpenPose. Although they have achieved impressive results on high resolution images, they have poor performance on low resolution images. [34] proposed a new bottom-up method, named PifPaf, which combines the part affinity fields designed in OpenPose with a part intensity field to predict the positions of body parts. This method outperforms other OpenPose-based approaches on low-resolution images.

In addition to the difference in dimensions and pipelines, there is also a difference in the way the human body is estimated. A human pose estimator can be either a skeleton-based, a contour-based or a volume-based model. A skeleton-based model represents a set of joint locations (typically between 10 and 30) and corresponding limb orientations that follow the skeletal structure of the human body [11]. It is also known as a stick figure model and can be described as a graph where vertices represent joints and edges represent constrictions or previous connections of joints within the skeletal structure [19]. This topology is very simple and flexible and is used in both 2D and 3D HPE. Despite the obvious advantages of simple and flexible representation, it also has many shortcomings, such as the lack of texture information, which means that there is no width and contour information of the human body. A contour-based

model contains the approximate width and contour information of body limbs and torso. Human body parts are represented by multiple rectangles that approximate the human body contours. It can capture the connection of body parts, which is not possible with skeleton-based models. The volume-based model is more advanced than the previous models and represents the human body as a 3D volume. Where the contour-based model is represented by multiple rectangles, the volume-based model consists of geometric shapes such as cylinders and cones to create a realistic representation of body poses.

4 DATA

In this chapter, Section 4.1 describes the process of collecting data from the various data sources used for the study. In order to get a better overview of what is contained in the collected data sets, the process of data exploration is described in Section 4.2, followed by an initial phase of data preprocessing to filter useful data from redundant data in Section 4.3.

4.1 Data Collection

This study employs two distinct data sources. The first originates from a paid service scoutings platform utilized by AFC Ajax to monitor potential future players and is described in Subsection 4.1.1. The second data stream consists of manually gathered data from different sources and is described in Subsection 4.1.2.

4.1.1 Wyscout event data. Wyscout is an Italian company that supports football scouting, match analysis and transfer dynamics. It provides video analysis tools and digital databases regarding performances and matches for football coaches, teams and players. Wyscout has worldwide coverage, play by play data on more than 600 competitions globally, from the biggest leagues to the most promising youth tournaments all over the world [82].

Wyscout offers different packages for different types of data and services. Currently AFC Ajax makes use of the Videos Pack [45], which allows accessing to the video footage API of Wyscout. With this package, basic statistics of a player's match performances can be displayed and video footage can be downloaded to be later reviewed by videoscouts.

For this research, a trial version of the "Events Pack" [81] was requested from Wyscout, which consists of a detailed analysis of every event that happens in a match. The data was obtained through Wyscout's API and consists of both the 'event data' and the video footage of almost all matches of the first 30 rounds of play of the Serie A season 2023/2024. The Serie A is Italian's top football division in the pyramid structure of four professional leagues in Italy [7]. This dataset amounts to a total of 295 matches. Since the video footage of a game takes up a lot of memory, and in most preprocessing of used methods in this research the resolution of the images is often reduced, the videos were not downloaded in the highest possible resolution, but in hd-format with a resolution of 1280x720 pixels.

In the course of this research, a trial version of the "Events Pack" [81], was requested from Wyscout. This comprises a comprehensive analysis of each occurrence within a given match. The data was obtained via Wyscout's application programming interface (API) and comprises both the event data and video footage of nearly all matches from the first 30 rounds of Serie A play during the 2023/2024 season. The Serie A is Italy's highest-level football division within the nations' pyramid structure comprising four professional leagues [7]. The dataset comprises a total of 295 matches. Given the considerable memory requirements of video footage and the prevalent reduction in image resolution during the preprocessing phase of most methods used, the videos were not downloaded in the highest possible resolution. Instead, they were downloaded in HD format with a resolution of 1280x720 pixels.

4.1.2 Manually collected data. In addition to the extracted datasets, this research also involved the collection of some manually-derived data. This was carried out in two distinct ways, corresponding to the team classification and field localization, respectively.

Manually collected data: Team Classification: The initial manual collection is designed to facilitate the classification of players into teams. This process is elaborated upon in Section 5.5. In order to facilitate the classification process, data has been collected on the outfits worn by the teams during each match. Each team in Serie A has a range of different kit options that can be worn during a match. The number of different kits varies between three and four for players and two and five for goalkeepers [20] . For a match, a selection of kits is made based on the distinctiveness of the different options, taking into account that the team playing on their home field can play in their first/home kit. For classification purposes, each match between team A and team B in the dataset is manually annotated with the jersey worn by the respective team, the goalkeeper, and the team of referees. The manually collected data on the worn kits can be found on the github page under data.

Manually collected data: Field Localization: The other manual collection is in correspondence with field localization, which is further explained in section 5.2. For precisely locating positions on the field during the process of field localization, the dimensions of every football field used in the dataset had to be annotated. According to FIFA's Laws of the Game [21], the in-field line markings have standard and non-changeable sizes, whereas the length of the touchline and goal line can differ per field of play and have to be in the range of 90-120m and 45-90m respectively. The different sizes for each playing field in the Serie A can be found in Table 1.

4.2 Data exploration

4.2.1 Event-Related Data Exploration. This section provides a more detailed examination of the Wyscout event data. Each match is represented by its own event data set, in which every event that occurred during that match has been annotated with detailed information. The mean number of events per match is 1681. Each event is identified by a unique event ID, the corresponding match ID, a description of the timestamp in-game (including a match timestamp and video timestamp), information on possible related events, the type of event, the location of the event, the players and team involved, and detailed statistics on the event. Additionally, information on the current play of ball possession is included. An instance of the manner in which an event is documented within the events dataset can be observed in Figure 63. This particular event is of the primary type 'pass' and initially presents the general information, followed by the detailed information regarding the pass (accuracy, angle, height, length, recipient and end location). This is

Team	Length touch line (meters)	Length goal line (meters)
AC Milan, Atalanta, Bologna, Fiorentina, Frosinone, Genoa, Hellas Verona, Inter Milan, Juventus, Lazio, Lecce, Monza, Roma, Salernitana, Torino	105	68
Cagliari	105	65
Empoli, Udinese	105	67
Napoli	110	68

Table 1: Dimensions of playing field for different teams in Serie A

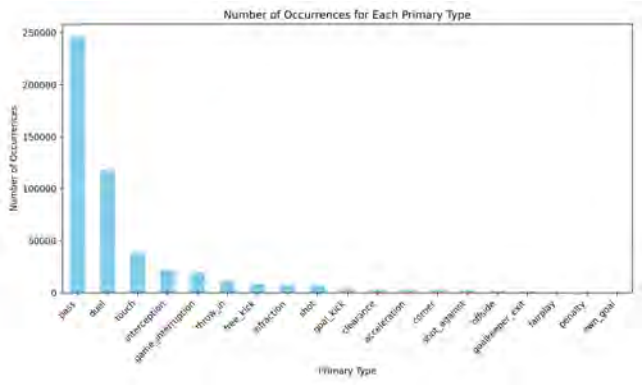


Figure 2: Number of occurrences of each primary event

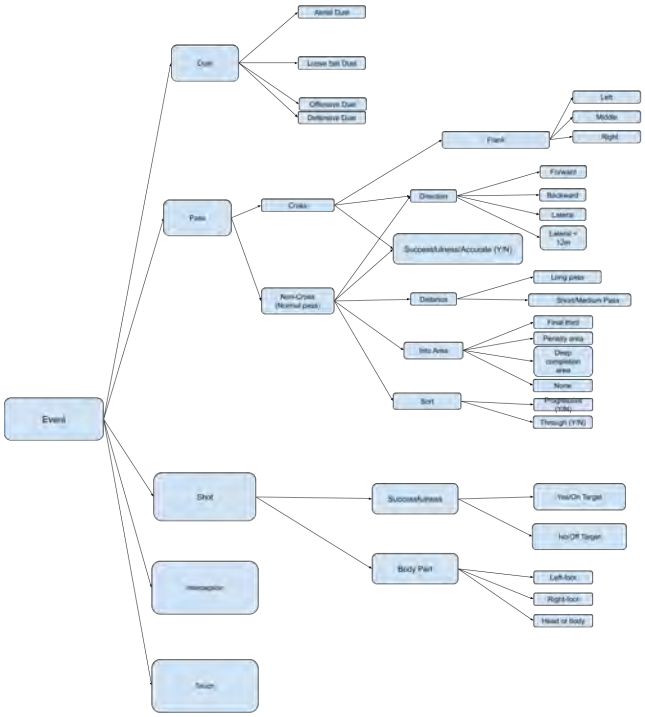


Figure 3: Visualization of classification hierarchy

subsequently followed by information on the possession statistics, which contains information on all events during this possession play, as well as the specific event's role within it.

The type of event is classified in primary types, and then described further by secondary types. An event can only have one primary type, which is always present, but may also contain one or more secondary types, including no secondary types at all. The dataset counts a total of 19 different primary types and 63 different secondary types. The different primary types and their occurences in the dataset are displayed in Figure 2. In the github repository for this project [78], the occurrences of different secondary types in combination with primary types can be found.

As previously stated, the event data comprises specific detailed statistics for certain primary types, classified according to a group of primary types (pass, shots, ground/aerial duels, infractions and carries). The aforementioned detailed statistics are presented in Table 11.

From these primary types, secondary types and detailed statistics, it is possible to identify a number of interesting distinctions that can be used to classify clips. The classification hierarchy employed in this research is illustrated in Figure 1. The actions to classify on have been selected based on three criteria: firstly, a threshold on the number of times they occur in the dataset; secondly, whether they occur during play of the game (and not when the game is interrupted); and thirdly, their importance in estimating a player's performance during a match. Subsequently, the descriptions of the various primary event classes are provided, based on the definitions of the actions as set forth in the Wyscout Glossary [83]. For additional details regarding each primary type, secondary type, or attribute, this glossary should be consulted.

- Duel: A challenge between two players to gain control of the ball, progress with the ball or change its direction. The duel is always a paired event, so for every offensive duel there will always be a defensive duel for another player. Possible types: offensive/defensive duel, loose ball duel, aerial duel.
- Pass: An attempt to pass the ball to a teammate. A pass can be either a cross or a non-cross pass.
- Shot: An attempt towards the opposition's goal with the intention of scoring.

- Interception: An act of player intercepting the ball by anticipating its movement when the opponent is shooting, passing or crossing.
- Touch: A touch (or missed touch) of the ball when the player is not doing a Pass or other clearly identifiable action.

4.2.2 TimeStamp-Related Data Exploration. An important decision within the classification process is the criteria used to determine the length of the input clips. The question thus arises as to the extent of the frames that must be included, both before and after the event timestamp, in order to facilitate the classification process. In order to arrive at an informed decision, it is essential to consider the following aspects: the degree of accuracy associated with the event timestamps; the manner in which the timestamps of consecutive events behave; the information that is necessary for specific classifications, and the extent to which the length of the clips affects this. This section seeks to examine and provide answers to these questions.

Accuracy of the event timestamps: The data pertaining to events observed by Wyscout is generated through the manual annotation of each event within the context of the game. A policy governing the placement of timestamps for different events during the annotation process has yet to be published by Wyscout. The annotation of different games is conducted by various individuals, who adhere to different standards for annotation, resulting in variations in the placement of timestamps. In addition, it is essential to determine the precise moment in time that the timestamp is being made. For instance, is the timestamp of a pass annotated as the moment that the ball propelles from the foot of the player passing the ball, or as the moment when the ball is halfway through the pass? To gain a comprehensive understanding of the general approach to annotation timestamps for events, the following test is conducted.

For each primary type event used in the classification process, 50 clips are randomly selected. The clips start at the timestamp of 2 events earlier and end at the timestamp of 2 events later. The clips are played frame by frame and a timestamp annotation is made when the event begins and ends according to the author's standards. The following standards are used as a policy for this:

- Pass: Timestamp start: when the ball leaves the foot of the passing player.
- Shot: Timestamp start: when the ball leaves the foot of the player taking the shot.
- Duel: Timestamp start: the moment when both players are fighting for the ball, or when one player decides to dribble against the other.
- Touch: Timestamp start: player touches the ball for the first time.
- Interception: Timestamp start: player touches the ball for the first time

The results are shown in Table 2 From these results, it can be concluded that the timestamps of the events are quite accurate, but most of the time are annotated a bit later than the event actually starting.

Behaviour of consecutive timestamps: To decide on the criteria for selecting the length of the input video clip, it is important to know how long the events last. How long does each event take before the next event occurs, and does this differ for different successive events? Table 3 and Table 4 show the mean and standard deviation of the duration for each primary event before the next timestamp and previous timestamp, respectively, is annotated for each of the other primary events.

Event	Mean Time Difference	Standard Deviation
Shot	0.0687	0.0693
Pass	0.1436	0.2078
Duel	0.2152	0.3195
Interception	0.0978	0.1159
Touch	0.0403	0.1156

Table 2: Mean time differences and corresponding standard deviations in seconds between the annotated starting timestamps from the dataset and the original start timestamps made by the author

As evidenced in Table 4, the mean time of duration when the previous event is a 'game interruption' is notably elevated. In accordance with the definitions provided in the Wyscout Glossary [83], a game interruption is annotated when the referee ceases play for reasons unrelated to the game itself. These include instances where an injured player requires medical attention, smoke is present, or fans invade the pitch. The elevated mean time required for the subsequent event to occur is an expected consequence. Furthermore, as can be found in Table 4, the mean time of duration when the previous event is an infraction is also considerable. According to the Wyscout Glossary, an infraction is defined as a foul. Following the occurrence of a foul, it often takes a brief interval before the match is continued the ball is returned to play.

The Wyscout Glossary defines the term "fair play" as a clearance of the ball when a player requires medical attention or when the ball is returned to the opposing team in accordance with the principles of fair play. These two occurrences invariably occur in the same order, thereby affording the annotator the option of choosing when to put the timestamp for 'fair play'. As illustrated in Table 3, the mean duration of the event "fair play" is 13.3 seconds, with a high standard deviation of 26 seconds when preceded by an interception. This elevated standard deviation is a consequence of giving the annotator's that choice. The 'fair play' annotations are occasionally made at the moment the ball is cleared, and at other instances at the moment the ball is returned to the opponent. In the latter case, this frequently occurs after the medical treatment has been administered, which often takes a while, thus resulting in a high standard deviation.

4.3 Data preprocessing

This section describes the steps taken to preprocess the data to make it ready for use in the contextual feature derivation methods, HPE methods and classification model. This section is divided into 3 subsections. In subsection 4.3.1 it is described how the information obtained in section 4.2 is used to edit the dataset to make it suitable for classification. Subsection 4.3.2 describes the various options for slicing the clips as input for classification, outlining the advantages and disadvantages of each approach. Subsection 4.3.3 describes the process used to make a selection of clips that are appropriate for the research goals of this study.

Next type	Duel			Pass			Interception			Touch			Shot		
	mean	std	count	mean	std	count	mean	std	count	mean	std	count	mean	std	count
Acceleration	1.72	0.81	630	1.89	0.91	1965	2.11	1.12	203	1.64	0.86	91			
Clearance	1.44	0.85	927	2.32	1.22	1047	1.62	0.89	449	1.33	0.79	375	1.78	0.82	27
Duel	0.37	0.79	69918	2.24	1.33	30185	2.21	1.51	3684	2.32	1.95	6401	2.94	1.91	18
Fairplay	5.44	1.49	6	4.03	2.49	109	13.3	26.50	9	5.72	3.21	25			
Game Interruption	4.10	5.04	4625	4.79	4.57	3451	4.15	4.33	4795	4.55	8.33	1255	4.82	4.02	2387
Goalkeeper Exit	0.75	0.77	108	1.74	0.69	340	1.73	0.94	23	0.45	0.37	10			
Infraction	1.68	0.94	6839	2.34	1.69	12	1.74	1.86	132	2.07	5.68	120	3.56		1
Interception	1.15	1.33	1226	1.37	1.17	15949	1.74	1.36	507	1.10	2.12	570	0.88	0.50	1792
Offside	1.96	1.25	48	2.60	1.88	720	2.48	2.26	67	2.18	1.46	22	4.15	1.29	4
Own Goal	0.26	0.08	2	0.56	0.31	7	0.71	0.22	5				0.90	0.53	5
Pass	1.94	2.28	28526	2.71	1.61	160198	2.92	4.04	9293	3.60	3.18	27984	4.44	3.05	56
Shot	1.10	0.71	2661	1.65	0.82	2269	1.64	0.88	538				1.42	0.69	29
Shot Against	1.31	0.99	16	1.36	0.57	32	1.13	0.54	33	0.68	0.48	14	1.23	0.65	2204
Touch	1.62	3.68	2752	1.72	0.94	29826	1.69	1.45	2167	1.75	2.98	1015	1.85	1.19	65

Table 3: Mean, standard deviation and occurences of time differences for every primary type and the type of the next action

Previous Type	Duel			Pass			Interception			Touch			Shot		
	mean	std	count	mean	std	count	mean	std	count	mean	std	count	mean	std	count
Acceleration	3.64	1.66	1010	4.32	1.70	1807	5.92	1.29	2	3.45	1.94	16	3.78	1.35	96
Clearance	2.61	1.59	245	4.31	2.48	646	1.73	4.74	63	2.72	1.85	153	1.70	0.65	142
Corner	1.50	0.86	1130	2.15	1.14	526	1.44	0.73	652	1.31	0.52	209	1.82	0.65	142
Duel	0.37	0.79	69918	1.94	2.28	28526	1.15	1.33	1226	1.62	3.68	2752	1.10	0.71	2661
Fairplay				8.22	13.87	147	5.51	13.13	44						
Free Kick	2.73	1.41	1384	3.17	2.08	4781	1.75	1.06	461	2.11	1.38	733	1.69	0.59	91
Game Interruption	25.15	23.00	4	52.28	46.72	328	47.34	41.80	21						
Goal Kick	3.31	1.01	1336	3.56	2.06	2380	3.63	1.62	89	2.20	1.41	665			
Goalkeeper Exit	1.56	3.63	227	9.69	7.30	362	0.50	0.63	8	4.22	5.64	31	1.87	0.89	26
Infraction	0.26		1	43.17	28.60	34									
Interception	2.21	1.51	3684	2.92	4.04	9293	1.47	1.36	507	1.69	1.45	2167	1.64	0.88	538
Pass	2.24	1.33	30185	2.71	1.61	160198	1.37	1.17	15949	1.72	0.94	29826	1.65	0.82	2269
Penalty	3.20		1	4.55		1							1.81	0.19	2
Shot	2.94	1.91	18	4.45	3.05	56	0.88	0.50	1792	1.85	1.19	65	1.42	0.69	29
Shot Against	12.04	15.94	111	41.72	37.82	1511	1.92	4.97	61	8.19	13.85	109	1.10	0.57	104
Throw In	1.78	1.15	2718	2.62	1.76	7094	1.67	0.63	572	1.85	1.22	736	2.74	0.92	14
Touch	2.32	1.95	6401	3.60	3.18	27984	1.10	2.12	570	1.75	2.98	1015	1.56	1.01	573

Table 4: Mean, standard deviation and occurences of time differences for every primary type and the type of the previous action

4.3.1 Editing the dataframe. The exploration done in 4.2 is used to edit the dataset. The subsequent modifications to the dataset are as follows:

- *Cross:* The Wyscout Data Glossary illustrates the various options for the 'flank' attribute (representing the flank at which the cross originates) as left, right, and centre. Conversely, an examination of the 8636 crosses present in the dataset revealed that 741 rows exhibited a 'nan' value for the flank attribute. Upon closer inspection of the starting coordinates associated with these crosses, the reason for the absence of a flank annotation was not found, given that they did, in fact, have starting coordinates within one of the flanks. Consequently, these rows were excluded from further consideration.
- *Pass:* The event data records the direction of the passes. A pass may be classified as either forward, backward, or lateral. The direction of a pass can be classified as forward (between -45 and 45 degrees, as seen from a player facing the opponents' goal), backward (between -135 and -180 and 135 and 180 degrees), or lateral (any area outside the aforementioned ranges). The Wyscout glossary specifies that lateral passes are defined as those exceeding a length of 12 metres. In the context of our dataset, all lateral passes that are 12 metres or shorter are also annotated as lateral. Of the 78,402 forward passes, 185 were not within the expected interval for a

forward pass. Similarly, 123 of the 36,181 backward passes and 1,343 of the 87,778 lateral passes were not within the expected interval. These passes were removed from the dataset to ensure clear distinction between the different types of passes.

- *Consecutive events*: As described in section 4.2.2 a few consecutive actions result in very high mean time difference between two events. These combinations of events are removed from the dataset to exclude long duration clips.
- *Long duration events*: After removal of the aforementioned long combinations, still a handful events in the dataset have a duration of more than 20 seconds. Since the events on which classification is performed almost never have a duration this long, these events are probably some kind of wrong annotation, and are dismissed from the dataset.

Following the aforementioned edits to the data frame, a random sample of 8,000 instances was selected to contribute to the training, validation, and test sets. This is due to the fact that, subsequent to the implementation of all requisite models, it became evident that not all 26,000+ clips originally sampled could be processed, given the computational resources that would be required. Figure 4 illustrates the distribution of the primary events of the 8,000 randomly sampled clips.

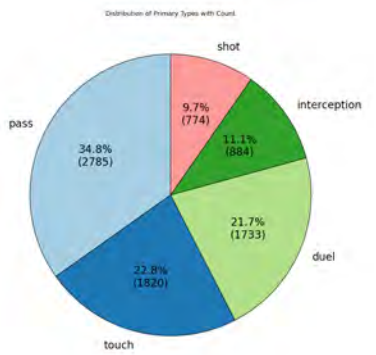


Figure 4: Distribution of the Primary Types of the 8000 random sampled events

4.3.2 Slicing Clips. As outlined in section 4.2.2, the duration of different actions varies, depending on the action itself and the action that follows it. In order to create an effective classification schedule, it is essential to consider the specific information streams from different moments in the clip, as each contributes to the overall classification process in a unique manner. To illustrate, in order to classify the degree of success of a shot, it is necessary for the clip to contain the moment at which the ball reaches the goal or the backline. In order to classify the body part, the clip must contain the moment at which the ball is released by the player who is taking the shot. An additional example of the necessity for specific information is the classification between a touch and an interception. In order to distinguish between these two actions, it is necessary to consider the preceding action. If a teammate passes the ball in the preceding action, the action is classified as a touch; however, if an opponent passes the ball in the preceding action, the action is classified as an interception.

In order to make an informed decision regarding the selection of clips from the game footage, it is essential to consider the aforementioned two aspects. Subsequently, three distinct slicing methods are presented, accompanied by an analysis of their respective advantages and disadvantages.

Slicing method 1: Stationary Slicing: The stationary slicing method is designed to treat the slicing of each clip in a uniform manner, irrespective of the temporal stamps associated with the preceding and subsequent events. The aforementioned method is illustrated in Figure 5. As can be observed, the clip is sliced at a cut-off timestamp that is x length units from the current event. The value of x is consistent for each event. The advantage of utilizing a stationary slicing method is that each training clip is of an identical length, simplifying the processing of the data through the pipeline. It should be noted that stationary slicing also has a number of disadvantages. The stationary slicing method may result in the occurrence of multiple events within a given time period, potentially complicating the process of identifying the event to be classified. Furthermore, the utilization of a stationary cut-off results in the incomplete execution of certain actions prior to the designated cut-off time. To illustrate, if the cut-off time for a shot is 1.5 seconds, but the shot exceeds this time threshold to reach the goal, the clip may lack clarity regarding whether the shot was successfully saved by the goalkeeper.

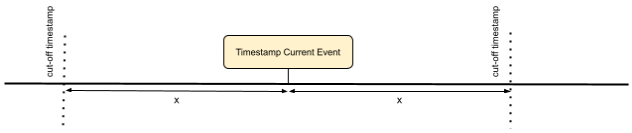


Figure 5: Timeline displaying the cut-off procedure of stationary slicing

Slicing method 2: Dynamic slicing method 0.5: A second slicing method, which can be seen in Figure 6, employs a dynamic approach whereby the cut-off point occurs exactly in the center, with reference to both the preceding and the subsequent event. This slicing method has the advantage of ensuring that the event on which the classification is based is always the only event visible in the clip. However, this approach presents a disadvantage in that the preceding action is not visible in the clips, and the conclusion of the ongoing action is frequently not observable as well. This slicing method is particularly useful for the classification of the three primary types of shot, pass and duel but is less effective for the primary types of touch and interception, as it is not possible to identify which team was in possession at the time of the previous event. Additionally, this cutting method is not optimal for classifying several secondary types, such as shot accuracy, due to the lack of visibility of the end of the action.

Slicing Method 3: Dynamic slicing method 1.25: The third slicing method is a dynamic slicing method, depicted in Figure 7, which extends the previous slicing method by performing the cut-off at $1.25x$ rather than at $0.5x$. This ensures that there will always

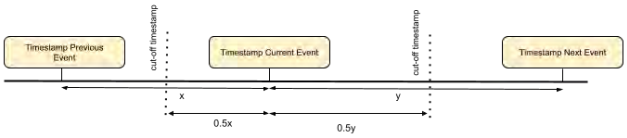


Figure 6: Timeline displaying cut-off procedure of dynamic slicing method: 0.5

be three events in the clip, and that the start and end of each event will always be fully visible.

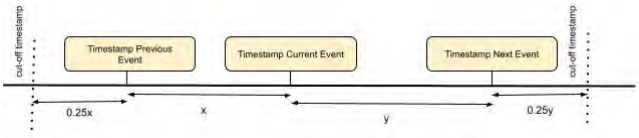


Figure 7: Timeline displaying cut-off procedure of dynamic slicing method: 1.25

In this research, the dynamic slicing method with a 1.25 cutoff is selected. All clips have their own length, which can be formulated as $1.25 \times x + 1.25 \times y$, with x being the duration in seconds between the timestamp of the previous event, and y the duration in seconds between the timestamp of the next event. The clips are processed through the pipeline with this clip length.

4.3.3 Classify Monocular View. As stated in the research objective, the classification of actions is to be conducted using a single-view monocular camera. The dataset obtained from Wyscout differs in that it comprises broadcast videos incorporating footage from multiple cameras, replays, and visual enhancements. To ensure the robustness of the classifying model with respect to the monocular viewpoint, the clips sliced in Section 4.3.2 undergo a selection process to exclude those containing footage obtained from sources other than the monocular viewpoint.

The selection procedure is as follows: 5 different frames are taken from each clip. The initial frame, the $\frac{1}{4}N$th frame, the $\frac{1}{2}N$th frame, the $\frac{3}{4}N$th frame, and the concluding frame (with N representing the total number of frames in within the given clip). The aforementioned five frames are then subjected to the keypoint detection algorithm delineated in Section 5.2.1. If all five frames return at least four keypoints with a confidence value exceeding 0.5, the clip is deemed to have successfully completed the selection procedure. The frames at different positions are selected because they are distributed evenly over the entire clip length. Furthermore, in the case that the preceding action results in the ball being taken out of play, or if the current action is of a similar nature, there is a high probability that a replay, zoom-in, or visual add-on will occur. It is therefore important to perform keypoint detection on both the first and last frame.

Figure 8 shows the distribution of the primary events after the appliance of the selection procedure. It is noteworthy that the distribution of the five primary events remained largely unchanged in terms of proportion, contrary to expectations that the proportion of shots would decline. The dataset did experience a slight reduction when compared to the sample size before applying the selection procedure. 65.4% of the samples passed the selection procedure.

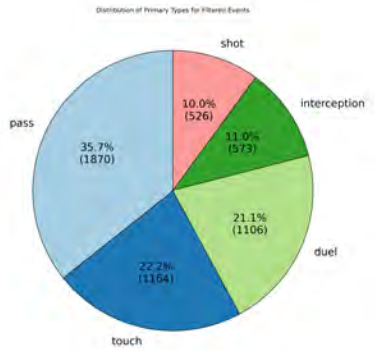


Figure 8: Distribution of primary types after selection procedure described in 4.3.3

5 CONTEXTUAL DATA DERIVATION METHODS

This Section is used to describe the different computer vision models used and how these are implemented on the data of this research to create data suitable for feature engineering for the classification model. Firstly in Section 5.1 the process of creating a suitable evaluation set is described. This is followed by Section 5.2, which discusses the process of field registration and homography mappings. This is followed by Sections 5.3 and 5.4 which describe the existing models used and their appliance in this research for the detection and tracking of respectively players and ball. Section 5.5 concludes by presenting a self-designed team classification model.

5.1 Creating the evaluation set

As the field localization algorithm, player detection, ball detection and human pose estimation algorithms are pretrained models, trained on datasets that differ from the one used for training the classification algorithm, it is necessary to assess their performance on the data used in that process. The evaluation set is representative of the data employed in the training process of the classification model. It is essential to evaluate the models against a range of scenarios and circumstances that may present visibility challenges.

The models must be tested against a variety of weather scenarios. This includes smog at the start of the game as a result of fireworks used by fans just before the start, heavy difference in shadows on the field, games lighten up by artificial light and games lighten up by daylight

Secondly, model specific difficulties need to be addressed and evaluated. Lets start with occlusions. Is the player detection model able to detect an occluded player? Is the human pose estimation still able to detect a visible arm while the rest of the player is occluded? Is the ball detection model able to detect a ball that is partly occluded by a player? Another difficulty that needs to be addressed is color similarity. Is the ball detection model able to detect a ball when

a player has it at its feet, where the ball and the shoes have the same color? Is the ball detection model able to detect the ball when it is in the air, and visually is in front of the public? Is the player detection model able to detect players wearing green kits, similar colored to the field itself? Another difficulty could be the detection of people who are not players or part of the referee staff. Is the player detection model detecting people in the stands? Or coaches at the sideline?

This evaluation set consists of 120 images that highlight these different scenarios/difficulties. By evaluating the different models on this evaluation set, we gain knowledge on when the models perform poorly and how we should anticipate on that to smooth-en the classification process. The different models will be tested on this evaluation set and the important findings will be mentioned. An example image from all seven different scenarios can be found in Figure 9

5.2 Field Registration

Many features, like velocity, distance to goal opponent and number of players within 10 meter radius, that are used within the action classification process are based on location. In this research, a difference is made between pixel location, and pitch location. Pixel location regards the absolute location of a detection on the image in pixel values. Pitch location is the pixel location transformed to a location on the 2D planar field that represents the pitch.

To transform the pixel location of a player detection correctly into a location on the 2D planar field, homography is used. [26] describe a homography as an invertible mapping from P2 to itself such that three points lie on the same line if and only if their mapped points are also collinear. In other words, a homography describes the relation between two images of the same plane. It can be used for image registration, image rectification and for calculating the movement of the camera that took the images. In the context of our research, homographies are used to map a frame of the footage, onto the top-down view of a football field.

[26] also gave an algebraic definition by proving the following theorem: A mapping from $\mathbb{P}^2 \rightarrow \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix H such that for any point in $\mathbb{P}^2$ represented by vector $\mathbf{x}$, its mapped point equals $H\mathbf{x}$. This tells us that in order to calculate the homography that maps each $\mathbf{x}_i$ to its corresponding $\mathbf{x}'_i$, it is sufficient to calculate the 3×3 homography matrix H [4].

Homographies can be estimated by finding feature correspondences between images, or in our case, feature correspondences between images and planar projection of a football field. Homography estimation algorithms can make use of point feature correspondences, or other features such as lines or conics. All three, points, lines and conics, are available on a football pitch. Since the broadcast view does not include the whole field in every frame, it is possible that the totality of lines and conics are not visible each frame. That is why in this research is chosen to estimate the homography using point feature correspondences.

This section is further divided into three subsections. In Subsection 5.2.1 the model used to detect keypoints in each frame is described and evaluated against our test set. This is followed by Subsection 5.2.2, in which the process of using the detected keypoints to create a homography matrix for different planar field sizes is described and examples of field localization are shown.

5.2.1 Keypoints Detection. The pre-existing model used in this research for the keypoints detection is a component of the SoccerNet Camera Calibration Challenge 2023 [62] winner, developed by Sportlight Technology team. The SoccerNet Camera Calibration Challenge 2023 aimed to generate accurate camera calibration parameters, including both intrinsic and extrinsic values, using individual frames extracted from football broadcast videos. The team constructed a hybrid approach that combines a keypoint detection model and a line detection model. In this research, their keypoint detection model has been used in the process of field registration.

The keypoint detection model of the Sportlight Technology team distinguishes itself from other field localization methods like [29] and [30], by making predictions on more keypoints on the pitch. While a minimum of four keypoints is required to create a homography matrix, this increased number of predicted keypoints makes the model more practical for use in broadcast view settings, where complete lines or conics are often not visible.

The model predicts 57 keypoints, categorized as follows:

- *Intersections*: These are the 30 visible intersections of the straight pitch lines. These intersections are marked with a red dot in Figure 10. These points were included in the annotation of the SoccerNet dataset.
- *Conic Intersections*: These are the 6 visible intersections between the straight and conic pitch lines. These intersections are marked with a blue dot in Figure 10. These points were included in the annotation of the SoccerNet dataset.
- *Tangent points*: These are the 8 tangent points of tangent lines from a known point to the circles. The tangent lines used are the lines from the top and bottom of both penalty boxes onto its circles, and the lines from the both the top as bottom intersection of the middle line with the sidelines onto both sides of the middle circle. These tangent points are marked with a purple dot in Figure 10. These points were not included in the annotation of the SoccerNet dataset, but were analytically derived using the ellipse equation and the known location of an external point.
- *Additional points*: These are the 13 dark-green dots in Figure 10. These points were not included in the annotation of the SoccerNet dataset but were derived using the homography created with the points from the previous categories, and via that way added as annotations to the dataset.

A detailed description of the network architecture can be found in Appendix A.1.

Performance of the keypoints detection model: The keypoint detection makes a prediction on the pixel location of every keypoint with an attached confidence score. This confidence score should ideally be close to 0 for keypoints that are not visible in the frame and close to 1 for keypoints that are. The keypoint detection algorithm was performed on the evaluation set. In Figure 11 the distribution of the confidence values is shown. As expected there are a lot of confidence values close to zero, since in every frame, most keypoints will not be visible. Two other things can be observed. Firstly,

(a) (b) (c)

(d) (e) (f)

Figure 9: Example images of evaluation set for different scenarios. With in 9a scenario day-light, in 9b scenario artificial light, in 9c scenario fog, in 9d scenario of ball occlusion, where the ball is vaguely visible since it as the foot of a player, in 9e an example of both scenarios player and keypoint occlusion, where both a few players are partly visible and a field-keypoint (touch of penalty box and its circle) completely occluded because of other players standing in front of them, and in 9f scenario green-kit, in which a team wears a partly green kit, making them hard to distinguish from the field.

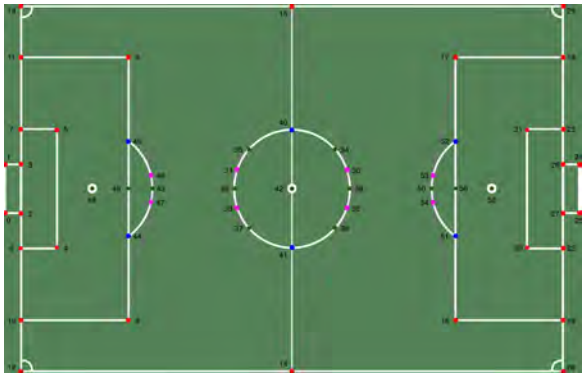


Figure 10: The 57 keypoints predicted by the model and their corresponding place in the prediction vector. Red dot: line-line intersection. Blue dot: line-conic intersection. Purple dot: conic tangent point. Dark-green dot: additional points projected by homography

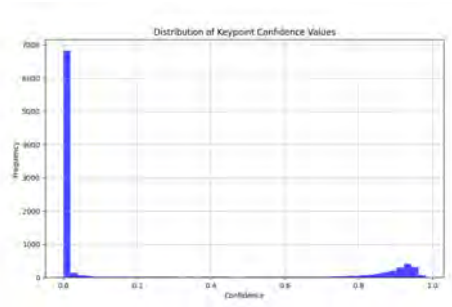


Figure 11: Distribution of the confidence values of the predicted keypoints on the evaluation set

there is a small cluster of higher confidence values (ranging approximately from 0.7 to 1.0) towards the right side of the histogram, these probably are the keypoints that are visible in the frame, and they are predicted with quite a high confidence. Secondly, there is a noticeable gap or low frequency of keypoints with midrange confidence values (between 0.1 and 0.6). This shows that most keypoints are either detected with very low confidence or very high confidence.

In order to create a homography matrix with high precision, it is necessary to identify as many keypoints as possible. This is described in greater detail in Subsection 5.2.2. However, it is also necessary to ensure a certain degree of reliability in the selection of keypoints for this purpose. It is therefore essential to consider the confidence value assigned to each keypoint. Different minimum confidence values to be selected as a keypoint are tested against each other. In Figure 12, two examples are given of prediction of keypoints on the same frame, with different confidence values to accept it as a keypoint. As is illustrated in Figure 12a, for a confidence value of 5 percent, multiple keypoints are predicted at the same location, of which one is the correct keypoint and the other is a similar keypoint (typically a mirrored keypoint in either horizontal or vertical mirror direction). With an increase in the confidence value to 0.2, as can be seen in Figure 12b, this double keypoint prediction on one location disappeared. An examination of Figure 12d, reveals that a confidence value of 0.05, predicts keypoints in

the stands. A number of these points appear to be similar to the keypoints on the goalposts, which is why they are considered as keypoints at this low confidence level. An increase in the confidence value to 0.2 (see Figure 12e) reveals that the least plausible of the predicted keypoints at 0.05 have been eliminated, although an invalid prediction persists. Further elevation of the confidence value to 0.5 (see Figure 12f) demonstrates the complete removal of this prediction.

To balance the reliability of the keypoints with the need for at least four keypoints for homography and for precizer homography (See Section 5.2.2), the confidence threshold to consider a prediction as valid is set to 0.5.

As mentioned in the detailed description on homography in Appendix A.1, the training dataset is very similar to the dataset used in this research. Despite this, it is always good to check its performance on the data used in this research. The keypoints detection method was tested on manual annotated data on the three scenarios of 'Fog', 'Natural Light' and 'Artificial Light'. Since this data is manual annotated, only the keypoints on the Intersections and Conic Intersections are annotated as they are detectable with the eye. These annotations also include keypoints that were not visible because of players standing in front of them. The results achieved by the keypoint detector are shown in Table 5

The keypoint detector achieves high confidence scores on all three scenarios. In these evaluation scenarios, keypoints that are occluded due to players standing in front of them are also included. Figure 13 is an example of such an occluded keypoint. As can be seen, the keypoint detector is capable of detecting the occluded keypoint, as it has sufficient information by recognizing the clearly visible lines that create the intersection of the keypoint.

5.2.2 Homography Estimation. The Kepypoint Detection algorithm, as described in Section 5.2.1, returns the prediction of the pixel-location of the 57 different keypoints, together with the confidence score of the prediction.

A homography is a 3x3 matrix H that describes a projective transformation between two planes. It allows you to transform points from one plane (e.g., an image) to another plane (e.g., a sports field). If $\mathbf{p} = (x, y, 1)$ are the homogeneous coordinates of a point in the image and $\mathbf{P} = (X, Y, 1)$ are the corresponding coordinates on the field, the relationship can be written as:

$$\mathbf{P} = H\mathbf{p}$$

where H is the homography matrix.

To compute H, we need at least four pairs of corresponding points between the two planes. Let (x_i, y_i) be the coordinates in the image and (X_i, Y_i) be the coordinates on the field for $i = 1, 2, 3, 4$. The relationship can be written as:

$$\begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} = H \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

Further detailed information and mathematical formulation of the homography matrix can be found in Appendix A.2

Subsequently, the homography matrix H can be employed to map the pixel positions of the identified player and ball objects onto the planar field. This enables the calculation of features, including ball velocity and distances to specific points, which are utilized in the classification process. It is important to consider the variation in field dimensions, as outlined in Section 4.1.2. The variations in the dimensions of the pitch, in terms of length and width, results in a distinct position within the coordinate system for each keypoint. Accordingly, a distinct pitch coordinate system is required for the homography mapping, depending on which team is playing at home in the given clip. In each coordinate system, a unit on the x and y axes represents a length of 1 metre in real life. This implies that the various field types have distinct maximum values within the coordinate system, and that the keypoints are situated at disparate coordinates for each type. By employing the official pitch dimensions as delineated in the FIFA official rules book [21] and the disparate width and height measurements for each type, the planar pitch coordinates can be derived. Table B.2 presents the various planar coordinates for the different field types. The keypoints numbered 0, 1, 24 and 25 are disregarded, as they correspond to the upper surfaces of the goal and fall outside the boundaries of the 2D planar field.

5.2.3 Appliance of the homography matrix. In this research project, the accurate estimation of the homography matrix for each frame in a video clip is a crucial component of the processing pipeline. This section outlines the procedure employed for the estimation of homography matrices.

Due to constraints in computational resources, keypoint detection is performed on every third frame of the video clip. Given a frame rate of 25 frames per second (fps), this approach results in keypoint detection occurring approximately 8.3 times per second, which subsequently leads to homography estimation at the same frequency. In order to obtain the homography matrix for the two intervening frames between each detected keypoint frame, the known homography matrices from the surrounding frames are interpolated.

The homography estimation process is contingent upon the detection of keypoints with a confidence score exceeding 0.5. The accuracy of the homography estimation is enhanced with the availability of a greater number of reliable keypoints. The estimation process is conducted using RANSAC [48], which chooses subsets of 4 points from the available keypoints, calculates the homography for those,count the number of inliers, and keep the homography if it is better than any homography yet found for that frame. This way, the homography estimation process is designed to be robust, ensuring that erroneously detected keypoints are identified as outliers and excluded from the homography computation.

In certain instances, some frames contain fewer than four predicted keypoints with a confidence score greater than 0.5. In these cases, a direct homography estimation is not feasible. For such frames, the homography is estimated by interpolating between the closest known homographies from previous and subsequent frames. If a frame lacks either a prior or subsequent known homography, the homography from the closest known frame is used directly.

As explained in Section 4.1.2, the dimensions of the football fields in the clip differ based on the home-playing team. For every match in the dataset, it is manually annotated in which stadium the match is played. So for every clip in the dataset the field dimensions

(a) confidence = 0.05 (b) confidence = 0.2 (c) confidence = 0.5

(d) confidence = 0.05 (e) confidence = 0.2 (f) confidence = 0.5

Figure 12: Keypoints detection at different minimum confidence values

Table 5: Statistics of Keypoint Predictions by Condition

Condition	Total Annotated Keypoints	Correctly Predicted Keypoints	Accuracy Percentage
Fog	54	52	96.30%
Natural Light	168	159	94.64%
Artificial Light	136	131	96.32%

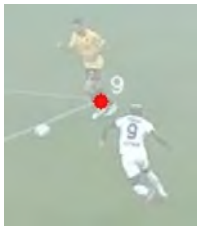


Figure 13: Illustrative example of the capacity of the keypoint detection model to identify keypoints despite their occlusion by a player.

of that field are known. These different field dimensions are also incorporated into the homography estimation.

The homography estimation is used to project the detections of objects, specifically the ball and players, into planar coordinates corresponding to the football pitch. This transformation uses a single pixel coordinate for each object.

For the player objects, the selected pixel is located at the centre of the bottom of the bounding box. This pixel corresponds to the point at which the player's feet make contact with the ground, thereby accurately representing the player's position on the two-dimensional pitch. It is of critical importance to select this particular pixel, as selecting the pixel corresponding to the player's head would inevitably result in an erroneous planar coordinate on the pitch. Given that players can be quite tall, selecting the head would result in the corresponding point on the pitch being positioned further away than the player's actual standing position in real life.

In the case of the ball, due to its relatively small size, the pixel at the center of the ball is used to represent its position on the 2D pitch. Some drawbacks that occur when projecting the detection of players and ball into the planar coordinates are discussed in Sections 5.3 and 5.4 respectively.

5.3 Player Detection and Tracking

It is crucial to ascertain the locations of players, as their trajectories, velocities, and distances between each other and with the ball contain vital information for the classification of certain actions. This section outlines the model employed for detection and tracking, its advantages and limitations, with illustrative examples from the evaluation dataset, and the manner in which the detections and tracking outcomes are processed for the classification model.

5.3.1 Used Model. As previously outlined in Section 3.2, a number of models have already been developed for the detection and tracking of soccer players. However, the majority of these models are not appropriate for the scenario under consideration in our research, which focuses on a challenging single-view camera setup.

Furthermore, the classification model is intended for use on camera footage from Ajax youth games, which often feature a more zoomed-out view compared to TV broadcasts. This results in player objects having lower resolution. The majority of existing player detection and tracking models are trained on broadcast footage with high resolution, which can present difficulties in detecting players of smaller sizes.

To address these challenges, this research utilizes the player detection and tracking model proposed by Hurault et al. in their paper "Self-Supervised Small Soccer Player Detection and Tracking Algorithm." Their model is well-suited for wide-angle video games as it is robust to small player sizes. The method is specialized for soccer and does not require any labeled data. The pipeline involves two steps: first, player detection, followed by tracking based on the detection results. The following subsections describe the process of training their detection and tracking algorithms.

Detection Algorithm: To develop an accurate soccer player detector, Hurault et al. employ a three-step process leveraging a pretrained object detection model without requiring any manual annotations on the target dataset. First, they apply a pretrained human detector, based on the Faster-RCNN network [54] in combination with the Feature Pyramid Network [38], referred to as the initial teacher network T_{init}. This model is used to automatically generate preliminary annotations on a set of unlabeled soccer images X taken from a subset of the SoccerNet dataset [22]. These initial annotations $T_{\text{init}}(X)$ contain noise in the form of false positives and false negatives.

The annotations are refined by removing false positives, such as detections of supporters and coaches, through a field detection process that uses green filtering, contour detection, and line extraction, as used in [9]. Conversely, missed detections are added using a blob detection strategy, which is realized via green filtering, contour detection, and human detection in the regions of the contours with appropriate sizes. This correction process results in a new set of labels Y_c, which is then used to fine-tune the initial teacher network, creating a more accurate teacher model T_f.

The fine-tuned teacher model T_f is subsequently used to train a smaller, more efficient student network S through knowledge distillation. Inspired by the idea of [54], the student network architecture is slightly modified to incorporate contextual information around detected objects, thereby improving detection accuracy for small players. This is achieved by concatenating feature maps from regions of interest with those of an enlarged surrounding region. To ensure robustness to varying player sizes, especially for smaller players that often appear in full-field views, the training data is augmented by randomly downscaling each image by a factor between 0.1 and 1, applying zero-padding to maintain the original image size. This comprehensive approach enables the final student network to effectively detect soccer players of different sizes and contexts. An illustration of the soccer player detection method proposed by Herault et al., can be found in Figure 14

Tracking Algorithm: To effectively track soccer players across video frames, Hurault et al. propose an unsupervised, fast, and accurate tracking framework that leverages the previously described player detector model. The goal is to associate player detections between consecutive frames in a video sequence $\{u_t\}_t$, where t denotes time. A player trajectory is defined as an ordered list of

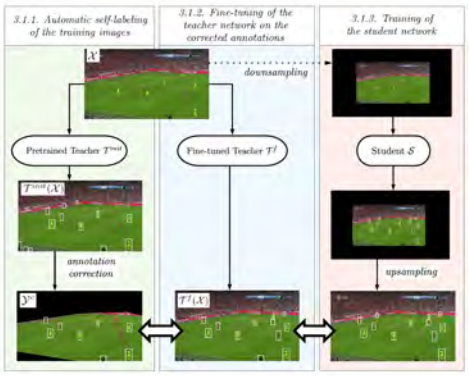


Figure 14: Illustration of the soccer player detection method. Each column corresponds to one step of the proposed method explained in Section 5.3

bounding boxes $T_k = \{b^k_{t_1}, b^k_{t_2}, \ldots\}$, with $t_1 < t_2 < \ldots$. At each frame u_t, the player detector S is used to compute a set of potential bounding boxes $B_t = \{b^{k1}_t, b^{k2}_t, \ldots\}$ by extracting detections with confidence higher than a threshold $\sigma_{\text{track}} > 0$.

The tracking approach then consists of two main steps. First, a spatial consistency association is applied, where bounding boxes from the current frame are matched to those in the previous frame based on their spatial overlap. Specifically, if the Intersection-over-Union (IoU) between a bounding box b^i_t in the current frame B_t and a bounding box b^j_{t-1} in the previous frame B_{t-1} exceeds a threshold τ_{IoU}, these boxes are considered part of the same track. If a bounding box from a previous frame is associated with multiple detections in the current frame, such associations are discarded to maintain consistency. However, as explained in [47] and [3], this criterion assumes significant overlap between tracked targets in successive frames, which may not hold in cases of low frame rate, fast player movements, or intense camera shifts.

To handle such scenarios, Hurault et al. implemented a visual consistency measure. This involves extracting visual embeddings from the player detector model and matching bounding boxes based on their visual similarity to previously deactivated tracks. This additional step improves tracking robustness in challenging conditions.

However, for the purpose of this research, where the frame rate is 25 fps, the spatial association measure is sufficient. To conserve computational resources, the visual embeddings measure is therefore omitted. To conserve lost identities of players by being not detected for a few frames, a different re-identification is used, which will be described in Section 5.3.3

5.3.2 Players detection on the evaluation dataset. As the tracking branch of the player detection and tracking is based on the existing detections, it is essential to understand the functioning of the detection algorithm on the data set in order to ensure the accuracy and reliability of the results. What are the shortcomings of this approach and how should they be addressed in subsequent processing before being incorporated into our classification model? The player detection algorithm has been evaluated using the aforementioned

evaluation set, as outlined in Section 5.1. The following findings are important to mention.

The detection algorithm can be set to a range of confidence scores to identify instances of detection. It is essential that the confidence score strikes a balance between the ability to detect players and the capacity to make accurate identifications. A variety of confidence scores have been subjected to testing. Figure 15 illustrates several instances of confidence equal to 0.2. A confidence score of 0.2 allows the detection algorithm to make a significant number of detections, which is beneficial in terms of identifying players across multiple frames.However, as can be seen in Figures 15c, 15c and 15c multiple detections are made on one player. Figure 15d displays that with a low confidence as 0.2 even the ball is detected as a player object.

Given that at confidence level 0.2 there is an absence of balance between the quantity of detected objects and the quality of the detected objects, it is evident that higher levels of confidence are required. Figure 16 shows two frames with its detections at different confidence level 0.95 and 0.8 respectively. As can be seen in figures 16a and 16c the detection of partly occluded players does not happen at that confidence level. In contrast, this does happen at a confidence level of 0.8 as can be seen in 16b and 16d.

Following an evaluation of the balance between quality and quantity of the detected player objects for varying confidence thresholds, the decision has been taken to pursue further research with a player detection confidence level of 0.8.

The detection model demonstrates robust performance across a range of weather conditions, including natural light, artificial light and fog. Figure 17a illustrates the effectiveness of the detection model in foggy weather conditions. The figure demonstrates that the algorithm is capable of accurately identifying players in such environments.

Nevertheless, an initial limitation of the detection algorithm is apparent in Figure 17a. It is evident that the linesman positioned in front of the advertising boards has not been identified. The detection algorithm displays inconsistency in the case of objects situated in front of the aforementioned advertising boards. This is illustrated in Figure 18, which depicts two scenarios. In scenario 18a, the algorithm identifies a player object with similar colours to its background advertising board. However, in scenario 18b, the algorithm is unable to detect a referee object with different colours to its background. However, in the same figure, another player object with a similar background is correctly identified. Overall, the detections in this region are somewhat inconsistent, with a higher frequency of correct detections than false negatives. As this often concerns the linesman, who is not involved in the game, the impact of this inconsistency will be minimal to the final classification model but still important to note.

Another potential limitation of the detection model is illustrated in Figure 17b. The image displayed in this frame was captured during a sequence in which the camera was in motion at a relatively high speed, resulting in a somewhat indistinct visual representation. It can be observed that the majority of players are correctly identified; however, some players situated at a greater distance and exhibiting less contrasting colours are not detected.

Given that players may occasionally be positioned differently from those typical of standing or running, for instance following a duel where they fall to the ground, it is essential to understand how the detection model responds in such circumstances. As illustrated in Figure 19a, the detection model demonstrates the capacity to accurately identify players in non-standard positions.

It is also important to consider the detection of non-player or non-referee objects, as this will have implications for subsequent processing. Firstly, it is notable that the detection model is able to successfully identify and exclude the public in the stadium from its object detection capabilities. However, as illustrated in Figure 20, the model does detect coaches situated adjacent to the field on the camera's side. In accordance with the regulations set by the FIFA [21], coaches are prohibited from entering the playing area and are required to maintain a designated position on the sidelines, as illustrated in the figure by the dotted lines. Such designated areas are situated at a distance from the playing area. Section 5.3.3 briefly explains how the exclusion of coaches is handled.

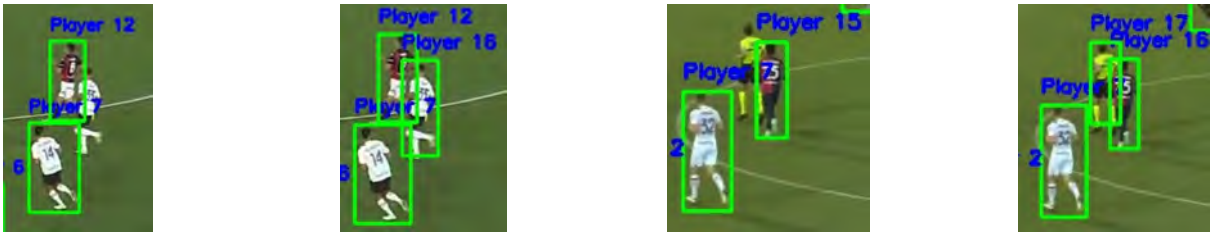
It is also noteworthy to observe the manner in which the detection method responds to instances of player occlusion. Given that players are frequently engaged in duels for the ball and positioned in close proximity to one another, it is not uncommon for one player to occlude the other from the camera's view. Figure 19 illustrates two of these aforementioned occlusions. In Figure 19b, it can be observed that players engaged in duels are accurately detected, provided that the occlusion is not excessive. However, in Figure 19c, where the player in the red kit is largely obscured by the player in the white kit, the player detector fails to recognize the red kit player as a player object.

A limitation of the tracking algorithm is its potential to misidentify player objects when two player objects move in close proximity to each other and then move away from each other. An illustrative example of this phenomenon can be observed in Figure 21. The figure illustrates this in four successive frames, with a particular focus on the two players situated in the two most rightward positions. The first frame illustrates that both players have distinct PlayerIDs, 6 and 11, respectively. In the subsequent frame, player 6 is occluded to such an extent by player 11 that it is no longer possible to detect it, resulting in the disappearance of the detection. Subsequently, in the next frame, the two distinct players are identified as a single entity and attributed to player 11 due to spatial consistency association with the detection of player 11 in the preceding frame. In the final frame, the two distinct players are once more identified individually. However, the player in red has now been assigned the detection ID of the player in white, who originally had the PlayerID 11. Consequently, the white player, which initially had the PlayerID 11, is now recognized as a new player object with the PlayerID 19.

5.3.3 Appliance of the detection algorithm. Once detected, the next goal is to establish a frame by frame positioning of the individual players in order to understand the play in total. Thus, the tracking algorithm described in Section 5.3.1 keeps track of the identified objects' movements. This tracking algorithm uses the information from the previous frame for initial conditions on tracking, and bases its location-wise re-identification on an IoU. But since this tracking algorithm bases this only on the previous frame, a problem arises in the following scenario. Once the players have been identified, the next objective is to establish a frame-by-frame positioning of each individual in order to gain a comprehensive understanding of the play as a whole. Accordingly, the tracking algorithm, as described in

(a) (b) (c) (d)

Figure 15: Illustration of player detections at confidence level 0.2. As can be seen, at this confidence level, multiple detections are made at one player



(a) confidence = 0.95 (b) confidence = 0.8 (c) confidence = 0.95 (d) confidence = 0.8

Figure 16: Illustration of player detections on two different frames with confidence score 0.95 and 0.8. As can be seen, the partly occluded players are not detected at confidence level 0.95, but are detected at confidence level 0.8

(a) Detections during fog (b) Detections during rapid camera movement

Figure 17

Section 5.3.1, maintains a record of the movements of the identified objects. The tracking algorithm utilises the data from the preceding frame to establish the initial conditions for tracking and bases its location-wise re-identification on an IoU. However, as this tracking algorithm is based solely on the preceding frame, a potential issue arises in the following scenario.

A player who had been previously identified by the detector and was tracked by the tracking algorithm was dropped since it was not detected for a frame. This can occur when players of the same team merge together, as described in reference 5.3.1, or if a player is occluded by another player. In such cases, re-identification via visual similarity should facilitate the re-identification of a specific player object and reconnect the new detection to the preceding tracking path. During the process of detection and tracking, it was observed that the visual similarity re-identification method was not particularly effective.

A minimum distance correlator, inspired by the methodology outlined in [57], is proposed as a solution to this problem. The locations of the bounding boxes are recorded for each individual frame. In the event of a player being dropped, the detection model will subsequently identify the player after a designated period of time. In the absence of a correlation between a box from the previous frame and the player in question, no established position can be attributed to him. The algorithm then determines whether the player who has been dropped before is situated in close proximity to the position of the player in question. The permitted distance is

(a) Player in front of advertising board succesfully detected

(b) Linesman (left) and Player (right) in front of advertising board, respectively unsuccessfully and successfully detected

Figure 18

a fixed value multiplied by the number of frames elapsed since the player was last observed. The maximum distance is 100 pixels, with a minimum frame difference of 2 and a maximum frame difference of 10. Furthermore, the algorithm determines whether the classification of the two player objects belong to the same team. If the player was within the distance of any previously dropped players, he is then correlated back to the original player, addressing the issue of players being dropped. Given that a back-and-forth movement of the camera could result in the loss of two older detections that have exited the frame and share the same last pixel position, the youngest lost detection will be the first in the sequence to be assigned to a new detection.

The pitch location of the player for the intermediate frames in which he is not detected is calculated by means of an interpolation process. This involves the use of the pitch location of the detection in the preceding frame, as well as that in the subsequent frame where re-detection and re-identification occur. The resulting location is interpolated on an even basis, taking into account the distance between the aforementioned locations and the number of frames during which the player object was not detected.

It should be noted that the re-identification method does, in fact, present a potential issue. An illustrative example of such a pitfall can be observed in Figure 22. In Figure 22a, the relevant players are indicated by red detection bounding boxes. In Figure 22b illustrates the failure of the detection process in the case of the linesman. Figure 22c is a subsequent frame, which demonstrates that the detection of the original player 10 has been unsuccessful. This is likely due to the occlusion caused by player 9. The track ID of player 10 is now available for re-identification. A new detection of the linesman is made, which is in pixel distance closer to the one of player 10 before he became lost than his own last known pixel position. As a result, the re-identification model now assigns the player object with PlayerID 10 to the linesman, as the new detection of the linesman. (In this example, the linesman was falsely classified as a team-mate of player 10)

The detected bounding boxes are transformed into pitch locations. The aforementioned computations are performed by converting the midpoint of the lower side of the bounding box. Given that the players are three-dimensional objects situated on a two-dimensional planar field, this is the point of connection between the two. If, in contrast to the aforementioned methodology, the upper side of the bounding box were to be employed, the homography transformation would direct the pitch location to the pixel of the grass situated immediately above the head of the player, which would be situated at a greater elevation along the pitch than the subject's actual position.

As mentioned in Section 5.3.2, the detection algorithm also detects the coaches standing outside of the field of play but within the view of the camera. To not influence the classification algorithm, the detected objects that have a pitch location of more than 0.5 meters off the field, are removed.

5.4 Ball Detection and Tracking

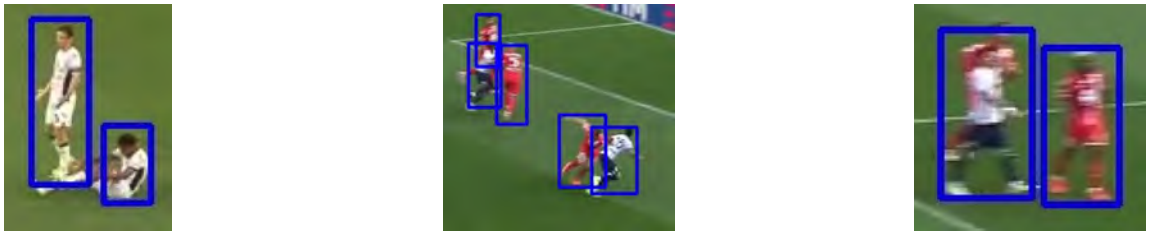
This section outlines the model employed for detection and tracking of the ball, its advantages and limitations, with illustrative examples from the evaluation dataset, and the manner in which the detections and tracking outcomes are processed for the classification model.

5.4.1 Used model. The model used in this project to perform ball detection and tracking is a YoloV8 model based detection algorithm trained by [33]. YOLOv8 or "You Only Look Once" is a prevalent multi-class object detection model created by Ultralytics [76]. The model works by splitting the image into m cells on a matrix and ascertains whether a given cell carries the central coordinates of a classifiable and identifiable object /. The preference of YOLO models over the existing R-CNN and Fast R-CNN models is due to its efficiency and speed at detecting smaller objects [77] which is especially useful as it involves the detection of a football which is of a relatively smaller size in comparison to the rest of the scene.

[33] trained the model to do both detection and classification on 4 different classes: ball, goalkeeper, player and referee, but for this research only the detections of the ball were used. The model was trained on 572 images for 120 epochs, leading to a total of 68,672 images processed. Each image was processed through the network multiple times, but the data augmentation methods in the form of HSV augmentation, translation augmentation, scale augmentation, horizontal flip and mosaic augmentation were applied before it was fed again in the network to prevent overfitting. The scale augmentation appliance makes the model robust to different situations, where some footage of football matches can be quite zoomed in, while other, especially from a monocular camera is quite zoomed out.

5.4.2 Evaluation of ball algorithm. While implementing the ball detection algorithm, a few implementation choices and challenges had to be made and overcome. Primarily, the selection of a suitable confidence value for a detection to be deemed valid was a crucial decision point. Additionally, the mitigation of errors in detection and the navigation of trajectory-related issues proved to be significant challenges.

Confidence score: The initial step is to identify the most appropriate confidence threshold. The confidence threshold represents the point at which a detection is deemed to be valid. The detection algorithm was evaluated using a series of confidence scores, including 0.01, 0.05, 0.1, 0.25, and 0.5, with each score tested on five distinct

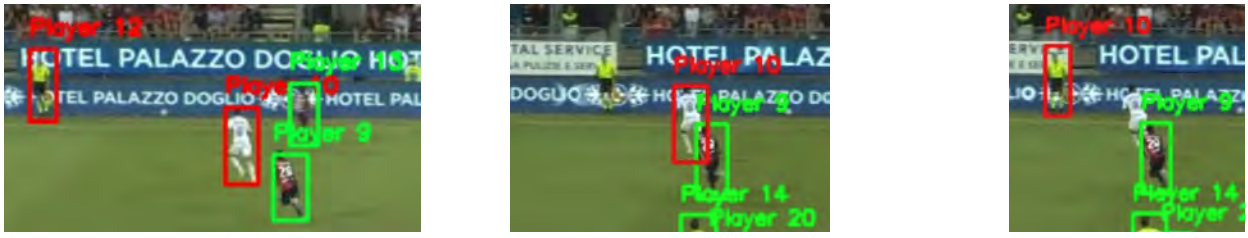


(a) (b) (c)

Figure 19: Three examples of challenging conditions for the player detection algorithm

Figure 20: Detections of coaches/people outside the dimensions of the field

Figure 21: Four subsequent frames, illustrating a failure in re-identification using spatial consistency within the tracking procedure



(a) (b) (c)

Figure 22: An illustrative example of a pitfall of re-identification

video clips. Table 6 presents the results of these different confidence scores. It is necessary to consider the number of correct detections in relation to the number of false detections. A confidence of 0.01 results in an average of 64.2% of frames having a correct detection (see Table 6), but also demonstrates a high incidence of false detections, (see Table7), which could potentially impede the accurate tracking of the ball's trajectory. With such a low confidence score, footwear worn by footballers is frequently misidentified as the ball, as illustrated in Figure 24b

In order to achieve optimal accuracy in detection, it is recommended that a high confidence score of 0.5 or 0.25 os used. As evidenced in Table 7 , this approach results in a minimal number of false detections. This consequently results in a reduction in the amount of noise that must be taken into account when estimating the trajectory of the ball. One disadvantage of such a high level of confidence is that it may result in a lower percentage of frames with a correct detection of the ball. The aforementioned high confidence score can be achieved when the object of interest

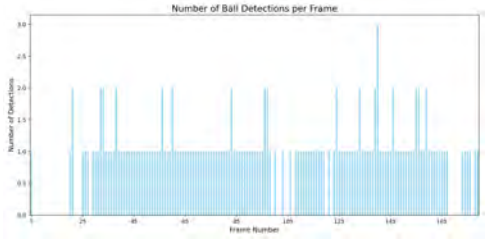


Figure 23: Number of ball detections per frame of event 2135530170

is easily discernible. However, in scenarios characterised by high visual clutter, the reliability of the detection may be compromised.

In order to balance the trade-off between the frequency of ball detection and the number of false positives, it is necessary to choose a confidence score between 0.10 and 0.05. The average rate of correct detections with a confidence score of 0.10 remains relatively high (57.2%), exhibiting only a slight decline in comparison to the confidence score of 0.05 (59.2%). However, in the five clips, the proportion of false detections is significantly lower (59.9%) when a confidence score of 0.10 is applied, as opposed to a confidence score of 0.05. Consequently, in the remainder of this research a confidence score of 0.10 is selected as a threshold for ball detection.

Detection challenges In order to demonstrate the challenges inherent in the detection of the ball, the clip from event 2135530170 (the complete sequence of frames can be accessed via the GitHub repository [78]) is subjected to further analysis. In these frames, the frame number is displayed in the upper right-hand corner of the screen. The colour of the frame number indicates the number of detections that occurred at that frame. Red indicates that multiple detections were made, yellow indicates the presence of one detection, and white indicates no detections.

Furthermore, the number of detections per frame is plotted in Figure 23. A review of the frames of the clip containing the detections and the figure reveals four distinct scenarios pertaining to the detection of the ball.

Initially, it can be observed that in some frames, a detection is made, but it does not correspond to the ball. An illustrative example is provided by Figure 24a Secondly, in some instances, multiple detections are made, of which one is identified as the ball. This is illustrated in Figure 24b Thirdly, as can be seen Figure 23 in some instances, no detection is made at all. Examples are frame 0 to 4. Fourthly, in some instances, a single, accurate detection of the ball is made. An illustrative example is shown in Figure 24c

Figure 25a depicts the pixel coordinates of the disparate ball detections, presented in a 1280 x 720 diagram, with the colour representing the temporal sequence of the frames. It is expected that the representation of the correct ball location will exhibit a linear correlation. In light of the fact that the camera is also in motion, it is possible that the ball may remain at the same pixel location as in the previous frame while moving in the real world. However, it is not possible for the ball to occupy a completely different pixel location in the subsequent frame. Upon closer examination of Figure 25a, a distinct separation becomes evident. Some data points are randomly distributed, while others demonstrate a clear correlation and appear to follow a linear relationship. Figure 25b illustrates the same data points, with the addition of coloured circles in the following cases: In the event of an erroneous detection in a given frame, this is indicated by a red circle. In the event of multiple detections within a single frame, the correct one is indicated by a yellow circle, while the incorrect one is marked with an orange circle. As illustrated in the figure, all points that do not exhibit correlation with the linear function are indicated by orange or red circles, signifying erroneous detections. However, the accurate representation of the ball in frame 5 does not align with the linear correlation, as no detections were made for the ball in preceding frames 0-4 and succeeding frames 6-19.

In order to eliminate the erroneous detections, an outlier detection procedure is employed, as outlined below. The data is sorted by frame number in order to maintain temporal order, and a moving window of size 10 is applied in order to compute rolling averages for the x and y centre positions. Deviations from the aforementioned rolling averages are calculated, and an interquartile range (IQR)-based threshold is employed to identify points that deviate significantly from their anticipated positions, thus marking them as outliers based on spatial movement. The aforementioned detections are indicated by a red colouration in Figure 25c. In addition to the aforementioned deviation-based method, a density-based clustering algorithm, DBSCAN, is employed for the purpose of grouping spatial-temporal data points. Any point not assigned to a cluster is labelled as noise or an outlier. The final outlier detection is a combination of both the deviation-based and DBSCAN-based approaches, capturing anomalies that either deviate significantly from their neighbours or fail to fit into clusters. Figure 25d illustrates the detections following the removal process.

It can be observed that, thus far, only two accurate detections have been erroneously removed. Some incorrect detections remain in the dataset; these are the detections that are in close proximity to the actual ball location and thus were not removed as outlier. Some of these erroneous detections also have a correct detection at the same frame. In order to distinguish between the correct and incorrect detections in those frames, the last known preceding and first known succeeding single ball detection are used to interpolate. The detection in the current frame with the pixel location closest to that interpolated position is selected as the true ball detection.

Now we are left with detections of the ball, from which outliers or double wrongfully detections are removed. We are left with detections that have a high probability of being right. As can be seen in table 6, not all frames have a (correct) detection. To still get information on an approximate ball position for the frames without a detection, interpolation is applied between the two frames with a known location before and after the frame. For frames that only have a previous or next known location, that location is used. The results of this interpolation on the clip can be found at the github page. To give an indication for feature calculation described in 7 that a ball location is interpolated and thus is not totally accurate, an attribute 'interpolated' is added to the ball locations whether a location is interpolated or not.

5.4.3 Trajectory of the ball. A new issue emerges when converting the pixel location of the ball detection into pitch location using homography. Given that homography performs the mapping on a

Clip	Frames	Correct Detections				
		0.5	0.25	0.10	0.05	0.01
1	180	102 **(56.7%)**	121 **(67.7%)**	140 **(77.7%)**	143 **(79.4%)**	154 **(85.6%)**
2	115	87 **(75.6%)**	103 **(89.6%)**	107 **(93.0%)**	107 **(93.0%)**	110 **(95.7%)**
3	88	22 **(25.0%)**	26 **(29.5%)**	33 **(37.5%)**	37 **(42.0%)**	42 **(47.7%)**
4	87	8 **(9.2%)**	16 **(18.4%)**	27 **(31.0%)**	31 **(35.6%)**	39 **(44.8%)**
5	67	0 **(0%)**	0 **(0%)**	0 **(0%)**	0 **(0%)**	0 **(0%)**

Table 6: Correct Detections for Different Clips at Various Confidence Levels (with percentages)

(a) Frame 20: Wrong detection of the ball **(b) Frame 21: Multiple detections, one correct** **(c) Frame 26: Correct detection of the ball**

Figure 24: Three different frames illustrating different kind of detections

Clip	Frames	False Detections				
		0.5	0.25	0.10	0.05	0.01
1	180	2	7	18	34	80
2	115	1	11	24	38	142
3	88	9	26	57	100	198
4	87	7	18	32	49	104
5	67	1	5	15	23	78

Table 7: False Detections for Different Clips at Various Confidence Levels

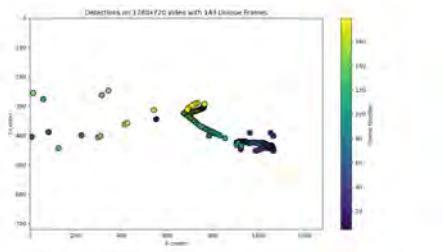
two-dimensional field, this approach is only applicable when the detection occurs within that same two-dimensional field. As player objects are bound by gravity and only move from the 2D plane for a limited time and height when they jump, the homography transformation is accurate for players. However, as the ball is less bound by gravity and can reach greater heights when shot into the air, the homography mapping is not accurate on the top-down position of the ball.

An illustration of this phenomenon can be found in Figure 26 , where in Figure 26a a frame is shown in which the ball is positioned in the air and off the 2D planar field. Figure 26b illustrates two distinct locations where the ball may be situated. The red location is the position recorded by the homography mapping as the pitch location, while the blue location shows the actual top-down position of the ball. Incorrect homography mapping is unavoidable when the ball is in the air. However, as described in Section 7.1, the addition of a feature provides supplementary information to the classification model, which can help tot distinguish between a ball being in the air and or being on the ground.

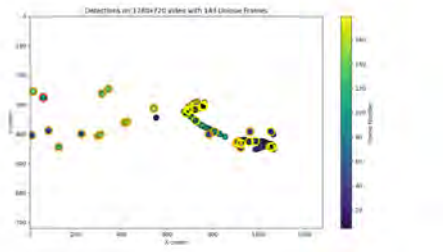
5.5 Team Classification

Team membership classification (i.e. labeling each person on a playing surface as a member of team A, team B, goalkeeper of team A, goalkeeper of team B or a referee) is an important task in sports video analytics: The majority of inferences and statistics are reliant on this information, as it determines which players are on each team and, subsequently, which teams are involved in a particular situation or event. In the context of our classification network, the team assignment of players is a crucial factor in almost every aspect of classification. A r example of this is in the classification of an event such as a duel. The presence of two players from different teams in close proximity indicates the potential for a duel, whereas when they are on the same team, it can't possibly be a duel.

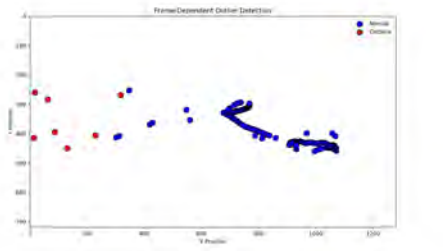
A number of team classification algorithms have been developed with the objective of completing this task. For example, [32] constructed a CNN to learn a descriptor that is similar for pixels depicting players from the same team and dissimilar when pixels correspond to different teams. The advantage of this approach is that it does not require per-game learning, thereby enabling efficient team discrimination for multiple games. The model was constructed for the purpose of team classification in basketball; however, the limitations of this approach become evident when it is applied to football. In basketball, there are two teams and a set of referees. All members of the same team are attired in identical jerseys, as are the referees. Given the smaller dimensions of the basketball court, there are typically more basketball players of either team than there are referees in each frame. Accordingly, the classification algorithm could effortlessly categorize the identified entities as belonging to Team A, Team B, or the Referee category. In contrast, in football, not all members of the same team wear the same jersey. The goalkeeper, for instance, wears a different



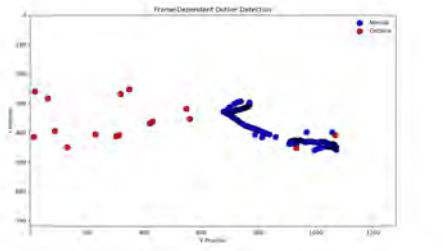
(a) Pixel locations ball detections



(b) Pixel locations with additional true/false markers



(c) Outliers detected by moving average



(d) Outliers detected with addition of DBSCAN

Figure 25: Depiction of the different stages in the ball detection and outlier removal process

(a)

(b)

Figure 26: Visualization of the challenge that arises with mapping the detection of the ball to the correct pitch coordinate when the ball is positioned in the air

jersey to the players. The goalkeeper is the sole individual wearing the aforementioned jersey, and thus, he is not part of a majority from which the classifier can distinct between referee or players. If the team classification algorithm of [32] were to be applied and increased to five classes (i.e., 'team A', 'team B', 'goalkeeper team A', 'goalkeeper team B', and 'referee'), it would be unable to determine whether the goalkeeper should be classified as 'referee', 'goalkeeper team 1', or 'goalkeeper team 2'. Furthermore, the distinct jersey worn by the goalkeeper from that of the rest of the team would also present an ambiguity for the team classifier, in terms of determining whether the goalkeeper should be classified as belonging to Team A or Team B. The typical process for analyzing a single game involves the detection of players in several frames, their classification into a team, and the application of k-means clustering to assign each detected object in other frames to a team. However, given that this project encompasses footage from approximately 300 distinct games, devising a unique classification model for each match would necessitate a vast amount of data collection, training, and model storage. To address this challenge, a customized team classification model tailored to this data set has been developed.

5.5.1 Classification in this project. In this project, the team classification algorithm is designed to accurately identify and differentiate between teams, goalkeepers and referee in each event video clip. The classification algorithms categorizes all detections into one of five distinct classes: 'Team A', 'Team B', 'Goalkeeper Team A', 'Goalkeeper Team B' or 'Referee'.

Instead of training individual models for each game, a single model is trained on the various kits worn across all games. For each clip, the model is then given five classification options it can choose from. According to Serie A regulations (and standard football rules), all team kits must be easily distinguishable from one another. While there is a slight possibility that a goalkeeper might wear a kit similar to the referee or another goalkeeper, this scenario

is uncommon. Therefore, the classification model should reliably distinguish between these classes.

This section provides a detailed explanation of the custom team classification algorithm. It will first describe the dataset used for training and validation, then delve into the model architecture, and finally, present the model's performance, including comparisons with simpler classification models.

5.5.2 Dataset. As previously mentioned, the choice of which kit a team wears is based on whether the team is playing at home and the colors that make up the home team's kit. Every team in Serie A designs its kits before the season starts, typically offering three different sets (Home, Away, and Third kits). Occasionally, teams design a fourth or even fifth kit, often for special occasions or celebrations. Goalkeepers generally have between 2 to 5 different kits, while referees have four. The player detection algorithm, described in Section 5.3, is used to collect bounding boxes of players wearing each kit. This results in a total of 132 different classes, with 50 images used for each class during training, yielding a dataset of 6,600 images.

In addition to building the dataset, it was necessary to gather information on which kits were worn in each match to ensure the classification algorithm functioned correctly. This specific information is not readily available online, so it had to be manually collected. This involved reviewing footage from approximately 300 games and annotating which kits were worn by each of the five classes (Team A, Team B, Goalkeeper Team A, Goalkeeper Team B, and Referee) for every match.

To further enhance the dataset, images were carefully selected to include various lighting conditions, camera angles, and player poses. This additional variety ensures that the model generalizes well across different scenarios, improving its robustness and accuracy of the model for the different lighting conditions the games are played in. Next to this, before feeding the data into the model, the following pre-processing steps were applied to even more diversify the training set:

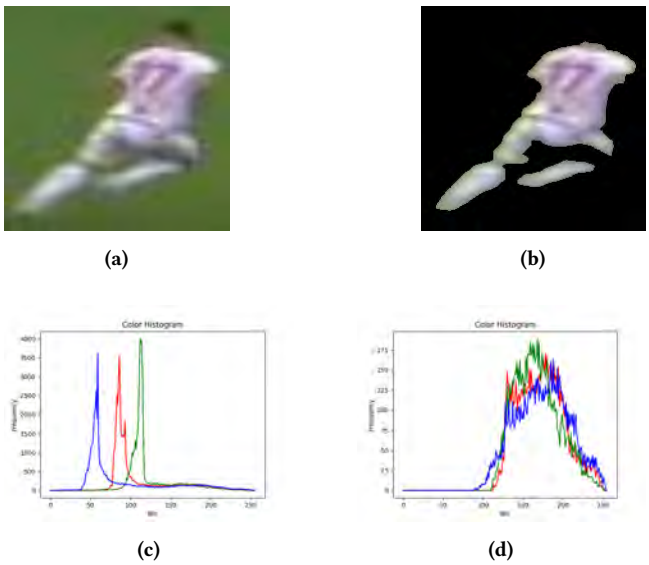
- Contrast: from the Keras library, the Random Contrast layer with lower bound 0.95 and upper bound 1.05 was applied.
- Rotation: from the Keras library, the Random Rotation layer with factor 0.05 was applied. Resulting in images being randomly rotated between -18 degrees and +18 degrees.
- Brightness: from the Keras library, the Random Brightness function was applied, with parameter 0.2, adjusting the brightness of the image randomly with a maximum relative change in brightness

5.5.3 Model architecture. The model is build of two different branches: a branch using the VGG16-model, and a branch using color histograms.

Color Histogram Branch: The first branch of the classification model relies on color histograms. A color histogram quantitatively represents the distribution of color intensities in an image, making it particularly useful for textured images that may not be easily segmented using traditional techniques. Color histograms are invariant to translation and rotation around the view axis and exhibit only gradual changes under variations in viewing angle, scale, and occlusion [68]. In this model, colors are represented in RGB space. A 256-bin histogram is generated, where each bin represents the number of pixels for each of the 256 intensity levels across the three RGB channels. An example histogram is shown in Figure ..., where the distribution of RGB colors is represented by separate lines for each channel.

The use of square bounding boxes for player detection results in a significant number of pixels within each detection representing the surrounding grass, thereby rendering green the dominant colour. Inclusion of this green within the colour histogram can result in distortion of the data, thereby reducing its representativeness of the actual kit colours. To address this issue, a green filter is applied to each detection image prior to the calculation of the colour histogram. The green filter is applied to every pixel within the RGB range (0, 70, 0) and (100, 255, 100). Figures 27 and 28 illustrate the application of green filtering to kits of different colours. As shown in 28, where a green filter is applied to a detection image of a player wearing a green kit, the green parts of the kit itself are also filtered. This filtering tends to make the histograms of the green kits look similar, which could be problematic if several green kits were among the classification options. However, as the five kits in each game are always different, this problem is unlikely to occur.

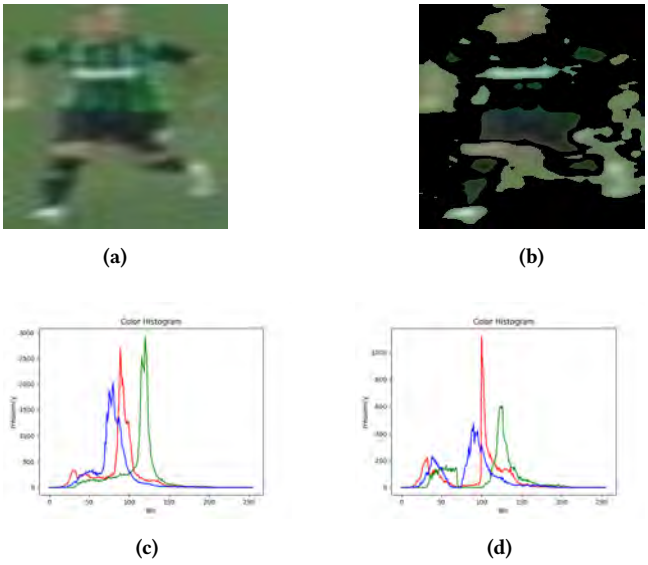
The histogram values of the images are fed into the network as a !shape! Tensor, which is fed through a series of fully connected (Dense) layers with ReLu activations, as well as a batch normalization and dropout layer (0.5 dropout).



(a) **(b)** **(c)** **(d)**

Figure 27: Green filter appliance on green jersey. 27a and 27b show the detection bounding box images before and after green filter appliance respectively, while 27c and 27d show the corresponding color histograms.

VGG16 branch: The other branch of the classification model is based upon a VGG16 architecture. VGG16 is a convolutional neural network architecture introduced by [60]. This architecture has proven to be highly effective, achieving a test accuracy of 92.77 percent on ImageNet, a dataset containing 14 million images across 10,000 classes. The architecture of VGG16, as shown in Figure 29,



(a) (b) (c) (d)

Figure 28: Green filter appliance on green jersey. 28a and 28b show the detection bounding box images before and after green filter appliance respectively, while 28c and 28d show the corresponding color histograms.

starts with input images of size 224 x 224 x 3, which pass through two convolutional layers followed by a max-pooling layer. This sequence is repeated twice more, with the number of convolutional layers increasing in each block, culminating in three fully connected layers and ReLU activation functions. The convolutional layers use 3x3 filters with a stride of 1, while the max-pooling layers have 2x2 filters with a stride of 2. The use of repetitive small convolutional layers (instead of large ones) allows the model to capture more detailed patterns. The VGG16 model used in this classification model is pretrained on the ImageNet dataset, which significantly reduces the need for extensive training data and computational resources. Where the color histogram branch is mainly used to distinguish the colors of kits from each other, the VGG16 branch is used to extract some deeper features and be able to distinguish shirts on for example design. The fully connected layers at the top of the VGG16 are excluded in this classification model to instead focus on extracting features from convolutional layers. The last 8 layers are set to be trainable. This allows the model to adapt to the new data while leveraging the powerful pre-trained features from earlier layers.

Combining of branches and loss function: The feature vectors from both the image and histogram branches are concatenated and passed through a Dense layer with 512 units and ReLU activation. After this, a Dropout layer with a rate of 0.2 is applied to reduce overfitting. The final layer is a Dense layer with 132 units and a softmax activation function, which outputs the class probabilities for classification.

Given that the model is designed to classify between 5 distinct jersey kits that are not visually similar in terms of color, it is crucial for the model to accurately distinguish colors. For instance, if the input is a white jersey, the model should predict a white jersey.

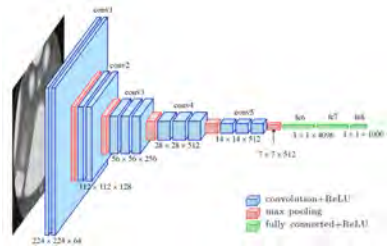


Figure 29: Architecture of VGG16

It is not necessary for the prediction to be the exact white jersey, but the probabilities of all white jersey classes should be relatively high compared to non-white jerseys. This ensures that the model effectively differentiates between jerseys based on color.

Using a standard categorical cross-entropy loss function, where every misclassification is penalized equally, would prevent the model from learning that, for example, when the true label is a white jersey, that predicting a different white jersey in training is better than predicting a red jersey. This would result in the model not understanding the importance of color. Therefore, a custom loss function is applied.

The custom loss function assigns different penalties based on the similarity between classes. This is achieved by creating a penalty matrix P, where each entry (i, j) represents the penalty for misclassifying class i as class j. Similar jerseys (e.g., different white jerseys) incur a lower penalty (factor 0.3), compared to dissimilar jerseys (e.g., a white jersey misclassified as a black jersey), which incur a higher penalty (factor 1.0).

The similarity between classes is determined by the main color of their jerseys. All jerseys are grouped into 13 different categories based on their main color, and for the entries (i, j) where i and j belong to the same category, the lower penalty is applied.

The penalty matrix $P(i, j)$ is defined as:

$$P(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are in different subgroups} \\ 0.3 & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are in the same subgroup} \end{cases}$$

- When $i = j$: $P(i, j) = 0$, meaning there is no penalty if the prediction is correct.
- When $i \neq j$ and i and j are in different subgroups: $P(i, j) = 1$, imposing a penalty of 1 for misclassifications across different subgroups.
- When $i \neq j$ and i and j are in the same subgroup: $P(i, j) = 0.3$, imposing a smaller penalty of 0.3 for misclassifications within the same subgroup.

This custom loss function helps the model understand the importance of color in training.

5.5.4 Model Training and Appliance. The model was trained for 15 epochs, and achieved a validation accuracy of 57.80 percent. This validation accuracy is for the prediction over all classes, thus the model predicts out of all classes including classes with similar jersey. As mentioned before, in this project, the team classification model only needs to predict from 5 quite distinctive classes. Figure

30 shows images of four player objects in the match Napoli - Monza In this match, Napoli was playing in their first kit (blue), Monza in their second kit (black), the keeper of Monza in his red kit, and the referee in the orange kit. The team classification algorithm made the following predictions on the bounding boxes with the following options [napoli 1, monza 3, napoli goalkeeper 3, monza goalkeeper 3, referee 3] (referring to the kit the teams were wearing during the match) For figure 30a the algorithm predicted: [5.1530376e-05 4.2695913e-01 1.8463358e-04 1.2978206e-04 7.3700268e-03]. A 42.69 percent prediction over all classes that it is Monza's third kit. The argmax function over the possible kits predicts Monza 3. For figure 30b the algorithm predicted: [1.1929249e-07 3.2544140e-05 2.2554076e-03 2.3877260e-03 9.9542014e-02]. The argmax function over the possible kits predicts referee 3, with a prediction of 9.95 percent prediction over all classes. For figure 30c the algorithm predicted: [2.6970711e-01 1.7771117e-09 3.1345323e-07 2.9126181e-08 3.6916969e-08]. Argmax concludes Napoli First kit. For figure 30d the algorithm predicted: 1.5567072e-06 6.7154865e-06 4.3437511e-05 2.6892889e-01 9.9851545e-03]. Argmax concludes Monza Keeper third kit.

(a) PL:Monza3. TL:Monza3 **(b) PL:Referee3. TL:Referee3** **(c) PL:Napoli1. TL:Napoli1** **(d) PL:MonzaGK3. TL:MonzaGK3**

Figure 30: Cropped bounding boxes from the match Napoli-Monza with their predicted label (PL) and true label (TL)

6 HUMAN POSE ESTIMATION

6.1 Research Objectives and Hypotheses

As stated in Section 2.2 the research question of this study is "How will the utilization of distinct 2D human pose estimation algorithms, encompassing top-down and bottom-up methodologies, coupled with models for field registration, player and ball detection and tracking, and team classification integrated in a machine learning system, impact the capacity to classify actions in football using footage from a monocular, action-tracking camera?". The two human pose estimation models that are to be compared are the bottom-up method, known as 'PifPaf', and the top-down method, known as 'HRNet'. In order to evaluate the two different HPE models, the following criteria will be considered: Firstly, the accuracy of the models is evaluated, which refers to their ability to correctly detect and localise the keypoints of the human body. Secondly, the robustness of the models will be evaluated, examining their performance

Figure 31: 17 body keypoints of COCO dataset

across diverse conditions, including variation in lighting, occlusion, different poses, and varying scales of the human body. Thirdly, the inference time, defined as the time taken for the model to process a clip, will be considered. Due to the lack of annotated data on human body estimates in our dataset and the low resolution of players in the background, which makes manual annotation of human keypoints impractical, the accuracy and robustness of the models will be evaluated qualitatively. In this section, both models will first be described in Sections 6.2 and 6.3 followed by an evaluation of their performance in Section 6.5.

6.2 Bottom-Up Pose Estimation Model

As described in Section 3.4, in the bottom-up approach, initially, all the key points of the targets are detected, and later in the optimization stage, the detected keypoints are associated with the corresponding targets. The bottom-up Pose Estimation Model used in this research is the method proposed by Kreiss et al. (2019) in their paper: "PifPaf: Composite Fields for Human Pose Estimation" [34]. Their method, PifPaf, uses a Part Intensity Field (PIF) to localize body parts and a Part Association Field (PAF) to associate body parts with each other to form full human poses. In their paper, they state that their method outperforms previous methods at low resolution and in crowded, cluttered and occluded scenes, by using (i) their new composite field PAF encoding fine-graied information and (ii) the choice of Laplace loss for regressionws which incorporates a notion of uncertainty. Their model performed on the same level with existing state-of-the-art bottom-up methods on the standard COCO keypoint task and produced state-of-the-art results on a modified COCO keypoint task on the transportation domain. This is interesting for this research, since the transportation domain includes often crowded images, in which pedestrians are occluded by other pedestrians, where bounding boxes clash and top-down methods particularly struggle. These occlusions of pedestrians can be compared to football players in duels or crowded areas excluding each other as well. A more detailed description of the PifPaf model can be found in A.3

The PifPaf model has been developed with the objective of predicting 17 body keypoints as defined by the COCO dataset. Figure 31 provides an illustration of these 17 keypoints. The OpenPifPaf documentation presents a comprehensive list of tunable parameters, including the seed and instance thresholds and the Cif-th and Caf-th parameters. The seed threshold establishes the minimum threshold for selecting candidate points, or "seeds," which serve as the initial point of body keypoint detection. The seeds act like anchors for detecting entire poses. A higher seed threshold ensures that only the most robust candidates are selected, thereby reducing the number of false positives while potentially overlooking weaker body keypoint predictions. The instance threshold serves to filter

entire detected poses based on the total confidence score of detected keypoints. Lower values permit a greater number of person detections, but this increases the probability of false positives.

The Cif-th parameter defines the threshold for Composite Intensity Fields (Cif), determining the strength of individual keypoint detections. Higher thresholds prioritise strong detections, but may result in the omission of weaker signals. The Caf-th parameter controls the threshold for Composite Association Fields (Caf), which link keypoints together to form complete poses. A higher Caf-th ensures more reliable connections, but may result in the exclusion of keypoint pairs. Conversely, a lower threshold allows for more connections, but may introduce inaccuracies.

A variety of parameter combinations are evaluated to determine their impact on the HPE process. Figure 32 depicts the pairing of the seed with instance thresholds and the Cif-th with Caf-th parameters. The figure presents the HPE results for the four distinct combinations of these pairs, for values of 0.25 and 0.05.

As can be observed in Figures 32a and 32b, a reduction in the values of the confidence thresholds for Cif-th and Caf-th results in the incorporation of additional keypoints into existing detections. An illustrative example can be observed in the incorporation of an additional head keypoint for the player with kit number 9 in Figure 32b.

A reduction in the seed and instance thresholds (see Figures (32b and 32d) indicates an increase in the number of body pose detections. However, the number of keypoints detected in previously identified body poses remains unchanged. To gain further insight into the individual contributions of the seed and instance thresholds, Figure 33 illustrates the same frame with varying values for each of these parameters. As can be observed in the comparison of Figures 33a and 32b, a reduction in the seed threshold does not result in an increase in the number of pose detections. This is because, when the instance threshold is maintained at a higher level, the total confidence score of the detected keypoints for the body pose does not rise sufficiently.

Figure 33b illustrates that the instance threshold exerts the greatest influence on the number of distinct poses that can be identified. However, a comparison with Figure 32d reveals that a reduction in the seed threshold, when the instance threshold is maintained at a low level, leads to an increase in the number of keypoint detections per pose. This is exemplified by the linesman in the top right corner, which exhibits a greater number of predicted keypoints for a lower seed threshold.

Based on the aforementioned comparisons and reviewing different threshold values than illustrated in this paper, the decision was taken to select the values of (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.05, 0.05, 0.05, 0.05). The results of these can be found in 32d. As illustrated in the figure, there are still instances where not all keypoints are detected at the specified confidence levels for some of the detected poses. This is why the PifPaf model incorporates an additional parameter, named 'force complete pose', which prompts to generate a complete pose even when certain keypoints have not been identified with sufficient confidence. In essence, the objective is to prompt the model to complete any missing or incomplete elements of a detected pose, utilising the available keypoint data. This can assist in situations where certain body parts are obscured or beyond the camera's field of view. It should be noted that the force-complete-pose does not replace the Cif and Caf thresholds; rather, it functions in conjunction with them, filling in incomplete poses subsequent to the application of the thresholds. It is important to note that the Cif/Caf thresholds continue to play a pivotal role in determining the quality and confidence of the initial detection. Figure 34 the application of the 'force complete pose' parameter. As can be observed, all previously detected poses have now been completed with predictions for every keypoint. However, there is a discrepancy in the top left of the figure, where a non-player object outside the field was detected. This issue can be resolved by applying the same procedure as for player detections outside the boundaries of the field, as described in Section 5.3.3.

6.3 Top-Down Pose Estimation Model

As described in Section 3.4, top-down human pose estimation is a single-person pipeline and relies on a detection algorithm for the bounding box of an object to estimate in. There are regression and heatmap-based methods. The top-down method chosen in this research is a heatmap-based model and will be described in further detail now.

The method used for top-down human pose estimation in this research is published on [61] and is an implementation of the algorithm presented in the paper 'Deep High-Resolution Representation Learning for Human Pose Estimation' [66]. Their top-down method, called HRNet, differs from existing classification networks that are build on Deep convolutional neural networks (DCNNs). Existing methods gradually reduce the spatial size of feature maps, by connecting convolutions from high resolutions to low resolution in series, and lead to a low-resolution representation, which is further processed for classification. For position-sensitive tasks as human pose estimation, high-resolution representations are needed, and thus adopt these methods a high-resolution representation recovering subnetwork, which is formed by connecting low-to-high convolutions in series. A representation of such a network can be found in Figure 35. HRNet differs from this in the fact that this architecture is able to maintain high-resolution representations through the whole process. The process starts with a high resolution convolution stream, and gradually high-to-low resolution convolution streams are added one by one, and connect the multi-resolution streams in parallel. The resulting network consists of 4 stages, where the nth stage contains n streams corresponding to n resolutions. Repeated multi-resolution fusions are conducted by exchanging the information across the parallel streams over and over, which boosts the high-resolution representations with the help of the low-resolution representations, and vice versa. A representation of their high-resolution network can be found in Figure 36

HRNet uses a heatmap-based framework, as described in 3.4, which estimates heatmaps of size $\frac{W}{4} \times \frac{H}{4}$, $\{H_1, H_2, \ldots, H_K\}$. Each heatmap represents the probability distribution of the corresponding body joint, with the peak of the heatmap indicating the most likely location of the joint. The key advantage of this approach lies in the ability to maintain high-resolution feature maps throughout the network. This leads to precise spatial predictions, which is essential for accurate human pose estimation, especially in complex poses or crowded environments. The implemented top-down

(a) (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.25, 0.25, 0.25, 0.25)

(b) (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.25, 0.25, 0.05, 0.05)

(c) (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.05, 0.05, 0.25, 0.25)

(d) (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.05, 0.05, 0.05, 0.05)

Figure 32: PifPaf HPE detection on the same frame for 4 different combinations of pairs of parameters

(a) (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.05, 0.25, 0.05, 0.05)

(b) (Seed threshold, Instance threshold, Cif-th, Caf-th) = (0.25, 0.05, 0.05, 0.05)

Figure 33: PifPaf HPE detection on the same frame for different values for seed and instance threshold

Figure 34: HPE detections with parameter 'force complete pose'

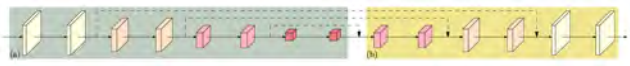


Figure 35: Representation Low to High

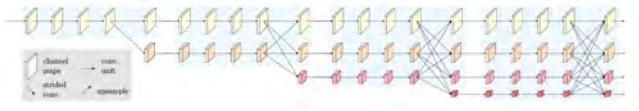


Figure 36: Representation HRNET

method was tested for various thresholds for the variables confidence thresholds and IOU-thresholds, but no changes were found in the detected keypoints when varying these.

6.4 Advantages and Limitations of both approaches

The existing research literature tends to favor bottom-up methods over top-down methods. Top-down methods are considered unreliable in situations where multiple individuals occlude each other, as the detection models, which are a key component of top-down methods, often have difficulty making accurate detections in such circumstances. Furthermore, when target individuals are in close proximity to one another, the top-down pose estimator may be misled by the presence of nearby individuals, resulting in the prediction of joints belonging to a nearby non-target individual. Bottom-up methods do not utilize any human detection, and thus are capable of producing results with higher accuracy when multiple individuals interact with each other. However, bottom-up methods are susceptible to scale variations. If the keypoints detected are at disparate scales, the model may encounter difficulties in grouping the keypoints correctly. Given that the discrepancy in scale from the broadcast view of football games is relatively minor, it is hypothesised that this does not exert a significant influence in this research.

6.5 Evaluation of the Human Pose Estimation Models

The two implemented models described above have been trained, as most HPE models on the COCO dataset. To compare the methods for their performance in the field of football, their performance on some clips from this research's data are evaluated.

The bottom-up method has the potential to accept whole images as input directly to perform HPE on. In contrast, the top-down method requires the detection bounding boxes output by the player detection algorithm, as described in 5.3, as input in order to estimate the human pose. This dependency on the detection model is both an advantage as a disadvantage. The bottom-up method is constrained by the performance of the detection model, and will not detect human body keypoints from the image that are not contained in these detections. Conversely, it draws upon the expertise of a robust algorithm, trained for the specific purpose of detecting football players, whereas the bottom-up method is trained on recognizing body parts from humans, not per se from football players. This enables the top-down method to retrieve information from a source that is well-suited to identifying keypoints. The advantage side of recipocitry can be found in the comparison of the Figures 37, 37. Those figures illustrate the HPE of player objects within the same frame. As can be observed, the bottom-up method does not identify keypoints on players situated on the opposing side of the field. Conversely, the top-down method obtained the bounding boxes from the player detection, thereby indicating the necessity for detection on the human key points in that part of the image, and successfully identified such points.

With regard to the precision of the predicted human body keypoints, a comparison is presented in Figure 39. Figure 39a depicts a section of a frame comprising a considerable number of players. Figures 39b and 39c illustrate the HPE detections in successive instances. It is evident that both models were unable to detect two partially hidden players. An examination of the precision with which both models predict the keypoints of the remaining players, reveals that the top-down model is considerably less accurate in its predictions and also fails to consistently predict entire body poses. The bottom-up method is accurate in all predictions, with the exception of a correct prediction of the knee and corresponding foot of the player in the middle. Based on the results of the clips in which both HPE models are evaluated, it can be concluded that the bottom-up method predicts keypoints with a bigger precision than the top-down method.

Figure 37: HPE performed by bottom-up method PifPaf

Figure 38: HPE performed by top-down method HRNet

In addition to a comparison of the models' accuracy in detection, a comparison of their processing times is also required. As previously stated in 3.4, for the top-down pipeline, the number of people

(a) (b) (c)

Figure 39: Example part of frame in 39a with the HPE detections of the top-down and bottom-up methods in 39b and 39c respectively

in the input image directly affects the computation time. This phenomenon does not occur with the bottom-up method. Consequently, the computational speed of bottom-up methods is typically superior to that of top-down methods. Table 8 presents the processing times for six different video clips, with data for both HPE models. It is noteworthy that the processing times of the top-down models are, in fact, lower for each clip than those of the bottom-up models, which is not in accordance with existing literature. It should be noted that the aforementioned processing times pertain solely to the estimation of human poses and do not include the detection of player objects. As player detection and tracking constituted a discrete branch of the pipeline, it can be stated that the top-down method exhibited superior processing speeds to the bottom-up method in this research.Furthermore, the processing times for the player detection and tracking model are included in Table 8. When considered outside the pipeline of this research, these should be incorporated into the top-down processing time to obtain the total time required for top-down HPE. It should be noted that, for top-down, the tracking component of the detection and tracking process would not be necessary, which would result in a slight reduction in the overall processing time.

7 CLASSIFICATION MODEL DEVELOPMENT

This section describes the process of the classification of the sliced clips. In Section 7.1 the process of turning the data obtained from the different implemented models into useful features is described. In Section 7.2 the outline of the hierarchical classification algorithm is explained, followed up by Section 7.3 in which its architecture is described. In Section 7.4 the training process is described.

7.1 Feature Integration

Once the pre-defined models have been applied to each data frame, the resulting information must be transformed in order to generate additional insights that are useful for the classification process.

Following the application of the field localization, player detection and tracking, ball detection and tracking, team classification and the two HPE models, the following information is available for each frame: In the case of the player/referee objects, the following information is available:

- Object location: The coordinates of the bounding box of each object are provided in the format (xmin, ymin, xmax, ymax), as well as the pitch coordinates transformed by the homography matrix. These coordinates refer to the location of the pitch, with (0,0) denoting the top left corner from the camera viewpoint.
- Object Human Pose Estimation: The human pose estimation is available for all objects. The various human pose keypoints identified are presented in a format comprising the x and y coordinates, along with the associated confidence value.
- Object team classification: The team classification of each object is also available and can be one of the following: ’"Team 1", "Team 2", "Goalkeeper Team 1", "Goalkeeper Team 2", "Referee

The following category is available for the ball object:

- Object location: As described in 5.4, a ball detection is available for every detection. It contains a pixel location bounding box in format (xmin, ymin, xmax, ymax), a pitch location in format (x coordinate, y coordinate) and an attribute called ’interpolation’ that is 0 if the ball was detected in that frame, and 1 if the ball was either inter- or extrapolated for that frame.

As previously stated, additional features were devised and calculated from this available data, which were then used as input into the classification model. These additional features can be classified into four distinct categories: player-related features, ball-related features, player-to-player related features, and player-to-ball related features.

The term "player-related features" is used to describe those features that are concerned with the object of a single player. The following features were identified and subsequently derived:

- The velocity features, comprising the distance travelled, mean velocity, and acceleration, were calculated for the different intervals of 1, 3, 5, 10, and 20 frames. These calculations were based on the pitch positions of the player.
- Direction features: These pertain to the angle of the player in relation to their position at the preceding 1, 3, 5 and 10 frames.
- Distance features: The distance, in pitch coordinate system units, of the player to specific points of interest on the field is calculated. The following keypoints were identified as being of particular significance: [2, 3, 26, 27, 8, 9, 16, 17, 42, 12, 13, 28, 29]. These are key points that represent points of significant importance. To illustrate, keypoints 26 and 27 represent the goalposts of the right goal, while keypoint 42

Event	Number of Frames	Bottom-Up	Top-Down	Player Detection
1742405812	310	27,202	21,198	24,48
2120719407	253	20,205	17,759	14,32
2135530621	56	8,486	3,751	14,25
2099524950	243	17,243	15,737	44,24
2106036784	100	10,310	8,221	24,34
2120719072	88	10,234	5,948	4,67

Table 8: Processing times of HPE methods for different events

represents the middle of the field. As an example to why these features are important: if a player in possession of the ball is in close proximity to keypoint 42 but distant from 26 and 27, the probability of the clip being a shot is minimal. The remaining keypoints represent other significant locations, such as corners or the boundaries of the penalty box.

Ball-related features pertain to the ball object and bear resemblance to player-related features. The following features were derived:

- Distance features: The distance of the ball to specific points on the field is calculated in pitch coordinates. These are the same key points that were employed for the players' distance features.
- Velocity features: Similarly to the player-related features, the distance traversed, velocity and acceleration of the various intervals of 1, 3, 5, 10 and 20 frames were calculated.
- Direction features: The angle of the ball from its preceding pitch positions, at 1, 3, 5 and 10 frames, is calculated. This feature is of greater importance for the ball than for players, as highlighted in Section 5.4, since the pitch coordinates of the ball do not represent its actual location when it is in the air. The direction features also allow the calculation of a further feature, namely the consistency of a ball's trajectory. The trajectory consistency categorizes the trajectory of the ball's pitch coordinates over an interval of 2, 6, 10, and 20 frames into one of three categories: 'straight', 'semi straight' or 'not straight'. This is achieved by comparing the direction of the first half of the interval to the second half of the interval. If the angle of the direction change is less than 5, it is labeled 'straight', between 5 and 15, 'semi straight', and greater than 15, 'not so straight'. This feature allows for a more accurate determination of whether a ball is in the air or on the ground. In the former case, the pitch coordinates may not be accurate, whereas in the latter, they are likely to be correct, given that passing or shooting often occurs in a relatively straight line.
- Area checks: These features state if a ball is in one or multiple of the following areas: middle third, left or right penalty area, left or right deep completion area. These location of these areas can be found in [83]

The term "player-ball related features" is used to describe those features that pertain to the relationship between the ball and a player object. These features are derived for each existing player object. The following features can be classified under this category:

- Object distance features: This is the distance from the player object to the ball, both in pixel distance as in pitch distance. For the purpose of calculating pixel distance, the distance to the lower middle pixel of the player object is used.
- HPE keypoints distance features: The HPE keypoints distance features concern the distance from every detected HPE keypoint of the player object to the ball, expressed in pixels. Only the pixel distance is derived, as the pitch coordinates of the HPE keypoints would not represent the actual value accurately, given the 3D nature of the player on a 2D planar field on which the homography estimation is conducted.

The aforementioned features are collectively represented by a single feature vector for each frame. This feature vector comprises a vector for each detected object within that frame, which encompasses 130 features. Features that are not pertinent to the object in question, such as ball-related features for a player object and vice versa, are masked.

7.2 Hierarchical Classification

The goal of the classification model is to classify clips into different classes following the classification hierarchy depicted in 3. The main and most important classification is done in the classification of the primary types 'Duel', 'Pass', 'Shot', 'Interception' and 'Touch', but it is interesting to see if the model can further classify these clips into more detailed descriptions.

To do this, the model for the main classification is trained on every instance of the dataset. The other classification models, which are on more detailed attributes, are only trained on the clips concerning that action. In Table 9, the lengths of the training, validation and test sets of the different classifications are shown.

7.3 Model Architecture

The model is a Long Short Term Memory (LSTM)-based classification model with attention and dropout mechanisms. The model starts with n (hyperparameter) LSTM layers, followed by an attention mechanism, which is used to make the model learn which frames to focus on. This is useful because the input of the clip is not only the action itself, but also the preceding and following actions. Using this attention mechanism, the model learns which frames to focus on during training, which hopefully leads to it recognising that it should not focus on the beginning and end of the clip, but rather the middle. The attention layer is followed by a dropout layer. The value of this dropout rate is a hyperparameter and is decided during hyperparameter tuning (see Section 7.4). The dropout layer is used to prevent overfitting, as the training data is quite

unbalanced. The dropout layer is followed by 2 fully connected layers that reduce the dimensionality of the data. The first applies a ReLU activation and dropout. The second outputs the logits for classification into the number of classes.

The data instances are fed into the model in batches. Since the model can only handle one specific input size per batch, all instances get the same number of objects and the same number of frames as the maximum of those found in that particular batch. These extra objects and frames are masked so as not to influence the training process.

7.4 Training process

Three distinct models have been developed. The initial model may be employed as a reference point for the comparison of the two distinct HPE methodologies, as it is founded upon characteristics that do not encompass any HPE detection. It should be noted that the aforementioned features do not encompass any information pertaining to the locations and distances of human pose keypoints. The model employs 79 distinct features for each player-object pairing.

The remaining two models are trained on the complete feature set and thus include the aforementioned HPE-based features. The feature sets of both models comprise 130 features per player object.

The base model is subjected to hyperparameter optimisation with respect to the primary classification. The optimal hyperparameters identified are subsequently employed in the training process for the other two models, thereby establishing an equivalent framework for comparison between the two. The aforementioned hyperparameters are employed in both the classification of the primary attributes and that of the more detailed attributes.

Since the training data consist of quite imbalanced data, a weighted data sampler is applied. The weighted data sampler assigns sampling probabilities to each data point based on class frequencies, ensuring that underrepresented classes are sampled more frequently. This helps address this class imbalance during model training by providing more balanced input batches.

The hyperparameter optimization is conducted via grid search. The grid search is constrained by the following values within its search space: Learning rates=(0.0001 0.001 0.01 0.1) Batch sizes=(32 64) Dropout rates=(0.5 0.3) Layers=(2 3 4) Hidden units=(64 128 256) Given that the search space of the grid search would entail 144 distinct combinations, a random grid search was employed, resulting in the generation of 32 unique hyperparameter configurations.

The model was trained for 100 epochs. The results are presented in Table 12. The impact of the various hyperparameters on the accuracy and F1 score can be observed in Figures 40 and 41, respectively.

Table 12 depicts the highest accuracy (40.11%) and F1-score (0.3808) for a configuration comprising 64 hidden units, four layers, 0.3 dropout, a batch size of 64, and a learning rate of 0.001. It is notable that the validation set is somewhat imbalanced. Consequently, there is greater interest in the F1 score than in the accuracy, as the former provides a more nuanced insight into performance. Although the aforementioned combination of hyperparameters yielded the highest F1-score, an examination of Figure 41 suggests that modifications to the parameters of the layers and dropout rate may be warranted. The application of a dropout rate of 0.5 has been observed to yield higher averages for the F1 score in comparison to a dropout rate of 0.3. Furthermore, the application of three LSTM layers has been observed to yield the highest average F1 score, while simultaneously exhibiting the lowest average training time.

It can be concluded that the following hyperparameters should be used during the training of the models on the detailed attributes with the basic features, as well as during the training of the models for both kinds of HPE features. The learning rate should be set to 0.001, the batch size to 64, the number of hidden units to 64, the number of LSTM layers to 3, and the dropout rate to 0.5.

Table 9: Classification Data: Training, Validation, and Test Splits for Three Models in format (base, top-down, bottom-up)

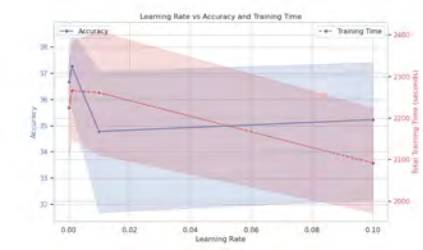
Classification Type	Training	Validation	Test
Primary	3168/3171/3168	905/905/905	452/452/452
Shot Body Part	304/320/320	89/83/80	51/41/44
Duel	691/675/687	170/203/177	106/89/103
Normal Pass or Cross	1098/1087/1084	326/333/329	159/163/169
Accuracy Cross	410/398/387	122/123/122	46/57/69
Direction Cross	410/398/387	122/123/122	46/57/69
Flank Cross	410/398/387	122/123/122	46/57/69
Accuracy Pass	688/689/697	204/210/207	113/106/100
Direction Pass	688/689/697	204/210/207	113/106/100
Distance Pass	688/689/697	204/210/207	113/106/100
Progressive Pass	688/689/697	204/210/207	113/106/100
Through Pass	688/689/697	204/210/207	113/106/100
On Target Shot	304/320/320	89/83/80	51/41/44

8 RESULTS

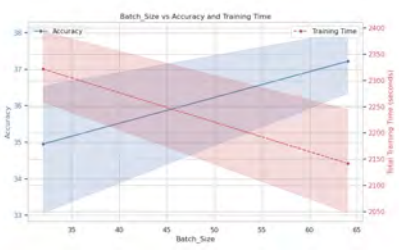
The various models for the different classification tasks were trained for 100 epochs. For each classification task, the epoch exhibiting the highest validation score was subsequently evaluated on the test sets. This section presents the results that were obtained.

8.1 Primary classification

With regard to the shot events, it is evident that a considerable proportion of the shots are classified as such for all three models, with the base model and bottom-up model outperforming the top-down model. It is also evident that a considerable number of false positives are predicted for shots. The predicted shot events of both the top-down and bottom-up models are presented in Figure 45. The subfigures illustrate the initial location of the event, as indicated by the event data, and its corresponding true label. In the event data, the location coordinates are indicated with the x-coordinate set to zero in relation to the goal of the team that takes the shot. Consequently, all genuine shots are located on the right side of the field. Both models demonstrate an understanding of the concept that for an event to be classified as a shot, it must be situated in close proximity to the goal. On the left side of the field, the predominant occurrences are duels and interceptions. Given that duels are notated in the event dataset for all players involved in the duel, it can be inferred that these events are likely associated with the defending player, resulting in low x-coordinates. It is also

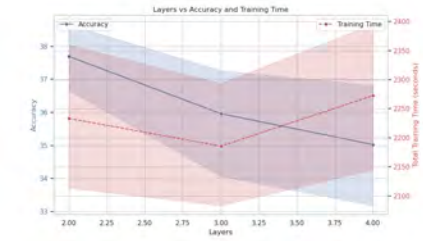


(a) Learning rate against accuracy

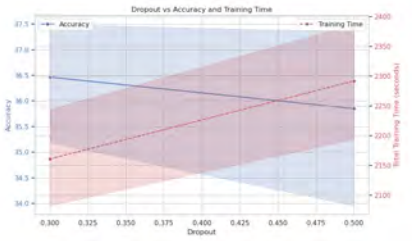


(b) Batch size against accuracy

(c) Hidden units against accuracy

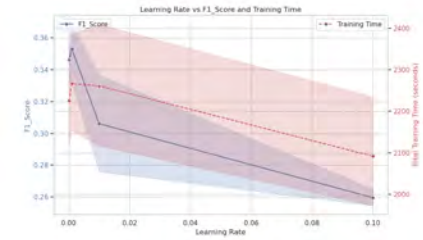


(d) Layers against accuracy

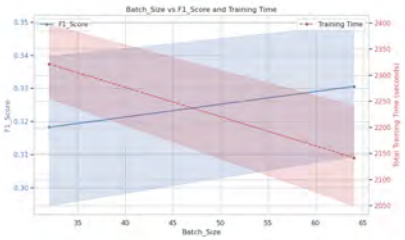


(e) Dropout against accuracy

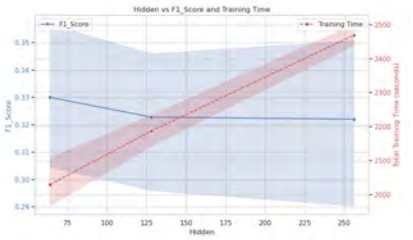
Figure 40: Influence of different performance measures on the accuracy



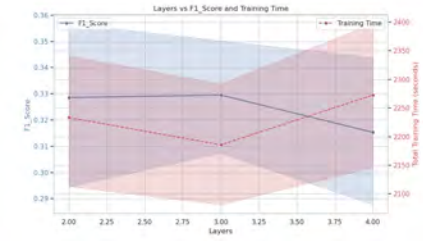
(a) Learning rate against F1-score



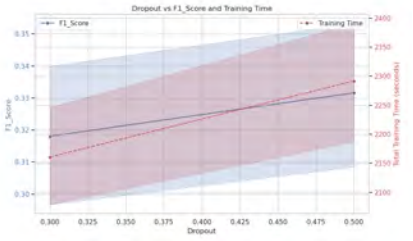
(b) Batch size against F1-score



(c) Hidden units against F1-score



(d) Layers against F1-score

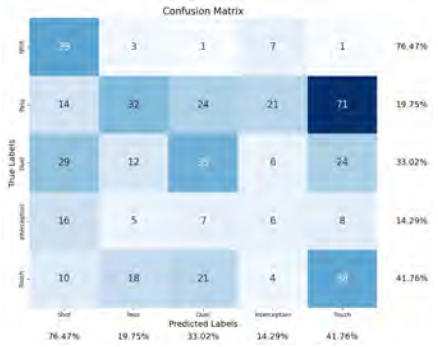


(e) Dropout against F1-score

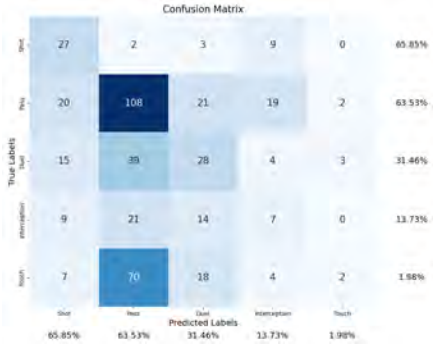
Figure 41: Influence of different performance measures on the F1-score

possible that interceptions may be linked to shots, given that interceptions are notated at the defender's coordinates, who have likely just blocked a shot from the opposing team. Therefore, although the event is classified as an interception, a shot may have occurred just before and was also visible in the clip, indicating that the model is capable of recognizing shots.

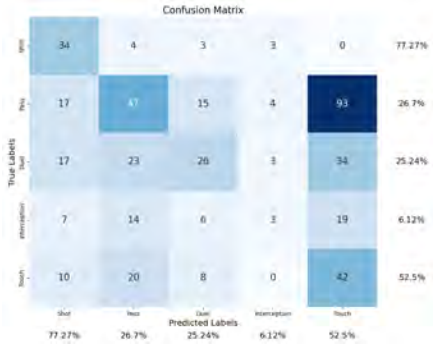
In regard to pass events, the top-down model demonstrates superior performance in comparison to the base and bottom-up models. The base and bottom-up models appear to encounter difficulties in differentiating between a touch and a pass. This is not entirely unexpected, given that the action of a touch is often immediately preceded and followed by a pass, which is then also observed in the clip due to the slice method employed.



(a)



(b)



(c)

Figure 42: Confidence matrices for base (42a), top-down (42b), and bottom-up (42c) classification models. In the cells under/next to the columns/rows the percentage of the correct classifications of the total within that column/row are shown.

With regard to duel events, all three models exhibit a lack of capacity to distinguish them from the other classes, resulting in a low accuracy score. The base and bottom-up models frequently categorise these as touches, whereas the top-down model often identifies them as passes.

With regard to touch events, the base and bottom-up models achieve considerably higher scores than the top-down model. The top-down model exhibits a striking lack of capacity to classify actions as touches, whereas the base and bottom-up models demonstrate a considerably higher propensity to do so. Moreover, this

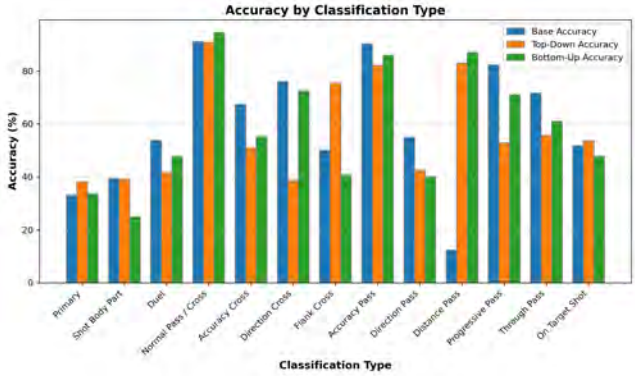


Figure 43: Accuracy's of the three models per classification type

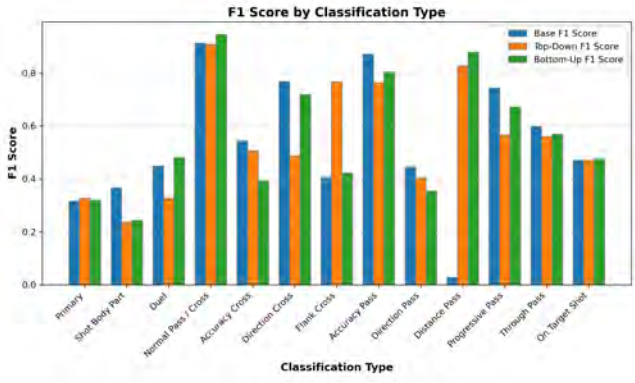


Figure 44: F1-scores of the three models per classification type

occurs with greater frequency than is necessary, as previously mentioned in the context of passes.

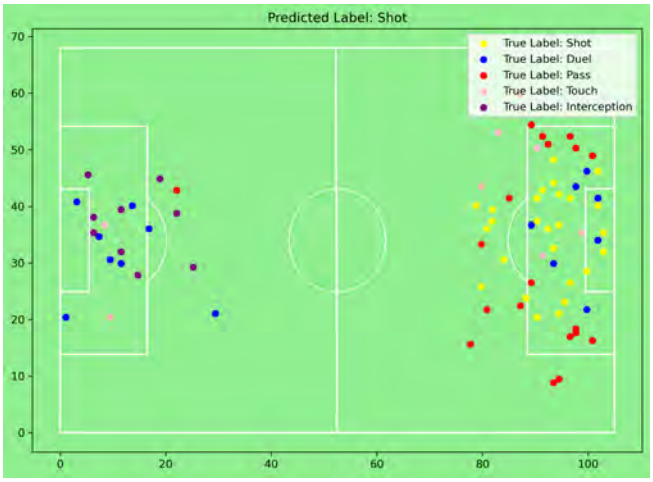
8.2 Duel-related classifications

In terms of duel-related classification, there is the classification between an 'aerial duel', 'loose ball duel' and 'ground duel'. Table 62 shows the confusion matrices of the three different models. Figure 43 shows that the base and bottom-up model show higher accuracy and F1-score than the top-down method. The confusion matrix shows that both base and top-down model do not do predictions on ground duels, whereas the bottom-up model does.

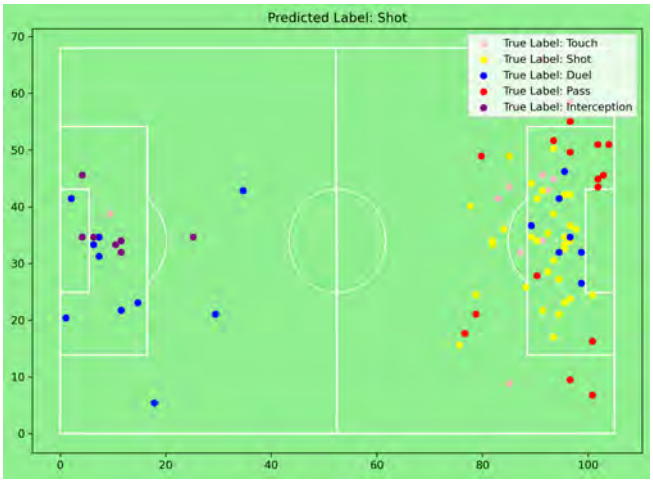
8.3 Shot-related classifications

In terms of shot-related classification, there are two different classifications to make. First of all the classification whether a shot is on target, or whether the shot goes wide or is blocked. In addition to that, the classification on with which body-part the shot is done.

In terms of assessing shot accuracy Figures 43 and 44 show similar accuracy's and F1 scores for all three models. Table 54 shows that the base model and the top-down model both have a preference to classify a shot as 'not on target', while the bottom-up model does



(a) Predicted shot events top-down model and its true labels



(b) Predicted shot events bottom-up model and its true labels

Figure 45: Main caption describing both subfigures.

consider both options, but often makes the wrong choice in doing so.

Figure 44 illustrates that the base model, contrary to expectations, attains a considerably higher F1 score than the two HPE-based models. However, as can be seen in Table 60, the base model selects only two options, namely "right" and "left." The top-down model demonstrates a clear preference for the left foot. The bottom-up based model, however, does in fact select from all three options, but appears to lack the ability to determine which is the correct option to choose. One potential explanation for these suboptimal predictions with these models, despite the fact that this should have been predictable for the HPE-based models, is that ball detection often fails when it is in close proximity to a player. Subsequently, the location is interpolated, which does not yield an entirely accurate result.

8.4 Pass-related classification

Regarding pass-related classification, a first distinction can be made between a cross and a normal pass. As evidenced in Table 57, all three models demonstrate high accuracy in differentiating between these two types of passes.

With regard to the categorisation of the distance of a pass, as illustrated in Table 53, the base model demonstrates a notable deficiency in predictive accuracy, classifying all passes as 'long'. In contrast, the other two models exhibit superior performance in this regard, achieving commendable results.

Upon examination of the classification of pass accuracy, it is evident that all three models exhibit high scores. However, an examination of Table 52 reveals that all three models exhibit a lack of capacity to predict the inaccuracy of a pass, thereby preventing them from discerning the distinction between accurate and inaccurate outcomes. This may be due to the fact that an accurate pass frequently occurs prior to a misplaced pass and is therefore also captured in the video footage. With regard to the classification of the success of a cross, it can be observed in Table 51 that both the base model and the bottom-up model predict all instances as unsuccessful. In contrast, the top-down model makes predictions on both classes.

With regard to the classification of through and progressive passes, an examination of Tables 55 and 56 reveals that the base model does not make a distinction and consequently classifies all instances as non-through or non-progressive. The bottom-up model also exhibits a strong tendency to classify both as "not". In contrast, the top-down model attempts to differentiate between the two options for both classifications, although this approach is not always effective.

With regard to the classification of the direction of the pass, it can be observed in Figure 43 that the base model achieves a high level of accuracy. However, an examination of the Table 59 reveals that this model predicts almost every instance as a lateral pass. The other two models exhibit a lesser tendency to make this error, but they frequently misclassify lateral passes as forward passes. Figure 46 illustrates the wrong predictions on direction of passes and their respective start and end points according to the event data. It can be observed that the direction of many of these passes does not even come close to the predicted direction, indicating that both models are unable to accurately identify the decisive factor in this classification task.

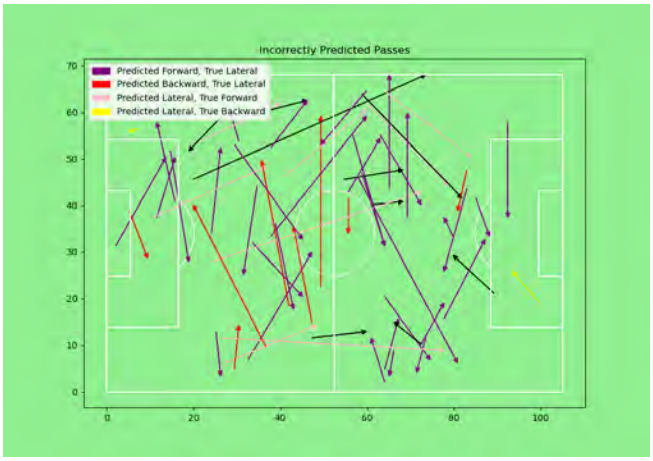
In terms of classifying the models on cross direction, Table 58 reveals that the base model and bottom-up model both achieve satisfactory results. However, the top-down model frequently confuses lateral crosses with backward crosses.

As illustrated in Figures 43 and 44, the top-down model demonstrates superior performance in predicting the flank from which a cross originates compared to the other two models. However, its confusion matrix in Table 61 indicates that this model exhibits lower accuracy in identifying crosses from the center of the field.

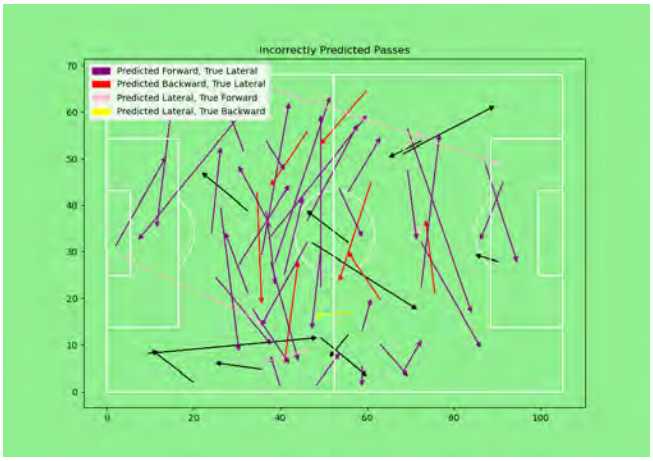
9 CONCLUSION AND DISCUSSION

9.1 Conclusion

The objective of this research was to establish the preliminary stages of developing an advanced data collection system based on



(a) False predictions of direction of passes top-down model



(b) False predictions of direction of passes bottom-up model

Figure 46

video footage of in-game football events obtained from a single, monocular camera. The objective was to develop and train LSTM classification models with the capacity to categorise brief video segments into five principal classes, along with an array of subclasses. To this end, a comprehensive pipeline was constructed, integrating a variety of existing and self-designed methods, including field registration, player and ball detection and tracking, team classification, and human pose estimation algorithms. The primary objective of this research was to evaluate the impact of diverse HPE techniques on the efficacy of clip classification. This was achieved by integrating a top-down and a bottom-up HPE method into the pipeline and comparing their results with one another and with a 'base' model that did not incorporate human pose estimation.

The results demonstrate that all three models exhibit significant challenges in classifying at both event and attribute levels. However, the classification models of both HPE-based models demonstrate superior performance compared to the base model in each type of classification. It can thus be concluded that the incorporation of HPE data facilitates the classification process. In comparing the two HPE-based models, the top-down based model achieves a slightly superior score on the hierarchically higher level of primary events. However, on the attribute level, the bottom-up model appears to perform slightly better with greater frequency. This research contributes to the scientific field by demonstrating the impact of employing two distinct types of HPE on the classification of actions. Based on the findings presented in Section 7 and the insights derived from Section 6, it is recommended to utilize bottom-up methodologies within the context of HPE models for the classification of football actions.

9.2 Discussion

The findings of this study provide a foundation for further research on the recognition of in-game football actions. The findings of this study can be used to inform the deployment of a specific HPE model at a specified depth of recognition of actions or attributes.

The classification of football actions from a single monocular viewpoint is a challenging endeavour, due to a number of practical considerations. Firstly, the image quality provided by a single monocular viewpoint from the opposing side of the pitch is insufficient for the accurate recognition of specific actions. Furthermore, the images captured from a single viewpoint frequently result in occlusions of the ball or player. Additionally, due to the limited angle of filming, it is challenging to accurately determine the location of the ball when it is in flight. To address this, it is essential to obtain footage from multiple angles to create a 3D model.

It is essential to acknowledge certain limitations in the methodology which may have influenced the outcomes. Firstly, in this study, the data set employed for the purposes of training, validating and testing the trained classification models is distinct for each classification model. The entirety of the dataset comprises the identical video segments; however, the manner in which these segments are distributed across the datasets differs for each classification model. The distribution in terms of numbers, as indicated in 9, is approximately equivalent. However, to facilitate a meaningful comparison between different models, it is essential that they are trained, validated and tested on an identical data set. This was not done in this study. It could be possible that a specific model has only been trained on simple and frequently occurring instances of a particular event, which may limit its ability to recognize slightly varied instances of this same event. Alternatively, it is conceivable that one model's test set of a particular event contains instances that are more easily recognisable, which may result in a higher test score.

A considerable proportion of the classifications were confronted with the challenge of imbalanced data. This issue can be attributed to two underlying factors. Firstly, the reduction in a particular type of data following the classification process, as outlined in (REF to classify clips), should have been anticipated on with the introduction of a new supply of the affected data until it reaches a level commensurate with the remaining data. Secondly, this can be attributed to the fact that some specific instances of a given event occur with much lower frequency than other instances. To illustrate, in the case of a cross-event, the vast majority of crosses are made in a lateral direction. Due to the methodology employed in this research, if an equal number of lateral, backward, and forward crosses

were to be sampled, to create a balanced dataset for the classficiation of the direction, the resulting data would be an inaccurate representation of the average cross for the classification between a pass and a cross. In this research, the issue is addressed through the use of a weighted sampler, which ensures an equal number of training instances for each class. However, this approach results in the repetition of training instances from less represented classes. It would be optimal for each classification task to have its own balanced dataset, which should also be representative of the typical appearance of that kind of event in a real match. To examplifie this, with regard to the classification between cross and pass, the ratio of the direction of the crosses in the training dataset could be represented as follows: (0.90; 0.05; 0.05) for (lateral, backward, forward). Conversely, for the classification of direction, the ratio should be as follows (0.33; 0.33; 0.33).

Besides this, the absence of cross-validation in this study represents a limitation in the generalisability of the models.

9.3 Future research

Although the proposed method provides a suitable means of fair comparison, the process of fixing the training hyperparameters of both the HPE-based classification models based on the optimals of the non-HPE-based model may have had an impact on the performance of the HPE-based models. Further research could investigate the impact on the performance of the various HPE-based classification models of tuning the hyperparameters on a model-by-model basis.

Furthermore, the choice of clip slicing method has a significant impact on the outcomes of this research. Future studies could explore whether more accurate results could be achieved by modifying the slicing clip method. Additionally, research could be conducted on the detection of the start and end of an action or event, which would represent a significant advancement in the ability to classify all actions throughout an entire match.

Finally, this research is based on footage from a single, monocular view camera of a match. Future research could investigate the performance of different HPE models within classifying or recognising football actions when footage is present from different angles, allowing the entire pitch to be imaged and creating a 3D representation instead of a 2D representation.

10 RECOMMENDATIONS

As evidenced by the results and the conclusion, utilising computer vision as a method for monitoring player performance from a monocular camera perspective is a challenging endeavour. In this study, the classification process is based solely on the analysis of feature information, rather than on the direct utilisation of camera footage as input to the classification system. The incorporation of camera footage into the classification model could potentially enhance the classification process; however, this would also result in an increase in computation time. However, this would only enhance it to a certain degree, as the feature extraction methods indicate that in certain situations, the detection or tracking is insufficient or invalid. The primary reason for this is the paucity of detail in the footage. The monocular viewpoint camera is insufficient for providing the requisite information and detail about the actions occurring on the field, thereby hindering consistent classification accuracy.

Therefore, it is not feasible to implement a player performance tracking system based on computer vision for the scouting process. It may be feasible to implement a player performance tracking system on the training grounds of AFC Ajax. However, a number of adaptations and additions are required. Primarily, additional cameras have to be positioned around the field, with the potential for supplementary cameras to be installed above the field as well. This allows for the capture of more detailed information from the cameras, which in turn facilitates the classification process. Furthermore, this allows a three-dimensional representation of the field to be generated, thereby enabling the precise location of the ball to be determined for each frame, even when it is in flight. This study focuses on action classification. However, to develop a performance tracking system, it is necessary to investigate the ability to identify the start and end points of actions as well.

Despite the lengthy process of implementing a functional performance tracking system in practice, given the multitude of diverse and intricate actions inherent to football, the computer vision algorithms employed can be utilized at an earlier stage for other, somewhat less complex applications. For example, they can be employed to identify specific formations and the compositions of positions of players on the field. Moments of high pressure, game replays, or switch-overs can be identified by a model. This can then be utilized as an automated process for video analysts or coaches, to quickly find certain situations from a variety of games. With regard to the utilization of data within scouting processes, it is recommended to utilize Wyscout's event-data package, and that all new event-data files undergo processing on a weekly basis. This will convert the event information into the necessary statistics, which can then be used in the scouting process.

REFERENCES

[1] Rockson Agyeman, Rafiq Muhammad, and Gyu Sang Choi. 2019. Soccer video summarization using deep learning. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 270–273.

[2] Sermetcan Baysal and Pinar Duygulu. 2015. Sentioscope: A Soccer Player Tracking System Using Model Field Particles. *IEEE Transactions on Circuits and Systems for Video Technology* 26 (01 2015), 1–1. https://doi.org/10.1109/TCSVT.2015.2455713

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. 2019. Tracking Without Bells and Whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[4] Matthew A. Brown and David G. Lowe. 2003. Recognising panoramas. *Proceedings Ninth IEEE International Conference on Computer Vision* (2003), 1218–1225 vol.2. https://api.semanticscholar.org/CorpusID:12554466

[5] Matija Burić, Miran Pobar, and Marina Ivašić-Kos. 2018. Object detection in sports videos. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 1034–1039.

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *CoRR* abs/1812.08008 (2018). arXiv:1812.08008 http://arxiv.org/abs/1812.08008

[7] Fiona Carmichael, Giambattista Rossi, and Denis Thomas. 2017. Production, Efficiency, and Corruption in Italian Serie A Football. *Journal of Sports Economics* 18, 1 (2017), 34–57. https://doi.org/10.1177/1527002514551802

[8] Chong Chen, Yao Shu, Kuang-I Shu, and Heng Zhang. 2018. WiTT:Modeling and the evaluation of table tennis actions based on WIFI signals. In *2018 24th International Conference on Pattern Recognition (ICPR)*. 3100–3107. https://doi.org/10.1109/ICPR.2018.8545854

[9] Jianhui Chen and J James. 2019. Little. 2019. Sports camera calibration via synthetic data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 0–0.*

[10] Xianjie Chen and Alan L Yuille. 2014. Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/8b6dd7db9af49e67306feb59a8bdc52c-Paper.pdf

[11] Yucheng Chen, Yingli Tian, and Mingyi He. 2020. Monocular human pose estimation: A survey of deep learning-based methods. *Computer vision and image understanding* 192 (2020), 102897.

[12] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. 2019. Bottom-up Higher-Resolution Networks for Multi-Person Pose Estimation. *CoRR* abs/1908.10357 (2019). arXiv:1908.10357 http://arxiv.org/abs/1908.10357

[13] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. 2017. Multi-Context Attention for Human Pose Estimation. *CoRR* abs/1702.07432 (2017). arXiv:1702.07432 http://arxiv.org/abs/1702.07432

[14] Steve Cornelius. 2023. The football club as media supplier: Media rights. In *Research Handbook on the Law of Professional Football Clubs*. Edward Elgar Publishing, 176–193.

[15] Carlos Cuevas, Daniel Quilon, and Narciso Garcia. 2020. Automatic soccer field of play registration. *Pattern Recognition* 103 (2020), 107278.

[16] Shradha Dubey and Manish Dixit. 2023. A comprehensive survey on human pose estimation approaches. *Multimedia Systems* 29, 1 (2023), 167–195.

[17] Elan Dubrofsky. 2009. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie* 5 (2009).

[18] Mehrnaz Fani, Kanav Vats, Christopher Dulhanty, David A Clausi, and John Zelek. 2019. Pose-projected action recognition hourglass network (PARHN) in soccer. In *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE Computer Society, 201–208.

[19] Pedro F Felzenszwalb and Daniel P Huttenlocher. 2005. Pictorial structures for object recognition. *International journal of computer vision* 61 (2005), 55–79.

[20] FootballKitArchive. 2024. *Serie A 2023-24 Tenues*. https://www.footballkitarchive.com/nl/serie-a-2023-24-kits/ Accessed: 2024-06-23.

[21] Fédération Internationale de Football Association. 2023. *Laws of the Game 2023/24*. https://www.fifa.com/technical/refereeing/laws-of-the-game/ Accessed: 2024-06-23.

[22] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. 2018. Soccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 1711–1721.

[23] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[24] Arpan Gupta, Ashish Karel, and M. Sakthi Balan. 2020. Discovering Cricket Stroke Classes in Trimmed Telecast Videos. In *Computer Vision and Image Processing*, Neeta Nain, Santosh Kumar Vipparthi, and Balasubramanian Raman (Eds.). Springer Singapore, Singapore, 509–520.

[25] Ankur Gupta, James J Little, and Robert J Woodham. 2011. Using line and ellipse features for rectification of broadcast hockey video. In *2011 Canadian conference on computer and robot vision*. IEEE, 32–39.

[26] Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

[27] Jean-Bernard Hayet, Justus Piater, and Jacques Verly. 2004. Robust incremental rectification of sports video sequences. In *British machine vision conference (BMVC'04)*. Citeseer, 687–696.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[29] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. 2016. Soccer field localization from a single image. *arXiv preprint arXiv:1604.02715* (2016).

[30] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. 2017. Sports field localization via deep structured models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5212–5220.

[31] Kristina Host and Marina Ivašić-Kos. 2022. An overview of Human Action Recognition in sports based on Computer Vision. *Heliyon* 8, 6 (2022).

[32] Maxime Istasse, Julien Moreau, and Christophe De Vleeschouwer. 2019. Associative Embedding for Team Discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[33] Noor Khokhar. 2024. *YOLOv8-football*. https://github.com/noorkhokhar99/YOLOv8-football Accessed: 2024-03-06.

[34] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. 2019. PifPaf: Composite Fields for Human Pose Estimation. *CoRR* abs/1903.06593 (2019). arXiv:1903.06593 http://arxiv.org/abs/1903.06593

[35] Simon Kuper and Stefan Szymanski. 2018. *Soccernomics: Why England Loses, Why Germany and Brazil Win, and Why the US, Japan, Australia, Turkey–and Even Iraq–Are Destined to Become the Kings of the World's Most Popular Sport*. Hachette UK.

[36] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. 2021. Pose Recognition with Cascade Transformers. *CoRR* abs/2104.06976 (2021). arXiv:2104.06976 https://arxiv.org/abs/2104.06976

[37] Sijin Li, Zhi-Qiang Liu, and Antoni B. Chan. 2014. Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network. *CoRR* abs/1406.3474 (2014). arXiv:1406.3474 http://arxiv.org/abs/1406.3474

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[39] Behzad Mahaseni, Erma Rahayu Mohd Faizal, and Ram Gopal Raj. 2021. Spotting football events using two-stream convolutional neural network and dilated recurrent neural network. *IEEE Access* 9 (2021), 61929–61942.

[40] Panagiotis Mavrogiannis and Ilias Maglogiannis. 2022. Amateur football analytics using computer vision. *Neural Computing and Applications* 34, 22 (2022), 19639–19654.

[41] Hiroaki Minoura, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi, Mitsuru Nakazawa, Yeongnam Chae, and Björn Stenger. 2021. Action spotting and temporal attention analysis in soccer videos. In *2021 17th International Conference on Machine Vision and Applications (MVA)*. IEEE, 1–6.

[42] Tewodros Legesse Munea, Yalew Zelalem Jembre, Halefom Tekle Weldegebriel, Longbiao Chen, Chenxi Huang, and Chenhui Yang. 2020. The progress of human pose estimation: A survey and taxonomy of models applied in 2D human pose estimation. *IEEE Access* 8 (2020), 133330–133348.

[43] Banoth Thulasya Naik, Mohammad Farukh Hashmi, and Neeraj Dhanraj Bokde. 2022. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. *Applied Sciences* 12, 9 (2022), 4429.

[44] Banoth Thulasya Naik, Mohammad Farukh Hashmi, Zong Woo Geem, and Neeraj Dhanraj Bokde. 2022. DeepPlayer-track: player and referee tracking with jersey color recognition in soccer. *IEEE Access* 10 (2022), 32494–32509.

[45] Niels Nederlof. 2024. Interview by Ivo van Miert. Personal Interview.

[46] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. *CoRR* abs/1603.06937 (2016). arXiv:1603.06937 http://arxiv.org/abs/1603.06937

[47] Guanghan Ning, Jian Pei, and Heng Huang. 2020. LightTrack: A Generic Framework for Online Top-Down Human Pose Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[48] George Nousias, Konstantinos Delibasis, and Ilias Maglogiannis. 2023. *H*-RANSAC, an algorithmic variant for Homography image transform from featureless point sets: application to video-based football analytics. *arXiv preprint arXiv:2310.04912* (2023).

[49] Yutian Pang and Yongming Liu. 2020. Conditional generative adversarial networks (CGAN) for aircraft trajectory prediction considering weather effects. In *AIAA Scitech 2020 Forum*. 1853.

[50] Preksha Pareek and Ankit Thakkar. 2021. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review* 54, 3 (2021), 2259–2322.

[51] Viorica Pătrăucean, Pierre Gurdjos, and Rafael Grompone Von Gioi. 2012. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*. Springer, 572–585.

[52] Nur Rahmad and Muhammad Amir As'ari. 2020. The new Convolutional Neural Network (CNN) local feature extractor for automated badminton action recognition on vision based data. *Journal of Physics: Conference Series* 1529 (04 2020), 022021. https://doi.org/10.1088/1742-6596/1529/2/022021

[53] Martin Rajchl, Matthew C. H. Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Bernhard Kainz, and Daniel Rueckert. 2016. DeepCut: Object Segmentation from Bounding Box Annotations using Convolutional Neural Networks. *CoRR* abs/1605.07866 (2016). arXiv:1605.07866 http://arxiv.org/abs/1605.07866

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence* 39, 6 (2016), 1137–1149.

[55] Haşim Sak, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*.

[56] Ryan Sanford, Siavash Gorji, Luiz G Hafemann, Bahareh Pourbabaee, and Mehrsan Javan. 2020. Group activity detection from trajectory and video data in soccer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 898–899.

[57] PK Santhosh and B Kaarthick. 2019. An Automated Player Detection and Tracking in Basketball Game. *Computers, Materials & Continua* 58, 3 (2019).

[58] Iqbal Sarker. 2021. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science* 2 (08 2021). https://doi.org/10.1007/s42979-021-00815-1

[59] Rahul Anand Sharma, Bharath Bhat, Vineet Gandhi, and CV Jawahar. 2018. Automated top view registration of broadcast football videos. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 305–313.

[60] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556* (09 2014).

[61] sithu31296. 2024. *Easy to use SOTA Top-Down Multi-person Pose Estimation Models in PyTorch.* https://github.com/sithu31296/pose-estimation?tab=readme-ov-file

[62] Soccer-Net. 2024. Camera Calibration. https://www.soccer-net.org/tasks/camera-calibration. Accessed: 2024-09-17.

[63] SoccerNet. 2023. *SoccerNet Action Spotting.* https://www.soccer-net.org/tasks/action-spotting Accessed: 2024-03-06.

[64] SoccerNet. 2023. *SoccerNet Ball Action Spotting.* https://www.soccer-net.org/tasks/ball-action-spotting Accessed: 2024-03-06.

[65] Khurram Soomro and Amir R Zamir. 2015. Action recognition in realistic sports videos. In *Computer vision in sports*. Springer, 181–208.

[66] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[67] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. 2017. Compositional Human Pose Regression. *CoRR* abs/1704.00159 (2017). arXiv:1704.00159 http://arxiv.org/abs/1704.00159

[68] Michael J Swain and Dana H Ballard. 1991. Color indexing. *International journal of computer vision* 7, 1 (1991), 11–32.

[69] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. 2013. Deep neural networks for object detection. *Advances in neural information processing systems* 26 (2013).

[70] Wei Tang and Ying Wu. 2019. Does Learning Specific Features for Related Parts Help Human Pose Estimation?. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1107–1116. https://doi.org/10.1109/CVPR.2019.00120

[71] Jonas Theiner, Wolfgang Gritz, Eric Müller-Budack, Robert Rein, Daniel Memmert, and Ralph Ewerth. 2022. Extraction of positional player data from broadcast soccer videos. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 823–833.

[72] Matteo Tomei, Lorenzo Baraldi, Simone Calderara, Simone Bronzin, and Rita Cucchiara. 2021. Rms-net: Regression and masking for soccer event spotting. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 7699–7706.

[73] Alexander Toshev and Christian Szegedy. 2014. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1653–1660.

[74] Karl Tuyls, Shayegan Omidshafiei, Paul Muller, Zhe Wang, Jerome Connor, Daniel Hennes, Ian Graham, William Spearman, Tim Waskett, Dafydd Steel, et al. 2021. Game Plan: What AI can do for Football, and What Football can do for AI. *Journal of Artificial Intelligence Research* 71 (2021), 41–88.

[75] Hayat Ullah and Muhammad Sajjad. 2018. Salient event detection in soccer videos using histogram of oriented gradient. In *Proc. 4th Int. Conf. Next Gener. Comput.(ICNGC)*. 231–233.

[76] Ultralytics. 2024. *YOLOv8.* https://github.com/ultralytics/ultralytics Accessed: 2024-03-06.

[77] Shankara V, Syed Ahmed, Sneha M, and Guruprakash Jayabalasamy. 2024. Object Detection and Tracking for Football Data Analytics. https://doi.org/10.4108/eai.23-11-2023.2343216

[78] Ivo van Miert. 2024. Master Thesis Repository. https://github.com/ivovanmiert/Thesis Accessed: 2024-09-29.

[79] Cor J Veenman, Marcel JT Reinders, and Eric Backer. 2001. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 1 (2001), 54–72.

[80] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. *CoRR* abs/1602.00134 (2016). arXiv:1602.00134 http://arxiv.org/abs/1602.00134

[81] Wyscout. 2024. *API Wyscout Events Data Package.* https://apidocs.wyscout.com/tag/Events Accessed: 2024-06-23.

[82] Wyscout. 2024. *Wyscout Data.* https://footballdata.wyscout.com/ Accessed: 2024-06-23.

[83] Wyscout. 2024. *Wyscout Glossary.* https://dataglossary.wyscout.com/

[84] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation. 3073–3082. https://doi.org/10.1109/CVPR.2016.335

[85] Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. 2021. Lite-HRNet: A Lightweight High-Resolution Network. *CoRR* abs/2104.06403 (2021). arXiv:2104.06403 https://arxiv.org/abs/2104.06403

[86] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. 2021. HRFormer: High-Resolution Transformer for Dense Prediction. *CoRR* abs/2110.09408 (2021). arXiv:2110.09408 https://arxiv.org/abs/2110.09408

[87] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. 2023. Deep learning-based human pose estimation: A survey. *Comput. Surveys* 56, 1 (2023), 1–37.

[88] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. 2023. Object detection in 20 years: A survey. *Proc. IEEE* 111, 3 (2023), 257–276.

A DETAILED DESCRIPTIONS

A.1 Network Architecture Keypoints Detection Model

The architecture for the keypoint detection is displayed in Figure 47. The network takes as input a batch of images, each with dimensions [3, 540, 960], representing RGB channels, height and width respectively. The network outputs heatmaps indicating the locations of various keypoints on the pitch. The network is composed of several layers and components, and its architecture can be described as follows: The input image passes through an initial convolutional layer, employing a 2D convolution with a kernel size of 3 and a stride of 2. The layer uses 64 filters and is followed by batch normalization and a ReLU activation function. The output of this layer either goes into a second convolutional layer identical to the first one, or awaits for later concatenation. The output from the first two convolutional layers is fed into the HRNetV2-w48 backbone. Wang et al. [2] proposed their High-Resolution Network (HRNet) as a framework that maintains high-resolution representation through the whole process, whereas other already existing frameworks first encode the input image as a low-resolution representation through a subnetwork that is formed by connecting high-to-low resolution convolution in series and then recover the high-resolution representation from the encoded low-resolution representation [2]. In this architecture, the backbone outputs a feature map of size [Batch, 720, 135, 240], which contains rich spatial and contextual information. The output of the HRNet-V2-w48 is subjected to a 2x upsampling operation to enhance the spatial resolution of the feature map. This feature map then is concatenated with the other output branch of the initial convolutional layer. This concatenation with the feature map from the earlier layer ensures that the high-resolution information from that layer is preserved and integrated into this later stage of the network. This produces a combined feature map size of [Batch, 784, 270, 480]. This feature map is then fed through two convolutional layers with 784 and 58 filters respectively. The final convolutional layer outputs a tensor of size [Batch, 58, 270, 480], where 58 corresponds to the number of keypoints plus one extra target channel, which ensures that the final target tensor sums to 1.0 at each spatial point. Softmax is employed as the final activation function to ensure the output is probabilistic and suitable for heatmap interpretation.

The network is initially trained using Mean Squared Error (MSE) loss, which compares the predicted heatmaps with the ground truth, and the Adam optimizer with a learning rate of 0.001, which is halved after 8 epochs of non-improvement, continuing until no further gains are seen over 32 epochs to prevent overfitting. It is then fine-tuned with Adaptive Wing Loss, applying the same learning rate strategy starting at 5e-4, with the best model selected based on a combined accuracy and completeness metric on the validation dataset. The model was trained on 16463 images of the SoccerNet Dataset. Accuracy measures on the keypoint detection were not mentioned since they were not all included in the SoccerNet Dataset, but the end model on predicting camera calibration parameter reached an accuracy (RMSE <5 pixels) of 76.675 finishing first on the leaderboard of the Soccernet Camera Calibration Challenge indicating a good keypoint estimation model. The images in the Soccernet Dataset are as in our dataset broadcast images, from various European Competitions covering three seasons from 2014 to 2017. So a little older than our dataset but very similar.

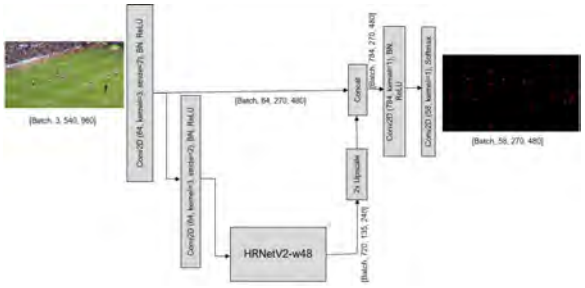


Figure 47: Architecture of the Keypoints Detection model

A.2 Homography Estimation

The Kepypoint Detection algorithm, as described in subsection 5.2.1, returns the prediction of the pixel-location of the 57 different keypoints, together with the confidence score of the prediction.

A homography is a 3x3 matrix H that describes a projective transformation between two planes. It allows you to transform points from one plane (e.g., an image) to another plane (e.g., a sports field). If $\mathbf{p} = (x, y, 1)$ are the homogeneous coordinates of a point in the image and $\mathbf{P} = (X, Y, 1)$ are the corresponding coordinates on the field, the relationship can be written as:

$$\mathbf{P} = H\mathbf{p}$$

where H is the homography matrix.

To compute H, we need at least four pairs of corresponding points between the two planes. Let (x_i, y_i) be the coordinates in the image and (X_i, Y_i) be the coordinates on the field for $i = 1, 2, 3, 4$. The relationship can be written as:

$$\begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} = H \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

Expanding this equation, we get:

$$h_{11}x_i + h_{12}y_i + h_{13} = X_i(h_{31}x_i + h_{32}y_i + h_{33})$$
$$h_{21}x_i + h_{22}y_i + h_{23} = Y_i(h_{31}x_i + h_{32}y_i + h_{33})$$

The above equations can be rewritten as a system of linear equations. For each point correspondence, we get two equations. For four points, we have eight equations. The system can be represented as:

$$X_i = h_{11}x_i + h_{12}y_i + h_{13} - X_i(h_{31}x_i + h_{32}y_i + h_{33})$$
$$Y_i = h_{21}x_i + h_{22}y_i + h_{23} - Y_i(h_{31}x_i + h_{32}y_i + h_{33})$$

Or in matrix form $A\mathbf{h} = 0$:

$$A = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1x_1 & -X_1y_1 & -X_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -Y_1x_1 & -Y_1y_1 & -Y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2x_2 & -X_2y_2 & -X_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -Y_2x_2 & -Y_2y_2 & -Y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -X_3x_3 & -X_3y_3 & -X_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -Y_3x_3 & -Y_3y_3 & -Y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -X_4x_4 & -X_4y_4 & -X_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -Y_4x_4 & -Y_4y_4 & -Y_4 \end{pmatrix}$$

And the vector $\mathbf{h}$ containing the elements of the homography matrix H:

$$\mathbf{h} = \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix}$$

To find the solution for $\mathbf{h}$, we use Singular Value Decomposition (SVD). The solution is the singular vector corresponding to the smallest singular value of matrix A. This vector represents the flattened homography matrix H.

Finally, the 9-element vector $\mathbf{h}$ can be reshaped into the 3x3 homography matrix H:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$$

A.3 PifPaf: Detailed Description

The PifPaf model representations can be found in figure 48. It can be described as a shared ResNet base network, a residual learning framework designed by [28] to ease the training that are substantially deep, with two head networks: a head network called Part Intensity Field (PIF), which predicts a confidence, precise location and size of a joint, and a head network called Part Association Field (PAF), which predicts associaton between those parts. The output feature maps of these two head networks are then decoded to create a set of 17 coordinates that compose into the human pose estimate. In this section, the two head networks are described in more detail, followed by a more detailed description of the decoding process.

Part Intensity Fields (PIF) are designed to detect and precisely localize specific body parts within an image by utilizing a composite structure. The PIF's combine multiple components: a scalar field p^c_{ij} representing the confidence value, a vector field $\{p^x_{ij}, p^y_{ij}\}$ pointing towards the closest body part, and another scalar field p^σ_{ij} representing the size of the joint. In this context, the indices i, j refer to the discrete spatial locations on the output grid of the neural network, while x, y represent real-valued coordinates within the image. The PIF at each location can be formally expressed as a composite field $p_{ij} = \{p^c_{ij}, p^x_{ij}, p^y_{ij}, p^b_{ij}, p^\sigma_{ij}\}$.
The confidence map generated by PIFs, represented by the scalar field p^c_{ij}, tends to be quite coarse. An example of a confidence map

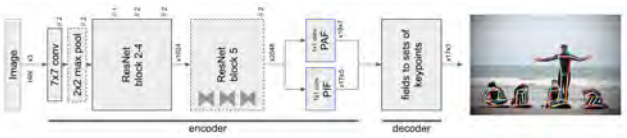


Figure 48: PifPaf Model architecture. The input is an image of size (H, W) with three color channels, indicated by "x3". The neural network based encoder produces PIF and PAF fields with 17×5 and 19×7 channels. An operation with stride two is indicated by "//2". The decoder is a program that converts PIF and PAF fields into pose estimates containing 17 joints each. Each joint is represented by an x and y coordinate and a confidence score.

on localizing the left shoulder generated by PIF's can be seen in Figure 49a. As can be seen, the confidence map is in very low resolution. To enhance localization, this confidence map is fused with the vectorial components $\{p^x_{ij}, p^y_{ij}\}$, which are shown in Figure 49b to create a high-resolution confidence map $f(x, y)$. This map, shown in figure c 49c is generated by convolving the regressed vector fields, weighted by the confidence scalar field, with an unnormalized Gaussian kernel. This emphasizes the grid-free nature of the localization process, improving the coarsiness of the confidence map generated before, allowing for precise identification of joint locations.

Part Association Fields (PAF) extend the concept of fields to associate detected joints into complete human poses. PAFs operate by predicting a composite field

$$a_{ij} = \{a^c_{ij}, a^{x1}_{ij}, a^{y1}_{ij}, a^{b1}_{ij}, a^{x2}_{ij}, a^{y2}_{ij}, a^{b2}_{ij}\}$$

at each output location (i, j). Here, a^c_{ij} is a scalar field representing the confidence of the association, $\{a^{x1}_{ij}, a^{y1}_{ij}\}$ and $\{a^{x2}_{ij}, a^{y2}_{ij}\}$ are vector fields representing the locations of the two joints being connected, and a^{b1}_{ij} and a^{b2}_{ij} are scalar fields representing the spatial precisions of these regressions.

The construction of PAFs involves identifying the closest joint of the required type to determine the first vector field component $\{a^x_{ij}1, a^y_{ij}1\}$, while the second vector component $\{a^x_{ij}2, a^y_{ij}2\}$ is determined by the ground truth pose, even if it points to a joint further away. This ensures that the fields correctly represent the association of parts, allowing for precise joint localization even in crowded scenes where multiple people may occlude one another.

Human Pose estimation algorithms tend to struggle with the diversity of scales that a human pose can have in an image. The regression outputs of the PIFs and PAFs are trained using an adaptive L1-type loss function that accounts for varying scales of human bodies within an image. This is important since the same absolute localization error for the joint of a large person can be minor, while for a small person in the background of the image, this error can be major. The L1-type loss is accountable for this since it is injected with a scale dependence with the SmoothL1 [23] loss. This loss allows to tune the radius r_{smooth} around the origin where softer gradients are produced, with r_{smooth} being proportional to $\sqrt{A_i}\sigma_k$,

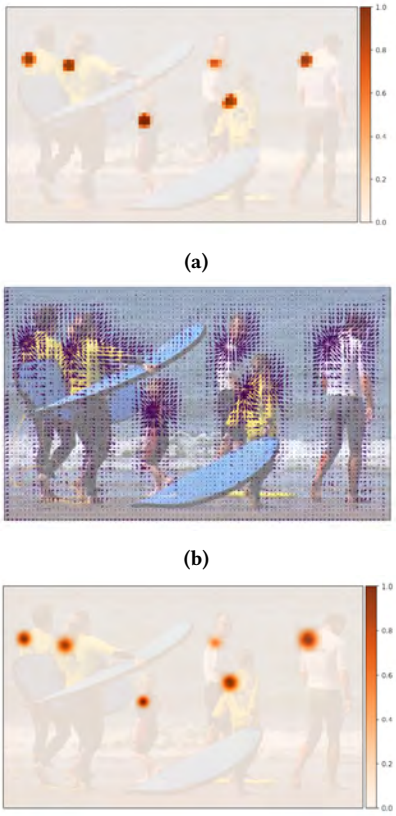


Figure 49: Visualization of the different components of the PIF of the left shoulder. In 49a the still coarse confidence map is shown. In 49b the vectorial components are shown. In 49c fusion of the convidence, vector and scale components are shown

where A_i is the area of the person instance's bounding box and σ_k is the keypoint size.

The decoding process converts the neural network's output feature maps into a set of 17 coordinates representing human pose estimates. This process begins by identifying PIF vectors with the highest values in the high-resolution confidence map $f(x, y)$ generated earlier. Starting from these high-confidence seed points, PAFs are used to form connections to other joints, effectively building the pose. The decoding algorithm is based on the fast greedy algorithm used by [34], meaning that once a connection is made, it is final. The scores $s(a, \mathbf{x})$ for PAF associations a are calculated by considering the confidence scalar field a^c_{ij}, the distance between the current joint position $\mathbf{x}$ and the vector field component $\{a^x_{ij}1, a^y_{ij}1\}$, and the high-resolution confidence at the target location $\{a^x_{ij}2, a^y_{ij}2\}$. Non-maximum suppression is applied at the keypoint level to finalize the pose, with the suppression radius being dynamically adjusted based on the predicted scale field p^σ_{ij}.

By leveraging these different fields, PIFs and PAFs provide a robust framework for human pose estimation, enabling precise detection and association of body parts even in complex scenes.

B TABLES

B.1 A

Primary Type	Secondary Type	Count	Secondary Type	Count
acceleration	carry	2047	progressive_run	1744
	linkup_play	45	third_assist	2
clearance	loss	973	head_pass	588
	under_pressure	341	recovery	146
	counterpressing_recovery	54	carry	19
	linkup_play	1		
corner	shot_assist	524	loss	489
	assist	26	opportunity	15
	carry	11	second_assist	10
	third_assist	5	shot	2
	progressive_run	1		
duel	ground_duel	78226	defensive_duel	39113
	offensive_duel	39113	aerial_duel	20566
	loose_ball_duel	19582	recovery	17894
	loss	17106	dribble	14614
	carry	10169	counterpressing_recovery	8530
	foul_suffered	6545	dribbled_past_attempt	5996
	linkup_play	2294	sliding_tackle	1677
	progressive_run	1318	interception	880
	second_assist	29	assist	17
	opportunity	10	third_assist	6
	shot_block	2		
fairplay	loss	153	hand_pass	7
	recovery	3	interception	2
free_kick	free_kick_cross	890	free_kick_shot	265
	shot	265	shot_assist	163
	opportunity	148	carry	37
	assist	19	goal	12
	progressive_run	3	second_assist	3
	third_assist	3		
game_interruption	ball_out	19045	whistle	451
goal_kick	loss	254	carry	53
	second_assist	1		
goalkeeper_exit	carry	58		
infraction	foul	7068	yellow_card	1290
	penalty_foul	106	red_card	54
interception	recovery	11248	pass	7522
	short_or_medium_pass	6757	loss	6157
	head_pass	4974	forward_pass	3910
	counterpressing_recovery	3863	shot_block	1829
	progressive_pass	1750	lateral_pass	1648
	carry	872	long_pass	757
	pass_to_final_third	753	back_pass	658
	hand_pass	236	touch_in_box	193
	acceleration	137	progressive_run	127
	pass_to_penalty_area	99	under_pressure	67
	deep_completion	41	through_pass	35
	shot_assist	33	key_pass	15
	cross	9	assist	5
	smart_pass	5	second_assist	4
	third_assist	3	cross_blocked	2
	deep_completed_cross	1		

Primary Type	Secondary Type	Count	Secondary Type	Count
own_goal	interception	16	recovery	10
	shot_block	5	pass	3
	short_or_medium_pass	3	back_pass	2
	counterpressing_recovery	2	head_pass	2
	under_pressure	1		
pass	short_or_medium_pass	213353	lateral_pass	94132
	forward_pass	78835	progressive_pass	37744
	back_pass	36345	loss	28135
	pass_to_final_third	27661	long_pass	24686
	recovery	16784	under_pressure	11959
	pass_to_penalty_area	11034	head_pass	9893
	cross	8627	counterpressing_recovery	8051
	linkup_play	6370	touch_in_box	4440
	deep_completion	4029	shot_assist	3474
	through_pass	3139	deep_completed_cross	2954
	key_pass	1915	hand_pass	1672
	smart_pass	1576	cross_blocked	1462
	carry	931	assist	361
	second_assist	179	third_assist	129
	progressive_run	94		
penalty	shot	106	goal	80
	penalty_goal	80		
shot	opportunity	4841	touch_in_box	3941
	head_shot	1126	shot_after_corner	997
	goal	649	shot_after_throw_in	480
	shot_after_free_kick	316	interception	38
	shot_block	7	assist	2
shot_against	save	1720	save_with_reflex	1055
	conceded_goal	762	penalty_conceded_goal	80
	penalty_save	15		
throw_in	loss	976	carry	101
	shot_assist	46	progressive_run	6
	third_assist	3	second_assist	1
touch	carry	26262	progressive_run	5908
	loss	3843	touch_in_box	1549
	under_pressure	483	opportunity	238
	second_assist	5	third_assist	4
	assist	1		

B.2 B

Primary Type	Attribute	Description
pass	accurate	True/False
	angle	degrees
	height	high/low/blocked/null
	length	Euclidean distance in coordinates
	recipient	ID, name, position
	end Location	x-coordinate, y-coordinate
shot	body-part	right foot/ left foot/ head or other
	isGoal	true/false
	onTarget	true/false
	goalzone	glt/gt/grt/gl/gc/gr/glb/gb/grb/olt/ot/ort/or/orb/ol/olb
	xg	0-1
	postshotxg	0-1
	goalkeeperaction id	ID

Primary Type	Attribute	Description
	goalkeeper	ID
ground duel	opponent	ID, name, position
	duelType	offensive, defensive, dribble
	keptPossession	true/false for offensive and dribble, null for defensive
	progressedWithBall	true/false for offensive and dribble, null for defensive
	stoppedProgress	true/false for defensive, null for offensive and dribble
	recoveredPossession	true/false for defensive, null for offensive and dribble
	takeOn	true/false
	side	right/left for dribble and defensive, null for offensive
	relatedDuelID	ID
aerialDuel	opponent	ID, name, position, height
	firstTouch	false/true
	height	centimeters
	relatedDuelID	ID
infraction	Yellow Card	True/False
	Red Card	True/False
	Type	regular foul/ protest foul
	Opponent	ID, Name, Position
	end location	x-coordinate, y-coordinate

Table 11

Keypoint	Type 1 (X, Y)	Type 2 (X, Y)	Type 3 (X, Y)	Type 4 (X, Y)
0				
1				
2	(0, 37.65)	(0, 37.15)	(0, 36.15)	(0, 37.65)
3	(0, 30.35)	(0, 29.85)	(0, 28.85)	(0, 30.35)
4	(5.50, 43.15)	(5.50, 42.65)	(5.50, 41.65)	(5.50, 43.15)
5	(5.50, 24.85)	(5.50, 24.35)	(5.50, 23.35)	(5.50, 24.85)
6	(0, 43.15)	(0, 42.65)	(0, 41.65)	(0, 43.15)
7	(0, 24.85)	(0, 24.35)	(0, 23.35)	(0, 24.85)
8	(16.50, 54.15)	(16.50, 53.65)	(16.50, 52.65)	(16.50, 54.15)
9	(16.50, 13.85)	(16.50, 13.35)	(16.50, 12.35)	(16.50, 13.85)
10	(0, 54.15)	(0, 53.65)	(0, 52.65)	(0, 54.15)
11	(0, 13.85)	(0, 13.35)	(0, 12.35)	(0, 13.85)
12	(0, 68)	(0, 67)	(0, 65)	(0, 68)
13	(0, 0)	(0, 0)	(0, 0)	(0, 0)
14	(52.50, 68)	(52.50, 67)	(52.50, 65)	(55, 68)
15	(52.50, 0)	(52.50, 0)	(52.50, 0)	(55, 0)
16	(88.50, 54.15)	(88.50, 53.65)	(88.50, 52.65)	(93.50, 54.15)
17	(88.50, 13.85)	(88.50, 13.35)	(88.50, 12.35)	(93.50, 13.85)
18	(105, 54.15)	(105, 53.65)	(105, 52.65)	(110, 54.15)
19	(105, 13.85)	(105, 13.35)	(105, 12.35)	(110, 13.85)
20	(99.50, 43.15)	(99.50, 42.65)	(99.50, 41.65)	(104.50, 43.15)
21	(99.50, 24.85)	(99.50, 24.35)	(99.50, 23.35)	(104.50, 24.85)
22	(105, 43.15)	(105, 42.65)	(105, 41.65)	(110, 43.15)
23	(105, 24.85)	(105, 24.35)	(105, 23.35)	(110, 24.85)
24				
25				
26	(105, 30.35)	(105, 29.85)	(105, 28.85)	(110, 30.35)
27	(105, 37.65)	(105, 37.15)	(105, 36.15)	(110, 37.65)
28	(105, 68)	(105, 67)	(105, 65)	(110, 68)
29	(105, 0)	(105, 0)	(105, 0)	(110, 0)

Keypoint	Type 1 (X, Y)	Type 2 (X, Y)	Type 3 (X, Y)	Type 4 (X, Y)
30	(61.33, 31.54)	(61.33, 31.04)	(61.33, 30.04)	(63.83, 31.54)
31	(43.67, 31.54)	(43.67, 31.04)	(43.67, 30.04)	(46.17, 31.54)
32	(61.33, 36.46)	(61.33, 35.96)	(61.33, 34.96)	(63.83, 36.46)
33	(43.67, 36.46)	(43.67, 35.96)	(43.67, 34.96)	(46.17, 36.46)
34	(58.96, 27.54)	(58.96, 27.04)	(58.96, 26.04)	(61.46, 27.54)
35	(46.04, 27.54)	(46.04, 27.04)	(46.04, 26.04)	(48.54, 27.54)
36	(58.96, 40.46)	(58.96, 39.96)	(58.96, 38.96)	(61.46, 40.46)
37	(46.04, 40.46)	(46.04, 39.96)	(46.04, 38.96)	(48.54, 40.46)
38	(61.65, 34)	(61.65, 33.50)	(61.65, 32.50)	(64.15, 34)
39	(43.35, 34)	(43.35, 33.50)	(43.35, 32.50)	(45.85, 34)
40	(52.50, 24.85)	(52.50, 24.35)	(52.50, 23.35)	(55, 24.85)
41	(52.50, 43.15)	(52.50, 42.65)	(52.50, 41.65)	(55, 43.15)
42	(52.50, 34)	(52.50, 33.50)	(52.50, 32.50)	(55, 34)
43	(20.15, 34)	(20.15, 33.50)	(20.15, 32.50)	(20.15, 34)
44	(16.50, 41.31)	(16.50, 40.81)	(16.50, 39.81)	(16.50, 41.31)
45	(16.50, 26.69)	(16.50, 26.19)	(16.50, 25.19)	(16.50, 26.69)
46	(19.94, 32.05)	(19.94, 31.55)	(19.94, 30.55)	(19.94, 32.05)
47	(19.94, 35.95)	(19.94, 35.45)	(19.94, 34.45)	(19.94, 35.95)
48	(11, 34)	(11, 33.50)	(11, 32.50)	(11, 34)
49	(16.50, 34)	(16.50, 33.50)	(16.50, 32.50)	(16.50, 34)
50	(84.85, 34)	(84.85, 33.50)	(84.85, 32.50)	(89.85, 34)
51	(88.50, 41.31)	(88.50, 40.81)	(88.50, 39.81)	(93.50, 41.31)
52	(88.50, 26.69)	(88.50, 26.19)	(88.50, 25.19)	(93.50, 26.69)
53	(85.06, 32.05)	(85.06, 31.55)	(85.06, 30.55)	(90.06, 32.05)
54	(85.06, 35.95)	(85.06, 35.45)	(85.06, 34.45)	(90.06, 35.95)
55	(94, 34)	(94, 33.50)	(94, 32.50)	(99, 34)
56	(88.50, 34)	(88.50, 33.50)	(88.50, 32.50)	(93.50, 34)

Hidden	Layers	Dropout	Batch_Size	Learning Rate	Epoch	Total Training Time	Accuracy	Precision	Recall	F1_Score
128	4	0.3	32	0.001	35	2300.90	35.58	0.3603	0.3558	0.3423
128	3	0.5	32	0.0001	91	2232.37	35.80	0.4000	0.3580	0.3405
128	2	0.3	32	0.1	57	2176.36	37.46	0.2168	0.3746	0.2605
128	2	0.3	64	0.1	46	2075.55	34.92	0.2130	0.3492	0.2631
128	4	0.5	32	0.001	44	2252.65	36.57	0.3722	0.3657	0.3609
64	3	0.5	64	0.0001	93	1945.63	36.24	0.3852	0.3624	0.3336
128	2	0.5	32	0.0001	51	2123.33	37.57	0.3905	0.3757	0.3583
128	3	0.5	32	0.0001	60	2238.44	37.24	0.3902	0.3724	0.3513
256	2	0.5	32	0.0001	63	2448.51	38.34	0.3917	0.3834	0.3787
128	4	0.3	32	0.1	12	2322.88	29.50	0.2312	0.2950	0.2501
64	3	0.5	32	0.0001	85	2179.55	37.68	0.3837	0.3768	0.3743
256	2	0.5	64	0.01	82	2433.99	39.78	0.3929	0.3978	0.3558
128	3	0.3	32	0.01	40	2196.80	36.46	0.2428	0.3646	0.2890
64	4	0.3	64	0.01	44	1990.86	35.03	0.3876	0.3503	0.3274
128	3	0.3	64	0.01	46	2110.83	35.25	0.3456	0.3525	0.3236
256	3	0.5	32	0.001	43	2477.11	35.58	0.3647	0.3558	0.3433
64	3	0.5	64	0.001	41	1997.77	38.90	0.3937	0.3890	0.3764
128	3	0.3	64	0.001	43	2115.79	37.24	0.3822	0.3724	0.3581
64	3	0.3	64	0.01	23	1972.81	36.91	0.3753	0.3691	0.3678
256	4	0.3	64	0.001	48	2471.69	38.12	0.3643	0.3812	0.3634
64	4	0.3	64	0.001	28	1995.35	40.11	0.3869	0.4011	0.3808
64	4	0.5	64	0.0001	95	2000.23	34.59	0.3916	0.3459	0.3176
64	3	0.3	64	0.1	61	1952.04	38.23	0.2166	0.3823	0.2681
256	2	0.3	64	0.0001	61	2403.12	36.35	0.3906	0.3635	0.3479
256	4	0.5	32	0.001	33	2549.18	31.71	0.3455	0.3171	0.2688
64	2	0.3	64	0.1	75	1932.72	36.02	0.2090	0.3602	0.2562
256	4	0.5	64	0.01	23	2510.86	36.24	0.1969	0.3624	0.2438
64	4	0.5	32	0.01	22	2329.71	32.82	0.3085	0.3282	0.2980
256	2	0.5	64	0.001	28	2403.31	38.78	0.4004	0.3878	0.3608
256	3	0.5	32	0.01	40	2542.79	25.75	0.2706	0.2575	0.2434
256	3	0.3	32	0.0001	34	2452.56	36.24	0.5029	0.3624	0.3145
128	2	0.3	64	0.001	21	2102.13	40.00	0.3755	0.4000	0.3758

Table 12: Training results of base models during hyperparameter tuning

	Pred Shot	Pred Pass	Pred Duel	Pred Interception	Pred Touch	%
True Shot	39	3	1	7	1	**76.47%**
True Pass	14	32	24	7	1	**19.75%**
True Duel	29	12	35	6	24	**33.02%**
True Interception	16	5	7	6	8	**14.29%**
True Touch	10	18	21	4	38	**41.76%**
%	**36.11%**	**45.71%**	**39.77%**	**13.64%**	**26.76%**	

	Pred Shot	Pred Pass	Pred Duel	Pred Interception	Pred Touch	%
True Shot	34	4	3	3	0	**77.27%**
True Pass	17	47	15	4	93	**26.70%**
True Duel	17	23	26	3	34	**25.24%**
True Interception	7	14	6	3	19	**6.12%**
True Touch	10	20	8	0	42	**52.50%**
%	**40.00%**	**43.52%**	**44.83%**	**23.08%**	**22.34%**	

	Pred Shot	Pred Pass	Pred Duel	Pred Interception	Pred Touch	%
True Shot	27	2	3	9	0	**65.85%**
True Pass	20	108	21	19	2	**63.53%**
True Duel	15	39	28	4	3	**31.46%**
True Interception	9	21	14	7	0	**13.73%**
True Touch	7	70	18	4	2	**1.98%**
%	**34.62%**	**45.00%**	**33.33%**	**16.28%**	**28.57%**	

Figure 50: Confusion matrices on primary classification

	Pred Accurate	Pred Not Accurate
True Accurate	0	15
True Not Accurate	0	31

	Pred Accurate	Pred Not Accurate
True Accurate	0	31
True Not Accurate	0	38

	Pred Accurate	Pred Not Accurate
True Accurate	15	4
True Not Accurate	24	14

Figure 51: Prediction on Accuracy of Cross

	Pred Accurate	Pred Not Accurate
True Accurate	101	1
True Not Accurate	10	1

	Pred Accurate	Pred Not Accurate
True Accurate	86	1
True Not Accurate	13	0

	Pred Accurate	Pred Not Accurate
True Accurate	85	1
True Not Accurate	18	2

Figure 52: Prediction on Accuracy of Pass

	Pred Long	Pred Short/Medium
True Long	14	0
True Short/Medium	99	0

	Pred Long	Pred Short/Medium
True Long	12	3
True Short/Medium	10	75

	Pred Long	Pred Short/Medium
True Long	12	10
True Short/Medium	8	76

Figure 53: Prediction on Distance of Pass

	Pred on Target	Pred Not on Target
True on Target	1	17
True Not on Target	1	32

	Pred on Target	Pred Not on Target
True on Target	11	5
True Not on Target	18	10

	Pred on Target	Pred Not on Target
True on Target	2	15
True Not on Target	4	20

Figure 54: Prediction on accuracy of shot

	Pred Through	Pred Not Through
True Long	0	32
True Short/Medium	0	81

	Pred Through	Pred Not Through
True Through	8	29
True Not Through	10	53

	Pred Through	Pred Not Through
True Through	23	21
True Not Through	26	36

Figure 55: Prediction on if a pass is a through pass

	Pred Progressive	Pred Not Progressive
True Progressive	0	20
True Not Progressive	0	93

	Pred Progressive	Pred Not Progressive
True Progressive	2	20
True Not Progressive	9	69

	Pred Progressive	Pred Not Progressive
True Progressive	12	12
True Not Progressive	38	44

Figure 56: Prediction on if Pass is progressive

	Pred Pass	Pred Cross
True Pass	103	10
True Cross	4	42

	Pred Pass	Pred Cross
True Pass	93	7
True Cross	2	67

	Pred Pass	Pred Cross
True Pass	98	8
True Cross	7	50

Figure 57: Prediction on Pass or Cross

	Pred Forward	Pred Backward	Pred Lateral
True Forward	1	0	4
True Backward	1	1	1
True Lateral	5	0	33

	Pred Forward	Pred Backward	Pred Lateral
True Forward	0	0	3
True Backward	2	0	6
True Lateral	8	0	50

	Pred Forward	Pred Backward	Pred Lateral
True Forward	1	1	0
True Backward	1	2	2
True Lateral	9	22	19

Figure 58: Prediction on the direction of the Cross

	Pred Forward	Pred Backward	Pred Lateral
True Forward	5	0	2
True Backward	3	0	17
True Lateral	4	0	57

	Pred Forward	Pred Backward	Pred Lateral
True Forward	28	4	5
True Backward	7	4	1
True Lateral	36	7	8

	Pred Forward	Pred Backward	Pred Lateral
True Forward	29	6	9
True Backward	3	4	2
True Lateral	33	8	12

Figure 59: Prediction on the direction of the Pass

	Pred right foot	Pred left foot	Pred head/other
True right foot	7	12	0
True left foot	12	13	0
True head/other	4	3	0

	Pred right foot	Pred left foot	Pred head/other
True right foot	4	6	5
True left foot	7	3	10
True head/other	0	5	4

	Pred right foot	Pred left foot	Pred head/other
True right foot	0	21	1
True left foot	0	16	1
True head/other	0	2	0

Figure 60: Prediction on the body part the shot is taken with

	Pred Right	Pred Left	Pred Center
True Right	19	3	0
True Left	18	1	1
True Center	1	0	3

	Pred Right	Pred Left	Pred Center
True Right	14	11	5
True Left	14	12	8
True Center	2	1	2

	Pred Right	Pred Left	Pred Center
True Right	23	2	4
True Left	0	19	3
True Center	3	2	1

Figure 61: Prediction on the Flank of the Cross

	Pred Aerial	Pred Loose	Pred Ground
True Aerial	25	7	0
True Loose	9	32	0
True Ground	25	8	0

	Pred Aerial	Pred Loose	Pred Ground
True Aerial	16	8	7
True Loose	13	21	2
True Ground	20	4	12

	Pred Aerial	Pred Loose	Pred Ground
True Aerial	19	9	0
True Loose	11	18	0
True Ground	22	10	0

Figure 62: Prediction on the kind of duel

C FIGURES

Figure 63: Example of event data as presented in the event data package. This event information is regarding the action of a pass, giving detailed information about the pass