FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Player Tracking System for Sports Events based on Computer Vision

Murilo de Mendonça Couceiro

Mestrado em Engenharia Informática

Supervisor: João Mendes-Moreira

Second Supervisor: Tiago Mendes-Neves

Sunday 31st July, 2022

Player Tracking System for Sports Events based on Computer Vision

Murilo de Mendonça Couceiro

Mestrado em Engenharia Informática

Approved in oral examination by the comitee:

Chair: Pedro Costa
External Examiner: Carlos Ferreira
Supervisor: João Mendes-Moreira

Sunday 31st July, 2022

Abstract

The broadcast of audio-video sports content is a field with increasingly larger audiences demanding higher quality content and involvement. This growth creates the necessity to develop more content to engage the users and keep this trend. Otherwise, it may stall or even revert. Therefore, enhancing the user experience, engagement, and involvement during live sports event broadcasts is of utmost importance.

This dissertation proposes a system that assesses multi-camera video content and provides visual representations to enhance the user experience on live sports broadcasts, resorting to Computer Vision techniques and Deep Learning algorithms. More specifically, the dissertation encompasses the definition and implementation of field registration, object detection and tracking tasks. Furthermore, a novel dataset combining several video sources, focusing on football sports events, was created and used for analysis and metadata extraction. This dataset reflects the defined acquisition protocol with two static cameras covering the entire field.

In particular, the proposed solution can detect and track players with acceptable precision using state-of-the-art methods, like YOLOv5 and DeepSORT. Moreover, resorting to the K-means unsupervised learning technique, the system provides team segmentation based on the colour of the players' kits. Finally, after collecting and fusing the metadata from both cameras, a series of visual representations on a top-view field is created regarding the extracted information, enabling broadcast enrichment and increasing user experience. This extracted information is stored as it can also benefit other more complex sports-related applications. The solution can be deployed in a cloud environment simplifying its access and utilisation.

This dissertation is framed in the H2020 DataCloud project, focusing on the MOGSports business case.

Keywords: Computer Vision, Deep Learning, Player Tracking, Broadcast Enrichment

Resumo

A transmissão de conteúdos audiovisuais desportivos é um campo com audiências cada vez maiores que exigem conteúdos e envolvimento de maior qualidade. Este crescimento cria a necessidade de desenvolver mais conteúdos para envolver os utilizadores e manter esta tendência. Caso contrário, pode estagnar ou mesmo diminuir. Por conseguinte, é da maior importância melhorar a experiência e o envolvimento dos utilizadores durante as transmissões em directo de eventos desportivos.

Esta dissertação propõe um sistema que avalia o video-conteúdo de várias câmaras e fornece representações visuais para melhorar a experiência do utilizador em transmissões desportivas ao vivo, utilizando técnicas de Visão por Computador e algoritmos de Aprendizagem Profunda. Mais especificamente, a dissertação engloba a definição e implementação das tarefas da estimativa da homografia e detecção e seguimento de objectos. Além disso, foi criado e utilizado para análise e extracção de metadados um novo conjunto de dados que combina várias fontes de vídeo, centrando-se em eventos desportivos de futebol. Este conjunto de dados reflecte o protocolo de aquisição definido para este contexto que inclui duas câmaras estáticas que cobrem todo o campo.

Em particular, a solução proposta consegue detectar e seguir jogadores com precisão aceitável utilizando métodos, como YOLOv5 e DeepSORT. Além disso, recorrendo ao algoritmo de aprendizagem não supervisionada K-means, o sistema segmenta a equipa dos jogadores detetados com base na cor dos seus equipamentos. Finalmente, após a recolha e fusão dos metadados extraídos de ambas as câmaras, é criada uma série de representações visuais numa visão aérea de um campo relativamente à informação extraída, permitindo assim o enriquecimento de uma transmissão e aumentando a experiência do utilizador. Esta informação extraída é armazenada uma vez que também pode beneficiar outras aplicações mais complexas relacionadas com o desporto. A solução pode ser implementada num ambiente *Cloud*, o que simplifica o seu acesso e utilização.

Esta dissertação está enquadrada no projecto H2020 DataCloud, centrando-se no *business case* MOGSports.

Palavras-chave: Visão por Computador, Aprendizagem Profunda, Seguimento de Jogadores, Enriquecimento para Transmissões

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores, professor João Moreira e Tiago Neves pela disponibilidade e ajuda durante o desenvolvimento desta dissertação. Foram, com certeza, um grande apoio para o avanço deste trabalho. Queria deixar uma nota de agradecimento mais pessoal para o Tiago, que pela proximidade ao tema me ajudou ainda mais e também por tentar envolver-me ao máximo nesta área que tanto nos interessa o que me motivou e ajudou bastante.

À MOG deixo uma palavra de agradecimento pelas condições e oportunidade de desenvolver este projeto, em especial à Inês Rito Lima pela disponibilidade e orientação às vezes para além do necessário, mas também ao Alexandre Ulisses, ao Vasco Filipe e à Claúdia Marinho por garantirem que tudo corria da melhor forma no projeto.

Aos restantes elementos da Academy, Helena Guedes, Luís Fernandes, Mafalda Santos, Matilde Alexandre, Nuno Ferreira e Paulo Araújo, quero também deixar o meu agradecimento pelo companheirismo e amizade que construímos durante este período. À Matilde Alexandre agradeço especialmente pela ajuda na gravação e edição dos vídeos e de muitos outros pedidos de ajuda.

Aos meus pais, irmão e avós por me permitirem traçar este caminho e por toda a força e apoio incondicional que me dão. Se estou no lugar onde estou, maior parte deve-se a eles.

Aos meus amigos que sempre me apoiaram e torcem pelo meu sucesso e felicidade.

À Isabel por ter sido o meu maior apoio nestes dois últimos anos, por ser o escape sempre disponível, pela sua preocupação e motivação constante. O meu maior obrigado por estares sempre presente e por fazeres parte desta etapa.

Gostaria de terminar com um agradecimento a todas as pessoas aqui não mencionadas e que fazem parte da minha vida, que de certeza que me ajudaram de uma maneira ou de outra e que por isso estou muito grato.

Murilo Couceiro

"Sometimes you just have to jump out the window
and grow wings on the way down."

Ray Bradburry

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Objectives . . . 2
- 1.3 Contributions . . . 3
- 1.4 Structure . . . 3

2 Computer Vision Techniques Applied to Sports **5**
- 2.1 Computer Vision . . . 5
  - 2.1.1 YOLO . . . 9
  - 2.1.2 Transfer Learning . . . 18
- 2.2 Computer Vision in Football . . . 20
  - 2.2.1 Camera Setup . . . 20
  - 2.2.2 Field Detection . . . 22
  - 2.2.3 Player and Ball Detection . . . 32
  - 2.2.4 Player and Ball Tracking . . . 35
- 2.3 Market Analysis . . . 40

3 Data Acquisition Protocol **43**
- 3.1 First Acquisition (Training Session) . . . 43
- 3.2 Second Acquisition (Training Session) . . . 44
- 3.3 Final Acquisition (Official Match) . . . 46

4 Extracting Tracking Data for Broadcast Enhancement with Computer Vision **48**
- 4.1 MOGSports . . . 48
- 4.2 Homography Estimation . . . 50
  - 4.2.1 Automatic Homography Estimation . . . 51
  - 4.2.2 Manual Homography Computation . . . 53
- 4.3 Player Detection . . . 55
- 4.4 Player Tracking . . . 58
- 4.5 Team Segmentation . . . 60
- 4.6 Metadata Fusion . . . 61

5 Results **64**
- 5.1 Visual Representations . . . 64
- 5.2 System Evaluation . . . 64
- 5.3 Discussion . . . 69

6 Conclusions and Future Work **74**
- 6.1 Future Work . . . 75

A Player Tracking System for Sports Broadcast Enrichment **88**

List of Figures

2.1 Object detectors architecture [18] . . . 8
2.2 Object detectors head differences . . . 8
2.3 IoU generic formula and representation . . . 9
2.4 YOLO detection grid model [79] . . . 11
2.5 Top view of the football field . . . 26
2.6 IoU metric representation [25] . . . 31
2.7 MOT tracking flows. . . . 35
2.8 Graph-based tracking representation . . . 37

3.1 First acquisition image examples . . . 43
3.2 Example of the cameras used . . . 44
3.3 Protocol one illustration . . . 45
3.4 Real overlap from the second acquistion images . . . 45
3.5 Protocol two illustration . . . 45
3.6 Images from the the final acquisition match footage . . . 46
3.7 Real overlap from the match images . . . 47

4.1 MOGSports pipeline . . . 48
4.2 Two-GAN model architecture [21] . . . 51
4.3 Two-GAN results for the first dataset images . . . 52
4.4 Two-GAN result example . . . 53
4.5 Feature extraction and matching example . . . 54
4.6 Match image transformation . . . 54
4.7 Homography warp example. . . . 55
4.8 Detections with the first acquired dataset . . . 56
4.9 Left camera detections and respective top-view transformions . . . 57
4.10 Occlusion examples . . . 58
4.11 Different kit colours . . . 60
4.12 Bonding boxes' color value . . . 61
4.13 Metadata Fusion strategy . . . 62
4.14 Metadata Fusion steps . . . 63

5.1 Visual Representations . . . 65
5.2 Number of tracks per frame - single-camera . . . 66
5.3 Left camera tracking results . . . 67
5.4 Right camera tracking results . . . 68
5.5 Setup occlusion handling advantages . . . 69
5.6 Number of tracks per frame . . . 70
5.7 Team Segmentation example . . . 71

5.8 Mean number of colours attributed per frame 72
5.9 Team Segmentation at the start of the second half 73
5.10 Top-view representation of both cameras' detections 73

List of Tables

2.1 Object detectors performance comparison 10
2.2 Field Registration methods comparison with IoU metric 31

Abbreviations

AI	Artificial Intelligence
AoC	Area of Coverage
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DLT	Direct Linear Transform
GAN	Generative Adversarial Network
HDR	High Dynamic Range
HT	Hough Transform
IoU	Intersection-over-Union
KF	Kalman Filter
MOT	Multi-Object Tracking
ORB	Oriented FAST and Rotated BRIEF
RANSAC	Random sample consensus
R-CNN	Region-based CNN
ReLU	Rectified Linear Unit
SIFT	Scale Invariant Feature Transform
SORT	Simple Online and Realtime Tracking
SSD	Single Shot Detector
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
YOLO	You Only Look Once
mAP	Mean Average Precision

Chapter 1

Introduction

1.1 Context and Motivation

Currently, football is the most watched sport in the world, with more than 265 million players in more than 200 countries [28]. Likewise, the broadcast of audio-video football content is a field with increasingly larger audiences that demand high quality and high involvement. The global football market was valued at $1,883.6 million in 2019. According to the global football market analysis, it is estimated to reach $3,712.7 million by 2027 [6]. These market values indicate immediately that this market is significant and needs support to keep growing. The market's and audiences' growth are also allied to these broadcasters' data growth, which must be put into good use. As Moeslund et al. [65] stated, "Sports is said to be the social glue of society. It allows people to interact irrespective of their social status or age. With the rise of the mass media, a significant quantity of resources has been channelled into sports in order to improve understanding, performance and presentation". One way to create this extra content and enhance the broadcast users' experience is to endorse new visual content from what is happening in the match. Furthermore, manual data creation describing details in sports events is tiresome, unreliable and, in many cases, simply unrealistic. Therefore, technology is becoming more and more crucial as the demand for improved understanding, performance, and presentation increases [65]. Meeting this demand is crucial for the broadcast audiences' growth trend as spectators expect these extras.

Computer Vision's goal is to extract useful information from images. It has made advances over almost four decades, but it has been a challenging task for thousands of minds that have investigated this area. Part of the problem is the complexity of the visual data [75]. We live in a world with a tremendous amount of data, and a lot of it is visual, e.g. images and videos, but as stated, the data is not easy to interpret for a computer as it is for a human, creating problems for this area of investigation. However, Computer Vision state-of-the-art already have the capacity and techniques to approach the video data from football events (matches or training sessions) and be able to extract valuable information. Moreover, it is already used in several applications related to sports, as Section 2 describes. Some of the best-known current application areas are in sports analysis for broadcast, for instance, showing players' position or the ball as 3D models to allow

the locations or trajectories to be explored in detail by a TV presenter [91].

The relevant information in a football match is related to the players and ball. Their positional information allied with its movement is named Tracking data. Tracking data intends to provide insights into players' positions and movements and create additional value, providing possible statistics and content enhancement. Object Detection can extract this positional information by processing the images. As the objective is to assist a broadcast, the object detection algorithm must be fast, working at least in near real-time. Hence, an algorithm like YOLO, which is fast and reasonably accurate, is a good fit for this application. Moreover, to add the movement information, the players can be tracked using Multiple Object Tracking (MOT) algorithms to identify or re-identify the players following them. Likewise, this step must be as fast as possible.

This dissertation is framed in the European H2020 DataCloud project that focuses on the computer continuum of Big Data. This project's overall vision is to create a novel paradigm for Big Data pipeline processing over heterogeneous resources encompassing the Computing Continuum, covering the complete lifecycle of managing Big Data pipelines. DataCloud has a diverse and relevant toolbox set that discovers, defines, simulates, deploys, and adapts Big Data pipelines. The project's consortium has 11 partners from 8 countries with the goal of lowering the technological entry barriers for incorporating Big Data pipelines in businesses in an accessible way. DataCloud's plan is validated through its various business cases. The one in focus is Automatic Live Sports Content Annotation, denominated MOGSports.

The MOGSports business case focuses on the media and entertainment business to enhance the user experience, engagement, and involvement during live sports broadcasts, to attend to the need within the broadcasting industry for automated video annotation and metadata extraction tools as mentioned above. As this is the focus of this dissertation, it is further discussed in Section 4.1

1.2 Objectives

This dissertation and MOGSports' goal is to enrich live sports broadcasts by creating graphical representations and overlays. In order to achieve this final objective, the development can be divided into several consequent sub-objectives, which create the desired information when integrated. Hence, this dissertation's different underlying objectives are:

- Robust Field Registration Methodology;
- Player and Ball Detection;
- Player Tracking;
- Metadata Fusion;
- Graphical Representations.

The definition and implementation of a field registration methodology are crucial as it will provide the mapping of the images and the objects comprised in them to a different view (e.g. 2D

top view see Figure 2.5) using a homography matrix allowing the game to be seen as a whole in another perspective. After this, another essential aspect is the Player and Ball detection that will retrieve their position. By joining these steps, it is possible to create metadata (i.e. the players and ball coordinates) in different views. Adding a sense of motion can create more valuable information and upgrade the previous positional information. For this purpose, a Player Tracking method is crucial. Furthermore, as there are different cameras and consequent metadata from different sources, there must be a Metadata Fusion module that associates the extracted metadata. Finally, the fused cameras' extracted metadata can be turned into various graphical representations, like a top view of the positional game where we can see the players and ball or heatmaps that present the most crowded field areas. Moreover, the metadata extracted can be stored and used for other applications, following the Big Data continuum proposed in DataCloud. Furthermore, following DataCloud's context, the solution should be deployed in a cloud environment, requiring extra effort when integrating the various modules.

Ultimately, MOGSports aims to develop an Event Detection module, which aims to create even more data to enrich the experience mentioned. The extraction of these events can be achieved with the metadata created or with dedicated Computer Vision algorithms. Although this is the end goal of MOGSports, this dissertation focuses on laying the groundwork to enable event extraction using computer vision in the future.

1.3 Contributions

The work developed in this dissertation aimed to develop the MOGSports business case. In the pursuit of this objective, this dissertation contributed to the state-of-the-art when proposing the integration of a manual homography computation, YOLOv5 as the players' detector and DeepSORT as the tracking algorithm toward a football environment. Also, a K-means algorithm was used to create value in the visual representations where it is possible to distinguish the players' teams. Furthermore, to test the developments and algorithms, this dissertation presents a novel dataset with around 40 minutes of a football match. It uses a two-camera system, using the defined novel protocol that creates visual representations to enhance it. Moreover, this system metadata and video data can be an important contribution to the sports-related data mining community as they are both critical resources for its applications, e.g. the metadata can be helpful for positional-related applications, and the video data for unlimited further research options. Also, a paper was written, presented in Appendix A, and submitted to the workshop, "9th Workshop on Machine Learning and Data Mining for Sports Analytics", hosted by the rank A conference ECML-PKDD.

1.4 Structure

Along with the present chapter, the dissertation is organised into five chapters:

- Chapter 2 presents the state-of-the-art description of the topics addressed in this dissertation.

- Chapter 3 illustrates the established data acquisition protocol and some other experimental ones to capture the football matches (or training sessions) and extract the required information at all times.
- Chapter 4 describes the experimental developments made when developing the proposed solution while seeking the objective of retrieving Tracking data and the corresponding visual representations to enhance the broadcast user's experience. First, the pipeline and data flow introduction is provided, and then the developments are demonstrated in the related modules.
- Chapter 5 presents and discusses the results obtained with the proposed solution using the work developed in the previous chapters.
- Chapter 6 summarises the work done for this dissertation and proposes further developments.

Chapter 2

Computer Vision Techniques Applied to Sports

This chapter introduces Computer Vision (CV) and its history. In the scope of this dissertation problem, this chapter also explores different football-related fields of research. In particular, it describes and compares different problem-related CV algorithms and data processing techniques. Finally, this chapter presents a market analysis to assess existing technologies that will compete with MOGSports.

2.1 Computer Vision

The first experiments of CV were conducted in the 1950s and continued to be studied until they were solidified in the early 1970s [9, 42]. The 1970s were marked by frequently mistaking the areas image processing and CV. There is a difference, and it is focused on the goal of each area, "Computer Vision aims to have the computer understand what is contained in an image, where image processing aims to simply recover the 3D structure of the image."[42]. This relativity drove the techniques used in image processing into CV [88]. The 1970s also saw some scene-understanding attempts by extracting edges and inferring 3D structures of an object or its topological structure. The area of edge detection was an active research area, as well. Another area being studied was the 3D modelling of non-polyhedral objects, using generalised cylinders to model the parts of the object and then arrange them hierarchically [42, 88].

The 1980s saw attention focused on more sophisticated mathematical techniques for performing quantitative image and scene analysis. Image blending and coarse-to-fine correspondence search improved by methods like image pyramids. Shaping techniques were also highlighted, with work in shape from shading, shade from texture, and shape from focus. These studies also helped improve edge and contour detection, and 3D models [88, 42].

In the decade of 90, the work mentioned earlier continued to be explored, some more than others. Problems such as motion, tracking, and image segmentation were explored extensively. Despite this, the most notable development in CV during this decade was the interaction with

computer graphics, leading to techniques to manipulate real-world imagery directly to create animations and models for graphics applications. Also, the automatic creation of realistic 3D models from a collection of images was being introduced [88, 42]. The new century has made many advances in CV algorithms, allied to machine learning and hardware advances.

The 00s decade continued to see a deepening interplay between CV and computer graphics. Topics like image stitching, new rendering techniques, and High Dynamic Range (HDR) image capture through exposure bracketing, re-launched as computational photography to acknowledge the increased use of such techniques in everyday digital photography. Other computational photography techniques are texture synthesis, quilting, and inpainting which re-combine input image samples to produce new photographs. Another area in vogue was feature-based techniques, combined with learning, for general recognition. Recognition was also targeted using contours and region segmentation. The trend that now dominates the area is the application of sophisticated machine learning techniques to CV problems. The increase of labelled data available helped these kinds of methods to grow and be effective.

There have been advances in the area, especially with Deep Learning (DL). Today we are in the era of DL. Citing Voulodimos et al. [94] "Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information, thus implicitly capturing intricate structures of large-scale data". The application of DL techniques helps solve many CV problems faster and more precisely, which raises attention and interest. Increasing data is a given fact, and these algorithms can only benefit from this. Here, the hardware also takes part, with the empowerment of parallel GPU computing, which enabled the transition from CPU-based to the much faster GPU-based training [94]. The number of current techniques demonstrates that a lot of commitment and investigation led to these results. CV is a trend, and more and more studies will be conducted to improve this area.

Briefly, when trying to perform CV tasks, the starting point is the image features that carry important information and the need to extract them to generate new important information. Machine learning, especially the subfield DL, allows the interpretation and conclusion to be made automatically.

Before DL, CV tasks required a lot of coding and manual effort to define the features extracted from images, with little automation involved. With DL methods such as Convolutional Neural Networks (CNN), much of that work related to the extracted features can be inferred automatically from data. Even though many features can be extracted automatically in the CNN framework for different tasks, manual feature extraction can still be helpful. It can be to either augment the automatically extracted features or to perform other tasks such as temporal segmentation of video or detection of active players [40]. DL uses various methods: neural networks, hierarchical probabilistic models, and various unsupervised and supervised feature learning algorithms. However, the method with the most common usage is Neural Networks (NN), especially CNNs. When using CNNs, the possible applications are manifold as this type of network's features are vital for image-related problems.

The CNN consists of several convolution layers, the activation and pooling layers, and one or more fully connected layers at the network's last layers. The convolutional layer takes a map of the features (or input image) and convolves it with a set of learned parameters resulting in a new two-dimensional feature map. The sets of learned parameters are called filters or kernels, which have a fixed-size square matrix, e.g. 3x3. The filter consists of real values that represent the weights that need to be learned so that the output feature map contains valuable information such as a particular shape, colour, edges, and textures to give the network the ability to compress the image's content in this reduced feature representation. Furthermore, according to the convolution operation, these filters' elements are connected to a local area of the input neurons instead of a full connection. This means that the number of parameters to be learned is reduced significantly, and a network can grow deeper with fewer parameters.

The pooling layer is usually inserted between two successive convolution layers to reduce a map resolution and increase spatial invariance or network insensitivity to minor shifts such as rotations and translations of features in the image. Some pooling methods are arithmetic mean and maximum, mixed pooling, stochastic pooling, and spatial pyramid pooling.

The activation function defines the output of a node given an input or set of inputs. In its simplest form, this function is binary and represents neurons' action potential by propagating its output value or stopping it. There is a broad range of univariate functions of a linear combination of the input variables acting as CNN activation functions, such as linear activation functions, jump functions, and sigmoidal functions. Recent architectures use activation functions behind each layer, like Rectified Linear Unit (ReLU) [55].

Another critical aspect of the CNN architecture is the loss function. Depending on the problem, a loss function must be defined so that the model can evaluate how the algorithm models the data. It performs this evaluation by computing the distance between the current output of the algorithm and the expected output. For CNN development, an important aspect is the network hyperparameters. These are the number of filters, the size of the kernels, the padding and the stride's size.

This dissertation's leading CV problem is object detection, which also uses CNNs as its core. An object detection model usually comprises three parts: a backbone, a neck, and a head (see Figure 2.1). The backbone is responsible for receiving an image for input and compressing features down through a CNN. The neck is a feature aggregator that mixes and combines the features from the backbone to feed into the network's head. The head is where the detection step happens, including localisation and classification for each bounding box.

Selecting the structure of the CNN network for feature extraction (backbone) plays a critical role in object detection because the number of parameters and types of layers directly affect the memory requirements, speed, and performance of the detector [40].

More specifically, there are currently two groups for object detection: two-stage and one-stage object detectors. The difference between the two is in the model head - object localisation and image classification - where one-stagers perform them as one and the two-stagers separate them. The most representative for each group are Faster R-CNN (Region-based CNN) and Mask R-CNN for two-stagers, and YOLO and SSD for one-stagers. Both have advantages and disadvantages,

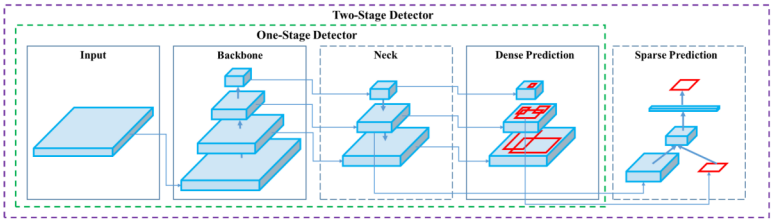

Figure 2.1: Object detectors architecture [18]

considering the objectives of their application. However, the known facts are that two-stage detectors provide an accurate localisation and identification of the objects but are relatively slow, whereas one-stage detectors can deliver high processing speed results with lower accuracy. The first stage of two-stage detectors is the region proposal generator, which proposes regions with possible objects. The second stage extracts the features from the proposed regions and executes the image classification and regression for the bounding boxes position. The one-stage detectors propose the bounding box locations and classifications directly without the generator (Figure 2.2).

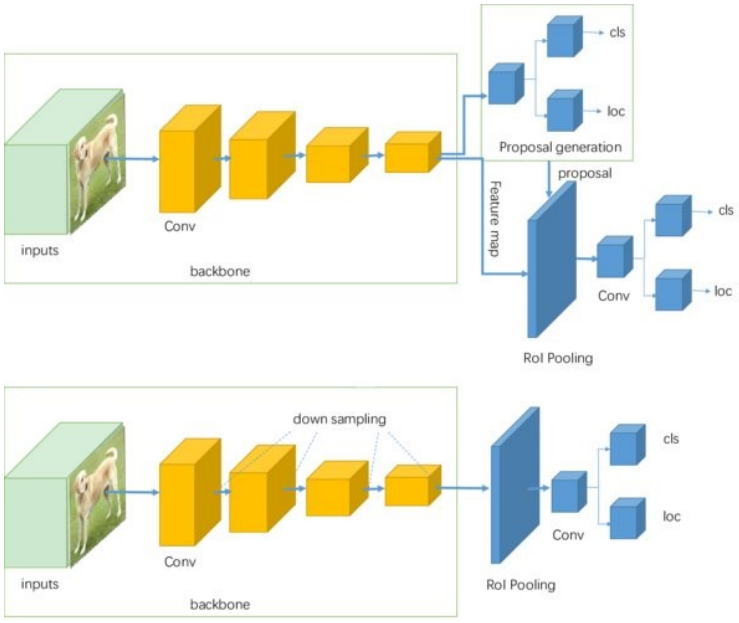

Figure 2.2: Object detectors head differences

The metric mean Average Precision (mAP) is widely used to evaluate these different object detection algorithms. In CV, mAP evaluates the bounding boxes' location and classification. Hence, the overlap between predicted and ground-truth bounding boxes is measured for each bounding box. This overlap can be calculated with the Intersection over Union (IoU) metric, which describes this exact overlap. Figure 2.3 illustrates the metric formula.

Another two important metrics related to mAP are Precision and Recall. Precision measures the fraction of correct predictions from all predictions, and Recall measures the fraction of positive instances obtained. For object detection evaluation, Precision and Recall use the IoU as a thresh-

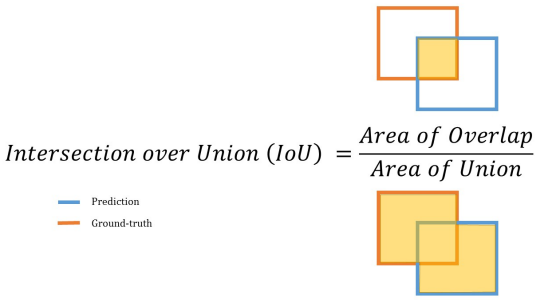

Figure 2.3: IoU generic formula and representation

old. For instance, if the threshold is 0.5 value and the bounding box prediction's IoU is bigger, it is classified as a correct prediction (True Positive); otherwise, it is a wrong prediction (False Positive).

The general definition for Average Precision (AP) is finding the area under the precision-recall curve and summarising it into a value. mAP is the average of AP. In some contexts, AP is calculated for each class and averaged to get the mAP; however, in the COCO challenge evaluation, there is no difference between AP and mAP. This means that the higher the score, the more accurate the model.

Since this dissertation relies on the speed of its algorithms, the one-stage object detectors are the most likely to be used. Among them, the YOLO is the one that is more used in the CV community. Section 2.1.1 describes how the YOLO algorithm works and how it has been redesigned throughout the years.

2.1.1 YOLO

In 2015, Redmon et al. [79] introduced an object detection system that performed all the essential stages to detect an object using a single neural network for the first time, the YOLO (You Only Look Once) algorithm. It reframed the object detection methodology into a single regression problem, straight from image pixels to bounding box coordinates and class probabilities, creating a much more simpler and faster way to train the model. As mentioned, it removed the previous region proposal step and does this by an ingenious methodology which divides the entire image into a grid and predicts bounding boxes at each grid cell.

Over the next five years, the YOLO algorithm was upgraded several times in different versions, with many of the most innovative ideas coming out of the computer vision research community. The first three versions were developed by the author of the YOLO algorithm, Joseph Redmon. The following were from different authors. YOLOv4 and YOLOv5 were released very close to each other, only 28 days apart. Despite using different methods and introducing advances, until YOLOv5, the methods followed a similar structure. Differently, the YOLOR and YOLOX algorithms have tried to incorporate the latest advances in Object Detectors in a more diverse manner. YOLOv5 is very close to the YOLOv3. A significant part of the work is converting the YOLOv3 algorithm to the PyTorch framework. Hence, this leads some to go against the YOLOv5

name as it does not have large contributions to the YOLO family. However, as Section 2.1.1.5 describes, YOLOv5 has other important contributions that help this algorithm improve the state-of-the-art. Table 2.1 reflects the state-of-the-art methods for the object detection task. It compares their performance related to the mAP and inference time. Different datasets are used; however, comparing the ones that use the same is still helpful. For the COCO dataset, the mAP metric is the $mAP_{0.5}$, which assumes the IoU threshold of 0.5 for classification. Using the COCO dataset, YOLOR has the higher real-time performance; however, YOLOv5 is the best algorithm regarding the accuracy/speed ratio.

Name	Dataset	mAP	Inferences (Frames) per Second
R-CNN	Pascal VOC	66%	0.02
Fast R-CNN	Pascal VOC	68.8%	0.5
Faster R-CNN	Pascal VOC	78.9%	7
YOLOv1	Pascal VOC	63.4%	45
YOLOv2	Pascal VOC	78.6%	67
SSD	Pascal VOC	74.3%	59
RetinaNet	COCO	61.1%	90
YOLOv3	COCO	51.3%	45
YOLOv4	COCO	65.7%	62
YOLOv5m	COCO	64.1%	**121**
YOLOR	COCO	**73.3%**	30
YOLOX-L	COCO	68.5%	57

Table 2.1: Object detectors performance comparison based [8]

2.1.1.1 YOLOv1

The main idea of YOLOv1 [79] is to apply a grid cell with the size of $S \times S$ (7×7 default) into an image. If the centre of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes with their parameters, x, y, h, w and the confidence scores for the boxes. The x and y represent the centre of the bounding box relative to the bounds of the grid cell, w the width and h the height. This confidence score reflects the presence or absence of an object in the bounding box. Formally it defines confidence as $Pr(Object) * IoU^{truth}_{pred}$. If no object exists in that cell, the confidence scores should be zero. Otherwise, the confidence score aims to equal the IoU between the predicted box and the ground truth [79].

Each grid cell also predicts C conditional class probabilities, $Pr(Class_i|Object)$, where C is the number of classes the network is trained. For YOLO was 20. These probabilities are conditioned on the grid cell containing an object.

This approach creates duplicate predictions because each cell predicts a bounding box. Hence, it also uses non-maximal suppression to clean the bounding boxes that contain the same object. Briefly, non-maximal supression chooses the highest probability score and suppress all the bounding boxes that have the largest IoU with it, then it repeats these stages in a loop until the final bounding boxes are obtained.

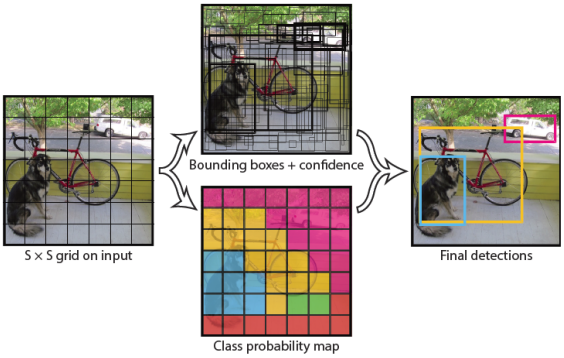

Figure 2.4: YOLO detection grid model [79]

The first version of YOLO network architecture was inspired by the GoogleNet model for image classification. It has 24 convolutional layers followed by two fully connected layers. The final layer uses a Linear activation function, and others use Leaky Rectified Linear Unit (leaky ReLU) activation. They have called it Darknet. The convolution layers in the front extract features, and the full connection layers in the back calculates the output probabilities and coordinates.

YOLO has difficulties in detecting small objects that appear in groups. This is because they may fall into one grid and one grid can only detect one object. It also has difficulties detecting objects with unusual aspect ratios because it learns them from the training data. The main limitation of YOLO is incorrect localisations.

This version of YOLO revolutionised the object detection paradigm breaking through the speed-precision tradeoff. The following versions worked on top of this idea and the limitations of each version with the latest computer vision latest ideas and got even better results.

2.1.1.2 YOLOv2

Picking up on some errors that the YOLOv1, Joseph Redmond and Ali Farhadi, added some features to the YOLO model to improve it. In December 2016, YOLOv2 [77] was released. The focus was mainly on improving recall and localisation while maintaining classification accuracy.The highlighted changes were:

Batch Normalisation: The authors added batch normalisation on all of the convolutional layers for this version. Fixing the distribution of the inputs to a ConvNet layer would bring advantages for the layers but, at the same time, is impossible to normalise the complete training set because the optimisation step operates with stochastic gradient descent (SGD). To address this, batch normalisation works as follows: Once SGD utilises mini-batches while training, every mini-batch generates estimations of each activation's mean and variance values. First, by computing the mean and variance value of the mini-batch of size m and then normalising the activations of number m to achieve a mean value of 0 and a variance value of 1. This way, activations have an equivalent distribution. Batch normalisation leads to significant improvements in convergence time effort and helps regularise the model.

High-Resolution Classifier: The original YOLO trains the classifier network at 224×224 and increases the resolution to 448 for detection. This premise means the network must simultaneously switch to learning object detection and adjust to the new input resolution. This adjustment is made by adding a fine-tuning procedure to the classification network at 448×448 for ten epochs on the ImageNet dataset. This adjustment gives the network time to adjust its filters to work better on higher resolution input. We then fine-tune the resulting network on detection.

Convolutional With Anchor Boxes: In original YOLO networks, coordinates of predicted boxes are directly generated by fully connected layers. Faster R-CNN uses anchor boxes as a reference to generate offsets with predicted boxes. In YOLOv2, the author adopts this prediction mechanism. An anchor box is a list of predefined boxes that best match the desired objects. The bounding boxes were not only predicted based on ground truth boxes but also predefined k anchor boxes. It firstly removed fully connected layers. Then it predicts class and objectness for every anchor box. An issue with the Faster R-CNN's anchor boxes strategy is that the network anchor boxes dimensions are hand-picked. To solve this, instead of choosing priors by hand, they run k-means clustering on the training set bounding boxes to automatically find suitable priors. The distance used is smaller if the IoU between the anchor and the centroids is higher. The number of clusters was set to 5.

Fine-Grained Features: While the 13×13 feature map is sufficient for large objects, it may benefit from finer-grained features for localising smaller objects. For localising smaller objects, higher-resolution feature maps can provide helpful information. Similar to the identity mappings in ResNet, YOLOv2 concatenates the higher and low-resolution features by stacking adjacent features into different channels.

Multi-Scale Training: For the algorithm to be robust to running on images of different sizes, we train this into the model, every ten batches, the network randomly chooses a new image dimension size from 320, 352, ..., 608. Hence, this means the same network can predict detections at different resolutions.

YOLOv2 also proposes a new classification backbone model, Darknet-19, inspired by the 19 convolutional layers. It also has five max-pooling layers.

2.1.1.3 YOLOv3

Two years later, YOLOv3 [78] was released. Following YOLOv2, the system predicts bounding boxes (centre, width, height) using dimension clusters as anchor boxes. YOLOv3 predicts an objectness score for each bounding box using logistic regression. This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior (anchor box). If the bounding box prior is not the best but does overlap a ground truth object by more than some threshold, it ignores the prediction. Unlike Faster R-CNN, the YOLOv3 system only assigns one bounding box prior for each ground truth object. If a bounding box prior is not assigned to a ground truth object, it incurs no loss for coordinate or class predictions, only objectness.

Also, to adapt to more complex datasets containing many overlapping labels, YOLOv3 changed the classification problem to a multi-label classification one. Now, each box predicts the classes

that the bounding box may contain. Instead of using softmax as until now, it now uses independent logistic classifiers for every class. The loss function is now a binary cross-entropy for the class predictions.

YOLOv3 contains now a multi-scale detector. It predicts boxes at three different scales. It extracts features from those scales using a similar concept to feature pyramid networks used in Mask R-CNN. At each scale, YOLOv3 uses 3 anchor boxes and predicts 3 boxes for any grid cell. Each object is still only assigned to one grid cell in one detection tensor.

The major change in YOLOv3 was its backbone. It now uses a new, deeper and better network for performing feature extraction. The new network is a hybrid approach between Darknet-19 and the residual network, so it has some shortcut connections. Darknet-53 is named after the 53 convolutions the network contains.

The results indicate that YOLOv3 is a powerful detector that excels at producing decent boxes for objects. However, performance drops significantly as the IoU threshold increases indicating YOLOv3 struggles to get the boxes perfectly aligned with the object. The localisation problem maintains itself. In the past, YOLO struggled with small objects. However, now we see a reversal in that trend. With the new multi-scale predictions, YOLOv3 has relatively high APS performance. However, it has comparatively worse performance on medium and larger size objects.

2.1.1.4 YOLOv4

YOLOv4 [18] was the first YOLO architecture release in which the creator, Joseph Redmon, did not participate.

This work's primary goal is to design a fast operating speed of an object detector in production systems and optimisation for parallel computations, rather than the low computation volume theoretical indicator (BFLOP). It aims to facilitate the training for custom domains and uphold real-time detections.

In essence, YOLOv4 is a series of additions of computer vision techniques that are known to work with a few small novel contributions. The main contribution is to discover how these techniques can be combined to play off one another effectively and efficiently for object detection [86].

The backbone network for an object detector is typically pre-trained on ImageNet classification [86]. The paper [18] considered different backbones: CSPResNext50, CSPDarknet53 and EfficientNet-B3, and after careful testing and theoretical justifications, the CSPDarknet-53 was determined as the optimal network and chosen to integrate the Yolov4 architecture.

The CSPResNext50 and the CSPDarknet53 are both based on DenseNet. DenseNet was designed to connect layers in CNNs with the following motivations: to alleviate the vanishing gradient problem (it is hard to backpropagate loss signals through a very deep network), to bolster feature propagation, to encourage the network to reuse features, and reduce the number of network parameters [86].

When choosing the backbone network, the objective was to find the optimal balance among the input network resolution, the convolutional layer number, the parameter number, and the number of layer outputs.

For the network's neck, YOLOv4 considered the following options: Feature Pyramid Network (FPN), Path Aggregation Network (PAN), NAS-FPN, BiFPN, ASFF, and SFAM

These layers are usually used to collect feature maps from different stages. Usually, a neck is composed of several bottom-up paths and several top-down paths. YOLOv4 adds an SPP block after CSPDarknet53 to increase the receptive field and separate the essential features from the backbone. Also, it chooses PANet for the feature aggregation from the different levels of the network.

As for the network's head, there is not much to add as YOLOv4 deploys the same head as YOLOv3 for detection with the anchor-based detection steps and three levels of detection granularity.

YOLOv4 introduces and tests methods to improve the network training and performance. These are the Bag of Freebies (BoF) and the Bag of Specials (BoS).

BoF are techniques that only change the training strategy or increase the training cost and improve the network's performance without adding inference time in production. The technique related to object detection methods that meet the definition of the bag of freebies is data augmentation. The purpose of data augmentation is to increase the variability of the input images so that the designed object detection model has higher robustness to the images obtained from different environments [18]. Many of these strategies were already known to the computer vision community, and YOLOv4 verifies their effectiveness [86].

There were also some new contributions like Mosaic data augmentation and Self-Adversarial Training. Mosaic data augmentation tiles four images together, teaching the model to find smaller objects and to pay less attention to surrounding scenes that are not immediately around the object. Self-Adversarial Training makes the neural network first alter the original image to create the deception that there is no desired object in the image. Then, the network is trained to detect an object on this modified image, forcing it to generalise new features that can help it detect this object. It follows the idea of the adversarial network.

The final BoF chosen for each part of the network were:

Backbone: CutMix and Mosaic data augmentation, DropBlock regularisation, Class label smoothing.

Detector: CIoU-loss, CmBN, DropBlock regularisation, Mosaic data augmentation, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler, Optimal hyperparameters, Random training shapes

BoS are those techniques that increase inference time by a small amount but can significantly improve object detection accuracy.

The BoS chosen for YOLOv4 were:

Backbone: Mish activation, Cross-stage partial connections (CSP), Multi-input weighted residual connections (MiWRC)

Detector: Mish activation, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS

In essence the YOLO v4 architecture is composed by CSPDarknet53 as the backbone, SPP and PAN as the neck and using the same head as YOLOv3

The YOLOv4 examines a large number of techniques and presents a large set of results. With the final configuration, YOLOv4 presents state of the art performance.

2.1.1.5 YOLOv5

YOLOv5 [44] was released just a few days after YOLOv4, meaning that YOLOv5 is not working on top of the v4 structure but rather on v3. Thus, both researchers essentially applied state-of-the-art computer vision innovations at that time [68].

YOLOv5 is a natural extension of the YOLOv3 repository from the author, where he was starting to develop new techniques to improve the model. The name YOLOv5 has been controversial but has been sticking for a few years. It was intended to be YOLOv4, but, to avoid collision, it was named YOLOv5.

Some more detailed aspects of YOLOv5 can not be presented as the author has not published a paper on this new architecture. This is the main problem with YOLOv5. Information is not demonstrated in a formal, detailed paper that can be reviewed with more awareness. However, some information is available through related works and the public repository documentation [44]. It is worth mentioning that the repository is in continuous development, and some of these specifications may become outdated.

As for the network architecture, the three parts are:

- **Backbone**: Focus structure, CSP network;
- **Neck**: SPPF block, PANet;
- **Head**: YOLOv3 head using GIoU-loss.

The structure is very close to YOLOv4's. For the backbone, it adds the Focus layer and maintains the CSP network, The purpose of the Focus layer is to reduce layers, reduce parameters, reduce FLOPS, reduce CUDA memory, and increase forward and backward speed while minimally impacting mAP. In essence, it is a tensor reshaping layer. It acts as a transformation from space to depth. The objective is to reduce the space (resolution) and increase the depth (number of channels). With this layer, the cost reduces the convolution computation using tensor shaping.

As for the neck, it is worth noting here that YOLOv5 borrows research inquiry from YOLOv4 to decide on the best neck for their architecture [68]. It introduces SPPF, a cascaded version of SPP that produces mathematically identical results to SPP with fewer FLOPs and faster speed. The head is also the same and uses the GIOU-loss function.

As for the activation function, PyTorch's SiLU is used throughout the mode. It has used leakyReLU and Hardswish functions; however, using SiLU simplified the model and improved its performance.

The differentiators for YOLOv5 are the auto-learning anchor boxes and the loss calculation. In its YOLOv3 repository, Glenn Jocher introduced learning anchor boxes based on the distribution of a custom dataset's bounding boxes with K-means and genetic learning algorithms. This learning is substantial for custom tasks. The aspect ratios may change with the custom datasets from the different users compared to the COCO dataset, where the models are pre-trained. So, these anchor boxes are auto-learned for the input (custom data) it receives.

The YOLOv5 has five different network sizes, n, the smallest, s, m, l and x, the largest. These are ordered in size. In general, larger models like YOLOv5l and YOLOv5x will produce better results in nearly all cases, but have more parameters, require more CUDA memory to train, and are slower to run. Jocher et al. [44] suggest the model size l for cloud deployments.

YOLOv5 is written in Pytorch, which becomes one of its advantages. This makes it much easier to integrate than YOLOv4, providing fast and easy installation and training. In addition, the PyTorch community is also more extensive than the Darknet community, which means that PyTorch will receive more contributions and growth potential in the future [92]. Hence, one of the largest contributions of YOLOv5 is to translate the Darknet research framework to the PyTorch framework. YOLOv5 repository [44] continues to evolve and has constant upgrades, which has a fundamental role for the community. Since its launch, YOLOv5 has had six versions where each one brought more integrations valuable to the users and, more importantly, model improvements. The last version is v6.1 and was released in February 2022.

2.1.1.6 YOLOv4 vs YOLOv5

YOLOv4 presented a considerable improvement to the YOLO series and was set to become the state-of-the-art algorithm for object detection. Just a few days later, YOLOv5 was released, and a "battle" between the two started to get the best results in the exciting area of object detection [69].

Nelson and Solawetz [69] presented an exciting benchmark between these two. It compares YOLOv4 and YOLOv5 Installations, Data Set Up, Training Configurations, Training Time, Evaluation, Inference Time and Model Storage Size. The dataset chosen was related to blood cell count and detection.

As for the installation, YOLOv5 proves to be more straightforward as it resumes to, cloning the repository with one command and installing the requirements in another. YOLOv4 requires the configuration of the makefile and a few extra installations among the repository.

Both are identical when setting up the dataset, although the YOLO5 format file has fewer lines and necessary directories. For training configurations, YOLOv4 Darknet provides more flexibility and is probably a better spot for research purposes. When Comparing YOLOv4 and YOLOv5 training time, the difference was enormous. YOLOv4 Darknet took a whopping 14hrs on our example dataset. However, it reached max validation evaluation well before that, which took about 3.5 hours. YOLOv5s, on the other hand, took 14.46 minutes to train on 200 epochs. The s version is the second smaller version for YOLOv5. The training length definition is also different in both. YOLOv4 works with the maximum number of batches that go through the model against the epoch definition. The size of the networks also changes drastically between the two. The

YOLOv4 default size sits between the YOLOv5 size x and l. YOLOv4 model weights 245MB, and YOLOv5's x version weighs 367MB. This training evaluation may be one of the essential parts of this evaluation. Since it is on a custom dataset and not the COCO, this is likely adjusted to what users will try, making it valuable but not generic for a complete comparison. For both networks, they have looked at the max validation $mAP_{0.5}$. We find that they perform similarly in this metric (0.91 mAP) for our task.

Lastly, [69] compares the inference time with the same configuration (Tesla P100). On single images (batch size of 1), YOLOv4 inferences in 22 ms and YOLOv5s inferences in 20ms. When in batch (size of 36), the YOLOv5s infers at 7ms, rounding the 140 FPS. Comparison with YOLOv4 is impossible as the authors could not implement batch inference for this network.

Overall, this benchmark is helpful to confirm how close these versions are. Throughout the diverse topics compared and the environment that they reside in, the YOLOv5 may come out on top. As for the precision, the difference is minimal, as shown above, but it could also be easily matched when training a custom dataset. For the rest of the fields, YOLOv5 shows that it is more easily integrated, runs faster and saves more space (dependent on the network size). Also, it has constant public improvements, resulting in constant improvements to users' applications.

2.1.1.7 YOLOX

YOLOX [32], released in 2021, aims to improve the YOLO series results with state-of-the-art object detection advances like anchor-free detectors, advanced label strategies and end-to-end detectors. Hence, the YOLOX improvements were:

Decoupled Head: YOLOX presents a decoupled head architecture for classification and localisation, which is widely used in most one-stage and two-stage detectors aiming to resolve the conflict between classification and regression problems.

Therefore, the decoupled head method, where classification and localisation operations are separated, is included in YOLOX. This increased the converging speed in YOLOX and is necessary for the end-to-end version, one of the innovations of YOLO. This version has the advantage of running end-to-end but slightly decreases the performance and the inference speed. Decreasing the inference time is positive but is not very significant as it reduces the performance.

Strong Data Augmentation: To boost performance, it uses data augmentation techniques - Mosaic and Mixup. Mosaic was initially introduced in YOLOv3 and subsequently used in v4 and v5. Mosaic data augmentation combines four training images into one in certain ratios. MixUp is a data augmentation technique that generates a weighted combination of random image pairs from the training data. With these changes [32] found that ImageNet pre-training is no more beneficial. Thus it trains all the following models from scratch.

Anchor-free Detection:

From YOLOv2 until YOLOv5, the model followed an anchor-based detection. However, the anchor mechanism has many known problems. First, to achieve optimal detection performance, one needs to conduct a clustering analysis to determine a set of optimal anchors before training. Those clustered anchors are domain-specific and less generalised. Second, the anchor mechanism

increases the complexity of detection heads, and the number of predictions for each image [32]. With anchor-free systems, similar success has been achieved, design parameters have decreased, and speed has increased. This has reduced the number of predictions for each location from 3 to 1. At the same time, YOLOX selects one positive sample for each object to avoid disturbing the consistency with the baseline YOLOv3. Hence, YOLOX, without an anchor box, selects a positive sample for each object (centre of the object). If it selects a single positive sample, it ignores other predictions, even if they are high-quality. However, optimising those high-quality predictions may also bring beneficial gradients, alleviating the extreme imbalance of positive/negative sampling during training. In essence, this positive sample is the central location of the object.

Advanced Label Assignment: YOLOX also presents an approach for the upcoming progress in advanced label assignments. Based on a personal study, the authors present simOTA for this matter. SimOTA shows positive results as it increases the network accuracy and reduces training time.

As for the results, the YOLOX paper presents some improvements against YOLOv5. However, with the YOLOv5 advances, these results have been surpassed, as shown in Table 2.1

2.1.1.8 YOLOR

Wang et al. [95] observes, "People "understand" the world via vision, hearing, tactile, and also the past experience. Human experience can be learned through normal learning (we call it explicit knowledge), or subconsciously (we call it implicit knowledge).".

Based on this idea, the YOLOR paper [95] describes an approach to combine explicit knowledge, defined as learning based on given data and input, and implicit knowledge learned subconsciously. YOLOR proposes a unified network to perform these several tasks at once.

Briefly, there are three notable processes by which this architecture is made functional: kernel space alignment, prediction refinement and multi-task learning in a CNN.

The main distinction is regarding the proposed CNN. While regular CNNs learn how to analyze inputs to get outputs, YOLOR's CNN tries to learn how to get outputs and what all the different outputs could be. Hence, rather than just one output, it can have many. Thus, to train the proposed unified networks, YOLOR uses explicit and implicit knowledge to model the error term and then uses it to guide the multi-purpose network training process. Furthermore, the kernel space alignment is used to fight the kernel space misalignment problem related to multi-task and multi-head neural networks.

In conclusion, YOLOR reaches state-of-the-art performance in real-time processing. However, this algorithm may be more than what this dissertation needs and, despite running in real-time, it may be slow for the next steps like tracking.

2.1.2 Transfer Learning

Some of the tasks required for this thesis are image registration, object detection, object tracking and action recognition, which are predominantly supervised learning tasks. Supervised learning

relies on labelled ground truth data, based on which the learning algorithm infers the mapping between the raw data and the desired labels in the training stage. Thus, a prerequisite for supervised learning is data preparation, which includes data collection and labelling, pre-processing and feature extraction, followed by splitting data into a training and testing set and selecting an appropriate learning algorithm and model structure for the specific task. After training and validating the model, the model needs to be tested, and the obtained results need to be compared with the ground-truth data to evaluate the model's performance [40].

Another important concept in machine learning that is used in this context is Transfer Learning. As it was shown above, deep neural networks became popular for object detection, which can also be applied for player detection using transfer learning [83]. The main purpose of transfer learning is to assist the training of the target domain model with a large number of training samples of the source domain, that is, to transfer a pre-trained model and parameters that have been trained in the source domain to the new model of the target domain. Pan and Yang [74] presents the following definition for transfer learning: Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T()$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.

Pan and Yang [74] categorises transfer learning under three sub-settings, inductive transfer learning, transductive transfer learning, and unsupervised transfer learning, based on different situations between the source and target domains and tasks. In inductive transfer learning, the target task is different from the source task, no matter when the source and target domains are the same or not. In action, some labelled data in the target domain are required to induce an objective predictive model $f_T()$ for use in the target domain. The source and target tasks are the same in transductive transfer learning, while the source and target domains are different. For this one, no labelled data in the target domain is available, while many labelled data in the source domain are available. Unsupervised transfer learning is similar to inductive transfer learning, but the focus is on solving unsupervised tasks in the target domain, so no labelled data is available. It also defines four different transfer learning approaches regarding what is transferred: instance transfer, feature-representation transfer, parameter transfer, and relational-knowledge transfer. These are the approaches used in some of the previous sub-settings. Briefly, instance transfer is to re-weight some labelled data in the source domain for the target domain. The feature-representation transfer idea is to find an adequate feature representation to improve the performance of the target task. So, the knowledge used to transfer across domains is encoded into the learned feature representation. The parameter-transfer approach focuses on the idea that the source tasks and the target tasks share some parameters or prior distributions of the model's hyperparameters, so the transferred knowledge is encoded into the shared parameters or priors. The relational-knowledge transfer applies to problems where the data is not independent and identically distributed as is traditionally assumed but can be represented through multiple relationships. Thus, the knowledge to be transferred is the relationship among the data [26].

In this case, ball detections would be direct, and person detection could be for the player detection, using a feature-representation approach. Still, in the source domain, person detection,

many source data is not directly related to our task. In contrast, in the target domain, there are only a small amount of target data that is directly related to the task, but the two kinds of data are related or similar to each other [103, 83]. So, when developing a detector, there is the opportunity to use transfer learning methods to speed the development and training steps.

2.2 Computer Vision in Football

As the most popular sport in our society, followed by millions of people, the high market value, and the growth in the amount of data generated, football needs the creation of automated tools to help clubs and coaches to stay at the top of the data analytics, but also for fans, by providing customised football games analysis services. Regarding these applications, and more not mentioned, many studies have been conducted in DL to perform the following tasks: action spotting, player counting and tracking, ball tracking, tactics analysis, pass feasibility, talent scouting, game phase analysis, or highlights generation [24]. There are also areas of investigation more related to field detection that helps these previously mentioned tasks, such as field segmentation and registration, camera calibration, and broadcast image registration [28, 91]. In conclusion, many issues are still open for research to fully automate the video analysis of sports events, and it is expected to keep it this way to keep upgrading these applications since the demand will not stop.

These previous areas of investigation are the glue to the high-level football video analysis applications. Suggested by [28], they can be classified in the following three areas:

- Detection/tracking of players and/or ball;
- Event detection;
- Game analysis.

Some applications tackle these areas separately, but multiple ones look at player and ball detection as the centre of the video analysis [28]. These detections, supported with field registration, to gather a real-world position is sufficient for most of this football video analysis area, which is this dissertation's focus. The numerous aspects of a sports event start when deciding how to capture the event; modelling the pose and viewpoint of a camera and calibrating large camera networks is still very challenging. There is then the critical challenge of detecting and analysing humans [91].

2.2.1 Camera Setup

Soccer videos can be categorised into those captured by static, dynamic/moving, where the extracted data from each setup category was obtained through single or multiple cameras. Different camera setups require different approaches for one task, i.e. to perform player tracking, we might use different techniques if there are one or more cameras [63].

Generally, much work has been done on broadcast videos, and multiple moving cameras has got little attention. In general, monocular approaches are affected by different problems, such as

small field of view (FOV), lack of 3D information, low resolution and complex occlusions. The broadcast cameras also partly cover the field, and all players are not visible all the time. For game analysis or other attempts to represent the game in another format, constant movement may obligate to an also constant homography calculation. This constraint is general to moving cameras and not only to the broadcast view. An alternative approach to improve soccer video analysis is to use multiple stationary cameras, which presents a series of advantages. Firstly, ambiguities can be reduced, particularly during crowded, cluttered or occluded scenes, and the analysis results (e.g. player labelling) can be improved. Moreover, the whole field can be covered all the time. These advantages make these setups more suitable for analysis of the game since the properties of the soccer match, such as predefined numbers of players or their teams, can also be applied. It is also more suitable for content enrichment, by analysing more than what the user may be seeing, broadcast view [63].

Furthermore, camera parameters remain fixed, and missed players can be compensated in some views. Moreover, the best camera for image analysis can be chosen since the FOVs of cameras can be defined. This setting also makes the background subtraction applicable to simplify foreground analysis. If the cameras are also static, the possible transformation of the image is also more relaxed since it is only calculated once (for each camera). Moreover, the size of the represented player can be big enough to provide information with better quality, e.g. possible kit number detection. This previous premise brings up the quality of the cameras used. Still related, another vital aspect is the camera's height which is also a trade-off. Very high cameras capture more information but with smaller objects, where the more level to the ground, the opposite. This positioning is also conditioned by the stadium where the game is recorded. 3D information of the scene can also be retrieved, and the collaboration can improve the accuracy of localisation among multiple views, which is highly required for detecting events such as offside moments [63].

Despite all these advantages, some limitations of these configurations consist of increasing hardware cost and processing time by an increase in the number of views, synchronising different views, considering robustness in configuration and collaboration among multiple views [63].

One crucial issue in multi-view configuration is the association and fusion across multiple simultaneous views. Association concerns assigning each measurement to its associated player, and fusion concerns fusing measurements from multiple views. Therefore, good observations in some views can compensate for poor observations in other views, and different tracks can be fused across different cameras [63].

The association and fusion of information from multiple views can be image-based. Accordingly, the geometrical relationship between cameras was made by homography or fundamental matrix while the result using the first outperformed. In order to fuse data, some researchers average the coordinates of the players from multiple views. Then the second average between close coordinates to the first average was considered the estimated player location. This approach can have a problem when players are very close to each other, which can be favourable to some errors [63]. Approaches for these tasks will be further presented in Section 2.2.4.

There are advantages and disadvantages for all possible setups, so a strategy must be set depending on the type of application, budget, and stadium capacities. Also, this strategy can be defined based on the chosen methods to perform the analysis of the images if done a priori.

2.2.2 Field Detection

When detecting a field, the logically first thing to do is recognise the area of the field and its lines. The images from that we observe the game are generally from a central position, from a camera that pans along the sidelines. This angle allows us to watch the game, but for other applications, we need other representations from what is happening in the game, e.g. field top view with player positions. Almost every image does not have a total view of the field, which allows mapping the player field position to a 2D top view. So, this creates the need to have a methodology to extract these exact positions [21]. This mapping can be done with the homography matrix. Various studies have been conducted to find the homography matrix of a game image so that the registration from that game image and the field top (or other) view becomes possible. This task is denominated as Field Registration. There are a few steps to perform it (not all necessarily), e.g. field and line detection and segmentation, feature detection, keypoint detection, homography regression.

Field Detection is normally the first stage in football video analysis. This detection reduces the noisy data from outside the field (e.g. advertisement, crowd, reporters, ball assistant) and so the number of pixels to analyse, but can also identify foreground objects - set of pixel intensities that drastically change their positions in the current frame concerning their positions in the previous frame (e.g. players, ball, field lines) - from the background by filtering the grass pixels. Some examples of the player detection are presented in Section 2.2.3.

Since the soccer field is predominantly green in long and medium shots, most researchers have tried to detect the field by detecting the dominant colour. The field colour varies as the game progresses and for different stadiums, depending on the weather, lighting conditions, shadows, and camera position. As a result, these variations must be considered for accurate field segmentation. Learning the field colour and holding it fixed without updating its statistics during the game is not an option [63].

The survey, [63] defines six main steps for field detection: feature selection, dominant colour extraction, playfield pixels detection, playfield extraction, shadow removal and model updating.

As previously pointed out, most researchers looked for the dominant colour on the shots to detect the field of play. Some researchers tried to utilise image features independent of illumination by transforming the RGB colour space to another. The transformations studied where to HIS (Hue, Intensity, Saturation), to YCbCr, where Y is the luma component, the brightness of the colour, Cb is the blue component relative to the green component, and Cr is the red component relative to the green component. Also, to Normalised RGB and YIQ. Yao et al. [106] as many others, transformed it into HSV (hue, saturation, value) [63].

In most previous approaches, the field colour was learned using supervised or unsupervised learning. For the supervised category, some works exist in which predefined thresholds were set

experimentally for detecting the playfield pixels. This approach has the drawback previously presented: the thresholds were too dependent on the grass colour, that variate with different stadiums and lightning. Several methods, such as mean, sliding window and k-means, were introduced to fix this issue by selecting the thresholds. The playfield colour can also be represented by the histogram's colour peak value. As the name suggests, it is based on a distribution of colours within an image where the data is obtained by counting the occurrence of each possible colour of the respective colour model within the image. The peak value is the value with more occurrences. Some methods calculated the mean value of each colour component around their respective histogram peaks or defined an interval around the histogram peak [63]. Homayounfar et al. [37] performs semantic segmentation of a broadcast image into grass vs non-grass by exploiting the prediction from a CNN trained using DeepLab for their binary segmentation task.

Field pixels detection can be divided into two main approaches: non-parametric and parametric methods. Gaussian Mixture Models (GMM) and colour histogram-based method were the most widely used parametric and non-parametric techniques, respectively [63]. Using GMM in RGB allowed, for example, the modelling of a field with a cast shadow from a stadium roof. Related to dominant colour extraction approaches, field pixels could be detected either by thresholding the distance between the colour of the testing pixel and dominant colour or by calculating the distance from the field colour class in the case of learning dominant colour by GMM. Some methods use the GMM for line correspondence, analysing their colours to find the most probable transition between the field and line colour classes within the search window.

In other approaches, the grass region was detected using background subtraction, followed by playfield pixels classification using Mahalanobis distance from the playfield model represented by the Eigen model and GMM. As a result, both models achieved similar levels of success; however, GMM had higher computational complexity. Alternatively, thresholding or clustering based on histogram peaks was applied for detecting playfield pixels [63].

Various algorithms can be used simultaneously to generate features to increase detection accuracy to achieve field detection. These features can be colour, motion, shape, texture, gradient, space and spatial coherence, local entropy, geometric masks, or other types of attributes that help the detection of the field. For instance, [99] combined colour, motion, and shape for field segmentation in two online and offline steps. Some approaches applied gradient with GM, while others used two binary playfield detection results from colour ratio and local entropy thresholding. In [47, 48], very similarly, hue and texture were used as features to model the grass as a Mixture of Gaussians (MOG), and the grass was segmented using multi-layer perceptron and Bayes theory. There is the chance to build the pitch mask as the intersection of colour-based and geometry-based masks. The geometric constraint, namely, area of coverage based on homography, was also applied by [70]. The Area of Coverage (AoC) is defined as the area on the playground that a particular camera captures. This definition is helpful for various broadcast or single camera applications. First, the system uses GMM to obtain a rough segmentation of the playfield area. Still worth mentioning for this method is that based on this initial segmentation, feature points of the field are detected for calibration using homography, and then the AoC of the camera view is calculated.

Finally, this AoC is back-projected to the image frame to provide a refine playfield detection [63]. These types of methodologies will be further explored next in Section 2.2.2.2.

The field colour should also adapt to the game's lighting changes. With this purpose, approaches propose the computation of the mean value of randomly selected pixels as the field colour and the final field colour obtained by the weighted sum of the colour of sample pixels in the previous and current frame. To complement it, some approaches use the extracted playfield to re-estimate the GMM or apply incremental training for the Gaussian field model. Finally, the field mask was often obtained by applying morphological operations on the resulting binary image from the playfield pixels detection, and the field was extracted by the convex hull of the green field [63].

Cuevas et al. [27] uses an unsupervised field segmentation strategy based on the estimation of the probability density function of the green chromaticity, $g = G/(R+G+B), 0 <= g <= 1$, of the image, based on the work of [76]. The output from the segmentation is used to obtain a binary mask corresponding to the field. It then uses the segment detection method, top-hat, to discard the vast majority of the unwanted edge data and help the next phase not detect false line marks. The Edge Linking and Line Segment Fitting (ELLSF) algorithms are applied to detect and label the line segments. To end the segmentation phase, the marginal utility principle removes the small segments that may still appear in the image, like billboards, players, goals. This removal proves to be a crucial aspect of this method's good results, as other strategies do not account for these false detections.

Finally, the field mask was often obtained by applying morphological operations on the resulting binary image from the field pixels detection. The field is extracted by the convex hull of the greenfield or boundary-following algorithm. The region growing method has also been employed to connect playfield pixels into regions [63].

After this field detection or segmentation, various methodologies use line detection and, consequently, keypoint detection to perform the registration to the 2D top view image presented next.

2.2.2.1 Homography Estimation

A homography is a mapping between two images of a planar surface from different perspectives. It maps the points in one point to the corresponding point in another image. The homography matrix is defined by a 3×3 matrix with eight degrees of freedom. The task of sports field registration aims at determining a homography matrix for the transformation of an image into two-dimensional sports field coordinates [90]. Nie et al. [72] define sports field registration as a special case of homography estimation.

Two traditional homography estimation approaches are direct and feature-based methods, which can also be used together to increase robustness. Also, inspired by the data-driven Convolution Neural Networks in CV, there has been an emergence of approaches for homography estimation using these supervised methods. For this extent, [72] regresses a set of dense features through a deep network, to compute the homography.

The direct methods, such as the seminal Lucas-Kanade algorithm, use pixel-to-pixel matching by shifting or warping the images relative to each other and comparing the pixel intensity values using an error metric such as the Sum of Squared Differences (SSD). These methods try a first guess for the homography parameters and then use search and optimisation techniques, like gradient descent, to minimise the error function.

The feature-based approach first extracts keypoints in each image using detectors like Scale Invariant Feature Transform (SIFT) or Oriented FAST and Rotated BRIEF (ORB). Then establish a correspondence between the two sets of keypoints, using feature matching, using methods like RANSAC or non-linear optimisation to find the best homography estimate [72, 71]. This approach is limited by the accuracy of the local feature detector, which is not robust to significant appearance variations and non-static objects.

Deep networks have stepped in for detecting keypoints for human pose estimation, using networks that have encode-encoder structures and skip connections to help the network capture context from a larger image than the local descriptors. Supervised approaches use ground truth labels and, as a result, are limited to synthetic datasets where the ground truth can be generated for free or require costly labelling of real-world data sets [72]. DeTone et al. [29] uses a CNN to regress the homography matrix, along with a 4-point parametrisation approach. For the dataset, as a large number of images is needed, they generate a nearly unlimited number of labelled training examples by applying random projective transformations to a large dataset of natural images.

Alternatively, the regression problem can also use unsupervised learning, like in [71]. Unlike the hand-crafted feature-based approaches or the supervised approach that needs costly labels, their model is adaptive and can easily learn good features specific to different datasets. Their method uses the 4-point parametrisation and predicts the homography to warp the image using a pixel-wise photometric loss function to compute the gradient for back-propagation. Those are also approaches that use a multi-scale network to estimate homography while pixels of moving objects are handled explicitly [72].

Field registration and homography estimation are essential for this type of application. These applications could be a very lucrative area, so the past years have seen much investigation for this area.

2.2.2.2 Field Registration

Field Registration, as it was mentioned before, allows the transformation of the game image to a top view of the field (Figure 2.5). It is on top of this model that the players and ball detections can be used to a better understanding of the game.

This task can be critical for the good behaviour of an application. A common approach is to estimate the homography matrix by detecting field points and lines in the image and associating them with points and lines in the model. Given these correspondences, the homography can be estimated in closed form using the Direct Linear Transform (DLT) algorithm [38, 35]. Martín and Martínez [64], and others use this method but select the points manually. Other techniques are

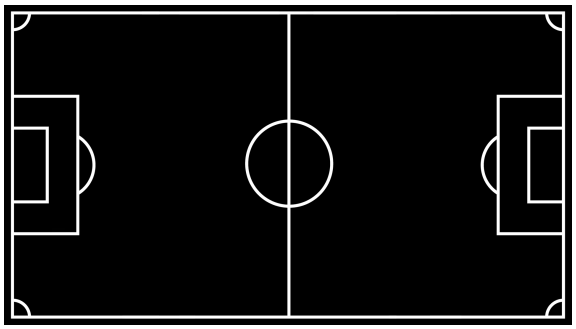

Figure 2.5: Top view of the football field

used. This section contains the state of the art methods to tackle this task and a table comparing the results of the most adequate and interesting methodologies with the IoU metric.

For line detection, general methods incorporating Hough Transform (HT) or Canny edge detector are applied. HT is a technique for mapping points in digital pictures into a parameter space where patterns of points are identified as peaks [13]. Chen et al. [20] presents a method for volleyball that first segments the court region consisting of the court lines using the dominant colour feature computed via colour histogram. The court lines are detected by the HT, which allows the extraction of the feature points via computing the intersection of court lines. This methodology can be used for most sports with plane fields, such as football.

Still using HT, there is also the option of using custom made ellipse detection for tangential feature point extraction to find feature points. The tuning of the homography matrix follows these methods, lastly. The HT for line detection suffers from the disadvantage of being slow [46].

For the broadcast soccer videos, the lines are long and sparse, and so, the solution can be in the form of a grid-based algorithm for line detection in the goal-mouth scene to tackle the speed problem. RANSAC based line detector for line parameter estimation is also an option instead of HT. The previous method required 1 second for initialisation, which is insufficient for real-time applications and increased the speed to 30 ms for the first frame and 6 ms during tracking. For broadcast videos, this is a remarkable improvement. For other types of camera setup, for example, fixed static cameras, the homography matrix can be detected even before the game, so this task would not have the time restriction [46].

These methods exposed are used as a pre-processing step for some presented next, to extract the field and players' real-world position. Some additional methods for line detection are still presented ahead, along with the homography estimation.

In [106], a self-calibration method for non-fixed cameras is proposed, which can register the images by comparing the intersections of the detected line marks with a set of 16 possible field crossing points patterns. This method uses a colour-based approach for field extraction by transforming the colour space into HSV and applying thresholds to produce a field mask, and also the work for line detection developed in [30].

In [93], once the detected lines have been classified into longitudinal or transverse, the homography matrices are estimated by analysing all the possible matches between the line intersections

and the field model keypoints.

Bozorgpour et al. [19] shares some problems in the line detection step like noise and blurred lines from camera motion but also points out that not every frame allows the opportunity to detect the points after the line detection. This may cause errors when estimating the homography. To tackle this problem, it performs the registration by establishing matches between the detected lines and the lines in the field model. It overlays the line mask obtained onto the football field model. Then it finds the nearest points of the projected line mask from some taken sample points on the field model. It then uses the RANSAC method to find the points' correspondence, which allows the homography calculation. It repeats this process until the frame line and the field map converge.

In the following three approaches [37, 36, 38], the sets of longitudinal (e.g. sidelines and its parallel lines) and transverse lines (e.g. end/goal lines and its parallel lines) are used to determine two vanishing points that are used to perform the registration. In [37], the circular-shaped lines are taken into account because of their unique shape/orientation and known locations. For the [36] methodology, the line detection method can be more relaxed because of their robust segment matching, which takes care of incomplete segments, missing segments, and spurious. This method has good results but has some trouble with images with not many lines, for example, from the centre of the field or more zoomed images.

Homayounfar et al. [38] presents a fully automatic method that formulates the problem as a branch and bound inference in a Markov Random Field (MRF), where an energy function is defined in terms of semantic cues such as the field surface, lines and circles obtained from a deep semantic segmentation network. The usage of MRF not only improves the registering process but also allows obtaining an accurate location of both straight and circle lines [28]. These strategies do not include any procedure to determine if the results are correct, as they have an unsupervised learning approach [27].

To register images with few visible lines - the main problem from the [37, 36, 38] work - some strategies [57, 105] include algorithms to relate these images to images that have been previously registered. Linnemann et al. [57] the main contribution is the view angle optimised processing, which is achieved by distinguishing three-view angle classes: goal; centre or intermediate view. Due to their diverse characteristics, it is appropriate to design dedicated tools for each type of view angle, so depending on a class assignment, the system determines the homography with the most promising estimation approach. The centre view detects the centre circle as an ellipse and computes the homography through points detected with tangents. The intermediate view, which even has fewer lines, uses the previous and after frame registered homographies to estimate the current. As a downside, the quality of their results decreases drastically as the number of consecutive images with few lines increases.

Yao et al. [105] addresses this by having an estimated first frame and tracking the feature points along video frames to construct a trajectory matrix of feature points that will help the registration of these next frames. This lack of visible lines and so key points was also thought by [22], which presents an approach where only two points are necessary for the estimation. It uses pan, tilt and zoom instead of homography to represent the transformations such that only two points are needed

for estimation. The limitation of this approach is that it assumes that the camera's base location and orientation are known a priori.

More strategies were developed regarding the centre circle and its relative ellipse in the image. Alvarez and Caselles [5] also tries to estimate the homography through the centre of the field as an ellipse. Oblique projection of the circular like objects makes an elliptical shape on a 2D space. The ellipse parameters give an estimate of the relative orientation between the object and the camera, so it can be detected with a 6-dimensional HT [5].

The main drawback of these strategies is their high computational and memory costs [27]. Consequently, some authors have developed strategies to reduce this cost. Cuevas et al. [27] presents some different approaches like probabilistic methods focused on reducing such costs [100, 13] which, since they do not consider all the significant edge points in the images, their accuracy is not high enough, or [98, 108, 109] that try to model the ellipses starting from some previously known ellipse points. However, their assumptions limit the number of images that provide successful results. An additional limitation of HT-based in these strategies is that they are not able of obtaining successful results in images where the centre circle does not appear complete [27].

Alemán-Flores et al. [3] uses HT to detect lines and realises that some of these lines are tangents of the centre circle, so they have proposed an algorithm based on the detection of straight lines tangent to detect the ellipse. They then have two cases to estimate the homography. If there are at least four visible lines in the image, they build homography candidates by putting in correspondent (matching) four visible lines in the scene with four lines in the actual court. Otherwise, the centre circle and the halfway line are visible in the image, and they assume that either a touchline or the centre spot is also visible. They then recover the homography. Their method is also robust to poor lighting conditions, shaded areas or lens distortion.

Cuevas et al. [27] presents an interesting approach that detects both straight and centre circle lines, which could be helpful as cameras views other than broadcast usually contain one or another. The results are presented as better than the state of the art's determined by its previous phase, line segmentation presented above in this document. It uses the HT method for straight lines and ellipse point detection for the centre circle line. The ellipse point detection is compelling and is done as follows: first, a list of interception points between the line segment, previously extracted, and the halfway line is computed. As there may be points that do not belong to the ellipse from this intersection, they are classified by the intersection ratio between the image in that point and a cross. The point that has the best ratio is the initial point of the ellipse. From here, they developed an algorithm to move between the pixels and register the points that are in the ellipse. This detection results in a new binary mask, indicating the positions of the pixels belonging to the ellipse is obtained. With the mask, more points can be discovered with the help of the LSF algorithm and auxiliary lines.

Some approaches rely on a database of images and related homographies to estimate new identical or similar ones to the input. This camera view database type of approach works by associating an image projection of a synthetic reference field model to a homography used to

produce said projection [24]. Chen and Little [21] extends the two-point method, [22], by relaxing the constraint of the camera location. The method has a camera pose engine, a deep feature extractor, a two-GAN model for field segmentation, subsequent field marking detection, and a camera pose refinement process. The camera pose engine has only three significant free parameters to generate many camera poses and corresponding edge images effectively. The two-GAN model first uses a segmentation GAN to segment the playing surface from the whole image and then use a detection GAN to detect field markings from the playing surface. Both GANs are firstly trained independently and then merged via joint training. Given edge images and their camera poses, the model learns compact features via a siamese network. The Camera Pose Optimisation first uses the siamese network and the feature-pose database for the first estimate and then the Lucas-Kanade algorithm on two distance images from the testing edge image and the retrieved edge image. This way, they can estimate the homography matrix from the nearest neighbour image to the testing image and use the chain rule to get the advanced camera pose [21]. The method works in the following sequence, generating diverse edge images and corresponding edges. Then, learn compact feature descriptors via a siamese network and build a feature-pose database. After that, use the GAN model to detect field markings in authentic images. Finally, query an initial camera pose from the feature-pose database and refine the camera pose by using distance images.

Still, in the camera views dictionary approach, [84] differs from most by using an area-based segmentation to associate between the images. It follows the same modules semantic segmentation, camera pose initialisation and homography refinement, but with the modules connected, allowing end-to-end training. The area approach is mainly used to fight images with heavy occlusion and motion blur caused by the camera. This type of approach is commonly used in other work that focuses on other specific objectives, like [24, 90] since they do not need prior manual annotation and are adaptable to different stadiums and conditions. Despite this, they may not perform in real-time if their dataset size grows and the processing time gets longer. Some applications need this large dataset to work with sufficient accuracy.

Jiang et al. [43] present a two-step DL approach. The first step is the initial registration network that provides a rough estimate of the registration, parameterised by a homography transform, and the second is the registration error network that regresses the error of the estimates. These networks are trained separately, decoupled training, unlike [84], to keep the network from overfitting. The registration network follows the 4-point parametrisation by directly regressing them of our template warped into a given view using the ground truth homography. For the registration error network, they refine the homography through differentiation. The model is trained to predict the error of metric, IoU. The drawbacks of this method are that it requires a relatively good estimation from the first network so that it converges. Also, it uses a warp step between the networks, making this approach slow.

Citraro et al. [25] suggest an approach that also takes into account the position of players. They present a first estimation with the detection and correspondence of specific line points and use the players' position to improve this estimation. The players' positions are back-projected from the image plane to world coordinates associated with the closest ground-truth positions. Then, the

new correspondences with the known semantic ones are used to estimate a new homography.

Nie et al. [72] tackle the challenge when no prior knowledge about the camera is available and propose a multi-task network to simultaneously detect a grid of key points and dense field features to estimate then and refine a homography matrix end-to-end. They consider that detecting sparse key points cannot perform an accurate homography estimation in every part of the field. To increase this chance, they propose the usage of uniform grid based key points that significantly increases the likelihood that a subset of grid-keypoints covers each part of the field. There can be frames that do not have grid-keypoints to estimate the homography. When this issue happens, they use their network to regress the dense frame features directly and align them with dense template features to further refine the initial keypoint based homography estimate. Its network counts with a dilated convolution and non-local block, and then DLT and RANSAC for the homography estimation. Another interesting step when estimating homography is verifying the temporal consistency of homography estimations. They compute the IoU measure between the binary maps projected using the first homography matrix estimate and its value of from the last frame.

The work stated above is mainly from monocular cameras, but the fundamentals can be utilised for other camera setups. If there are multiple moving cameras, the broadcast approaches can be very interesting. If the cameras are static, line detection and feature extraction can also be essential. In conclusion, estimating the homography for the game images has various possible approaches that must be weighted according to its advantages and disadvantages regarding the different cameras setup, resolution, and time available.

Various recent studies present different methodologies for this problem and different techniques and mainly use the IoU metric to measure the calibration accuracy.

For this case, the IoU is taken to be the shared area between the ground-truth model of the field and the re-projected one divided by the area of the union of these two areas (Figure 2.3). The IoU is one if the two coincide exactly and zero if they do not overlap at all [25]. The IoU metric can be divided into two sub-metrics, the IoU_{whole}, where all field is considered and the IoU_{part}, where only the area of the field visible in the image is taken into account. Consensus in which is better depends on the task itself and in the position of the researchers [25], presents that IoU_{whole} is better because it compares the whole field, even if invisible. However, [72], defend IoU_{part}, by stating that registration error of the invisible field cannot be measured because the ground truth homography is obtained using the visible field area and only guarantees the projection correctness of the visible area. Figure 2.6 is a good example for this division of opinions. The grey area shows the correspondent area captured in the frame, the blue lines show the ground truth model, and the red lines show the projection according to the calculated homography. As it is visible, IoU_{part} would be almost perfect, while IoU_{whole} would be very poor.

For most approaches, many training samples are required to achieve the desired accuracy. However, training data is not publicly available except for the World Cup 2014 dataset, which consists of broadcast images.

Table 2.2 demonstrate the results for most articles presented above regarding field registration, using the World Cup 2014 dataset. IoU_{part} has better results than IoU_{whole}, which may result from

Figure 2.6: IoU metric representation [25]

the lack of training data and closed images in the dataset. The best results in IoU_{whole} are from [25] with the help of manual player annotations, which confirms the benefit of detecting the players for better registration. From the ones that calculated the IoU_{part}, the best one was [72], with both their methodologies having similar results.

Table 2.2: Field Registration methods comparison with IoU metric

Method	IoU_{whole}%		IoU_{part}%	
	Mean	Median	Mean	Median
[38] Branch and Bound	83	-	-	-
[21] Two-Gan + Syntethic Data	89.4	91	94.5	96.2
[43] Two-step DL	89.8	92.9	95.1	96.7
[43] SFF [1]	83.9	85.7	90.3	92.1
[43] FFR [2]	84.0	86.2	90.3	92.1
[84] End-to-End	88.3	92.1	93.2	96.1
[25] FCN [3]	90.5	91.8	-	-
[25] FCN + player annotations [4]	**93.9**	**95.5**	-	-
[72] w/ alignment	91.6	93.4	**95.9**	97.1
[72] keypoints only	91.5	93.3	95.8	**97.2**
[24] CCBV [84] teacher [5]	79.8	81.7	88.5	92.3

2.2.2.3 Field Registration vs Camera Calibration

The main concepts stated in the state of the art articles are camera calibration and field registration. These two different areas are brought up throughout these articles as different applications for this case. Those are different concepts, but for this purpose, they have a common objective, homography estimation.

Sha et al. [84] remarks, "Field registration is a critical component of camera calibration in sport". Camera calibration is essential for the ball and player tracking systems and for any systems that need to render graphics into the image that appear locked to the real world (or 'tied-to-pitch') [91]. In other words, for a frame, it will allow the projection of any point in the field to its real-world coordinates in a plane football field. The camera may be parametrised using the

[1] Single Single-stage feed-forward network (SFF)
[2] Two- stage Feed Forward Refinement (FFR)
[3] Fully Convulutional Deep (FCN) Architecture
[4] with manual player manual annotation
[5] not fine-tuned for World Cup dataset

full perspective projection model but also using a homography model. Indeed, the field being most often planar, it is a convenient calibration rig to estimate the homography between the field plane and the image. Hereafter, camera calibration means estimating the internal (intrinsic) and external (extrinsic) camera parameters. External parameters of the camera include its position and orientation with respect to the world coordinate system, whereas the internal parameters include the image centre, distortion coefficients, and focal length [46, 24]. This area, camera calibration for football games, has been extensively studied in the last twenty years, generally improving its results and incorporating trending algorithms [22].

Same as for the camera calibration, some classify field registration as it is a method that requires the mapping of image pixels of a sports field to their corresponding real-world locations [72] or as belonging to a phase for camera calibration.

In this context, field registration and camera calibration, intend to do the same thing, estimate the homography between a plane field and the game image [72, 22, 24, 84, 21]. Field registration mathematically may be labeled as a homography estimation problem, which its task is to find the homography that can map the field from the observed camera perspective to a known overhead perspective [72, 84] . As for camera calibration, the calibration parametrization can be the homography between the field and the image, as well, which is valid as the playing surface is plane [22, 24, 35, 21].

2.2.3 Player and Ball Detection

Player detection is a crucial stage for soccer video analysis. As a low-level image processing task, player detection provides measurements that can be used for initialising the subsequent tracking step. In the same line of thought, the detection of the ball is also a crucial step for obtaining game information. For example, after performing this detection, each player and the ball can be represented by a bounding box or a blob. For the players, its location can be determined by the position of the players' feet [63].

When performing player detection, several associated challenges must be faced to obtain successful results, such as the occlusions between players, the abrupt movements of the camera, the changes of lighting, the lack of resolution in very distant players, the blurring of the players that are moving, or the players remaining long static periods of time [28]. The ball detection also comes with many challenges, mainly caused by its size compared to players and its even faster movement. These challenges are: in images with a low zoom level, the ball is represented by few pixels; the players may occlude it; its detection is difficult when it is on a line of the field of play; its appearance may vary in size, shape, and colour; and its speed is significantly changing.

Starting with the player detection, various techniques and methods are employed in this area, such as Background Subtraction (BGS), Region-Based Segmentation (RBS), and Supervised Learning [28].

The strategies typically started with colour-based foreground segmentation since there is an obvious colour difference between the players' kits and the field colour. Noisy data can disrupt this method. There are various methods following this segmentation strategy, like analysing the

histogram of the H component of the HSV colour space in search of the dominant mode, which is assumed to correspond to the green colour of the field [82, 49, 73]. Other colour models were used for this purpose, in [4, 61], like the RGB (Red, Green, Blue), where the field pixels were expected to have a distinguished green colour, following the rule, $G > R > B$. In, [61], primarily, the field ground and the edges of the original image are detected. Then, to identify the lines, they use the HT algorithm. The lines are then eliminated from the original image so that only detected players and the ball remain in the scene. This turned out useful for ball and line occlusion [28, 63].

Another type of approach, like [101], proposes a multi-camera system and a GMM to detect moving objects and separate them from the field. In [31, 66], morphological operations are applied, iteratively, using colour information, to segment the players, the referees and the ball. The obtained results are then improved in a second stage through a Watershed Transform, based in [81]. The method proposed in [53], detects players by performing a frame differencing BGS algorithm that is improved by a second-stage analysis based on foreground edge information [51, 28, 63].

Other strategies used methods like RBS. For example, [52, 12], used a k-means algorithm to group and extract pixels that segmented the types grass, players, ball. A method of player detection was proposed, in [104], using a fusion of the Otsu Algorithm and an edge detection method. In [87], after applying a BGS, the Basic Sequential Algorithm Scheme (BSAS) [102] is used to classify the players by grouping pixels according to the colour of their kit and to compare them with known samples of this kit. There is the problem of the frequent kit change using blobs, [62], presents a method to detect and label multiple players that combines a two-step blob detection with optimisation mechanisms that are capable of detecting and labelling multiple players. The two steps for the blobs are a colour-based grass detection to attribute the blobs to the player and a multi-scale edge-based detector to remove false positives. It accounts and works with occlusion [28, 63].

Colour-based segmentation benefits from the field lines' white colour, as they contrast with the field's green. The problem is that these segmentation methods require additional algorithms to remove the lines from their obtained results. Some authors propose analysing the density and size of the segmented objects to detect and discard lines [11]. As an alternative, other works [4, 61] use the HT for detecting lines and separating them from the rest of segmented objects [28, 63].

Supervised Learning methods were also applied, like Boosted Cascade Learning [58, 93], or Support Vector Machines (SVM) [7].

More recently, CNN's have taken over as one of the most used and competitive methods for object detection in general. As players and the ball are objects, these algorithms have been used extensively for this purpose. Lu et al. [59] proposes a method of a cascaded CNN that detects and classifies the players from different sports in their teams. It shows the ability to detect players under challenging conditions such as varying illumination, camera movements, and motion blur.

Zhang et al. [110] proposes an end-to-end Reverse Connected Convolutional Neural Network (RC-CNN), which is able of detecting players in multiple scales and several challenging scenes, such as motion blur, heavy occlusion, and multi-scale players images. It advantages its architecture of reverse connected module integrated to the convolution neural network that gets the

convolutional computation from deeper layer back to the shallower layer. With this, the former layers with lower receptive fields could obtain more semantic information and exploit the valuable local context in the playing games. The reverse module can be positioned in a regular CNN and in different layers that produce different results.

An interesting approach for this purpose is not to detect generic players on the field but to automatically recognise the number on their shirts [33]. It starts with an initial player detection based on oriented gradients (HOG) and linear SVMs, followed by a CNN trained to recognise numbers that receive as input the cropped images resulting from the previous detections [28, 63].

Rematas et al. [80] uses a Faster R-CNN to obtain an initial bounding box on the players and then uses a pose estimation to refine these same boxes. To separate the players, they use DBSCAN clustering. They have accounted that the estimated poses could better separate the players than just the bounding boxes. For this pose estimation, they use a convolution pose machine, from [96]. Komorowski et al. [54] developed an integrated player and ball detector, called FootAndBall, using a deep neural network.

The network uses a Feature Pyramid Network (FPN) where lower level features with the higher spatial resolution are combined with higher-level features with a bigger receptive field. This helps the ball's detection as larger visual contexts are considered. However, it still struggles with false detections from the players' heads or feet. This method is more suitable for high resolution, long shot, video recordings. Hurault et al. [39] use a Faster R-CNN with an FPN for soccer player detection and tracking using a self-supervised technique. It uses the transfer learning approach teacher-student. Law and Deng [56] presents their object detection approach where they detect an object bounding box as a pair of keypoints, the top-left corner and the bottom-right corner, using a single convolution neural network. This work is not developed for the particular application of player detection but could be applied to do so.

The one-stage detectors are also used for this purpose. Acuna [2] uses YOLOv2 to detect players in RGB images. It focuses on real-time detection, and this architecture has good results.

Ivašić-Kos et al. [40] compares a Mask R-CNN with a YOLOv3 detector for a handball dataset, where both were successful and had the predicted results: YOLO is faster, with good results, but the Mask R-CNN has a better recall, precision and F1 score. Naik and Hashmi [67] also present a DL-based YOLOv3 model for ball and player detection.

Şah and Direkoğlu [83] presents a comparison between a Faster R-CNN, YOLO and SSD for field hockey player detection. They apply transfer learning for player detection using Faster R-CNN. As a feature extraction, it is used a pre-trained ResNet-50 network. Features are extracted from activation-40-relu and fed into RPN and detection networks. Training is done via their own data. They also use SSD 300 with a VGG16 as a backbone architecture. YOLO was set with the pre-trained feature extractor, ResNet-50 network, extracted with an activation-40-relu. This comparison ended with observing that these complex neural networks fail to detect occluded players and players with low resolution (far from the camera). This misdetection happens because these networks are specially designed for large-sized objects with high resolutions, so after several hidden layers, information about small-sized players are lost in the network.

2.2.4 Player and Ball Tracking

The detectors deliver observations corresponding to the ball and the players, but they lack temporal correspondence. This correspondence can be obtained with a tracker to relate this detection data and form the respective trajectories. Tracking is a crucial aspect of content enrichment as it can provide many data to enrich and complement the game footage. It is also an essential aspect of game analysis, by the ability (joined with the topics mentioned above) to provide aspects like distance covered, top speed and other interesting statistics.

Player and Ball Tracking is an example of Multi-Object Tracking (MOT). Its objective is to identify and track the position of multiple objects present in a video so that the same unique ID is assigned to each in every frame it appears [40]. If done separately, the ball's tracking can be an example of Single Object Tracking. When MOT is narrowed to this football scenario, there are several related problems, such as occlusion and a close similarity between the objects [85]. Luo et al. [60] presents the MOT as a multi-variable estimation problem. It gives the following definition: Given an image sequence, we employ s_t^i to denote the state of the i-th object in the t-th frame, $S_t = (s_t^1, s_t^2, ..., s_t^{Mt})$ to denote states of all the Mt objects in the t-th frame. We employ $s_{is:ie}^i = s_{is}^i, ..., s_{ie}^i$ to denote the sequential states of the i-th object, where *is* and *ie* are respectively the first and last frame in which target i exists, and $S_{1:t} = S_1, S_2, ..., S_t$ to denote all the sequential states of all the objects from the first frame to the t-th frame. Note that the object number may vary from frame to frame.

Despite considering a hard job, because of the amount of other possible criteria, [60], differentiate MOT methods groups according to the following three criteria: a) initialisation method, b) processing mode, and c) type of output.

The MOT works can be grouped regarding the initialisation method into two sets: Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT). Briefly, DBT, commonly referred to as tracking-by-detection, happens when objects are first detected and then linked into trajectories. Most DBT focuses on specific targets, such as pedestrians, vehicles or faces. It is highly dependent on the object detector performance. DFT requires manual initialisation of a fixed number of objects in the first frame, then localises these objects in subsequent frames. The difference between DBT and DFT is whether a detection model is adopted (DBT) or not (DFT) (Figure 2.7).

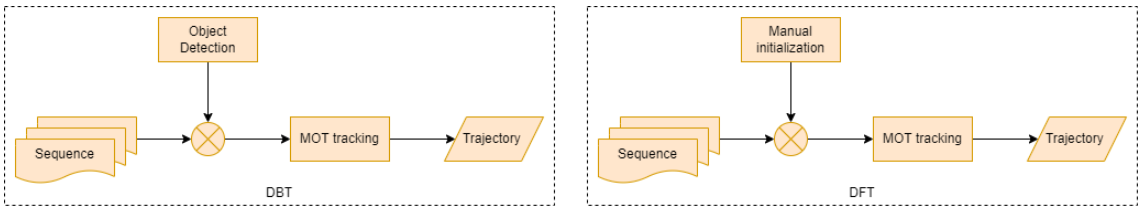

Figure 2.7: MOT tracking flows. Left: Detection-Based; Right: Detection-Free Tracking [60]

DBT is more popular than DFT because new objects are discovered, and disappearing objects are terminated automatically. This popularity is also connected to the better quality of the current object detectors. Nowadays, the attention is rising rapidly on applying CNN as a detector in

the tracking-by-detection method, since it can achieve a significantly fast speed and high accuracy when dealing with visual recognition tasks [85]. DFT cannot deal with the case that objects appear. However, it is free of pre-trained object detectors [60].

Regarding the processing mode, they can be considered Online Tracking or Offline Tracking. The difference is whether observations from future frames are utilised when handling the current frame. In Online Tracking, the image sequence is handled step-wise. Thus, online tracking is also named sequential tracking, so no further frames are considered. The results are represented by the object's location and its ID. Based on the up-to-time observations, trajectories are produced on the fly. In Offline Tracking, the image sequence is handled in a batch, so observations from all the frames are required to be obtained in advance and are analysed jointly to estimate the final output [60].

As for the type of output criteria, it classifies methods as deterministic or stochastic ones. The difference between these two methods primarily results from the optimisation methods. As the name suggests, the output is constant when running the methods multiple times in deterministic tracking. As for stochastic tracking, the output results of stochastic tracking vary from time to time [60].

Once the players have been detected, it is possible to track them to register their positions during the video sequences. When trying to track the detected objects, [28] aggregates the methods in three groups: Key-Point Tracking, Template Matching and Silhouette Tracking. Their according definitions are:

- Key-Point Tracking (KPT): The objects detected in consecutive images are associated by establishing relationships between their keypoints. These relationships are then used to estimate such objects' position in subsequent frames. Some typical KPT algorithms are the KF [45] and some extended methods, the Particle Filter, the Joint Probabilistic Data Association Filter (JPDAF), the Markov chains [63, 28].

- Template Matching (TM): These techniques aim to track the players based on their appearance. The players have a template initially associated, manually annotated or detected, used for the similarity comparison of the rest of the frames. Some examples of typical TM methods are those based on Cross-correlation and Normalized cross-correlation, or on Sum of squares differences [63, 28].

- Silhouette Tracking (ST): These techniques track by estimating the region occupied by the object inside their detected silhouette. Approaches like Contour Tracking and Shape Matching are the most common. This approach has various disadvantages: the shape of the players' variation, occlusion, and camera resolution. As these problems are difficult to contradict, this method is not commonly used [63, 28].

Various approaches use the Kalman Filter (KF) to perform this tracking step [10, 41, 50]. KF is used to estimate the state of a linear system where the state is assumed to be Gaussian distributed. Kalman filtering is composed of two steps, prediction and correction. The KF is a

recursive estimator, which means that only the estimated state from the previous time step and the current measurement is needed to compute the estimate for the current state. Particle Filter, also known as condensation or sequential Monte Carlo (SMC), has been widely applied, based on the limitation of KF in the case of non-linear movements. Typically, particles representing possible solutions to the tracking problem (i.e. player candidates) is initialised, multimodal distribution [34], and their next states are predicted for propagating them. It is robust to inaccurate segments. Each particle is weighted and the mean state of the player is estimated [63, 28].

Markov Chain Monte Carlo (MCMC) is also another option for keypoint trackers. MCMC is a data association that approximates a solution to a combinatorial optimisation problem by randomly searching the space instead of enumerating all possible associations. Multi-player tracking with MCMC was presented by [93, 58] in which the complete detection and labelling results were taken as observations. Then, the best association was represented by configuration on a neighbourhood graph which optimised Gibbs distribution. For this purpose, Metropolis-Hastings was adopted to estimate the optimal solution, and observations of missed players were interpolated. Other methods for data association are Multiple Hypothesis Tracking (MHT), and Nearest Neighbour Association (NNA) [63, 28].

Somewhere between some KPT and TM, there is also some proposals for graph-based tracking (Figure 2.8), which has been one of the most used solutions for player tracking, with good performance on handling occlusion. For graph-based tracking, nodes in the graph represent the segmented players, while each edge represents the distance between nodes in consecutive frames. Moreover, each node stores blobs' features (e.g. size, area, centre and colour). Different approaches use different strategies for the information stored in the edges, like the blob's information (e.g. velocity, orientation, and colour) or the correlation between players corresponding to the edge. After the representation, player tracking was usually performed by searching the optimal path in the graph. The methods using TM [63] have the challenge of the object changing with its movement, especially if the camera is moving and occlusions. The template can be manually annotated like, [89], or it can be detected [107].

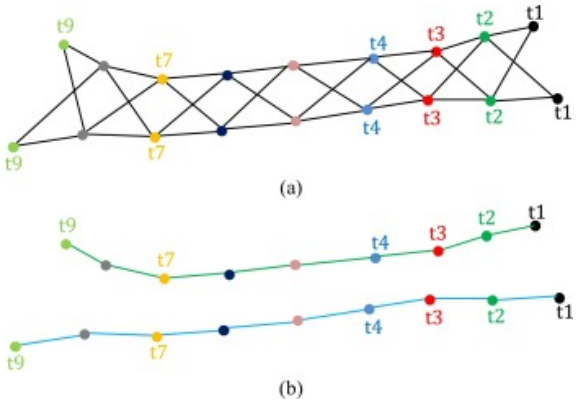

Figure 2.8: Graph-based tracking representation [63] a) Graph representation for two near by players (tn denotes detected player in n-th frame), b) extracted trajectories using the graph representation.

The work presented above only concerns player tracking, but ball tracking work also uses these same methods. It follows the same logic; tracking algorithms can be applied after detecting the ball. KPT algorithms are used, but also the Lucas-Kanade and overlapping bounding boxes analysis. Data associations can also be achieved using the Markov Chain assumption and a predictive motion model. This method can be a computationally complex problem if the detections used as inputs contain noisy data. Alternatively, ST-based methods are also used, with some Particle Filter inspired algorithms, to fight noisy data.

More recent MOT algorithms use these previous algorithms, but their real focus is on detecting as accurate as possible, tracking-by-detection. Thanks to improvements in the performance of object detectors, and thanks to the ability to deal with challenges such as cluttered scenes or dynamics of tracked objects, tracking-by-detection has become a leading paradigm for MOT [40]. These methods make the tracking step an association step for the detections from the detectors. Consequently, the application relies on the detection step to be as good as possible.

Bewley et al. [15] presents SORT, an approach to multiple object tracking where the main focus is associating objects efficiently for online and real-time applications. The first step for the SORT algorithm is detecting the desired objects. It is founded on a CNN-based detector and mainly focuses on frame-to-frame association, combining the KF and the Hungarian algorithm. The authors approximated the inter-frame displacements of each object with a linear constant velocity model [85]. Bewley et al. [15] reinforces that the detection quality significantly impacts tracking performance. The comparison to other trackers is similar to the object detectors. The state-of-the-art methods that achieve better results are slower, whereas high-speed algorithms have lower accuracy. Citing [15] "SORT combines the two desirable properties, speed and accuracy, without the typical drawbacks.". Shi [85] tested the SORT algorithm with a CNN core and obtained a good performance.

However, while achieving overall good performance in tracking precision and accuracy, SORT returns a relatively high number of identity switches. These switches happen because the employed association metric is only accurate when state estimation uncertainty is low [97]. Consecutively, it may have difficulties tracking through occlusions. This complication is a frequent event in football matches, so it needs a solution to this problem. DeepSORT [97] was developed to overcome the SORT's high number of identity switches problem. Briefly, the developers do this by replacing the association metric with a more informed one that combines motion and appearance information. This way, DeepSORT receives the detections' features as input and improves robustness against occlusions and missed tracking whilst keeping the implementation simple.

DeepSort's track handling is broadly identical to the original SORT version. The tracking scenario is defined on the eight-dimensional state space, $(u; v; \gamma; h; \dot{x}; \dot{y}; \dot{\gamma}, \dot{h})$. These are the bounding box centre position (u, v), aspect ratio γ, height h, and their respective velocities in image coordinates $(\dot{x}; \dot{y}; \dot{\gamma}, \dot{h})$. It then uses a standard KF with constant velocity motion and a linear observation model with the bounding coordinates as the direct observation of the model state. For each track, DeepSORT counts the number of frames since the last successful measurement association a_k. This counter is incremented during Kalman filter prediction and reset to 0 when the track mea-

surement is associated again. To this extent, tracks that exceed a pre-defined maximum age A_{max} are considered to have left the scene and are deleted from the track set. Thus, the number of tracks can be limited to recent tracks, lightening the algorithm associations. New track hypotheses are initiated for each detection that cannot be associated with an existing track. These new tracks are classified as tentative during their first three frames and during this time, we expect a successful measurement association at each time step. Tracks that are not successfully associated with a measurement within their first three frames are deleted [97].

The assignment step aims to associate the predictions state from the Kalman Filter with the new measurements from that frame. Accordingly, DeepSORT uses the Hungarian algorithm, same as SORT. Additionally, DeepSORT integrates motion and appearance information through two appropriate metrics. To incorporate motion information, it uses the (squared) Mahalanobis distance, also known as covariance distance, between predicted Kalman states and newly arrived measurements as expressed in Equation 2.1, where y_i and S_i represent the mean location and the covariance matrix of bounding box observations for the i-th track, and d_j is the j-th detected bounding box. Tracks have three possible states: Confirmed, Tentative and Deleted. Fundamentally, confirmed tracks are the ones associated with some detection; tentatives are detections that were not associated yet to already tracks; deleted are tracks that do not associate with any detection for some time and are considered to have left the scene.

The second metric measures the smallest cosine distance between the i-th track and j-th detection in appearance space as shown in Equation 2.2, where r_j is the appearance descriptor extracted from within the j-th detected bounding box, and R_i is the set of the last 100 i-th track appearance descriptors, r_k^i. These features are extracted by a CNN that returns the best features from a player detection bounding box and concatenates them into a feature vector.

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \tag{2.1}$$

$$d^{(2)}(i,j) = min\{1 - {r_j}^T r_k^{(i)} | r_k^{(i)} \varepsilon R_i\} \tag{2.2}$$

The two metrics complement each other serving different aspects of the assignment problem. On the one hand, Mahalanobis distance provides information regarding the possible objects' location based on motion. On the other hand, the cosine distance supplies information about the measurements' appearance. Therefore, the Mahalanobis distance affects short-term predictions since the motion information is valid only for the subsequent close frames and the cosine distance is beneficial for long-term predictions as it compares the appearance of tracks over time as these are not likely to change during a match.

The two are combined in a weighted sum (Equation 2.3), controlled by the λ parameter.

$$c_{i,j} = \lambda d^1(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{2.3}$$

Another exciting aspect of this method is the matching cascade strategy used. The Kalman

filter can not accurately predict the target state when the object is occluded for a long time because when an object is occluded for a more extended period of time, subsequent Kalman filter predictions increase the uncertainty associated with the object location. In order to resolve some errors allied with this uncertainty, [97] implements a matching cascade that prioritises targets who appear more frequently, reducing track fragmentations and unstable tracks. Furthermore, in a final matching stage, IOU association, similar to [15], is used on the set of unconfirmed and unmatched tracks of age $n = 1$. This helps account for sudden appearance changes, e.g., partial occlusion or oversized bounding box, and increases robustness against erroneous initialisation [97]. Ivašić-Kos et al. [40] used DeepSORT in a handball dataset and achieved good results. It does not help conclude much about its application to football datasets, as the distance between the player and camera is bigger, making a difference in the appearance information; however, it is good to know its relevance to sports environments.

Other interesting methods that follow this line of thought are TRACKTOR++ [14], FairMOT [112], TransMOT [23], ByteTrack [111], IoUTrackers [16, 17].

As discussed in Section 2.2.1, when multiple views cover the same area and detections, there is the need to have methods that can perform the fusion of these detections. There are two approaches for generating the fused trajectories: fuse and then track and track and then fuse [64].

2.3 Market Analysis

As this area is trendy and has various monetary benefits, there are already some solutions that allow the automatic extraction of useful metadata in sports. The following list presents some of these solutions already in the market and the according functionalities:

- **Catapult Sports** - is a sports performance analytics solution capable of reviewing sports events using multi-angle footage and a specific tagging and labelling system, with enhanced playback controls support for key moments review. It also allows integrated wearable-based data to monitor athletes' performance. It lacks an easy way to deploy the solution in the cloud continuum and simulate the Big Data workflow before the event starts, like the DataCloud project aims to allow. It is also not integrated with an ingest system, which means that the resulting videos need to be later transformed before being made available online.

- **LongoMatch** - is a video analysis tool that allows live video capture and review enables to analyse and respond in real-time during games and practices. It is possible to capture video footage using a webcam, IP, HDMI or SDI camera or import it directly from the hard drive. Review live tagged events and share them with the team over Wi-Fi. Essentially, it offers manual annotation and the capacity to review the matches using up to 8 cameras.

- **VO Sports** - is a multi-sports video analysis software that allows video capture with customised solutions for various sports in-real time during matches and training sessions. The

registered data is stored and processed in a central system that automatically generates dynamic match reports, players and scouting reports, as well as creating and editing the videos online.

- **Spiideo** - is also a multi-sport video analysis software that allows video capture with live tagging for desktop, mobile, iOS app and Apple Watch. Drawing tools on live or recorded video gives players and coaches visual feedback immediately during practise or games. Its intelligent camera system and cloud platform automatically capture the entire field. It features an AI game tracker and allows event detection. Spiideo provides its own camera systems, which are scalable and flexible to meet any recording situation.

- **Nacsport** - is another multi-sport video analysis tool that allows live video capture directly onto a computer from a video camera, IP camera or YouTube and collect real-time data. Allows content annotation. Use the tools on offer to produce a more advanced analysis. Interactive Dashboard creates eye-catching charts and graphics with the Nacsport Dashboard, which is linked directly to the user videos for consultation in real-time.

- **Kinovea** - is a free, open-source video annotation tool designed for sports analysis that allows live video capture and content annotation. The content annotation feature is manual with some semi-automated tracking to follow the trajectories of points on the video. It is also essentially used for performance analysis and not for the analysis of the matches themselves.

- **DARTFISH** - provides performance analysis solutions with AI event tracking and data analysis in real-time. The application allows the user to identify key actions and tag the game using a preconfigured soccer tagging panel. It is also possible to review and edit key actions in real-time while recording the event. Based on it, the tool allows the user to illustrate what athletes need to improve using frame by frame playback, slow motion, drawings and instant replay.

- **Viz Libero** - is the sports analysis solution inside VIZRT, combining 3D virtual enhancements with powerful telestration tools. It creates captivating analysis and provides the perfect perspective to get viewers into the game. Viz Libero automatically tracks players without the need for special cameras and can enrich 3D photos of the sports action with analysis graphics. No infrastructure is required in the stadium, making Viz Libero's file-based workflow perfect for remote productions, studio applications, or digital platform clip creation. In many cases, Viz Libero can simply run from a laptop.

- **StatsPerform/Qwinn** - one of the functionalities that this system provide is SportVU, football analysis and performance statistics solution. This allows tracking data and event data synchronisation to provide: comprehensive match coverage; data collection for both teams plus the ball, providing a holistic view of the game, helping develop more tactical match strategies and in-game player tracking coupled with off-field training data to help reduce

injury by highlighting performance deviations. It works using 4k resolution cameras to provide the most accurate data. One of their goals is to control and extend the media/broadcast experience far beyond the stadium. Whether written or video, content can be generated based on individual fan preferences and the subjective excitement of each event to create a truly personalised experience on an individual basis.

Chapter 3

Data Acquisition Protocol

Generally, a sports broadcast does not cover the entire field. In order to augment a broadcast audience's experience, the entire field should be visible to bring the maximum information possible for the user. As with all machine or deep learning-related solutions, the available dataset is essential for the final product and demonstrates the results obtained. Hence, the protocol definition and process designed to capture is an important work that this dissertation presents.

This chapter presents each data acquisition's different phases and objectives, the final solution, and its advantages and disadvantages.

All of the match and training footage was recorded by members of MOG, including the author, at the Estádio Municipal de Aveiro.

3.1 First Acquisition (Training Session)

The first data acquisition was a rough start to understanding the best position for the cameras. At the time, automatic homography estimation was the project's current area of development, which led the dataset to be focused on that matter. Regardless, it was still valuable for the further modules. This acquisition had various intricacies, such as not being able to go to the upper level of the stands (social communication/media area) and the hardware for the footage. The footage was obtained from personal phones with some tripod help but not entirely.

Figure 3.1: First acquisition image examples. Left: Parallel option. Right: Diagonal

At this stage, diagonal and parallel shots were taken from the stands in more than one place (see Figure 3.1). Diagonal shots give a wide-angle and much information, but they turned out to be very distant from the field. This strategy would obligate more cameras to be used compared to a parallel camera strategy. Similarly, the parallel strategy also had issues. Parallel shots needed to be higher since the occlusions were intense, and some objects covered parts of the field, like the bench on the left image in Figure 3.1. In conclusion, this acquisition was highly influenced by the negative variants and so the resulting images were not the best.

3.2 Second Acquisition (Training Session)

Having solved the homography problem, with a manual approach, the aim of this capture was more accurate, test various camera setups to help detect the players and minimise the associated problems, occlusion mainly. The footage was obtained in the upper level of the stands (media area). The manual process also led to the definition of a setup with fewer cameras to minimise the calibration effort. Naturally, the height difference made an immediate difference in the cameras' AoC and the occlusion handling. The cameras were located 20 meters above the field and had a tilt of around 15° (see Figure 3.2).

Figure 3.2: Example of the cameras used

For this acquisition, two video cameras were available along with tripods. However, there was still some issues with the hardware provided. The cameras used differed substantially in image quality, mainly caused by the different lenses. One was a CANON 700D and the other a CANON 600D. The videos were captured in Full HD with 30 frames per second.

As for the protocol, essentially, two setups were tested. Both have the cameras along one side-line. Protocol one has two cameras requiring that each camera catches each midfield. This strategy has various advantages, like covering the entire field with only two cameras and facilitating and accelerating the fusion of extracted data from each camera. Also, its position and orientation can be helpful to fight the players' occlusion since they have significant overlap (see Figure 3.4), and their different view is sensitive to it.

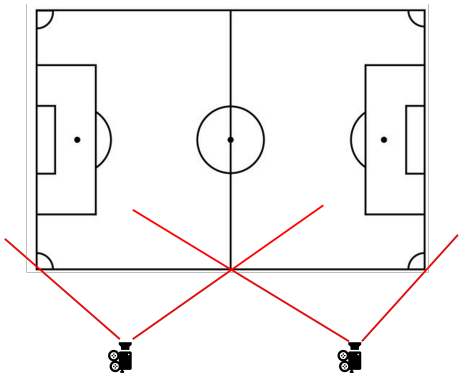

Figure 3.3: Protocol one illustration

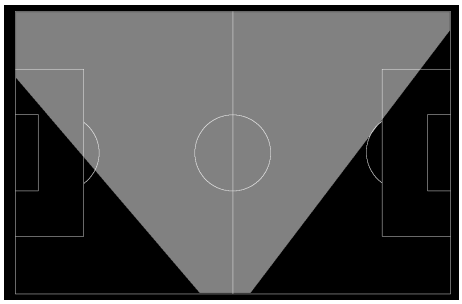

Figure 3.4: Real overlap from the second acquistion images (protocol one)

The other strategy, protocol two (see Figure 3.5), had three cameras planned, the same two for each side but one extra in the middle. This extra camera could be helpful if all of the midfield could not be captured with the lateral cameras and to bring more accurate detections. The cameras would have a smaller coverage area and, therefore, be better at distinguishing players.

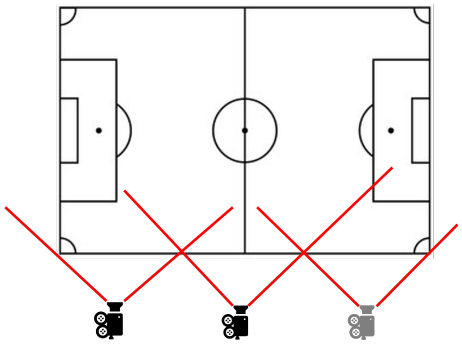

Figure 3.5: Protocol two illustration (right camera is faded because only two cameras were available)

When testing protocol one, the field had total coverage, which was a good indicator that it was admissible for the final protocol. Also, the players were noticeable, which was a critical element. Protocol two was tested but not thoroughly since protocol one appeared to be successful, and the number of available cameras for this acquisition (two cameras) only allowed to test half of the setup.

3.3 Final Acquisition (Official Match)

Whilst using more cameras enables detailed field coverage, it also increases the computation effort when matching the extracted data. Hence, the counterbalance between the number of cameras and the fusion algorithm complexity must be evaluated.

Therefore, with the previous footage acquisition, protocol one (Figure 3.3) was chosen as the final protocol. After this, a complete match was recorded to test the various modules developed and help develop the following. After some hardware troubles, the first half of the game was lost, so this capture returned 45 minutes of game footage. The cameras were still different for this occasion: a CANON 250D and a CANON 700D. This distinction is noticeable in the image quality, which may affect the results obtained.

This acquisition emphasises one of the problems related to the broadcast and recording of a live football match, which is the illumination present in the scene. This condition is demanding since it is uncontrollable and can vary during and for different matches. As the game hour changes regularly, their scene may be very well illuminated and have no complications for the recording. Contrarily, it can also have shades in the images or excessively bright areas that make capturing the footage difficult. This last exposed problem occurred at the date of this acquisition, where the cameras could not cope very well with it (one better than the other), having a big shaded area and a consequent very bright field area. This field's bright area vanished throughout the match, so in the final minutes of the footage, the field has an almost heterogeneous brightness. Figure 3.6 shows the brightness issue, where the shadow area is moving through the match. Likewise, it exhibits the difference between the cameras in managing the scene's brightness.

Figure 3.6: Images from the the final acquisition match footage. Top: Beggining of the 2nd half. Bottom: Midlle of the 2nd half.

A disadvantage of having more than one camera in the setup is their synchronisation. Because the metadata is extracted for each camera, it must be associated with a timestamp when fusing, creating the necessity of video synchronisation, so the timestamps are correct. The synchronisation is done a priori as the videos are pre-processed before entering the pipeline for testing.

The cameras were not placed in the exact place as the previous acquisition date, though in a very similar one, following the description provided above. As noticeable in Figure 3.7 the strategy rules were maintained, but the cameras were closer to the field, creating a smaller overlap while capturing a more detailed image of the on-field action.

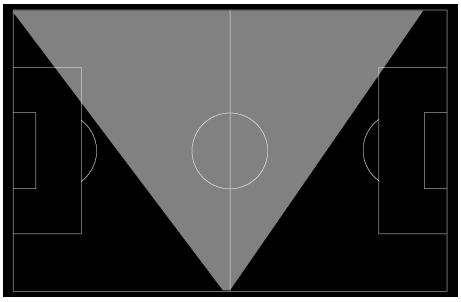

Figure 3.7: Real overlap from the match images

Chapter 4

Extracting Tracking Data for Broadcast Enhancement with Computer Vision

This chapter demonstrates the details of the developed modules in this dissertation to reach MOGSports's goal. As mentioned above, the objective is to create visual representations to enrich a sports broadcast. In order to extract the information for this objective, the efforts can be divided into various more specific modules, as mentioned. Section 4.1 describes in detail these different modules' flow and communication. The development of the individual modules is presented in detail further.

4.1 MOGSports

This section describes the MOGSports modular pipeline. As mentioned, MOGSports business case focuses on the broadcast user experience enhancement by creating an automated video annotation and metadata extraction tools. Figure 4.1 illustrates the pipeline of the proposed system to achieve these enhancements.

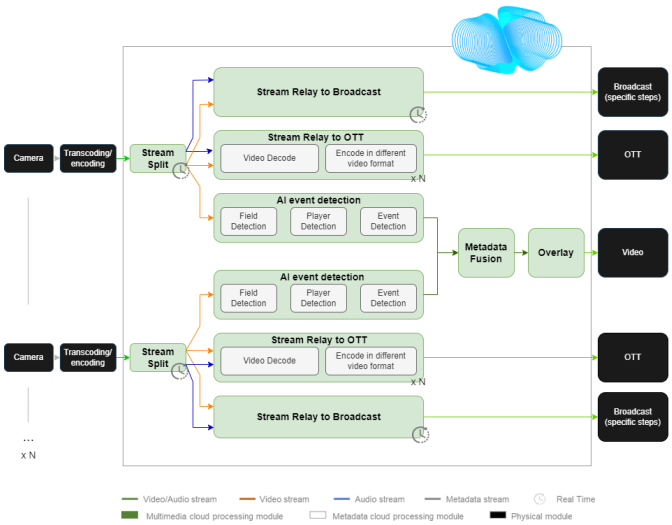

Figure 4.1: MOGSports pipeline

The pipeline (Figure 4.1) portrays a multi-camera system moulded to accept *N* cameras whose content is distributed through diverse modules, such as various stream forwards to assist the usual broadcast camera as well as the AI Event Detection and Metadata Fusion to create the enhancements. The stream forwards are slightly out of the scope of this dissertation and will be presented shortly. Conversely, the focus will be on the AI Event Detection and Metadata Fusion modules, where this dissertation's development concentrates. The tasks of the various modules present in this pipeline are:

- **Transcoding/encoding**: Physical equipment that receives the uncompressed media data from the professional camera via SDI connection and processes it to compress and wrap the data in standardised IP/UDP/TCP.
- **Stream Split**: Virtualised module that receives a media standardised data stream and splits it on each media essence stream (Video and Audio). The split essence streams still go through IP/UDP/TCP connections.
- **Stream Relay**: Virtualised module that receives compressed essence streams and wraps them in a standardised data stream that then exits the DataCloud platform.
- **Video Decoder**: Virtualised module that handles a compressed video data stream and decodes the video to provide an uncompressed data stream.
- **Video Encoder**: Virtualised module that creates a new compressed video data stream from the uncompressed video stream. This compressed stream can be different from the original one. This module is optional.
- **AI Event Detection**: Virtualised module to extract relevant information in the uncompressed video data stream. The data created by this module is associated with the respective video frame timestamp and camera identification. It composed by three smaller modules:
  - **Field Detection**: Virtualised module to identify the mapping between the input image and an image representing an areal-view of the same field. This mapping translates into a homography matrix.
  - **Player Detection**: Virtualised module to detect the players on the field and retrieve that information.
  - **Event Detection**: Virtualised module that aims to detect certain events occurring during the match to create valuable metadata.
- **Metadata Fusion**: Virtualised module to aggregate and process the data created from the individual cameras.
- **Overlay**: Virtualised module to annotate the chosen visual representations into the original video.

To summarise, the dissertation's critical stages are the following: The Field Detection module computes the homography of the associated camera with a top-view field image and passes it to the detection module so it can map the detections. Next, the detection model outputs the detected on-field players, passing them to the tracking algorithm. After this, the tracking algorithm identifies or re-identifies the detected players' bounding boxes and maps their position into the top-view image with the computed homography. With the players' representations, this top model allows a better understanding of the game. Finally, the tracking module sends this information to the Metadata Fusion module, which combines it with the other metadata from the other cameras. In this sense, the fusion module must adjust accordingly, accompanying the camera scalability. The proposed solution describes an acquisition protocol (chapter 3) that uses only two cameras to extract information from the entire field at all times.

Event Detection is the last module and problem that MOGSports intend to solve. It comes last in the pipeline because it is a slightly different problem and can be developed with the previous metadata extracted along with video data. Thus, this module only makes sense to develop once the previous have achieved all requirements.

As mentioned, the solution must be deployed in a Cloud environment representing an extra effort to integrate the proposed solution. The virtualisation of the modules implies the need for a particular communication method. In this domain, the communications must go through the network. The Stream Split and the AI Event detection can communicate through various types of streaming and their different protocols, but the chosen was Real-Time Messaging Protocol (RTMP), using the Transmission Control Protocol (TCP). As the stream plays, the AI Event Detection loads and processes the frames, sending their extracted data to the Metadata Fusion afterwards. The communication between the AI Event Detection and the Metadata Fusion modules is conducted through WebSockets. As the metadata fusion module initialises, a TCP socket is created that is actively listening, waiting on the previous modules to send the metadata extracted through this channel.

4.2 Homography Estimation

The problem and proposed solution presented in this section relate to the MOGSports' Field Detection module and the task of Field Registration. As mentioned in Chapter 2, the latter is associated with the homography matrix estimation between two 2D images that allows the transformation from points to one another. Game footage and field top view images have the field of play in common; however, they are projected differently in both, meaning that there is a homography that maps these two images. This module receives the first eligible frame from the stream split and uses it as the image to map.

Since the setup cameras are static, the homography computation only needs to happen once at the match's beginning. The premise of a constant homography, one of the main advantages of the setup, makes the registration process exponentially cheaper when compared to a moving camera, where it can change several times per second.

This problem had two different approaches, automatic and manual. An automated solution is the objective of MOGSports; however, the manual user approach was implemented in sight of the automatic approach results, the complexity of the problem, and the time to solve it.

4.2.1 Automatic Homography Estimation

The Homography Estimation was pursued with the two-GAN model approach, used by [21] and using a similar but more straightforward approach to find the homography matrix. Two-GAN is a popular, efficient and accurate algorithm that performs the field line segmentation. Allied with its quality, it has an open-source version, which facilitates its employment. The work from [21] utilises the two-GANs to detect field markings in a football field image. After this segmentation, they extract its features with a siamese network, using it to query a camera pose related to it. After that, they refine the camera pose by the distance image and the Lucas-Kanade algorithm. Then, they estimate the homography matrix with the nearest neighbour algorithm to the testing image with this new image.

The proposed method performs the homography estimation differently, using a feature-based method. The two-GAN model returns the line segmentation in a binary image - white lines and a black background representing the field. Similar to the two-GAN output, the field top view image (see Figure 2.5) has a black background and white lines. Therefore, since it is the same object (football field always has the same format), projected in another view, matching features, most probably corners, can be discovered in these images. Thus, the homography matrix can be calculated with at least 4 pairs of corresponding points.

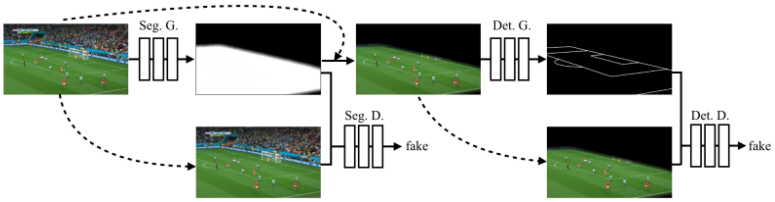

Figure 4.2: Two-GAN model architecture [21]

The two-GAN strategy combines a GAN that segments the field of play and another that draws the line markings as the final output (see Figure 4.2). As for the model training, the images generated can be controlled to be more naturalistic since some notion of the authentic images is an input for the generator. Both the networks train separately and are then merged via joint training. Along with the training implementation, an open-source dataset, World Cup 2014 [38], additionally annotated for this problem, is used. Likewise, various non-broadcast images are also part of the training dataset.

Chen and Little [21] report that the two-GANS have good results in avoiding other white lines that may be present in the image, something that can happen if a classic threshold operation is performed. This is also broadly useful for the implemented acquisition protocol, as it catches a large amount of the stadium other than the field of play.

The segmented output then has to be processed to find the homography. By analysing this method's outputs, it had many incomplete lines. So, a line detector could be helpful to complete them and increase their similarity to a more real field marking image. For this purpose, the HT method is applied to join the missing parts of the drawn field lines. This addition led to more robust lines, but even though it helped, it still required a better performance from the two-GAN network.

Figure 4.3: Two-GAN results for the first dataset images. Top: Original image. Bottom: Field Line Segmentation

Figure 4.3 shows the poor results of this technique regarding the images from the first data acquisition (Section 3.1). The two-GAN network performance does not entirely cause these results. The images are not from the best angles, and the lines do not have much contrast due to the image brightness. Furthermore, the distance to the field is significant, and the position was low, leading, in some cases, for players to block the lines and difficult the Two-GAN job. When developing this module, the available dataset was the content captured in 3.1, so other images considered similar to those considered as the final protocol were also tested. These pictures were taken from fans in football stadiums and were extracted from [1]. Figure 4.4 reflects the results for these images. The lines were not accurately segmented, but the results improved with these images. It is noticeable that the model performs a satisfactory field segmentation but has some difficulties for further lines with less visibility. In essence, it does not reach the requirements yet.

After the segmentation, the images must go through the homography estimation. Using a feature-based method to estimate the homography means that a feature detector tries to find features that it considers somewhat meaningful in both images to match each other. A few detectors were tested, and SIFT was the most promising. Unlike others, it is invariant to various geometrical operations, which may help extract good feature points. For matching these features, the method RANSAC is the most suitable for its proven help in this homography computation task. Figure 4.5 shows the steps for the proposed automatic approach. The example uses a video-game image as the contrast is very good, and the field view is similar to the proposed. It shows a good penalty area segmentation; however, the remaining lines are not very well segmented. This affects the

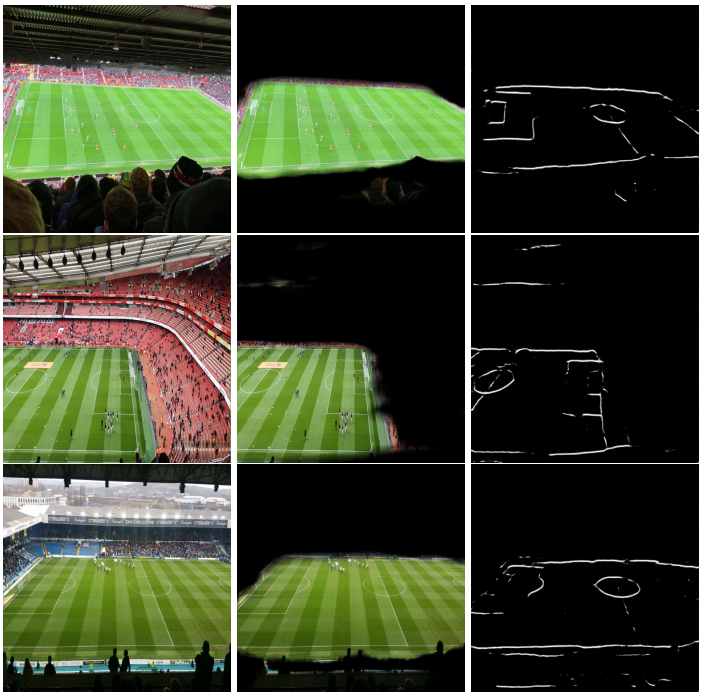

Figure 4.4: Two-GAN results on images similar to the proposed. Left: Original image. Middle: Field Segmentation after first GAN. Right: Field Line Segmentation

following steps as the field is not represented correctly which is necessary for the feature matching. Nevertheless, the features extracted and consequent matching are not ideal, leaving room for improvements.

After the incomplete results, various reasons led to the conclusion that the training dataset was not fit for this problem. The two-GAN training data is mainly from a broadcast video, and despite having non-broadcast images, they are not like the images similar to the protocol proposed in Section 3.3. Hence, as GAN networks require a tremendous amount of data to create a satisfactory generative model, and the dataset used is not generic enough to provide the training necessary, the results are unsatisfactory. The extra labelling effort would be too much to conciliate with the rest of the project's objectives, so it was not pursued. Identically, the matching step was also not very successful as most features were hard to distinguish due to the failed segmentations. The two images represent, in theory, the same entity, but one of them has a noisy representation making it challenging to match.

Also, despite these results, this methodology is thought as the best when analysing the state of the art and the proposed methodology, so no further methodology was tested.

4.2.2 Manual Homography Computation

As previously stated, the homography matrix calculation needs at least four corresponding points from each image, meaning eight points. Therefore, this assignment becomes lightweight for manual calibration, as a one-time eight-point selection is not very time-consuming for a system user.

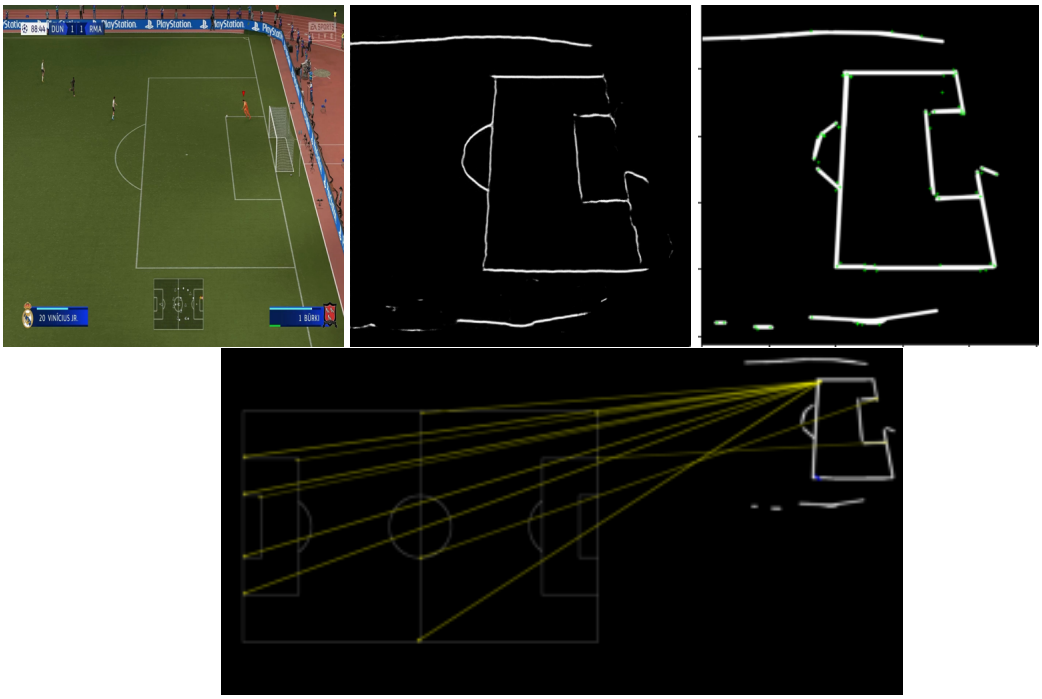

Figure 4.5: Feature extraction and matching example. Top Left: Original Image. Top Middle: Two-GAN Segmentation. Top Right: Feature Extraction (green). Bottom: RANSAC Feature Matching

After the corresponding points are selected, the homography matrix is computed, which is kept throughout the whole match. Assuming that the camera does not move, it will be accurate at every moment. Figure 4.6 demonstrates the mapping from the footage image to the top-view, especially the player position in this new field representation.

Figure 4.6: Match image homography transformation. Left: Original match image. Right: Warped image

It must be highlighted that the eight points mark is just the minimum number of points that one must select for the homography calculation. It has good results and requires minimum manual work. More corresponding points would make the homography more robust, but it takes longer.

Identically to the number of points, the distance between the selected points also weights for the homography matrix robustness. For example, a higher image would have more field coverage

leading to more spaced points which would benefit the homography calculation (see Figure 4.7). This happens because the points that lie far from the plane the four selected points create are considered occluded, leading to associated errors. These errors must be considered for further detection mapping and location comparison to the other camera.

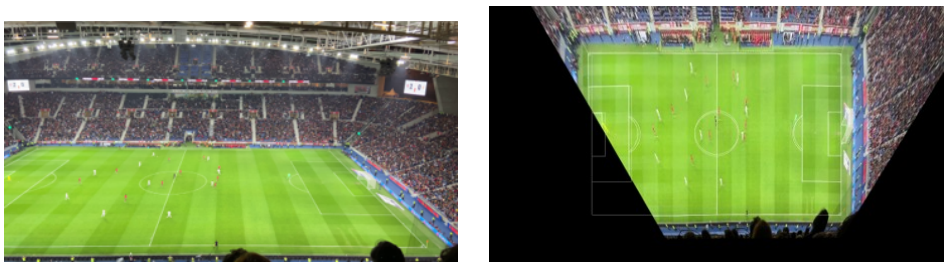

Figure 4.7: Homography warp example. Left: Original match image. Right: Warped image

4.3 Player Detection

Detection is vital for this methodology because it underpins other tasks, such as tracking and its use of re-identification.

Detecting the players requires a fast object detection algorithm. As presented in chapter 2, the one-stagers detectors are faster but have reduced precision compared to the two-stagers. This project is targeted at a real-time cloud environment, where the fastness of the methods is an integral aspect. With this in mind, the YOLO algorithm was selected. YOLO is an algorithm with multiple versions, with discoveries and state-of-the-art improvements emerging (see Section 2.1.1).

Some effort went into choosing the YOLO algorithm for this development. The two algorithms taken into consideration were YOLOv4 and YOLOv5. Considering the arguments present in 2.1.1, YOLOv5 was the chosen algorithm. The proposed solution aims at being fast, and YOLOv5 makes it happen on all fronts. Most notably in the detections, its integration and possible training or fine-tuning. It also maintains active discussion and improvements on its repository, making a difference compared to other versions. Despite not being as flexible in structure as YOLOv4, having an in-depth training effort was not expected, so this decision did not have a substantial weight in the final decision.

Coupled with these causes, exploiting the main difference in YOLOv5, auto-learning anchor boxes made this algorithm more attractive. Persons bounding boxes are not a class that requires very different anchor boxes as their positions are predominantly identical. The same does not happen with players. Players adopt positions during a match that may not embody what an average person would look like, which can bring a perfect match with this version contribution. The bounding boxes can also be small for distant players benefitting from this strategy. Player Detection, as with any object detection problem, has its related issues. For a usual scenario, occlusion is a typically moderate problem. However, this becomes a tremendous problem in a football match where players have multiple challenges and confrontations fighting for the ball. Moreover, there

are parts of the match with enormous agglomerations of players, like corner kicks or free kicks near the penalty area.

Likewise, the speed at which the players move can become a problem for their detection. Swift movements from the player can create motion blur in the image, which presents a problem for object detection.

Furthermore, the players are not the only persons on the field. Various entities (e.g. security guards, ball boys, cameramen, coaches) stand around the field. As the model is directed to detect persons, it is expected and not incorrect that the model detects them. However, these detections should be filtered so that the metadata released relates only to the in-field action.

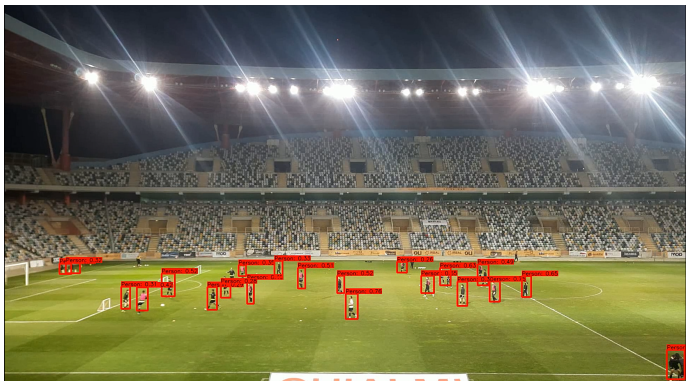

Figure 4.8: Detections with the first acquired dataset

Further training may be a solution for these problems. For this type of object detection, fine-tuning may be influential so that the model can find the high-level features that help it perform better. This extra training would require labelled data from football matches, presumably from similar images to those presented as the protocol for this solution (see Figure 3.3). Despite the likely improvements, this was not pursued as it would mean a considerable labelling effort that was not considered essential for this phase.

Finally, players are no more than persons in a field playing a sport. This knowledge of what a player looks like can be valuable for the detection problem, as person detection has a variety of fine-tuned models that can accelerate and help this specific task. Along with the state-of-the-art research, an exciting YOLOv5 model fine-tuned to crowded human environments was discovered, which inclined, even more, the algorithm choice. This model holds the YOLOv5m architecture and is fine-tuned to identify persons and their heads. Initially, the head identification seemed promising as it could help with some partial occlusions but was rejected after the first tests.

The YOLOv5 model usage has two main inputs, the IoU and confidence thresholds. IoU is used to measure the overlap of an object's predicted versus the actual bounding box. Its threshold value determines if that bounding box is chosen. The confidence is a value used to identify the network's confidence that a class object is inside a bounding box. Its threshold sets the minimum confidence the network has to have to identify that bounding box as that class.

The results from the first acquisition, shown in Figure 4.8, were promising and confirmed the assumptions of the chosen model. This dataset allowed the IoU and class confidence definition.

No labelled data was available, so the tests made were from human perception. The confidence value chosen was 0.35, as the players move rapidly or stand in running positions where the model tends to have low confidence in these cases but detects the player correctly. As for the IoU, the value was 0.45, which was the minimum for accurate bounding boxes.

As for the model utilisation, the system's input is a stream, a sequence of frames. Hence, for every frame, the algorithm outputs the respective detection bounding boxes that are then processed. These bounding boxes are represented with their bottom left and upper right coordinates. The players' position in the field is set where they stand, at their feet. Foreseeing that the player is in the centre of the retrieved bounding box, its position is defined with the bounding box's bottom centre coordinates.

After computing the homography matrix in the previous module, the system can transform the detection coordinates into the top-view representation. Therefore, once mapped by the homography, this point is the player's position in the top-view representation.

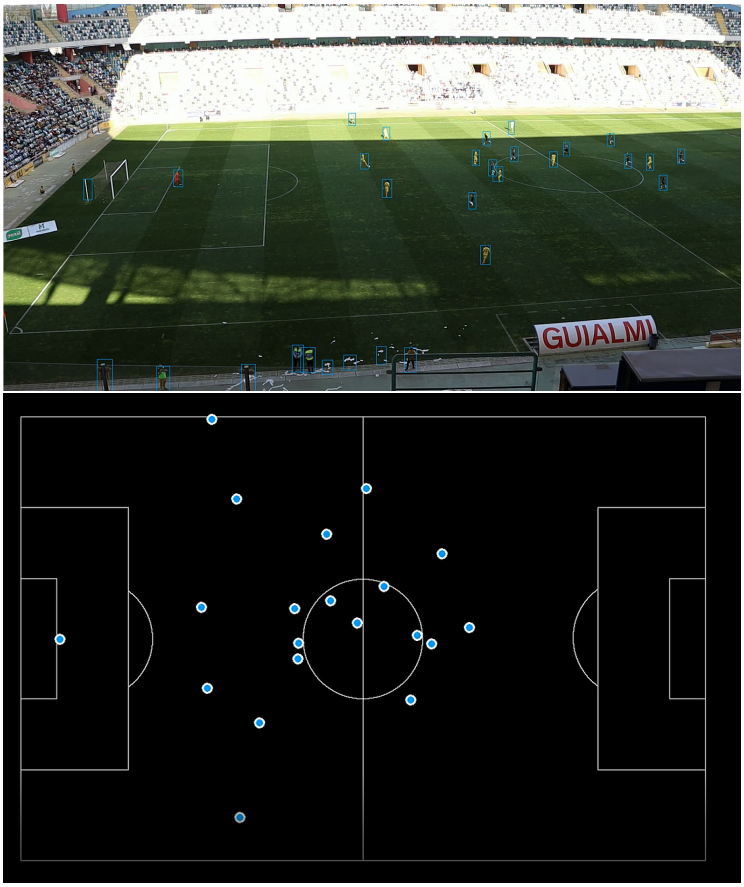

Figure 4.9: Left camera detections and respective top-view transformions

Furthermore, the top-view field image is predetermined, meaning that the field lines coordinates are known. Thus, the transformed coordinates can be easily filtered around the area of the field by their image coordinates. This effort is an essential step. In the first place, it updates the

detections to those that matter to the final objective. In the second place, it simplifies the following steps by reducing the detections and minimising its efforts. Hence, the off-field detections are filtered before being forwarded to the tracking algorithm. Figure 4.9 exemplifies this filtering with the security guards at the bottom of the image that are not present in the visual representation.

In Figure 4.9 there are no signs of occlusion, so all the players are correctly detected and transformed. However, it exists and presents serious challenges for this solution. These occlusions not only prevent us from detecting the players in this situation as it can ruin the visible player bounding box. This happens very quickly, creating unstable detections and consequently unstable transformed points and visual representations. In Figure 4.10, the bounding boxes' position vary fast from one frame to another, confirming the unstable detections' problem.

Figure 4.10: Occlusion examples. Each line represents a situation (ordered).

Analysing Figure 4.10 it is an issue that needs special attention. For the current protocol, the solution may be to train the model for these situations or use the other camera's view to fix these erroneous detections. Another option is to add more cameras so that the different perspectives would allow the occlusion handling. Despite this, there are a few examples where the model handles this problem with some success. The partial occlusion is handled reasonably in the last row of Figure 4.10, but the bounding box size has prominent errors.

This module's output covers the primary data needed to obtain the proposed visual representations; however, no temporal information is associated with the detections. Thus, a tracking algorithm can associate these detections over image sequences and provide further information beneficial for visual representations. This way, this module can output the on-field detections' transformed top-view coordinates to the metadata fusion module or send them to the tracking algorithm to obtain more complex information.

4.4 Player Tracking

Player Tracking allows the identification and re-identification of the detected players throughout the frames. Identifying a detection throughout frames also allows other information associated with it, like its kit colour, which can be calculated once or a few times rather than in all frames. Tracking can also create and represent more valuable information that can be used to augment the

user experience, e.g., more visual representations or information that creates player statistics. For this job, the author chose the DeepSORT algorithm. The chosen tracking algorithm is based on tracking-by-detection, so player detection is a fundamental step. Without a good detection model, the tracking algorithm will perform poorly.

The DeepSORT algorithm was selected to perform this task based on the state-of-the-art MOT algorithms. Despite not being the most accurate tracking algorithm, it is the one that offers the best time-quality relation. Furthermore, it offers an open-source implementation. DeepSORT's deep association intends to use feature extractors with a CNN to help with the detections' re-identification. The re-identification problem has been researched thoroughly, which sets the opportunity to find fine-tuned networks whose output features may be of great help.

The proposed methodology follows a single-camera tracking strategy that is then fused between cameras after application. The individual cameras do not have full field coverage with the specified acquisition protocol. Hence, it is natural that players leave the scene and can be re-identified as a new player (new id) when entering the field coverage again. These id swaps can be fought with the A_{max} parameter. However, as it is not an occlusion and the other camera presumably tracks that player, supposing that these tracking will be fused, it is not troubling. This way, this hyperparameter definition does not need to consider this case.

Furthermore, the hyperparameter λ allows the control of the weighted sum responsible for the matching assignment. This sum relates, to the motion and the features from the detections, as shown in detail in Section 2.2.4. These two fundamental aspects support the algorithm and are challenging to overput one for another. Motion is the base of the SORT algorithm and a vital component for player tracking, as the players' body gives much away from where they will move next. So, this hyperparameter's definition falls on the extracted features' importance. One may argue that the camera is far from the field, so the features may not influence the tracking as much as expected. However, in a match scenario, occlusion often occurs with the cameras' proposed height. DeepSORT's deep association is implemented for occlusion handling and the advantage it brings when objects are occluded for some time. So this way, this parameter will remain equal, and the weights for the two components, motion and features, are considered equally important.

Nevertheless, there are a few a priori problems with this algorithm. The camera distance from the field can, in fact, be prejudicial for the features association. Initially designed for long-term help, it is challenging that the features extracted may describe the player in this case. As the players are far away, use the same kit as at least nine players, and the camera may not have the best quality possible, this element's importance diminishes over time, i.e. the features extracted will have fewer unique features, so the older extracted features may be of no use. Furthermore, some occlusions may happen for a long time, as the player may run side by side for many meters. This lengthy occlusion makes it challenging to detect and track the occluded player. The motion difference will be very high, and the features extracted may not be exclusive enough, leading to an ID switch. Likewise, as mentioned before, the players leave the scene, which is also an ID switch problem; however, this does not lead to losing the track as it is presumably being detected and tracked in the other camera.

The feature extractor used for this DeepSORT version was not the same as the one presented in the original paper. A recent library that also provides re-identification models was implemented so that the features extracted were the best possible for the objective. This library allowed an easy integration as it is developed in PyTorch as it is the object detection module. The network used was a OSNet [113] trained for person re-identification purposes. It is a lightweight (636,520 parameters) network that presents good results in re-ID benchmarks.

In essence, the Player Tracking module workflow is: the input is the filtered players bounding boxes. These go through the DeepSORT algorithm, which returns the bounding boxes with the predicted ids. Next, it transforms the players' feet coordinates (x, y) into the top-view with the computed homography matrix (x_t, y_t) and calculates the bounding boxes colour value used for the team segmentation described in Section 4.5. Finally, the information created is sent to the metadata fusion module. It is sent frame-by-frame with an array of elements like $[camera, timestamp, [[id_0, (x_{0t}, y_{0t}), colour], ..., [id_n, (x_{1t}, y_{1t}), colour]]]$.

4.5 Team Segmentation

The players' positional representations must illustrate their respective teams, which are recognised through their outfit's colour. Thus, the objective is to analyse the bounding boxes' colours and distribute them to the teams. For this, a K-means clustering algorithm allows the aggregation of team players based on their outfit colours.

However, there are some problems with this approach. There are three teams in a classic football match: the home team, the visitor team, and the referee team. All of which have different kit colours and distributions - there are more detections with some colours than others. There are five different kit colours—two for each team, as the goalkeeper has a different kit than the field players, and one for the referee. So, in theory, five clusters would be defined for the five colours.

Figure 4.11: Different kit colours. 1. Home team 2. Away team goalkeeper 3. Away team 4. Referee 5. Home team goalkeeper

The noticeable problem is that the high number of clusters may create false predictions. As the obtained footage presented in Section 3.3, sometimes the conditions are not favourable, creating possible errors. For instance, the field has different brightness zones, which may alter the colour perception, e.g., different players from the same team have alleged different kits because of the zone they stand. Likewise, the player is not always in a position to favour the kit colour extraction strategy. In essence, various causes may affect the bounding box plainness to notice the kit colour and create false detections by having a high number of clusters which augment this possibility. As

the field players' colours are the most meaningful to differentiate, the number of clusters is two to diminish these errors.

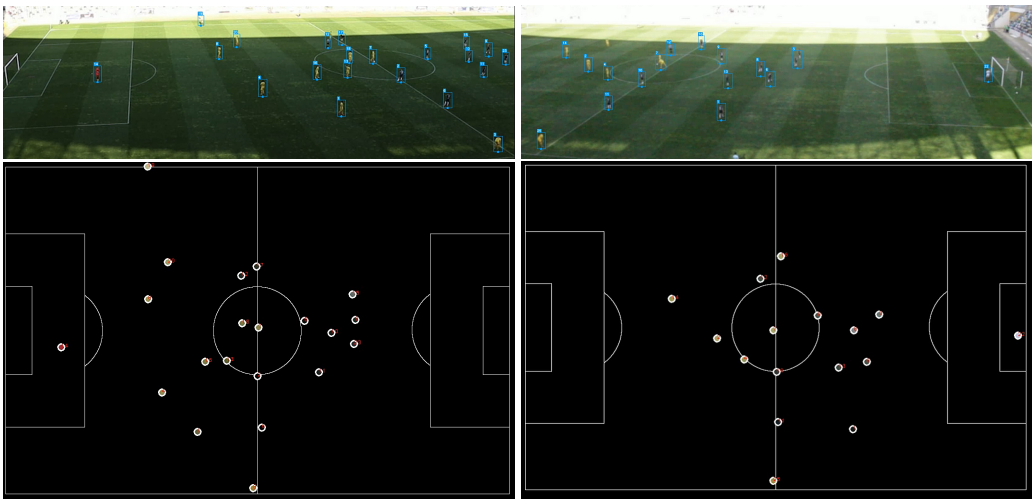

Figure 4.12: Bonding boxes' color value

Regularly, as the players are in the centre of the bounding boxes, the value used to represent their team is the average of the pixels' colour of a 5x5 centred area to capture only the player's kit. The model is built with the colour values of the bounding boxes from both cameras' first frames. After fitting the model, the following frames' bounding boxes' colour values are fed to the model to predict its cluster and, consequently, its team. After predicted, the colour is the same throughout the tracking ID lifetime. Figure 4.12 illustrates the computed colour values. The difference between the computed values is relevant, confirming the problem that the high number of cluster may have.

Having a two-camera setup adds another minor issue for this module. The colours must be the same for both camera tracks. Hence, this algorithm operates within the fusion module, accounting for the bounding boxes from the two cameras. Thus, in the tracking module, the new tracks bounding boxes' colour values are sent to the Metadata Fusion, which predicts their team according to the K-means model.

4.6 Metadata Fusion

This module is responsible for receiving the cameras' metadata and merging them into a complete context. It is also responsible for the team segmentation, creating visual representations, and storing the information extracted for other possible objectives.

In order to fuse the metadata, this module assumed a filtering approach to fuse both cameras' information. After their top view transformation, the detections are separated according to their side midfield (see Figure 4.13). This strategy sets the need to define a pixel in the top view representations that permit separating the input from both cameras. This manual annotation removes the instant expansion of the camera setup that may be available and sets a camera's obligation to

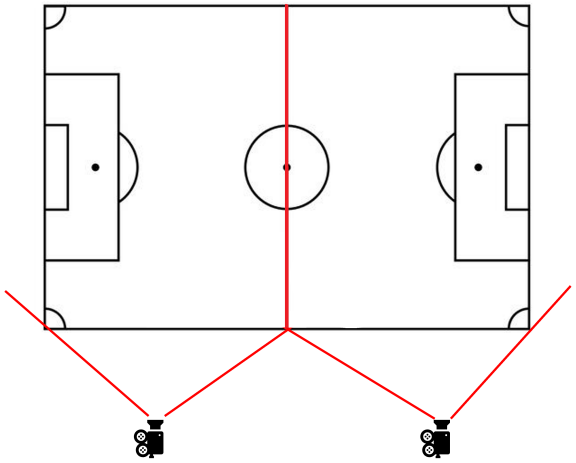

Figure 4.13: Metadata Fusion strategy

inform its side when sending its metadata. Figure 4.14 illustrate the fusion pipeline. First, single-camera detections and tracking take place. Next, the tracked players' positions are mapped to the top-view representation. This representation has the track computed colour value displayed for each player. Finally, the fusion method filters the players' positions according to the camera and their position side and predicts the colour used to represent them.

For the fusion purpose, this module has a WebSocket actively listening for the single-camera information. Consequently, the data must come in a format that allows the information to be rapidly processed. For each frame, the module receives the following metadata: timestamp, camera and the tracking information containing the coordinates and colour if it is a new track.

As the metadata comes through the socket communication, this module filters them according to their side, labels the new tracks team, and updates the visual representations described in Section 5.1. Furthermore, it saves all of this information as it can be valuable for applications other than visual representations.

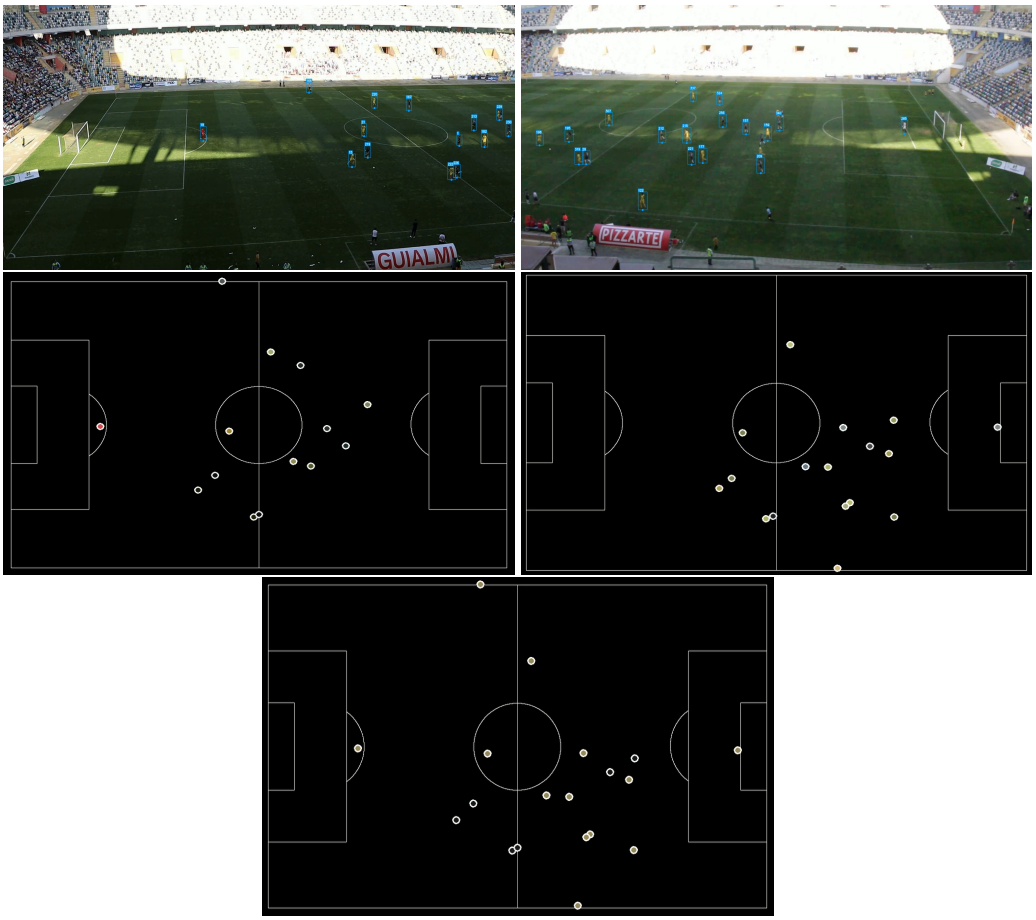

Figure 4.14: Metadata Fusion. Top: Single-Camera tracking. Middle: Top View single-camera with colour value representation. Bottom: Fused Top-View Representation

Chapter 5

Results

This chapter presents the obtained results from the implementation of the steps discussed in the previous chapter. Whereas the outcome of each step was included while discussing it, this chapter reflects the combination of the information retrieved and focuses on the results, which directly address the project goal of providing visual representations of sports games. Furthermore, the results are evaluated and discussed, identifying the proposed methodologies' strengths and limitations.

5.1 Visual Representations

The main goal of the proposed solution is to enhance the user's experience by creating visual representations of what is happening in a match. Three types of visual representations are drawn based on the players' positional information.

The first two graphics illustrate a spatial-temporal representation of the players' position. Both pretend to represent the player's position at a given moment. However, the second also represents the past positions of that player, giving it a notion of direction. Also, the second representation is more consistent as it keeps the tracks in the representation for a few frames even if it is not being tracked. The third is a heat map that indicates the field areas with more presence.

However, there are more possible uses for the information extracted with the proposed solution. First, this information is helpful for another MOGSports objective, the Event Detection module, which can be solved with this information in association with video data. In the same way, the applications of the extracted positional data are myriad since it is the basis for most sport-related game analysis. Finally, another example for the tracking data, which ideally follows a player for most frames, is that it can provide exciting statistics. For instance, a player's distance covered, maximum speed, and the number of sprints.

5.2 System Evaluation

The results are promising while unveiling technical and scientific challenges that need to be addressed in the future. In order to evaluate the system, this section presents various plots that

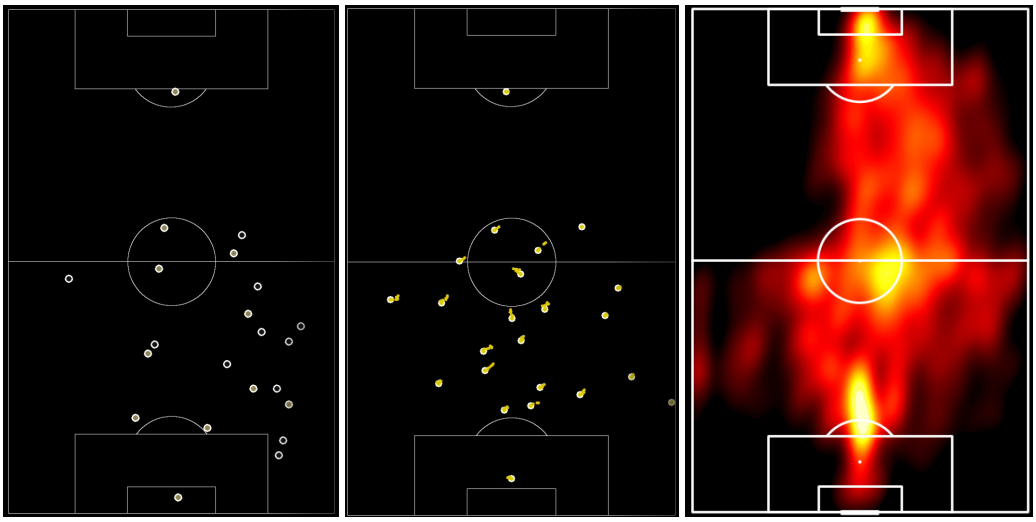

Figure 5.1: Visual Representations. Examples from a five minute sample of the recorded match mentioned in Chapter 3 in the middle of the second half

evaluate the modules individually and the overall system. These plots represent the results of the final solution applied to a sample of around three minutes from the final dataset acquired (see Section 3.3). The sample is reduced, but it is enough to evaluate the system, as it presents it's strengths and weaknesses.

Comprehensively, the proposed solution can perform player detection and corresponding representation, namely the positional plot and heatmap, in real-time. However, the system can not perform online tracking due to hardware limitations. The tracking algorithm was tested with two different GPUs: T600 and RTX4000. The first got 4 frames per second (fps), whereas the second got around 12 fps. Despite these results, it is expected that the tracking algorithm can run in real-time with a more advanced GPU. Nevertheless, the visual representations can still work with 12 fps. The positional representation and heatmap would work almost the same, but with less frame rate and positional information as there would be fewer frames. As for tracking, it could also work, but the tracking capacity would not be as good from the loss of information, and more ID switches would occur.

Figure 5.2 illustrates both cameras individual tracking results. The orange line represents the number of tracked players (also referred to as tracks) per frame, whereas the blue represents the number of new IDs in that same frame. The average number of tracked players is around 19, indicating that the players are correctly detected in most frames. With an unlabelled dataset, it is harder to evaluate the modules, but knowing that the cameras don't cover the entire field, 19 detections is a good indicator for the Player Detection and Tracking modules.

The number of tracks suffers many variations throughout the sample video. Consequently, there are often new IDs when these variations happen. The number of tracks is related to the number of detections as it is a tracking-by-detection algorithm. Essentially, the tracking algorithm can not track what the detection algorithm can not detect. Thus, these variations are the player detection module's fault. As mentioned previously, various possible problems trigger the miss

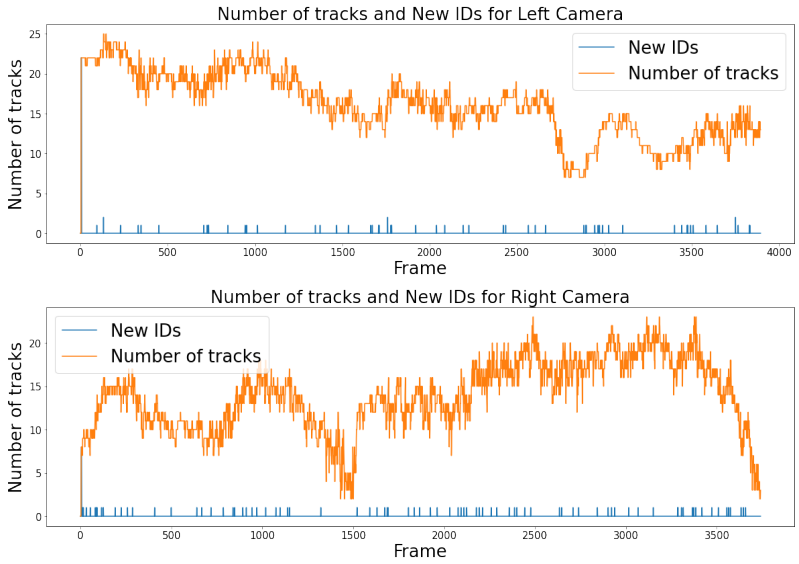

Figure 5.2: Number of tracks per frame - single-camera. Top: Left camera. Bottom: Right camera.

detections, such as motion blur and occlusion. Moreover, a single camera does not cover the entire field, which is why there are moments when the number of tracked players is lower as the players are condensed into one midfield, leaving one of the cameras' field coverage area. Furthermore, this is another challenge for the tracking algorithm to track the players who leave and enter the scene after several later frames. However, the variations are more drastic in the right camera due to a video quality discrepancy between cameras since the right camera quality is significantly lower than the left.

Correspondingly, as this disparity between the cameras is noticeable in the variations of the detection algorithm, it affects the tracking module and its capacity to associate the current frame's detections with the prior. For this sample, the left camera could track the players for a mean of 2035.9 frames, while the right camera did it for 1085.5 frames, revealing the inferiority of the right camera detection and consequent tracking capacity.

To complement these plots and illustrate its interesting moments, Figures 5.3 and 5.4, demonstrate match images and respective tracks for these specific moments. Figure 5.3 exhibits that the detection algorithm successfully detects most of the players in the scene with the left camera, and the players who are not detected are either far from the camera or occluded. The top and middle images represent the initial frames from the top plot in Figure 5.2 where most players are inside the camera's covered area. Conversely, the bottom image from Figure 5.3 is an example of when the number of tracked players is low, and the number of ID switches increases for the following frames as players start entering the scene (for frames [2500-3000]). Furthermore, this image shows that the player detection module is working correctly as the players are far away from the camera and still detected. Figure 5.4 portrays the moments when the object detections work successfully and when it does not work in the right camera. The top image is an example of frames around 1500, with a low number of tracks where these miss detections and consequent miss tracks

Figure 5.3: Left camera tracking results

result from the camera's poor image quality. For this illustration, the players are distant, which, allied with the image's quality, assemble a severe challenge for the Player Detection model. The middle and bottom images demonstrate satisfactory results as most players are correctly detected, although the model has hardships when players are distant.

Looking closely at occlusions, one of the significant problems that create the number of detections instability, these are very hard to challenge with a single view. Therefore, the camera setup designed and illustrated in Section 3.3 offers some advantages to oppose this problem. The cameras have a significant overlap area, meaning they detect the same players from different angles. So, different angles can be helpful when one of the cameras encloses occlusions the other may not, as illustrated with several examples in Figure 5.5.

To evaluate the Metadata Fusion module, the plot in Figure 5.6 presents the number of tracked players in each camera and the total after the filtering method used as the fusion method. For most of the sample, the number of tracks is near 20, which is a good indicator. The mean number of tracks is 17.5. However, this plot reveals variations in the detection and tracking models as presented previously. These variations occur when the left camera has fewer detections and the right camera can not cope with the players on its side, decreasing the total number of players tracked as expected since it has worse results for the detection and tracking models.

The Team Segmentation module has satisfactory results. It assigns the expected colour for many tracks, with occasional errors, especially when noise is in the bounding box. Furthermore,

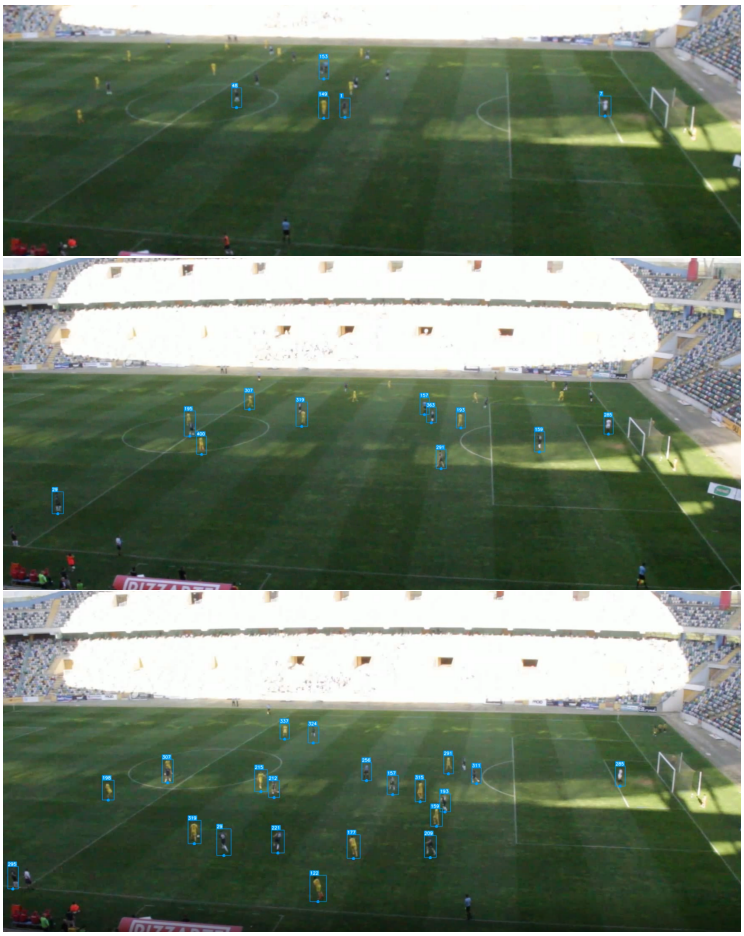

Figure 5.4: Right camera tracking results

the bounding boxes from the goalkeepers and referees add outliers to the model created, creating slight errors in the two formed clusters. Figure 5.8 presents the mean number of the attributed colours to the players per frame, revealing an expected difference in this value as five tracks (goalkeepers and referees) can create erroneous results. The precision of this method calculation needs labelled data, allowing the comparison of the attributed colour and the ground truth. As that dataset does not exist, these results are significant but are not an absolute assurance that this method works.

Furthermore, the sample does not include the beginning of the match as expected, which eases the model created as the players are in a slightly fixed and regular position as the game is on the edge of starting. Figure 4.5 confines various attribution errors. It shows six errors when attributing a colour to tracked players. These mislabelling are caused by the camera difference in saturation and brightness due to the climatic environment and the camera's quality. Moreover, the bounding boxes' content is prominently noisier when the match is being played against the start. Figure 5.9 shows the improvement in detecting at the beginning of the match. Also, it only uses the bounding boxes from the left camera, demonstrating the right camera's harmful effects.

Figure 5.5: Setup occlusion handling advantages

5.3 Discussion

As previously mentioned, the presented solution is a work-in-progress in the scope of MOGSports and DataCloud. This section discusses some of the implemented solution's positive and negative aspects and possible improvements. The current solution encompasses modules that perform Field Registration, Player Detection and Tracking, Team Segmentation and Fusion.

Manual field registration is a straightforward method for computing the homography matrix, which benefits from static cameras. Despite being user-friendly for a small number of cameras, automation is required for scalability.

Accordingly, an automated approach is being implemented and tested. The Two-GAN model [21] is used for field line segmentation. Then, the solution attempts a feature-based approach, comparing the top-view and the segmented image. Along with the tests conducted until now, the Two-GAN line segmentation is expected to improve with more training data, primarily similar to the ones acquired in the proposed setup. Another integral aspect that led to the preference for this feature-based approach was the type of images imagined going through the proposed solution. The camera angles from the dataset do not make the lines drastically change their direction, which could help preserve the corners and the points that could lead to the homography calculations. However, as they are both black and white images, the features may not be the best. Furthermore, many features are similar corners, adding some additional issues to the matching method.

The Player Detection model presents promising results by detecting the majority of the players throughout the match; however, it still requires further fine-tuning to improve the miss detections in some frames. Players' unusual positions or fast-changing movements increase the difficulty of accurate detection. Likewise, brightness can be an issue for very bright or dark field areas, which difficult the detection task. Furthermore, the many occlusions during a match lead to unstable positional information retrieval and unstable visual representations. These miss detections may be difficult to contradict in the real-time representations, but the metadata can be post-match processed if used for additional applications. Especially for tracking, the miss detections positions can be an average point between the two closest track positions (before and after).

As for the Player Tracking module, due to hardware constraints, DeepSORT is not yet thoroughly tested for real-time tracking. Nonetheless, an unwanted ID switching issue has been identi-

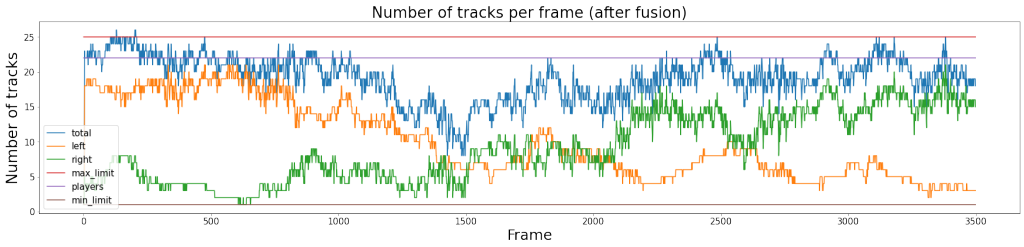

Figure 5.6: Number of tracks per frame. Fused tracks (blue). left camera (orange). Right camera (green). Results obtained with reduced sample (around 3 minutes) of a custom dataset with the proposed camera setup.

fied. Consequently, the algorithm can suffer some modifications to strengthen this module. Other state-of-the-art methods for MOT, such as the latest version of DeepSORT (StrongSORT), may be adopted. Team information could also be a valuable metric for the association step, neglecting possible matches with players from the opposite team. Additionally, this strategy would require a very robust team labelling model. This team association metric could replace or complement the deep feature metric, depending on the results of the possible implementation. However, analysing the results obtained, the deep association has difficulties extracting meaningful features with the current camera setup, so examining additional features than the colour could be ineffective. As presented in Section 5.2, the deep metric may have a lower impact than initially thought. Hence, the DeepSORT hyperparameter λ can be studied in more detail. However, as no labelled data is available to support a precise classification of the algorithm, it is hard to compare the different options.

Alternatively, tracking can occur after data fusion, resorting to algorithms like graph-based tracking using the transformed top-view detection coordinates. This strategy would demand a robust fusion algorithm as a previous step.

Despite being accurate for numerous players, team segmentation also has limitations. As players move rapidly, their bounding box may temporarily not fit its body. Hence, background or other artefacts can influence the colour value calculation. Likewise, the camera difference influences the computed colour value and can lead to errors. In Figure 4.14, it is noticeable that the right camera is much brighter, meaning the computed value is consequently brighter, which, in this case, leads to errors in the segmentation. In this sense, the method can improve by assigning each player to the team associated with the mode of a periodic check prediction or of the N initial frames predictions. This way, it allows some miss classifications for imprecise detections moments and increases the robustness of the module. Furthermore, as the goalkeepers have a generally fixed position, their colour could be assigned by it.

Fusion is critical, and a scalable solution is necessary for this module. The cameras provide an excellent overlap area that can be used for occlusion handling and to complete miss detections from one another. Incorporating a matching algorithm into this module could leverage the full potential of the proposed camera setup. One possible strategy would be associating the players from each camera in each frame. This method could consider all players, calculate the euclidean

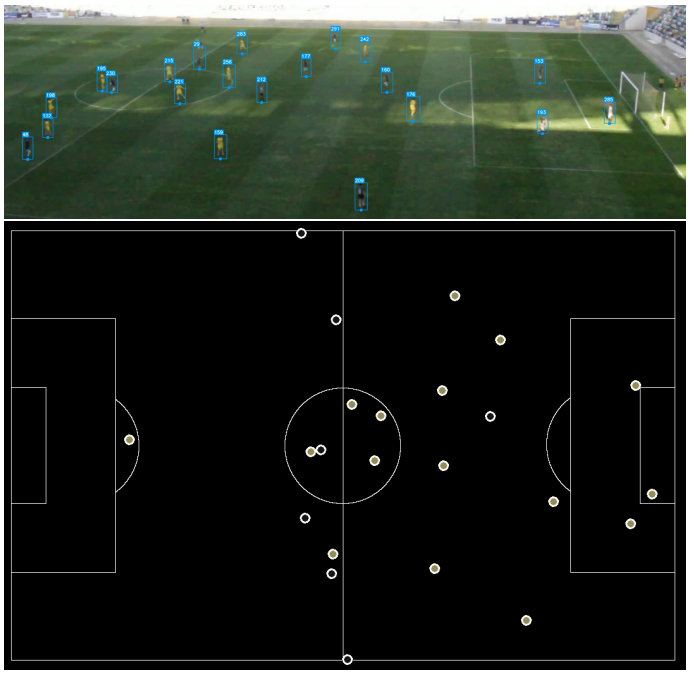

Figure 5.7: Team Segmentation example

distance between the transformed points, and then associate them optimally.

As a computationally demanding task, other parameters, such as the team label, could reduce the distance calculations and consecutive matching. Figure 5.10 illustrates the top view representation with the detections from each camera drawn with different colours, blue for the left camera and red for the right. It is a simple example with no occlusions to demonstrate that both cameras' detections may be easy to associate as they are mapped to a very close position. So, by using a matching algorithm, no detection would be lost in this frame.

This association is also crucial for Player Tracking. With the current approach, tracking is valuable for the representations. However, for further use, e.g. statistics and event detection, it is not yet as valuable as the IDs attributed do not reflect the match but only the camera scene. For instance, the player with ID 1 in one camera can have the ID 5 in the other, and by not being associated, these are effectively different players for the fusion module. Hence, creating a multi-camera association algorithm would provide a multi-camera tracking system, creating more relevant metadata and further advantages for the solution. Associating single-camera tracks has the advantages of a previous association in the tracking step. Thus, this matching with the other cameras can be prompter as IDs are attributed, i.e. if the ID 1 from one camera is associated with ID 5 from the other in a frame, the following frame does not need to associate them, assuming the single-camera tracking is working correctly.

Overall, the results obtained and presented in this chapter are promising and indicate specific problems that are hard to challenge, though this section presents possible solutions. However, the quality discrepancy between the cameras used for the data acquisition affects the results negatively. Consequently, the debated problems may be softened when considering a dataset with equally

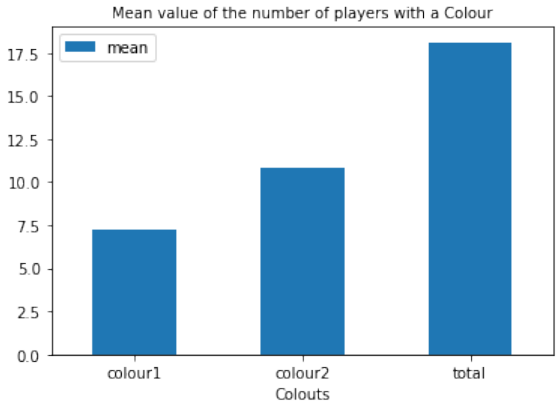

Figure 5.8: Mean number of colours attributed per frame

high-quality images.

As for the complete pipeline, the proposed solution is ready to create in real-time the representations regarding the detections, player positional top-view and heatmap; however, the overlay has not been tested and placed into an input stream or video. In essence, these visual representations can be created using a stream but are saved to a file and not yet overlayed in a broadcast or other image.

The tracking algorithm, as mentioned, is not yet in real-time, but the objective is to reach 20 to 25 fps so that at least near real-time is possible. Furthermore, the Homography Estimation is another crucial aspect of automating the content enrichment. This task is done manually; however, its automation is vital for the scalability and automation of the system. This manual effort is one of the changes in this dissertation, and the project's expectations led by the complexity of the problem and previewed time to accomplish it, but also by the wish to move to the next stage and start building the entire pipeline. The other aspect was the metadata fusion approach. The advantages of the setup, mainly the speed related to the filtering approach, were a factor in this decision comparing it to a multi-camera association algorithm approach.

In essence, some open issues do not allow the system to be fully operational, such as the speed of the tracking algorithm and the manual homography estimation method. However, both have proposed solutions that leave confident expectations for future developments. With these advances, the system would be operational and fully automatic, reaching the main goal.

In addition, Event Detection and Ball Detection can upgrade even more the representations and metadata created. Nevertheless, Event Detection needs to be implemented from the beginning, leaving the final solution with no confident finish date prevision. This module has not been developed as the previous have some open, unresolved situations. However, when developed, it would mean that MOGSports has reached the objective and is fully operational.

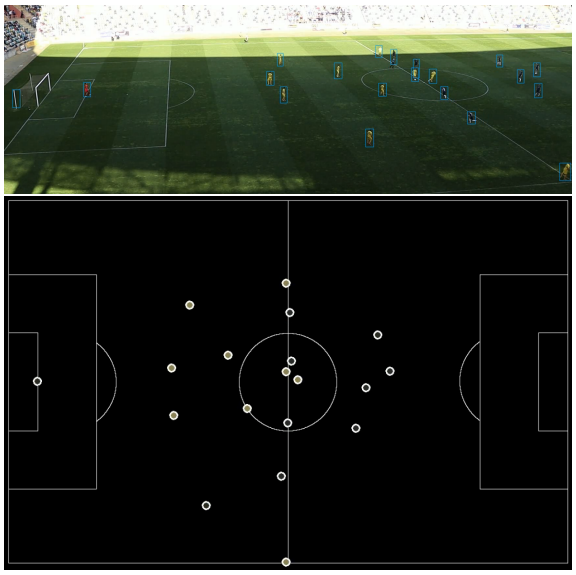

Figure 5.9: Team Segmentation at the start of the second half

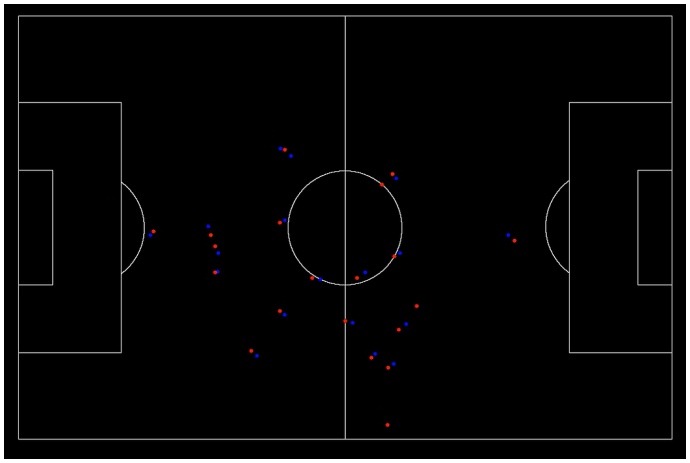

Figure 5.10: Top-view representation of both cameras' detections

Chapter 6

Conclusions and Future Work

In response to the audience demand, user experience enhancement is an imperative action for the sports broadcasting industry. This dissertation presents a step forward to accomplish this enhancement based on players' positional data extraction and representation. This chapter summarises the work developed and expected future developments to enhance a broadcast user experience.

The proposed solution can perform the required tasks with satisfactory and promising results, creating good expectations for the upcoming developments and the potential of the MOGSports business case within the media and entertainment industries.

The core of the Field Detection module is the homography matrix estimation. This task was achieved with a manual procedure that provides an accurate homography for the complete match. Automatic Homography Estimation was considered, resorting to the Two-GAN field segmentation model. However, the available data for training was not representative of the different match image conditions and, consequently, was unable to generalise to the new environments (camera views, distance and brightness).

The Player Detection module resorts to the YOLOv5 model, especially a fine-tuned model for human crowd environments. This fine-tuning was considered crucial as the standard pre-trained YOLOv5 models did not have satisfactory results for match environments. Football matches pose various challenges like occlusion, motion blur and the image distance to the field, thus difficulting the detector inference. Therefore, the Data Acquisition Protocol plays a pivotal role. Additionally, the retrieved dataset unveiled that camera specifications, positioning and exposure are factors that influence the detector performance. In particular, the results associated with the right side of the field led to unstable detections due to the respective camera's lower image quality. Despite the presented challenges, the Player Detection module presented satisfactory results, detecting the majority of the players (and off-field persons) in the scene.

The tracking module depends on the detection, as it is a tracking-by-detection algorithm. Thus, if the detection module does not detect, the tracking module can not track. Overall, the tracking module achieves the delimited objectives with some inconsistencies. Conversely to the detection module, the tracking module does not operate in real-time due to hardware constraints.

The Metadata Fusion module combines the metadata extracted from the two cameras used in

the setup, which were found sufficient to provide full-field coverage. The results assessment after fusion showed the smooth transition of players between cameras while maintaining a fairly stable number of total detected players. However, the filtering method occasionally led to a slight loss of information at the midfield line.

Additionally, the Fusion module also segments and represents the tracked players for their team with some margin of error. This task occurs in this module as the used k-means model must be the same for both cameras' tracks. Using a 5-pixel pad around the bounding box centre point gave a good representation of the team colour value. Despite football matches having five different kit colours on the field (two goalkeepers, two field player colours and the referees), the most valuable representation focuses on-field players, using 2 clusters, achieving meaningful results. Similarly to the detection module, the camera's specification strongly affects the results obtained. Results portray this disparity between cameras, as the right side camera has a more bright image leading to some incorrect decisions.

As for the Visual Representations created, they can illustrate the player's current position, the player's current and last few positions, creating a notion of motion, and also heatmaps (field areas with more activity).

The proposed methodology and results obtained are a step forward to a novel paradigm of metadata extraction and sports broadcasting enrichment in order to fulfil the MOGSports' project vision.

6.1 Future Work

The proposed solution reflects ongoing work to achieve MOGSports and DataCloud's objectives. Extracting the proposed metadata has complex associated problems, as the results reflect. Accordingly, future work is anticipated. Section 5.3 has one or more proposed alternatives for each problem encountered.

Future work is expected to improve the used YOLOv5 model performance by using the acquired footage as a training dataset. Furthermore, representing the ball is also a project goal. In order to represent it, the system must detect it, which is a more challenging problem as the ball is smaller and moves faster than the players.

The tracking algorithm is set to be tested with more recent hardware and with slight modifications as described in Section 5.3.

Leveraging the advantages of the camera setup and integrating the information extracted from each camera, the Metadata Fusion strategy may address the occlusion challenges previously described.

Regarding the performance of the team segmentation algorithm, the collection of more data with higher image quality can not only lead to less misidentification of team players due to brightness but also help increase the density of each cluster while maximising the distance between them. This may also enable the detection of more than 2 clusters to reflect goalkeepers.

For system deployment into the broadcasting industry, MOGSports' next steps will include implementing an interface for manual calibration. Finally, Event Detection is a comprehensive problem that MOGSports intends to solve, which reflects the culmination of the developments from the previous modules and the valuable metadata extracted. It may comprise match highlights (e.g. goals) and statistical insights.

The combination of the presented work with future developments will enable the extraction of even more valuable data during a football matches broadcast, opening the door for new configurable workflows and cost efficiencies for the media producers while unconstraining and expanding the content consumers' user experience.

Bibliography

[1] Football Stadiums & Arenas seating views. See your seat view before you buy Soccer tickets. URL https://aviewfrommyseat.co.uk/sports/Soccer/.

[2] David Acuna. Towards real-time detection and tracking of basketball players using deep neural networks. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA*, pages 4–9, 2017.

[3] M. Alemán-Flores, L. Alvarez, L. Gomez, P. Henriquez, and L. Mazorra. Camera calibration in sport event scenarios. *Pattern Recognition*, 47(1):89–95, 2014. ISSN 0031-3203. doi: 10.1016/j.patcog.2013.05.011.

[4] MM Naushad Ali, Mohammad Abdullah-Al-Wadud, and Seok-Lyong Lee. An efficient algorithm for detection of soccer ball and players. *Proc. 16th ASTL Control and Networking*, 16, 2012.

[5] Luis Alvarez and Vicent Caselles. Homography estimation using one ellipse correspondence and minimal additional information. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4842–4846, 2014. doi: 10.1109/ICIP.2014.7025981.

[6] Kumar Shridutt Anand and Roshan Deshmukh. Football Market by Product Type (Training Ball, Match Ball, and Others), Manufacturing Process (Hand Stitched, Machine Stitched, and Thermally Bonded), and Distribution Channel (Online and Offline): Global Opportunity Analysis and Industry Forecast 2021–2027.

[7] Xuefeng Bai, Tiejun Zhang, Chuanjun Wang, Ahmed Abd El-Latif, and Xiamu Niu. A fully automatic player detection method based on one-class SVM. *IEICE Transactions on Information and Systems*, E96.D:387–391, February 2013. doi: 10.1587/transinf.E96.D.387.

[8] Abhishek Balasubramaniam and Sudeep Pasricha. Object detection in autonomous vehicles: Status and open challenges. arXiv, 2022. doi: 10.48550/ARXIV.2201.07706.

[9] Dana Harry Ballard and Christopher M. Brown. *Computer Vision*. Prentice Hall Professional Technical Reference, 1st edition, 1982. ISBN 0-13-165316-4.

[10] Lluis Barceló, Xavier Binefa, and John R. Kender. Robust methods and representations for soccer player tracking and collision resolution. In Wee-Kheng Leow, Michael S. Lew, Tat-Seng Chua, Wei-Ying Ma, Lekha Chaisorn, and Erwin M. Bakker, editors, *Image and Video Retrieval*, pages 237–246, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31678-7.

[11] Michael Beetz, Nico Hoyningen-Huene, Jan Bandouch, Bernhard Kirchlechner, Suat Gedikli, and Alexis Maldonado. Camera-based observation of football games for analyzing multi-agent activities. volume 2006 of *Proceedings of the International Conference on Autonomous Agents*, pages 42–49, January 2006. doi: 10.1145/1160633.1160638.

[12] Michael Beetz, Suat Gedikli, Jan Bandouch, Bernhard Kirchlechner, Nico Hoyningen-Huene, and Alexander Perzylo. Visually tracking football games based on TV broadcasts. IJCAI International Joint Conference on Artificial Intelligence, pages 2066–2071, January 2007.

[13] James R Bergen and Haim Shvaytser (Schweitzer). A probabilistic algorithm for computing Hough transforms. *Journal of Algorithms*, 12(4):639–656, 1991. ISSN 0196-6774. doi: 10.1016/0196-6774(91)90037-Y.

[14] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. *CoRR*, abs/1903.05625, 2019.

[15] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, September 2016. doi: 10.1109/icip.2016.7533003.

[16] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-Speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017. doi: 10.1109/AVSS.2017.8078516.

[17] Erik Bochinski, Tobias Senst, and Thomas Sikora. Extending IOU based multi-object tracking by visual information. November 2018. doi: 10.1109/AVSS.2018.8639144.

[18] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection, 2020.

[19] Afshin Bozorgpour, Mehran Fotouhi, and Shohreh Kasaei. Robust homography optimization in soccer scenes. In *2015 23rd Iranian Conference on Electrical Engineering*, pages 787–792, 2015. doi: 10.1109/IranianCEE.2015.7146320.

[20] Hua-Tsung Chen, Wen-Jiin Tsai, Suh-Yin Lee, and Jen-Yu Yu. Ball tracking and 3D trajectory approximation with applications to tactics analysis from single-camera volleyball

sequences. *Multimedia Tools and Applications*, 60(3):641–667, October 2012. ISSN 1573-7721. doi: 10.1007/s11042-011-0833-y.

[21] Jianhui Chen and James J. Little. Sports Camera Calibration via Synthetic Data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2497–2504, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72812-506-0. doi: 10.1109/CVPRW.2019.00305.

[22] Jianhui Chen, Fangrui Zhu, and James J. Little. A Two-Point Method for PTZ Camera Calibration in Sports. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 287–295, Lake Tahoe, NV, March 2018. IEEE. ISBN 978-1-5386-4886-5. doi: 10.1109/WACV.2018.00038.

[23] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. TransMOT: Spatial-temporal graph transformer for multiple object tracking. *CoRR*, abs/2104.00194, 2021.

[24] Anthony Cioppa, Adrien Deliege, Floriane Magera, Silvio Giancola, Olivier Barnich, Bernard Ghanem, and Marc Van Droogenbroeck. Camera Calibration and Player Localization in SoccerNet-v2 and Investigation of their Representations for Action Spotting. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4532–4541, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-899-4. doi: 10.1109/CVPRW53098.2021.00511.

[25] Leonardo Citraro, Pablo Márquez-Neila, Stefano Savarè, Vivek Jayaram, Charles Dubout, Félix Renaut, Andrés Hasfura, Horesh Ben Shitrit, and Pascal Fua. Real-Time Camera Pose Estimation for Sports Fields. *Machine Vision and Applications*, 31(3):16, March 2020. ISSN 0932-8092, 1432-1769. doi: 10.1007/s00138-020-01064-7.

[26] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, 36(3):537–556, September 2013. ISSN 0219-3116. doi: 10.1007/s10115-013-0665-3.

[27] Carlos Cuevas, Daniel Quilón, and Narciso García. Automatic soccer field of play registration. *Pattern Recognition*, 103:107278, 2020. ISSN 0031-3203. doi: 10.1016/j.patcog.2020.107278.

[28] Carlos Cuevas, Daniel Quilón, and Narciso García. Techniques and applications for soccer video analysis: A survey. *Multimedia Tools and Applications*, 79(39-40):29685–29721, October 2020. ISSN 1380-7501, 1573-7721. doi: 10.1007/s11042-020-09409-0.

[29] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *CoRR*, abs/1606.03798, 2016.

[30] Dirk Farin, Susanne Krabbe, Peter With, and Wolfgang Effelsberg. Robust camera calibration for sport videos using court models. volume 5307, pages 80–91, January 2004. doi: 10.1117/12.526813.

[31] Pascual Figueroa, Neucimar Leite, and Ricardo Barros. Background recovering in outdoor image sequences: An example of soccer players segmentation. *Image and Vision Computing*, 24:363–374, April 2006. doi: 10.1016/j.imavis.2005.12.012.

[32] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021, 2021.

[33] Sebastian Gerke, Karsten Müller, and Ralf Schäfer. Soccer jersey number recognition using convolutional neural networks. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 734–741, 2015. doi: 10.1109/ICCVW.2015.100.

[34] Raffay Hamid, Ram Krishan Kumar, Matthias Grundmann, Kihwan Kim, Irfan Essa, and Jessica Hodgins. Player localization using multiple static cameras for sports visualization. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–738, 2010. doi: 10.1109/CVPR.2010.5540142.

[35] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, second edition, 2004. doi: 10.1017/CBO9780511811685.

[36] J.-B. Hayet, J.H. Piater, and J.G. Verly. Fast 2D model-to-image registration using vanishing points for sports video analysis. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–417, 2005. doi: 10.1109/ICIP.2005.1530417.

[37] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Soccer field localization from a single image, 2016.

[38] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4012–4020, 2017. doi: 10.1109/CVPR.2017.427.

[39] Samuel Hurault, C. Ballester, and Gloria Haro. Self-supervised small soccer player detection and tracking, November 2020.

[40] Marina Ivašić-Kos, Kristina Host, and Miran Pobar. Application of deep learning methods for detection and tracking of players. February 2021. ISBN 978-1-83962-374-5. doi: 10.5772/intechopen.96308.

[41] S. Iwase and H. Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 751–754 Vol.4, 2004. doi: 10.1109/ICPR.2004.1333881.

[42] Colby Jeffries. Sports Analytics With Computer Vision. *Senior Independent Study Theses*, January 2018. URL https://openworks.wooster.edu/independentstudy/8103.

[43] Wei Jiang, Juan Camilo Gamboa Higuera, Baptiste Angles, Weiwei Sun, Mehrsan Javan, and Kwang Moo Yi. Optimizing Through Learned Errors for Accurate Sports Field Registration. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 201–210, Snowmass Village, CO, USA, March 2020. IEEE. ISBN 978-1-72816-553-0. doi: 10.1109/WACV45572.2020.9093581.

[44] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, Imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, Tkianai, YxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. Ultralytics/yolov5: V6.1, February 2022.

[45] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960. ISSN 0021-9223. doi: 10.1115/1.3662552.

[46] Paresh R. Kamble, Avinash G. Keskar, and Kishor M. Bhurchandi. Ball tracking in sports: A survey. *Artificial Intelligence Review*, 52(3):1655–1705, October 2019. ISSN 0269-2821, 1573-7462. doi: 10.1007/s10462-017-9582-2.

[47] Kaveh Kangarloo and Ehsanollah Kabir. Grass field segmentation, the first step toward player tracking, deep compression, and content based football image retrieval. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 818–824, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-30125-7.

[48] Kaveh Kangarloo and Ehsanollah Kabir. Sequential probabilistic grass field segmentation of soccer video images. In Reinhard Klette and Joviša Žunić, editors, *Combinatorial Image Analysis*, pages 639–645, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-30503-3.

[49] Hirokatsu Kataoka, Kiyoshi Hashimoto, and Yoshimitsu Aoki. Player position estimation by monocular camera for soccer video analysis. *SICE Annual Conference 2011*, pages 1985–1990, 2011.

[50] Seyed Hossein Khatoonabadi and Mohammad Rahmati. Automatic soccer players tracking in goal scenes by camera motion elimination. *Image and Vision Computing*, 27(4):469–479, 2009. ISSN 0262-8856. doi: 10.1016/j.imavis.2008.06.015.

[51] Changick Kim and Jenq-Neng Hwang. Fast and automatic video object segmentation and tracking for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):122–129, 2002. doi: 10.1109/76.988659.

[52] Hanjoo Kim, Siwo Ok, and Jaihie Kim. Player segmentation evaluation for trajectory estimation in soccer games. January 2003.

[53] Wonjun Kim, Sung-Won Moon, Jiwon Lee, Do Nam, and Chanho Jung. Multiple player tracking in soccer videos: An adaptive multiscale sampling approach. *Multimedia Systems*, 24, November 2018. doi: 10.1007/s00530-018-0586-9.

[54] Jacek Komorowski, Grzegorz Kurzejamski, and Grzegorz Sarwas. FootAndBall: Integrated player and ball detector. *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2020. doi: 10.5220/0008916000470056.

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[56] Hei Law and Jia Deng. CornerNet: Detecting Objects as Paired Keypoints. *International Journal of Computer Vision*, 128(3):642–656, March 2020. ISSN 1573-1405. doi: 10.1007/s11263-019-01204-1.

[57] Antje Linnemann, Sebastian Gerke, Sven Kriener, and Patrick Ndjiki-Nya. Temporally consistent soccer field registration. In *2013 IEEE International Conference on Image Processing*, pages 1316–1320, 2013. doi: 10.1109/ICIP.2013.6738271.

[58] Jia Liu, Xiaofeng Tong, Wenlong Li, Yimin Zhang, and Hongqi Wang. Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30: 103–113, January 2009. doi: 10.1016/j.patrec.2008.02.011.

[59] Keyu Lu, Jianhui Chen, James J. Little, and Hangen He. Lightweight convolutional neural networks for player detection and classification. *Computer Vision and Image Understanding*, 172:77–87, 2018. ISSN 1077-3142. doi: 10.1016/j.cviu.2018.02.008.

[60] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, April 2021. ISSN 0004-3702. doi: 10.1016/j.artint.2020.103448.

[61] Upendra Rao M. and Umesh C. Pati. A novel algorithm for detection of soccer ball and player. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 0344–0348, 2015. doi: 10.1109/ICCSP.2015.7322903.

[62] M. Manafifard, Hamid Ebadi, and Hamid Abrishami Moghaddam. Multi-player detection in soccer broadcast videos using a blob-guided particle swarm optimization method. *Multimedia Tools and Applications*, 76:12251–12280, 2016.

[63] M. Manafifard, H. Ebadi, and H. Abrishami Moghaddam. A survey on player tracking in soccer videos. *Computer Vision and Image Understanding*, 159:19–46, June 2017. ISSN 10773142. doi: 10.1016/j.cviu.2017.02.002.

[64] Rafael Martín and José M. Martínez. A semi-supervised system for players detection and tracking in multi-camera soccer videos. *Multimedia Tools and Applications*, 73(3):1617–1642, December 2014. ISSN 1380-7501, 1573-7721. doi: 10.1007/s11042-013-1659-6.

[65] Thomas B. Moeslund, Graham Thomas, and Adrian Hilton, editors. *Computer Vision in Sports*. Advances in Computer Vision and Pattern Recognition. Springer International Publishing, Cham, 2014. ISBN 978-3-319-09395-6 978-3-319-09396-3. doi: 10.1007/978-3-319-09396-3.

[66] M. Naemura, A. Fukuda, Y. Mizutani, Y. Izumi, Y. Tanaka, and K. Enami. Morphological segmentation of sport scenes using color information. *IEEE Transactions on Broadcasting*, 46(3):181–188, 2000. doi: 10.1109/11.892154.

[67] Banoth Naik and Mohammad Farukh Hashmi. Ball and player detection & tracking in soccer videos using improved YOLOV3 model, April 2021.

[68] Jacob Solawetz Nelson, Joseph. YOLOv5 is Here, June 2020.

[69] Joseph Nelson and Jacob Solawetz. Responding to the Controversy about YOLOv5, June 2020.

[70] V.A. Ngo, W. Yang, and J. Cai. Accurate playfield detection using area-of-coverage. ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, pages 3441–3444, 2010. doi: 10.1109/ISCAS.2010.5537853.

[71] Ty Nguyen, Steven W. Chen, Shreyas S. Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. doi: 10.1109/LRA.2018.2809549.

[72] Xiaohan Nie, Shixing Chen, and Raffay Hamid. A Robust and Efficient Framework for Sports-Field Registration. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1935–1943, Waikoloa, HI, USA, January 2021. IEEE. ISBN 978-1-66540-477-8. doi: 10.1109/WACV48630.2021.00198.

[73] Juan Rafael Nunez, Jacques Facon, and Alceu de Souza Brito. Soccer video segmentation: Referee and player detection. In *2008 15th International Conference on Systems, Signals and Image Processing*, pages 279–282, 2008. doi: 10.1109/IWSSIP.2008.4604421.

[74] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

[75] Simon JD Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

[76] Daniel Quilón, Raúl Mohedano, Carlos Cuevas, and Narciso García. Unsupervised high-quality soccer field segmentation. In *2015 International Symposium on Consumer Electronics (ISCE)*, pages 1–2, 2015. doi: 10.1109/ISCE.2015.7177808.

[77] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. Number arXiv:1612.08242, December 2016. doi: 10.48550/ARXIV.1612.08242.

[78] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. Number arXiv:1804.02767, April 2018. doi: 10.48550/ARXIV.1804.02767.

[79] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. Number arXiv:1506.02640, May 2016.

[80] Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, and Steve Seitz. Soccer on Your Tabletop. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4738–4747, Salt Lake City, UT, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00498.

[81] Jos B.T.M. Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000. ISSN 0169-2968. doi: 10.3233/FI-2000-411207.

[82] Houari SABIRIN, Hiroshi SANKOH, and Sei NAITO. Automatic Soccer Player Tracking in Single Camera with Robust Occlusion Handling Using Attribute Matching. *IEICE Transactions on Information and Systems*, E98.D(8):1580–1588, 2015. doi: 10.1587/transinf.2014EDP7313.

[83] Melike Şah and Cem Direkoğlu. Review and evaluation of player detection methods in field sports. *Multimedia Tools and Applications*, June 2021. ISSN 1573-7721. doi: 10.1007/s11042-021-11071-z.

[84] Long Sha, Jennifer Hobbs, Panna Felsen, Xinyu Wei, Patrick Lucey, and Sujoy Ganguly. End-to-End Camera Calibration for Broadcast Videos. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13624–13633, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.01364.

[85] Suping Shi. Comparison of player tracking-by-detection algorithms in football videos. Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2020.

[86] Jacob Solawetz. YOLOv4 - An explanation of how it works, June 2020.

[87] P. Spagnolo and P. L. Mazzeo. UNSUPERVISED ALGORITHMS FOR SEGMENTATION AND CLUSTERING APPLIED TO SOCCER PLAYERS CLASSIFICATION:. In

Proceedings of the Second International Conference on Signal Processing and Multimedia Applications, pages 129–134, Barcelona, Spain, 2007. SciTePress - Science and and Technology Publications. ISBN 978-989-8111-13-5. doi: 10.5220/0002139501290134.

[88] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[89] Tsuyoshi Taki, Jun-ichi Hasegawa, and Teruo Fukumura. Development of motion analysis system for quantitative evaluation of teamwork in soccer games. volume 3 of *IEEE International Conference on Image Processing*, pages 815–818, 1996.

[90] Jonas Theiner, Wolfgang Gritz, Eric Muller-Budack, Robert Rein, Daniel Memmert, and Ralph Ewerth. Extraction of Positional Player Data From Broadcast Soccer Videos. page 11.

[91] Graham Thomas, Rikke Gade, Thomas B. Moeslund, Peter Carr, and Adrian Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18, 2017. ISSN 1077-3142. doi: 10.1016/j.cviu.2017.04.011.

[92] Do Thuan. EVOLUTION OF YOLO ALGORITHM AND YOLOV5: THE STATE-OF-THE-ART OBJECT DETECTION ALGORITHM. page 61.

[93] Xiaofeng Tong, Jia Liu, Tao Wang, and Yimin Zhang. Automatic player labeling, tracking and field registration and trajectory mapping in broadcast soccer video. *ACM Transactions on Intelligent Systems and Technology*, 2(2), February 2011. ISSN 2157-6904. doi: 10.1145/1899412.1899419.

[94] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:1–13, February 2018. doi: 10.1155/2018/7068349.

[95] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. arXiv, 2021. doi: 10.48550/ARXIV.2105.04206.

[96] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.

[97] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. 2017.

[98] Yonghong Xie and Qiang Ji. A new efficient ellipse detection method. In *2002 International Conference on Pattern Recognition*, volume 2, pages 957–960 vol.2, 2002. doi: 10.1109/ICPR.2002.1048464.

[99] J. Xing, H. Ai, L. Liu, and S. Lao. Multiple player tracking in sports video: A dual-mode two-way Bayesian inference approach with progressive observation modeling. *IEEE Transactions on Image Processing*, 20(6):1652–1667, 2011. doi: 10.1109/TIP.2010.2102045.

[100] Lei Xu, Erkki Oja, and Pekka Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990. ISSN 0167-8655. doi: 10.1016/0167-8655(90)90042-Z.

[101] M. Xu, L. Lowey, and J. Orwell. Architecture and algorithms for tracking football players with multiple cameras. In *IEE Intelligent Distributed Surveilliance Systems*, pages 51–55, 2004. doi: 10.1049/ic:20040098.

[102] Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005. doi: 10.1109/TNN.2005.845141.

[103] Chan-Yun Yang, Hooman Samani, Nana Ji, Chunxu Li, Ding-Bang Chen, and Man Qi. Deep learning based real-time facial mask detection and crowd monitoring. *Computing and Informatics*, November 2021.

[104] Ying Yang and Danyang Li. Robust player detection and tracking in broadcast soccer video based on enhanced particle filter. *Journal of Visual Communication and Image Representation*, 46:81–94, 2017. ISSN 1047-3203. doi: 10.1016/j.jvcir.2017.03.008.

[105] Qiang Yao, Keisuke Nonaka, Hiroshi Sankoh, and Sei Naito. Robust moving camera calibration for synthesizing free viewpoint soccer video. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1185–1189, 2016. doi: 10.1109/ICIP.2016.7532545.

[106] Qiang Yao, Akira Kubota, Kaoru Kawakita, Keisuke Nonaka, Hiroshi Sankoh, and Sei Naito. Fast camera self-calibration for synthesizing Free Viewpoint soccer Video. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1612–1616, 2017. doi: 10.1109/ICASSP.2017.7952429.

[107] Ho-Sub Yoon, Young-lae Bae, and Young Yang. A soccer image sequence mosaicking and analysis method using line and advertisement board detection. *Etri Journal - ETRI J*, 24: 443–454, December 2002. doi: 10.4218/etrij.02.0102.0005.

[108] Xinguo Yu, Changsheng Xu, Hon Wai Leong, Qi Tian, Qing Tang, and Kong Wah Wan. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, MULTIMEDIA '03, pages 11–20, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1-58113-722-2. doi: 10.1145/957013.957018.

[109] Xinguo Yu, Hon Wai Leong, Changsheng Xu, and Qi Tian. A robust Hough-based algorithm for partial ellipse detection in broadcast soccer video. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 3, pages 1555–1558 Vol.3, 2004. doi: 10.1109/ICME.2004.1394544.

[110] Lijing Zhang, Yao Lu, Ge Song, and Hanfeng Zheng. RC-CNN: Reverse connected convolutional neural network for accurate player detection. In Xin Geng and Byeong-Ho Kang, editors, *PRICAI 2018: Trends in Artificial Intelligence*, pages 438–446, Cham, 2018. Springer International Publishing. ISBN 978-3-319-97310-4.

[111] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. 2021.

[112] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, September 2021. ISSN 1573-1405. doi: 10.1007/s11263-021-01513-4.

[113] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. arXiv, 2019. doi: 10.48550/ARXIV.1905.00953.

Appendix A

Player Tracking System for Sports Broadcast Enrichment

Player Tracking System for Sports Broadcast Enrichment

Murilo Couceiro[1,3], Inês Rito Lima[3][0000−0002−9681−4740], Alexandre Ulisses[3], Tiago Mendes-Neves[1,2][0000−0002−4802−7558], and João Mendes-Moreira[1,2][0000−0002−2471−2833]

[1] Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
{up202003377,up201406104,jmoreira}@fe.up.pt
[2] LIAAD - INESC TEC, Porto, Portugal
[3] MOG Technologies, Department of Innovation, Maia, Portugal
{ines.lima,alexandre.ulisses}@mog-technologies.com

Abstract. The broadcast of audio-video sports content is a field with increasingly larger audiences demanding higher quality content and involvement. This growth creates the necessity to develop more content to engage the users and keep this trend. Otherwise, it may stall or even diminish. Therefore, enhancing the user experience, engagement, and involvement during live sports event broadcasts is of utmost importance. This paper proposes a solution to extract event's information from video, resorting to Computer Vision techniques and Deep Learning algorithms. More specifically, the project encompassed the definition and implementation of field registration, object detection and tracking tasks. Focusing on football sports events, a novel dataset combining several video sources was created and used for analysis and metadata extraction. In particular, the proposed solution can detect and track players with acceptable precision using state-of-the-art methods, like YOLOv5 and DeepSORT. Furthermore, resorting to unsupervised learning techniques, the system provides team segmentation based on the colour of the players' kits. A series of visual representations regarding the players' movements on the field enables broadcast enrichment and increased user experience. The presented solution is framed in the H2020 DataCloud project and will be deployed in a cloud environment simplifying its access and utilisation.

Keywords: Metadata Enhancement · Player Tracking · Computer vision. · Sports Broadcast

1 Introduction

As the most popular sport in our society, with millions of fans, football has very high market value and has a growing amount of visual data generated, leading to a growth in the research of this topic. Football clubs, coaches and fans demand customised football match analysis services, creating the need within the broadcasting industry for automated video annotation and metadata extraction tools.

The proposed work is framed in the DataCloud project[4], particularly on the MOGSports business case. MOGSports focuses on the media and entertainment business and aims to enhance the user experience, engagement, and involvement during live sports event broadcasts.

This paper explores the development of a system that assesses multi-camera video content and provides visual representations to enhance the user experience on live sports broadcast. The proposed data acquisition setup includes two cameras to extract information from the entire field at all times. The system extract metadata by a detection and tracking algorithm for each camera. This uni-camera information is then fused, creating a final multi-camera global result. The solution must be deployed in a Cloud environment along with the project objectives. Moreover, the project relates to complementing a broadcast, so real-time or near-real-time extraction is necessary. Thus, the size and velocity of the methods used are critical aspects.

Football matches video contains prominent and diverse information that can be extracted to enrich the content of a broadcast. This paper presents a solution for extracting player position and team information, in near real-time, throughout the match, providing the equivalent positional representations as output. The proposed methodology is crucial to enable further complex analysis such as goals, passes, player velocity and other statistics extraction. Furthermore, the applications of the retrieved positional data are manifold since it is the basis for most sport-related Machine Learning work, making it an excellent contribution to the Sports Data Mining Community.

Following the context provided in the current section, a literature overview on sports analytics, computer vision and machine learning is given in Section 2. Section 3 presents the pipeline of the project at hand and detailed information about its modules. Section 4 presents the results obtained with the proposed solution, and section 5 describes its positive and negatives sides as well as recommendations and ideas for future work. Finally, section 6 summarises the work done for this paper.

2 Related Work

Investigation into sports-related areas has been getting much attention. Many studies have been conducted in Computer Vision and Deep Learning to extract football matches information through video. This paper covers the areas of Field Registration, Object Detection and Object Tracking. All to extract information from a video of a football match.

The task of sports field registration aims at determining a homography matrix H for the transformation of an image into two-dimensional sports field coordinates [9]. The homography is defined by a 3x3 matrix with eight degrees of freedom that performs mapping between two images of a planar surface from different perspectives. There are two traditional homography estimation approaches,

[4] Horizon2020 grant agreement nº 101016835

direct and feature-based methods, which can also be used together to increase robustness [5, 6]. The direct methods use pixel-to-pixel matching by shifting or warping the images relative to each other and comparing the pixel intensity values using an error metric. The feature-based approach first extracts keypoints in each image using local feature detectors to establish correspondences between the two sets. Also, inspired by the data-driven Convolution Neural Networks (CNN) in CV, approaches for homography estimation have emerged using supervised methods. For this extent, [6] regresses a set of dense features through a deep network, to estimate the homography. Another approach is to use unsupervised methods [5]. Generative Adversarial Networks (GAN) and synthetic data have also been proposed [1]

Player detection is a case of object detection but with additional challenges. When performing player detection, several associated challenges must be faced to obtain successful results: the occlusions between players, the abrupt movements of the camera, the changes in lighting, the lack of resolution in very distant players, the blurring of the players that are moving, or the players remaining long static periods [3]. When [8] compares object detection models such as Faster R-CNN, YOLO and SSD using different backbones and other parameters for a field hockey dataset, it concludes that these complex neural networks fail to detect occluded players and players with low resolution (far from the camera).

Player Tracking is a Multi-Object-Tracking (MOT) problem. The MOT methods can be grouped regarding the initialisation method into two sets: Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT). Briefly, DBT, commonly named tracking-by-detection, happens when objects are first detected and then linked into trajectories. Thanks to improvements in the performance of object detectors and the ability to deal with challenges such as cluttered scenes or dynamics of tracked objects, tracking-by-detection has become a leading paradigm for MOT [4]. This approach is followed by methods like SORT, DeepSORT, TRACKTOR++, FairMOT, TransMOT, and StrongSORT. These methods make the tracking step an association step for the detections received by the detector.

The game analysis applications have a variety of topics: action spotting or event detection, player and ball tracking, tactics analysis, statistics collection, pass feasibility, talent scouting, game phase analysis, or highlights generation [2]. Furthermore, there is work on the enhancement and experience area, like [7] that uses augmented reality to present a match in a 3D model on a flat surface.

3 Proposed Solution

This section describes the MOGSports modular pipeline shown in Figure 1, centralising the AI Event Detection and Metadata Fusion modules.

The pipeline was designed to enable the use of N cameras, each undertaking an AI Event Detection module. This module includes three steps: Field Detection, Player Tracking and Event Detection. This paper presents the first two modules.

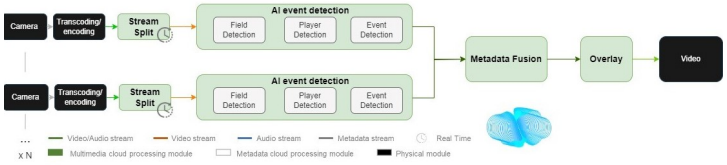

Fig. 1. MOGSports pipeline

Field Detection encapsulates the task of Field Registration. Resorting to a homography matrix, Field Registration allows the transformation from the game image to a top view perspective of the field. With the players' detection representations, this top model allows a better understanding of the game.

The Metadata Fusion module combines the information retrieved from each camera video processing. In this sense, the fusion module must adjust accordingly, accompanying the camera scalability.

3.1 Camera Setup

Whilst using more cameras enables detailed field coverage, it also increases the computation effort when matching the extracted data. Hence, the counterbalance between the number of cameras and the fusion algorithm complexity must be evaluated.

In this sense, the proposed setup comprises two static cameras placed along one side-line, as shown in Figure 2, guaranteeing the coverage of the entire field with significant detail. Furthermore, this setup allows the design of innovative strategies to fight occlusion, one of the problematic aspects of payer detection and tracking task

This setup was tested at Estádio Municipal de Aveiro stadium, and the cameras were around 20 meters high and had around 15°tilt.

Field Detection The Field Detection problem relates to the homography matrix computation between the captured image and a top-view field image.

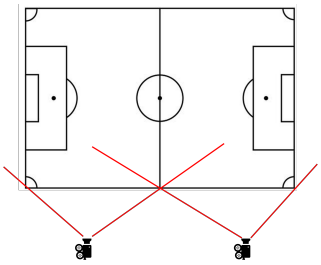

Fig. 2. Proposed Camera Setup

Since the setup cameras are static, the homography computation only needs to happen once at the match's beginning. The premise of a constant homography makes the registration process exponentially cheaper when compared to a moving camera, where it can change several times per second.

Furthermore, this premise also enables a lightweight manual registration. Thus, the proposed solution integrates a manual registration process, requiring the user to select four or more corresponding points in one video frame and on the top view field image.

3.2 Player Tracking

Player Tracking allows the identification and re-identification of the detected players throughout the frames, as shown in 5. Tracking can create and represent more valuable information that augments the user experience, such as other visual representation options or individual player statistics. The chosen tracking algorithm is based on a tracking-by-detection methodology, so player detection is a fundamental step. Without a robust detection model, the tracking algorithm will perform poorly.

Player Detection : The detection algorithm must be as accurate and fast as possible, enabling the pipeline to run in real-time. Also, it must be precise to strengthen the tracking step. These requirements drove the choice of the one-stage algorithm YOLOv5.

The players' position in the field is set where they stand, at their feet. Foreseeing that the player is in the centre of the retrieved bounding box, its position is defined with the bounding box's bottom centre coordinates. Once mapped by the homography, this point is the player's position in the top-view representation. The top-view field image is predetermined, with known field lines coordinates. Thus, the players projected coordinates can be easily filtered around the field area. Hence, the off-field detections are removed before being forwarded to the tracking algorithm.

DeepSORT : The tracking algorithm integrated with the solution is DeepSORT. Despite sporadic accuracy issues, it is one of the most suitable tracking algorithm for the time-quality balance among the state-of-the-art.

This algorithm contains a motion and a deep association metric. The proposed solution uses an OSNet, trained for person re-identification, to extract the features needed for the deep association metric.

3.3 Metadata Fusion

This module is responsible for receiving the cameras' metadata and merging it into a single, complete output. It is also responsible for assigning the tracked players to their teams, creating visual representations and storing the information created for other possible objectives.

To fuse both cameras' information, the detections are filtered after mapping the players' position into the top view according to their field side - the left camera detects the players on the left side and right on the right side. This strategy requires the definition of the midfield line pixels, in the top view representations, as a separator of the cameras' input.

Team Labelling The players' positional representations must illustrate their respective teams, which are recognised through their outfit's colour. Thus, the objective is to analyse the bounding boxes' colours and distribute them to the teams. For this, a K-means clustering algorithm allows the aggregation of team players based on their outfit colours. As the field players' colours are the most meaningful to differentiate, the number of clusters is two. Regularly, as the players are in the centre of the bounding boxes, the value used to represent them is the average of the pixels' colour of a 5x5 centred area. The model is built with the colour values of the bounding boxes from both cameras' first frame. After fitting the model, the following frames' bounding boxes' colour values are fed to the model to predict its cluster and, consequently, its team. After predicted, the colour is the same throughout the tracking ID lifetime. The team labelling algorithm operates within the fusion module as the k-means model should be the same for both cameras.

Visual Representations Three types of visual representations are defined for the proposed solution based on the players' positional information, as shown in Figure 3. The first two provide a spatial-temporal representation of the players' position at a given moment. However, the second also represents the past positions of that player, giving it a notion of direction. The third is a heat map that indicates the field areas with more presence throughout the game.

4 Results

The results of the proposed solution are promising, while unveiling technical and scientific challenges that need to be addressed in the future. The proposed solution can perform player detection and according representation, namely the positional plot and heatmap, in real-time. However, due to hardware limitations, the system can not perform online tracking.

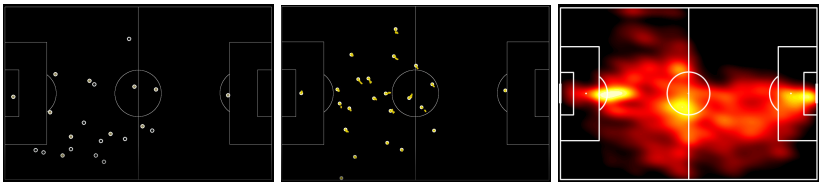

Fig. 3. Visual Representations

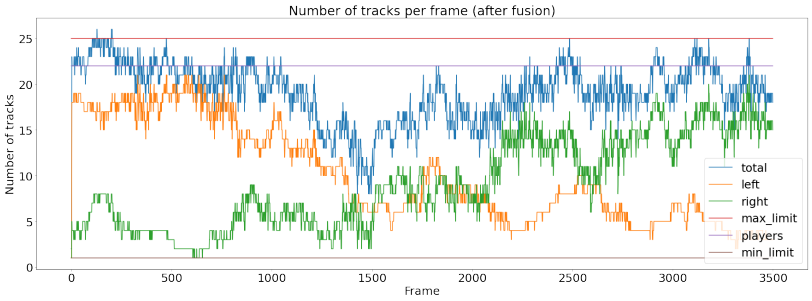

Fig. 4. Number of tracks per frame, fused tracks (blue), left camera (orange), right camera (green). Results obtained with reduced sample (around 3 minutes) of a custom dataset with proposed camera setup.

To evaluate the overall system behaviour, Figure 4 illustrates the number of tracked players per frame for the right, left and fused outputs, during a significant time span. This plot shows many variations in the detection and tracking models. Overall, the number of tracks is around 22 persons, which is a good indicator. However, when the left camera has fewer detections, the right camera can not cope with the players on its side, decreasing the total number players tracked. This is a consequence of a video quality discrepancy between cameras, since the right camera quality is significantly lower than the left (see Figure 5 and 6). In fact, all results are affected by this limitation, which will be taken into account for future data acquisition experiments.

The main consequence of these variations is the number of ID switches. DeepSORT was selected based on its ability to fight this problem. In light of these results, the deep metric used may be less relevant than expected. The auto-encoder used for this metric is explicitly trained for re-identification, so this is presumably not the faulty step. The reason may be the distance between the players and the camera. Despite the camera placement advantages relative to the field coverage, the players become small, which may affect the deep metric precision. Another aspect affecting the number of ID switches is the players switching midfields.

Another cause for variations are occlusions, illustrated in the left side of Figure 6. These are are hard to oppose with a single view. Therefore, the proposed

Fig. 5. Player tracking example for a frame inside the interval [1400-1600] in each camera (cropped image). Blue: Predicted tracks. Red: Miss detections manually annotated

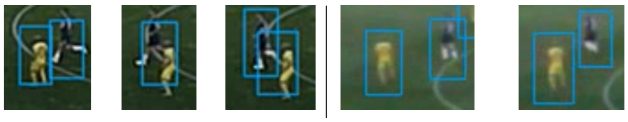

Fig. 6. Camera Setup advantages for occlusion handling

camera setup can help with some of these occlusions, shown on the right side of Figure 6.

Regarding the current fusion approach, even though the detection works suitably, some loss of information may occur. As Figure 5 shows, if one camera detects the players that the other does not, if on the opposite midfield, these detections are discarded.

Team labelling has satisfactory results. It assigns the expected colours for a high number of tracks, with occasional errors, specially when noise is in the bounding box.

5 Discussion and Future Work

As previously mentioned, the presented solution is a work-in-progress in the scope of MOGSports and DataCloud. In this section are discussed some of the positive and negative aspects of the implemented solution and of the next steps. The current solution encompasses modules that perform field registration, player detection and tracking, team labelling and fusion. Manual registration is a straightforward method for computing the homography matrix, which benefits from static cameras. Despite user-friendly for few cameras, automation is required for scalability. Accordingly, an automated approach is being implemented and tested. The Two-GAN model [1] is used for field line segmentation. Then, the solution attempts a feature-based approach, comparing the top-view and the segmented image. Along with the tests conducted until now, the Two-GAN line segmentation is expected to improve with more training data, primarily similar to the ones acquired in the proposed setup.

The Player Detection model presents promising results, still requiring further fine-tuning.Players' unusual positions or fast changing movements increase the difficulty of accurate detections. Furthermore, the many occlusions during a match lead to unstable positional information retrieval and consequent unstable visual representations. Future work is expected to improve the used YOLOv5 model performance resorting to the acquired dataset. Detecting and representing the ball is also a project goal.

As for the Player Tracking module, due to hardware constraints, DeepSORT is not yet thoroughly tested for real-time processing. Nonetheless, an unwanted ID switching issue has been identified. Consequently, the algorithm can suffer some modifications. As presented in Section 4, the deep metric may have a lower impact than initially thought. Other stat-of-the-art methods for MOT, such as the latest version of DeepSORT (StrongSORT), may be adopted to

strengthen this module. Team information could also be a valuable metric for the association step, neglecting possible matches with players from the opposite team. This strategy would require a very robust team labelling model. Tracking can also occur after the data is fused. Algorithms like graph-based tracking on the transformed top-view detection coordinates can be an accurate option. This strategy would need more development time and a robust fusion algorithm as a previous step. Alternatively, tracking can occur after data fusion, resorting to algorithms like graph-based tracking using the transformed top-view detection coordinates. This strategy would demand a robust fusion algorithm as a previous step.

Despite being accurate for numerous players, team labelling also has limitations. As players move rapidly, their bounding box may temporarily not fit its body. Hence, background or other artefacts can influence the colour value calculation. In this sense, the method could be improved by assigning each player to the team associated with the mode of a periodic check prediction or of the N initial frames predictions. Furthermore, as the goalkeepers have a generally fixed position, their colour could be assigned by it.

Fusion is critical, and a scalable solution is necessary for this module. The cameras provide a good overlap area that can be used for occlusion handling and to complete miss detections from one another. Incorporating a matching algorithm into this module could leverage the full potential of the proposed camera setup. One possible strategy would be associating the players from each camera in each frame. This method could consider all players, calculate the euclidean distance between the transformed points, and then associate them optimally. Being a computationally demanding task, other parameters, such as the team label, could be used to reduce the distances calculations and consecutive matching. The quality discrepancy between cameras used for the data acquisition affects the results negatively. Consequently, the debated problems may be softened when considering a dataset with equally high-quality images.

Event Detection is a comprehensive problem that MOGSports intends to solve, which reflects the culmination of the developments from the previous modules and its valuable metadata extracted. It may comprise match highlights (e.g. goals) as well as statistic insights.

6 Conclusion

In response to audience demand, user experience enhancement is an imperative action for the sports broadcasting industry. This paper presented a step forward to accomplish this enhancement, based on players' positional data extraction and representation. Extracting this data has complex associated problems and the results reflect this complexity. However, there are one or more proposed alternatives for each problem encountered. The proposed solution reflects ongoing work. The results achieved so far create good expectations for the upcoming developments and for the potential of the MOGSports business case within the media and entertainment industries.

Acknowledgements This work has been supported by the H2020 DATACLOUD project, funded by European Union's Horizon 2020 research and innovation programme under grant agreement No 101016835, as well as by the National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

References

1. Chen, J., Little, J.J.: Sports camera calibration via synthetic data (2018). https://doi.org/10.48550/ARXIV.1810.10658
2. Cioppa, A., Deliège, A., Magera, F., Giancola, S., Barnich, O., Ghanem, B., Van Droogenbroeck, M.: Camera calibration and player localization in SoccerNet-v2 and investigation of their representations for action spotting (2021). https://doi.org/10.48550/ARXIV.2104.09333
3. Cuevas, C., Quilon Gonzalez, D., García, N.: Techniques and applications for soccer video analysis: A survey. Multimedia Tools and Applications **79** (Oct 2020). https://doi.org/10.1007/s11042-020-09409-0
4. Ivašić-Kos, M., Host, K., Pobar, M.: Application of deep learning methods for detection and tracking of players (Feb 2021). https://doi.org/10.5772/intechopen.96308
5. Nguyen, T., Chen, S.W., Shivakumar, S.S., Taylor, C.J., Kumar, V.: Unsupervised deep homography: A fast and robust homography estimation model (2017). https://doi.org/10.48550/ARXIV.1709.03966
6. Nie, X., Chen, S., Hamid, R.: A robust and efficient framework for sports-field registration. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1935–1943 (2021). https://doi.org/10.1109/WACV48630.2021.00198
7. Rematas, K., Kemelmacher-Shlizerman, I., Curless, B., Seitz, S.: Soccer on your tabletop (2018). https://doi.org/10.48550/ARXIV.1806.00890
8. Sah, M., Direkoglu, C.: Review and evaluation of player detection methods in field sports: Comparing conventional and deep learning based methods. Multimedia Tools and Applications (Jun 2021). https://doi.org/10.1007/s11042-021-11071-z
9. Theiner, J., Gritz, W., Müller-Budack, E., Rein, R., Memmert, D., Ewerth, R.: Extraction of positional player data from broadcast soccer videos (2021). https://doi.org/10.48550/ARXIV.2110.11107