ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

University of Piraeus
Department of Digital Systems

Automatic Mapping of Football Team Formation using Computer Vision

by
Filimon Trastelis

Thesis submitted in partial fulfilment of the requirements
for the degree of MSc "Information Systems & Services"
in the Area of Study "Big Data and Analytics"

April 2022

Acknowledgements

I would like to thank a number of people for their encouragement and assistance throughout the writing of this thesis.

First and foremost, I would like to thank my supervisor, Dr. Ilias Maglogiannis, Professor and Head of the Dept of Digital Systems in the University of Piraeus, for the constant guidance and support during this period. I would also like to express my gratitude to the Senior Teaching Fellow Konstantinos Moutselos for his valuable help and advice.

This thesis would not have been possible without my parents, who are always there by my side. Last but not least, I would like to thank my sister and brother for all their love and support.

Automatic Mapping of Football Team Formation using Computer Vision

by

Filimon Trastelis

ABSTRACT

The integration of artificial intelligence in the area of sports is showing an upward trend the past few years. There is now a vast amount of generated data that can be processed and transformed into useful information in various aspects of a high demand sport such as football. In this thesis, primary target is the extraction and visualization of tactical statistics from broadcast videos of football matches. Continuous video frames are being processed and analyzed, in order to detect players positions and project both teams formations on a static top view of a football field. The emerging knowledge from the results can be valuable for the examination of the tactical team's performance during the game. Deep learning methods from the area of computer vision are applied to fulfill the tasks of players identification and camera pose estimation. A state of the art data set is used for the evaluation of different frameworks and the whole procedure is validated using image sequences acquired from Greek football matches. Ground truth data were generated manually for the two main tasks and the predicted results were compared with them to measure their accuracy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sports industry yields a huge amount of money every year and millions of people earn a living in this business. The popularity of sports reaches a high percent of the overall world population, making this industry one of the largest ones in the United States and Europe. Sports companies are constantly seeking new ways for the evolution and growth of sports that are interested in, especially the latest two years where the global pandemic of coronavirus took place and broadcasting turned to be mandatory for the industry's survival. The Deloitte Annual Review of Football Finance in 2021 [1], assesses the financial shrinkage of European football market to 13% in 2019/20, presenting thus the first recession in sports since the global financial crisis in 2008. Despite this downturn, according to Sports Global Market Report in 2021 [2], the effort of companies to recover from the consequences of pandemic expected to lead the global sports market to an annual growth rate of 13,5%. The leading factor for an effective restructuring could be data monetization.

More than a few decades, data are collected in a variety of sports, representing most times discrete features, such as goals or points scored from a player or a team. The derived information are giving a slight advantage to the people in charge of teams, not producing however sufficient valuable knowledge and raising the need for more advanced information.

The past few years, the rapid enlargement of Internet of Things has a great impact in sports. An ideal example is the GPS trackers on the ball and players that has generated high volume of data and has given the ability to many people that are involved in an activity or experience focused on a sport to collect data in real time and gain useful insights for the evolution of their sport. Sports fans and media coverage are also benefited from these data and the statistics created from them. The application of visualization methods to sports data has brought out a whole new fan experience.

Sports analytics nowadays is the key for most players and their teams to gain a competitive edge over their opponents. Players are able to track how many kilometres they have run and how much energy they have expended. Coaches in team sports have the ability to observe each player's movements during a game from both teams and get an overall view of their team's effectiveness.

The primary reason for this expansion in advanced statistics is the entry of Artificial Intelligence (AI) and, most precisely, of deep learning techniques that are implemented in a field of AI named computer vision. Main objective of computer vision is the extraction of information from images or video frames of a game in the case of sports. A

diverse set of tasks in this area is making use of methods based on convolutional neural networks. Some of these tasks are object detection, real time action recognition and object tracking. Major professional sports leagues, like the National Basketball Association (NBA) and the National Hockey League (NHL), utilize these techniques as part of sports systems for ball tracking or shot recognition.

1.1 Problem Outline

In this thesis, the main target is to provide the means to people in charge of a football team to observe their team and their opponent formation during a game. It is examined the process of developing a tool for a sports application that needs the least inputs from a user and is able to project the formation of both teams from the input video of a football game. Specifically, the video frames from a main camera view are being processed and the final result is the projection of the players and the area that they are covering on a top-down view of a football field template.

In order to accomplish this project, it is a necessity to carry out in the first place two main tasks. The first one is the object detection in every frame. Through this task, players have to be localized and along this, to be assigned a label of the team that each of them belongs. A deep neural network-based detector, named FootAndBall [3], is used for the players and ball recognition. Team identification is posing as a classification problem where primary color features are extracted from players jerseys, K-means classifier is applied and each label corresponds to the players of a team. Furthermore, an additional process is taking place with purpose to predict the primary colors for both teams and correspond each team to a label based on its colors. Subtraction of the referee and the goalkeeper, when they are detected in the frame, enhances the effectiveness of this process.

Subsequently, players positions have to be projected on a football field template. This goal can be attained through the generation of the ideal homography. The visible football field from each broadcast frame should be transformed to top view frame and this demands the extraction of a highly accurate homography. This piece of work can be executed manually with the selection of at least four points across both the frames.

However, this is not a practical method, as the system must convert the players coordinates automatically. Consequently, two methods are examined one at a time for the field registration challenge. Initially, an automatic method for sports camera calibration [4] that detects features such as grass field or field markings and implements a camera pose refinement process. In the second place, a framework [5] that utilizes a deep neural network to get from a pose a first estimation of the homography, with which the football field template is warped and it is concatenated with the original pose. Then, this image is used as input to a second deep neural network, where the homography is optimized through the estimation of the warping error, completing this way the refinement process.

After the homography estimation is integrated with team identification in a uniform module, the principal idea for the extraction of team formation is to find the regions on the football field where the players for each team have been most detected. For this

reason, it is essential the split of the field in regions, forming in effect a grid on the field template with equal cells. Then, the number of players from all the frames is computed for every region, showing where each team is most dominant in the field.

Additionally, an alternative concept is to find a way to transform the input video in a new one with the same frame rate that displays the players from each team and the area on the field that they are covering. For the purpose of this task, it is required each frame to be processed and to project on the field template the area that each team rules with a heatmap. In this way, form of the map renders the team formation. All the resulted images shall be combined in a final video which is the output. Finally, the estimation of players and ball coordinates on the template can be exploited to measure the ball possession for each team over the whole match and with the use of a heatmap to show these regions where a team controls most the ball.

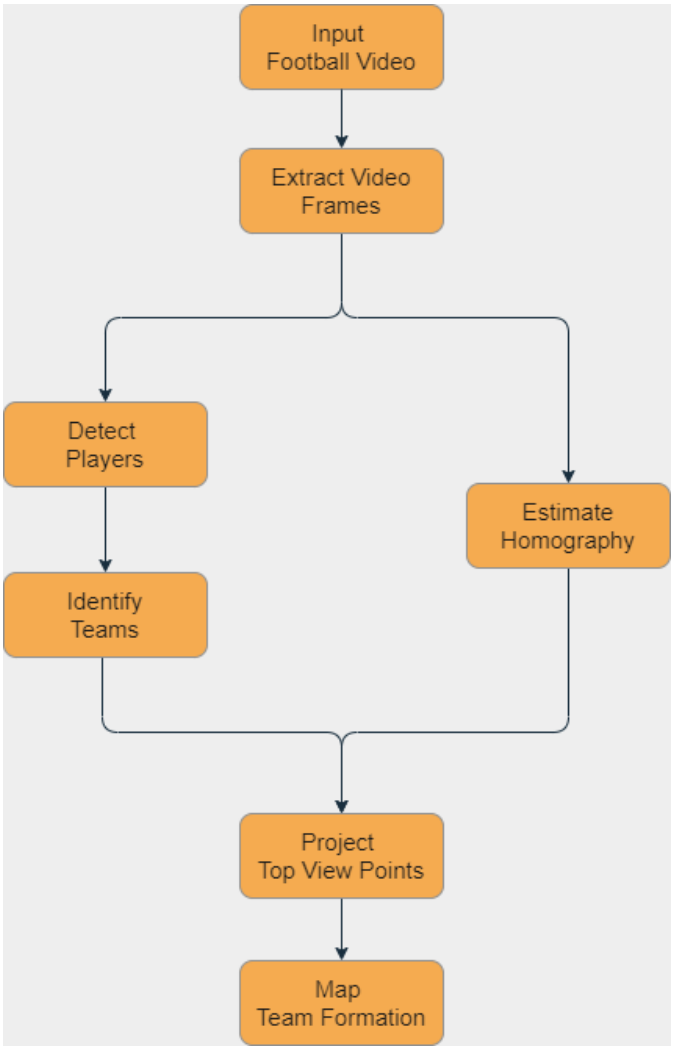


Figure 1: Abstract Process Flow

1.2 Challenges in Football Video Analysis

The technical analysis of a video from football match occasionally leads to different challenges that emerge during the following phases of the procedure.

Homography Estimation

Extraction of the right homography is the most likely phase to reveal crucial problems. Video frames with low resolution may affect notably the recognition of the field, as the lines intersections or the field corners could not be identified. Moreover, these features will not be recognized if the weather or the field conditions are not the appropriate. Sunlight at the early afternoon is possible to cause long shadows and adversely affect the quality of football videos. If the weather is too rainy or a snowfall is occurring, then long parts of the side lines may not be visible and the shades of the field's green will not be the same as in other games.

In a football video, there are often frames with irrelevant information, like a product's advertisement, that can raise an error during the court detection phase. In addition, there are times when the camera zooms enough to get a closely look and frames then, do not contain the necessary features of the field that will generate an accurate homography.

Player Detection

Frequently, during a football match, players from both teams are trying to obtain the possession of the ball. Because of this action, they are captured in some frames closely enough, making their detection impractical. A player may also be in close contact with the referee, affecting negatively the process of referee's subtraction, besides players detection. In other occasions, players recognition is extremely hard when they have fallen on the pitch or their poses are uncommon. To overcome former problems, a possible solution is the utilization of multiple cameras during a game or the tracking of players from previous frames.

Ball Detection

The recognition of the ball during a football match is an extremely demanding task because most of the times, the ball is positioned quite near to a player's feet and it is occluded. Moreover, the ball may be in a constant movement for a long duration and in combination to its small size, it is possible to be displayed in many frames as blurred. In these cases, the detection can be improved with a ball tracking method that keeps a record for the motion state of the ball and uses a robust model to predict its next positions. Another problem that must be pointed out, is when the ball has been kicked and it is in the air, making the estimation of its coordinates extremely challenging, especially when it is detected from a 2D image, where it may seem that the ball is outside the field. The most efficient way to face this issue is to measure the 3D position of the ball, a solution that definitely needs the coordinates of the ball from different angles, but it can also be tackled with the measurement of the motion features.

Team Classification

Color extraction follows after object recognition from its player image. This task can turn to be problematic, due to shadows or other weather conditions that would lead to inaccuracies in the color identification. After the application of classifier, there are times when the same label is corresponded to players from both teams. This incorrect clustering is caused often, because of the erroneous subtraction of the referee or the goalkeeper. Other cause of the mislabelling, is when a detected player is too close with an opponent and is not extracted the correct color feature. Consequently, a cluster with different players is not clear which team represents, thus a secondary process has been developed for the assignment of the correct color to each label.

Team Formation

Before a football match, the coach of a team chooses specific areas where each player has to be, but during the match, players are often in different areas, especially when there is a corner kick or a foul where the ball is stationary and most players are in one side of the pitch. Due to these type of events, it is more essential to display the areas on the field where the players of each team have most been over the match and project on a football field template how players moved during a specific case, like a conducted shot or a goal scored. This final stage of the procedure is heavily dependent on the previous ones, as it will fail if there is an inaccurate estimation of the homography or a mistaken classification of teams.

1.3 Contributions

This thesis examines all the stages of the procedure that are necessary for the fulfilment of the project. Methods from different papers are analyzed and integrated in a sequence for the composition of the process. Videos of Greek football matches are used for the evaluation of these methods in each phase, assessing with this way how valuable can be this procedure for both professional and amateur Greek leagues.

For the homography estimation, two methods are compared and tested on the World Cup dataset, the same one as in the relative papers. First, Jianhui Chen and James J. Little [4] propose an automatic method on sports camera calibration that is used for field recognition and camera pose estimation. In contrast to this method, it is tested an alternative framework [5] which predicts a roughly homography estimation, before it refines it with the use of registration error. In this work, the methods are evaluated further with Greek football videos from leagues of different levels.

After the object detection phase, primary color features are extracted from the players and the referee, if the latter is in the frame. Images with a referee or a goalkeeper in them are manually annotated and the process for their subtraction is evaluated for its effectiveness. Players color features are classified in two clusters with a K-Means classifier. Players from various frames are annotated from scratch also and the accuracy of the classifier's predictions is evaluated. After the teams labelling, a model constructed and trained from scratch, predicts the color from the players features for each label and helps to find the primary colors for both teams.

The proposed system in this thesis can be used for the production of accurate heat maps that display the players positioning during a football match and gain helpful statistics about the ball possession. Furthermore, it might be valuable for the development of a more compound sports application which shall be a valuable tool for coaching analysis.

In summary, primary contribution of this work in the sports analysis as a field of research, is the comparison of two methods for the task of sports field registration, based on a published data set and on frames from Greek football videos. A process for extracting the primary features from players uniforms is also evaluated and a model is created for the prediction of teams colors from the players features. Additionally, the overall unified procedure for generating tactical heat maps is applied on the Greek football data sets, providing thus useful insights and enhancing the effort for sports systems growth in Greek sports market.

Chapter 2

Related Work

Sports analysis is an area of study with increasing number of scientific papers published every year. A variety of researches and methods, from sub-fields of the Computer Vision related to the tasks that are examined in this thesis, are cited below.

Sports Camera Calibration

A sports application that generates players statistics from input videos has as a requirement the estimation of camera pose. Most of the time during the broadcast of a football game, the camera follows every action on the court. Camera angle has a vital role for the observation of a sports event, as only with the knowledge of its parameters the details can be interpreted into game statistics.

The calculation of camera parameters from a video frame is a powerful tool for every system that transforms visible court and players positions to coordinates on the model of the playing surface. A successful camera pose is extracted through the computation of an accurate homography. Homography is a 3×3 matrix with main function to align images in the case of pure camera rotation or a common planar scene. In detail, features are selected from each image along with the right matching algorithms to make the correspondence between the images [6]. At least four points from each image are necessary for a precise matching. With a known homography, players can be observed through a top view angle and their positions are far more obvious.

A wide range of researches look into the problem of registering the field from a broadcast video frame on a top view model. Over a decade now, the registration task is examined using feature detection techniques, like in [7] and [8], with which the local features of a video image are matching with the ones of the selected model images. Hess and Fern [9] later, present a method that detects invariant image features in the broadcast image using the SIFT descriptor [10] and matches them to features of the model images, facing though the problem of few visual field features. They are trying to minimize the manually annotation of the correspondences and enhance the approach of Okuma, Little and Lowe [11], who used the Kanade Lucas-Tomasi (KLT) tracker and the RANSAC method [12] for the image features detection and the estimation of the homography in hockey games. Human intervention is noticed also in [13], where is initialized the first frame's homography. However, the manual selection of the right points to produce homography is not a practical technique for any sports system that processes video sequences.

Another issue to be encountered is when the number of field markings in the image is limited. In [14], is proposed as a solution a two point method that needs only two points

to calibrate cameras, with a crucial limitation the demand of a pan-tilt-zoom camera's base location and orientation.

An alternative approach [15] utilizes a deep semantic segmentation network to derive from a broadcast image all the primary field markings, like the field surface, lines and circles, that define an energy function. Then, achieves energy minimization in a Markov random field and performs inference with a branch and bound algorithm. This method is applied to the sports of football and hockey and is tested on a data set of football images taken from the World cup games in 2014. Later, on the same data set, is tested the efficiency of the method in [16]. This paper proposes a pipeline where from a football image is obtained a binary edge map, before it is used as an input for a nearest neighbour search over a generated dictionary of images with synthetic edge maps and corresponding homographies.

Sports frames are indicated often as challenging when most of the lines and intersections are obstructed, especially in sports as basketball. Framework suggested in [17] to face this problem, starts with semantic segmentation and then the camera pose initialization is following, using a siamese network to retrieve the field template from a pre-built dictionary. For the homography refinement as a final step, it is introduced the spatial transformer network to handle any large non-affine transformation.

A similar framework is proposed in [4] by Chen and Little, evaluated in the already known World cup data set. A camera pose engine and a siamese network are used to form a database with camera poses and their related feature vectors. The markings on the field are detected with two generative adversarial networks (GAN) merged together. Given a detected edge image and a camera pose from the database, two distance images are retrieved and the refined homography is computed with the Lucas-Kanade algorithm [18], a well known method for image alignment. Another algorithm implemented in this work for the homography refinement process is the Enhanced Correlation Coefficient (ECC) [19]. It is proven to be beneficial for image registration, increasing the structural similarity, quality and registration accuracy between adjacent images.

The former method is examined in this thesis and is compared to a method with optimization process for a more robust sports field registration [5]. This research introduces a deep neural network to estimate the homography from a football frame and warps the top view image of football field according to it. The warped image is merged with the original frame and the generated image is fed to a second deep neural network to evaluate the registration error. With the known error, the homography matrix can be updated and iteratively reach an efficient registration result.

A contemporary framework, also for sports field registration [20], faces the challenge of few recognizable features on the field with the detection of a grid from key points on the field. The key points are used to estimate the initial homography and a larger receptive field is integrated with the deep network for a more accurate detection. When the homography needs optimization, the network detects dense-frame features and aligns them with dense template-features, with purpose to enhance the homography estimate based on the key point. For the evaluation of the aforementioned method, a new data set was compiled with camera poses from five different sports.

Player Detection

Players recognition is a fundamental task in a sports analytics system based on computer vision. Deep neural networks are implemented in applications with aim to locate bounding boxes with players in a broadcast frame. The bounding box is a frame on the image with rectangular shape and describes the spatial location of an object. Many of the detectors are trying to locate on the frame and the ball along with the players.

Object detection has been the subject of many research studies over the past few decades. Traditional methods that have been applied in sports images, are using the field extraction as a primary process. In [21], the conversion of images color features into gray level values enhances the locating of players in white regions. A labelling algorithm and the Hough transform are processing the images and based on the minimum bounding rectangle, the average gray level, the Hough value and the compactness are distinguish the players regions from lines segments. Hough transform turned out to be valuable for the circle detection algorithm in [22], where the main purpose is to identify the circle in the football image that represents the ball.

Moreover, an alternative proposed framework [23] starts the process of players identification with a field detector based on color histogram for background extraction. Then, the Euclidean distance transform is applied to locate the frames with the highest potential to represent an object, before it continues with the further shape analysis. Another paper [24] introduces a framework for real time objects recognition in sports videos. This method tries to overcome the temporal constraint, using in parallel regions with accurate boundaries to refine the results of motion estimation.

Later, a trained deformable part model (DPM) was introduced in 2008 [25], making use of more latent information for object detection. The system presented in [13] uses the DPM to recognize players in basketball videos. The DPM consists of 6 parts and 3 aspect ratios and achieves high efficiency, fails though to separate the referees or to recognize occluded players.

More detection models came up the past few years and one of them is the R-CNN. This model feeds into convolutional neural networks (CNNs) a specific amount of region proposals in order to recognize and segment objects [26]. After it detects potential bounding boxes and classify them, a refinement process occurs to the bounding boxes and any duplicates are subtracted.

Because the train of this model takes a huge amount of time, the same author introduced an enhanced version of the former network, called Fast R-CNN [27]. The principal reason the new one is faster, is because it takes as an input the image instead of the region proposals, before it generates the convolutional feature map.

In contrast to the former two networks that use selective search to identify the region proposals, a more recent method predicts proposals with a separate deep neural network. Faster R-CNN [28] uses the convolutional feature maps to construct Region Proposal Networks (RPNs) by adding on top of them two additional convolutional layers. This model turns to be much faster than its predecessors as the detection time is improved drastically.

In addition to the Faster R-CNN, a later model called Mask R-CNN [29] was in-

troduced. This one is actually, an extended version of Faster R-CNN, with the only difference to be an added branch. Purpose of this branch is the prediction of an object mask along with the bounding box, giving the ability to estimate human poses.

A contemporary model is presented in [30], quite different to the aforementioned networks. You Only Look Once (YOLO) is an object detector based on a deep convolutional neural network. It predicts the bounding boxes and the class probabilities for these boxes, without separating region-proposal generation module. YOLO achieves a much higher performance in contrast to DPM or R-CNN and makes much less background errors compared to Fast R-CNN.

Another method with a single deep neural network is the Single Shot MultiBox Detector (SSD) [31]. It is similar to the YOLO and it takes only one shot to detect multiple objects present in an image. Both YOLO and SSD are large scale networks with an enormous amount of trainable parameters with a crucial limitation on detection of small objects.

Players and ball detection eventually achieves remarkable efficiency levels with the FootAndBall detector [3], an extended work of the DeepBall detector [32], a model with specialty only on the ball identification task. FootAndBall is a neural network-based object detector and it produces ball confidence map, indicating thus the position of detected ball, player confidence map encoding player's position and player bounding box tensor encoding coordinates of a player's frame. The model's design pattern is based on a Feature Pyramid Network [33], creating this way an architecture with rich semantics at all levels.

Ball Detection

The subject of the ball estimation over a sequence of football video frames has been used in a large number of studies, trying to face different issues that emerge during the detection and tracking phase.

In [34], the proposed method predicts the 3D position of the moving ball during a broadcasting football video and it can estimate the ball coordinates even when it is in the air. The method uses the Viterbi decoding algorithm to search the regions in adjacent frames with the highest potential to include a ball. After the ball is detected, the Kalman filter is used for the ball tracking, until it is lost and the detection phase starts again.

An other approach [35] introduces a method to detect the ball in challenging situations, like when it overlaps with players and lines. The method creates a transition graph to display all the possible routes of the ball based on the spatio-temporal relationships between players, lines, and the ball. The best route is selected according to the generated data that indicates the ball presence in every potential route.

The framework in [36], detects the ball from long shot video frames in real time. In every frame, after its transformation to a binary map, all the possible positions of the ball are detected using a rule based algorithm and then, they are filtered with the Kalman filter to find the best location.

The ball route is predicted further with the method in [37], where multiple fixed cameras are used to detect and track the ball from different angles, producing an accurate

estimation of the ball trajectory. Another research [38] uses a dynamic Kalman filter algorithm to track the ball, even when it is occluded for a long time, by controlling the velocity of the state vector.

Player and Team Identification

Players and referees recognition is not sufficient for the generation of sports team statistics. It is essential the correct correspondence of every player to each team. A diverse set of researches have look into this problem, either focusing on each player's identification based on prior knowledge or applying deep learning models for the teams classification.

The procedure followed in [39] uses the entire bodies or body parts along with a shared network to put a name on each player. Another approach [40] exploits the distinguishing features of a face to tackle this issue.

A well known strategy, implemented in many papers, is the jersey numbers recognition. In [41], number is separated from jersey with the use of HSV color space and the localization of number occurs with internal contours detection. Similar in [42], after the image segmentation, a recognition process is executed, including a K-NN classifier for the detection of most possible numbers in jersey. Recent approaches [43] and [44] are using deep learning knowledge to locate the jersey number in the player image. A residual network (ResNet) combined with a long short-term memory (LSTM) layer are forming the deep CNN in [45] as a proposed solution.

A different point of view in researches have tried to identify the players as members of a team considering the fact that players of the same teams have the same colors. An unsupervised procedure [46] creates the classes with a clustering algorithm and assigns the normalized color histograms to the correct class. With the same way in [13], image frames are corresponded to team labels where each frame is represented by RGB color histograms.

More recent papers are using methods for the label assignment to players, from the area of deep learning. The CNN trained in [47] generates pixel-wise descriptors that are close based on pixels depicting players from the same team, and the opposite when pixels correspond to different teams. A clustering algorithm then, classifies the players in two teams. Another CNN [48] is designed in a cascaded architecture for team classification, achieving to save memory with efficient performance. Both networks have as a requirement annotated data for the training process.

Team Formation Estimation

A modern aspect in football analysis is the observation of team formation in specific game events. An early research [49] is making use of spatiotemporal data and generates a model for the expected team behavior with the help of a codebook with past performances. An other one [50] computes the positions of each team's players and with a hierarchical clustering, after measuring the distances between formations with the Wasserstein metric, splits the offensive and defensive formations. Measuring the distribution of positions from a sequence of frames enhances the procedure in [51] to extract motion features and recognize a team's behavior during the game. The analysis of spatiotemporal correlation between players gives an accurate look of motion patterns and benefits this research [52] to distinct winning and losing teams. In [53] the methods

[4] [3] are combined to assess the side of the field that a team controls the most and detect if the team has the ball possession.

Formation recognition is often useful to observe offensive strategies of a team. A proposed solution [54] identifies the attacking team and trains a linear SVM with features from the offense side to distinct team formations. An alternative approach [55] focuses on the behaviour analysis of players with the same role and identifies the formation of these subgroups during attacking plays.

A recent paper [56] introduces a procedure where each player is assigned to a unique role using the expectation maximisation (EM) algorithm, instead of clustering each data point. Then, the probability distributions are computed, providing thus the formation over a match-half that represents best the team's data. Furthermore, challenging structures detected during a game are overcome by calculating the distance between the frame and the formation template. This approach is the extension of a previous work [57] and is trying to resolve the changing of roles between the players during specific time intervals when a team is constantly attacking or the opposite.

Chapter 3

Methodology

The procedure described in this chapter is composed from different stages. Each one of them is essential for the automatic execution of the proposed framework and the accomplishment of the final target, which is the projection of team formation on the football field template. The fundamental tasks are the homography estimation and the player and team identification. Prior to this phases, foremost need is the extraction and individual analysis of every frame from the input football video.

The initial phase is related to the homography estimation, a matrix that is needed to transform the coordinates of players from a frame to the ones on the template. Two major methods are examined separately for this task, an automatic sports camera calibration method [4] and a general framework for sports field registration [5].

Following the steps of the first method, a two-GAN model, which is based on a Pix2Pix network [58], is used to detect the field lines from each frame. Then, the image is fed into a siamese network that generates a feature vector. A database with camera poses and their related feature vectors is already built with the same siamese network and a camera pose engine. The closest vector to the one generated earlier indicates the camera pose that is most similar to the frame. The detected image and the camera pose are input to a refinement process, where the Enhanced Correlation Coefficient method is utilized to extract an accurate homography.

As for the alternative approach, a basic homography estimation of the input frame is generated with an initial registration network. The homography is then used to warp the football field template and the result is concatenated with the original frame. The created image is inserted in a registration error network, where the prediction of a registration error metric is used to optimize the homography parameters. The refinement process is conducted repeatedly until a highly accurate result is produced.

In parallel to the former phase, an object detector named FootAndBall [3] localizes the players and the ball in the frame. After the players detection, the primary color features from their uniforms are extracted. A created model predicts the color that each feature represents, so to be easy to identify and subtract any referees or goalkeeper from the detected players. Then, only the players are divided into two clusters based on their corresponding color feature. A process is taken place afterwards to predict with the use of the model the top color for each label and identify the home and away team.

With the knowledge of homography and the players positions in the frame, the coordinates on the football field template now can be computed. For every frame, the distance between the ball and the players is measured, so to find which team has the control of

the ball. A heat map for each team also projects on the template the occupied regions of the field. In addition, all the coordinates of players on the template are accumulated and based on a grid on it, the number of players on every region is calculated for each team. Thus, the areas with the highest percentage are the ones where the players of a team are most detected over the whole game. A similar process occurs also in the end to find these regions where each has most the control of the ball. Figure 2 displays analytically the steps of the process examined in this research to achieve the final target.

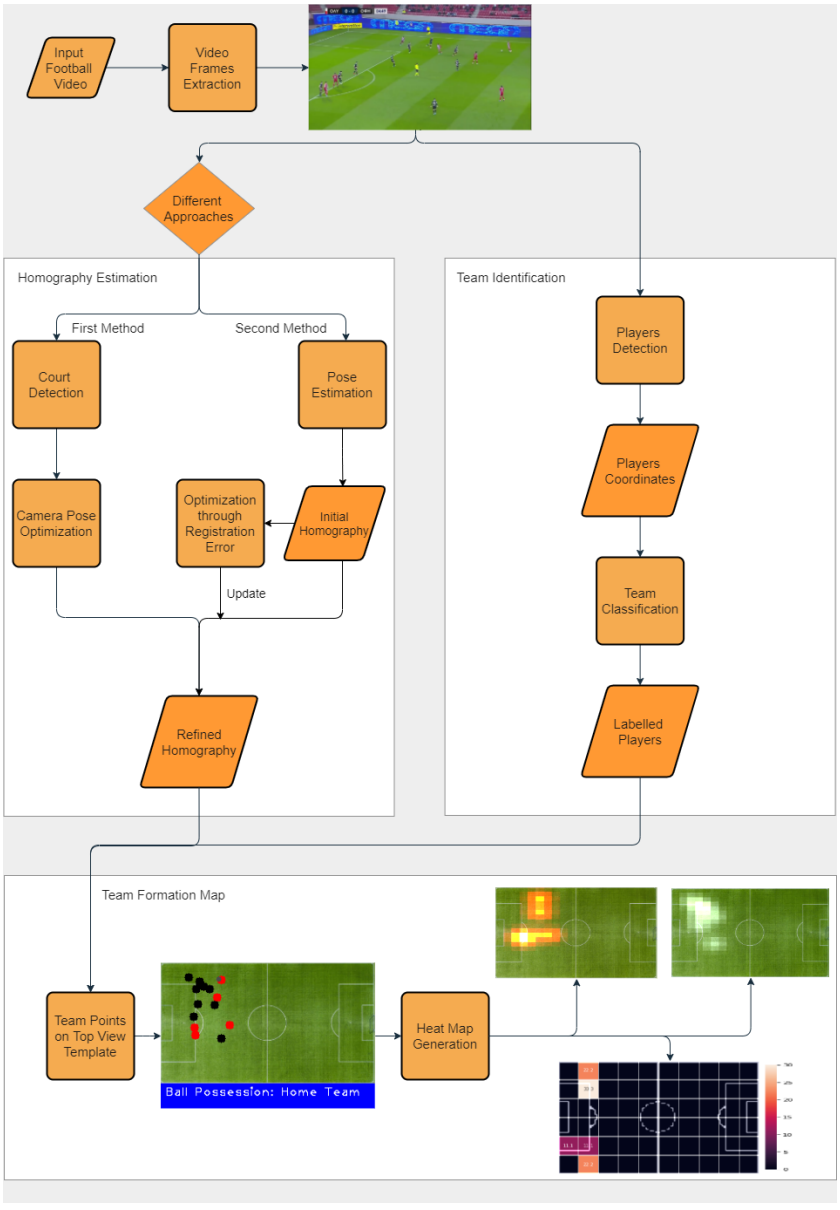


Figure 2: Research Methodology Flow Chart

3.1 Technology Infrastructure

For the needs of this thesis Google Colaboratory is used, an online cloud-based Jupyter notebook environment. All the training and testing procedures and any other necessary computation for the completion of the process are performed in Google Colab.

A requirement for the implementation of neural networks from the examined methods is to use Pytorch on NVIDIA GPU along with the CUDA Toolkit. Torch is a machine learning library which supports deep learning applications using GPUs and CPUs. The CUDA packages are essential for the development of applications accelerated with NVIDIA GPU. Pytorch version 1.10.0 and CUDA 1.11 are imported.

Various algorithms and programming functions of the OpenCV library are also used. OpenCV library, version 4.1.2, is an open source computer vision library for video analysis and image processing. Other important libraries installed are the NumPy version 1.19.5, SciPy version 1.1.0 and scikit-learn version 1.0.2. Another one, especially for the first method in homography estimation task, is the pyflann version version 0.1.0, a python implementation of FLANN - Fast Library for Approximate Nearest Neighbors.

During the training stages and experiments, the type of GPU provided is Tesla P100-PCIE-16GB and the model name of the available CPU is Intel(R) Xeon(R) CPU @ 2.20GHz.

3.2 Homography Estimation

The transfer of the players positions from the frame to a top view field is only feasible with the computation of the homography matrix. Two methods from different papers are integrated in this procedure with few modifications wherever it is necessary. They are analyzed and compared in terms of their effectiveness. The first method comes from the research in [4] and the second one is proposed in [5].

3.2.1 Initial Approach

Every frame of the input football video is inserted into a two-GAN model for the detection of the lines that define the football field. The output of this model is an image which displays only the visible lines of the field, namely an edge map. A siamese network transforms the detected edge map to a features vector. From the features-camera pose database is selected the camera pose with the closest vector to the former one and then, the corresponding edge image and homography are computed. The retrieved edge image along with the detected one are transformed into two distance images from which a refined homography is calculated with the ECC method. The process flow is projected in Figure 3.

The aforementioned database is already formed with the usage of the camera pose generator and the siamese network. Camera pose generator plays further a major role to the training of the siamese network as it produces sampled camera poses and their related edge maps. After the camera pose selection from the database, this tool retrieves the homography and edge map of the camera pose.

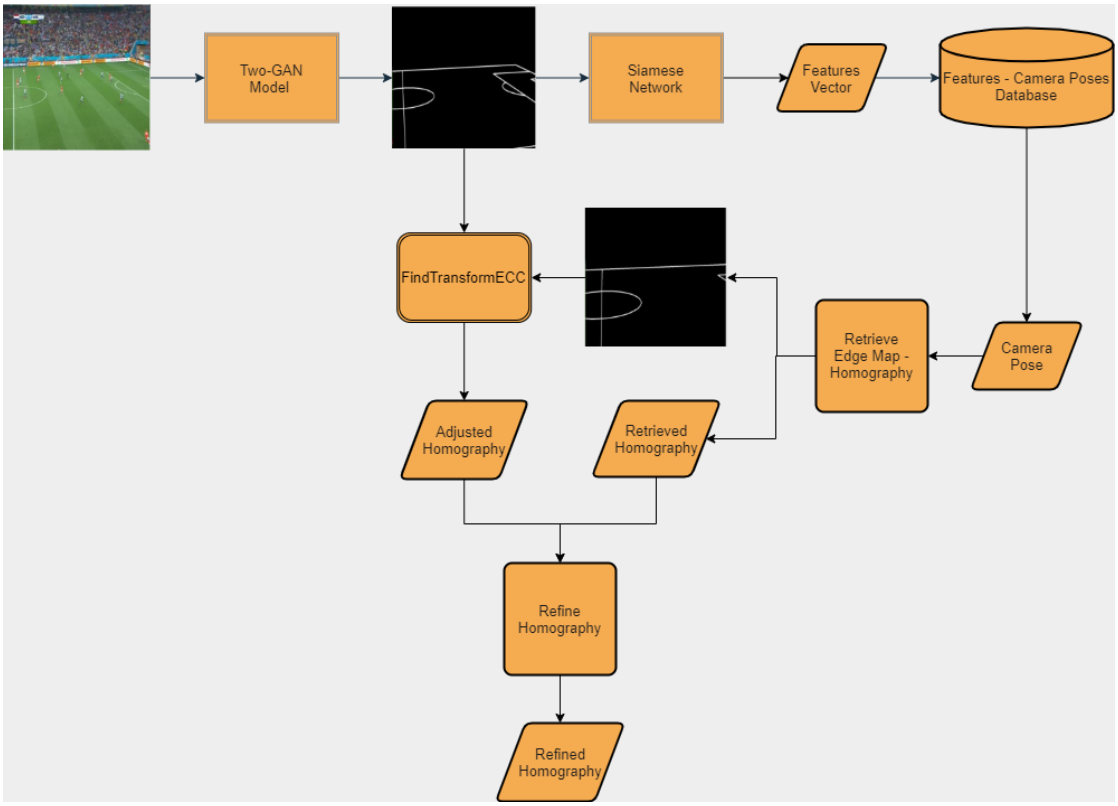


Figure 3: First Method Process

Camera Pose Generator

The cameras used to track down every event on the field during a football match are pan-tilt-zoom (PTZ) type, as they rotate very fast and can cover a wide range of an area. In order to obtain the camera configurations without knowing exactly the camera location and orientation, the pinhole camera model is used. Main characteristic of this model is that it can be helpful for the mapping of a three-dimensional scene to a two-dimensional image. The model describes a PTZ camera as

$$P = KR[I| - C], \tag{1}$$

where K is the intrinsic matrix, R is a rotation matrix from world to camera coordinates, I is an identity matrix and C is the camera's center of projection. Figure 4 displays the pinhole camera model according to the coordinate systems of the 3D world, the 3D camera and the 2D image.

Based on the assumptions of square pixels and a principal point at the image center, the K matrix is a combination of three parameters, the center (u,v) and the unknown focal length f. The rotation matrix R expresses the rotations from world to camera coordinates and is composed of Qϕ, Qθ and S. The metrics Qϕ, Qθ are counting how much the camera pans and tilts correspondingly, after the rotation of camera to the PTZ

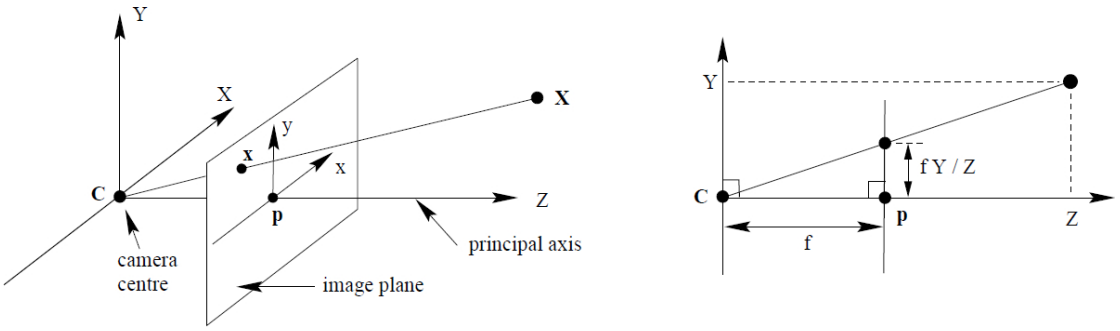


Figure 4: The Pinhole Camera Model

camera base by S. Matrix C describes the coordinates of camera center within world. After the decomposition of the base rotation S, the resulted camera model is

$$P = KQ_\phi Q_\theta S_\rho S_\varphi [I| - C], \tag{2}$$

where ρ and φ are roll and tilt angles of the camera base. To have a better understanding, Figure 5 describes the camera and world coordinates with black and red color respectively. The left bottom of the soccer template is defined as the origin of the world coordinate. The Y-axis in the world coordinate and the Z-axis in the camera coordinate are aligned when the values of pan and tilt are zero [4].

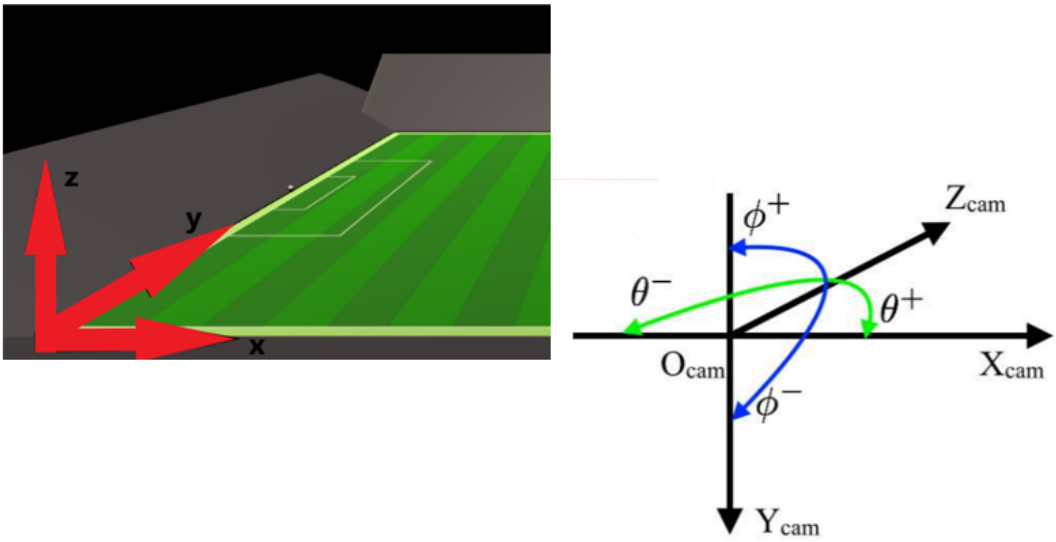


Figure 5: World and Camera Coordinates

After setting φ equal to -90°, and with ρ to fluctuate in a small range [-0.1°, 0.1°], sampled camera poses and their edge maps are produced by the parameters ϕ, θ and f. The edge maps are generated using a binary static top view model of the football field and they have resolution 1280 x 720. In this way, it is created the data set for network training and any edge map or homography from a retrieved camera pose.

Sampled Camera Poses Data Set is already built and is essential for the creation of the siamese training dataset and the forming of features-pose database. Images used as a basis for the sampling process come from the World Cup data set [15]. The processing of these images derived the camera location distribution $N(\mu, \sigma^2)$ and the pan, tilt and focal length ranges as camera configurations. The distribution metrics are $\mu \approx [52, -45, 17]^T and\ \sigma \approx \pm[2, 9, 3]^T$ and the respective ranges are [-35°, 35°], [-15°, -5°] and [1000, 6000] pixels.

The camera pose generator sampled 90,000 camera poses. The camera centers are sampled from the Gaussian distribution $N(\mu, \sigma^2)$. The pan, tilt and focal length values are sampled from the uniform distributions of [-35°, 35°], [-15°, -5°] and [1000, 6000] pixels, respectively. The tilt of camera base ϕ, as mentioned before, is -90°and the roll angle varies in a range $\pm(0.1°)$.

Features Extraction via a Siamese network

The siamese model is used to extract features from edge images, after its training with the edge maps of sampled camera poses. More specifically, the network takes as input a pair of edge maps. The network learns to define as similar or not, pairs of edge images according to their pan, tilt and focal length differences. The decision if two images x_1 and x_2 are close enough results from the loss function

$$L(w, x_1, x_2, y) = yD_w(x_1, x_2) + (1 - y)max(0, m - D_w(x_1, x_2)), \tag{3}$$

where $D_w(x_1, x_2) = ||f_w(x_1) - f_w(x_2)||_2^2$ represents the Euclidean distance between features vectors x_1 and x_2, y is the similarity label and m is a margin that extents the distance between dissimilar features vectors. In this way, it extracts more similar features vectors from close edge maps. This network has used to produce features vectors from all sampled camera poses and build the features-pose database.

Network Structure is composed of two convolutional neural networks f_w and each of them forms a separate branch. The network consists of 5 stride-2 convolutions (kernel size 7, 5, 3, 3, 3) followed by a 6 × 10 convolution and a L2 normalization layer. Based on experiments, the learned feature dimension is set as 16. The input edge maps are tensors with shape 1 x 180 x 320 and the output is a features vector with shape 1 x 16 that describes the input image.

Training Procedure needs a data set with edge maps to be put into effect. The input to the model is pairs of edge images along with their labels. If two edge maps are

seemed to be similar enough then their corresponding label as a pair is 1, differently the label takes the value 0. This process is inspired mainly from [59]. A pair of images is processed in the two branches and the outputs are the two features vectors. After the calculation of their distance, if the two images are similar, the result value is small, otherwise it is large. The loss function minimizes the distance if the two images are close and maximizes it if they are from different categories. The parameter w is computed with the stochastic gradient method, using the sum of the gradients contributed by the two branches.

The input images, after they are transformed into tensors, they are normalized with values for mean equal to 0.0188 and 0.128 for standard deviation. The epochs for training phase have been set to 10, with 128 batches of 64 image pairs per epoch and 0.01 value for the static learning rate. The only modifications to the code has been made with purpose to save model after each epoch and continue training from selected checkpoints.

Siamese Training Data Set includes 50,000 pairs from the corresponding edge maps of camera poses created with the camera pose generator. Camera poses are selected randomly from the sampled camera poses data set. The edge maps are created from a pose and a sampled one from the initial pose with the pinhole camera model and they have 320 x 180 resolution.

Camera Pose Retrievement

The selection of the suitable camera pose and the estimation of its homography need at first the extraction of the edge image features. With the use of siamese network, the edge map turns into a features vector. Then, the output of the model is matched with the nearest vector from the database that contains camera poses and their respective features. The search within database to find most similar features occurs with the Fast Library for Approximate Nearest Neighbors (FLANN) [60]. It is a library that contains a collection of optimized algorithms for performing fast and efficient searches in large datasets and for high dimensional features.

Therefore, after the search, the most similar camera pose is chosen. Now, with the estimated camera pose and the pinhole camera model, the respective homography and edge image can be produced. The visible field of an input image with detected its edge features and with a known homography can easily be registered on a top view model, as it is illustrated in Figure 6.

However, the retrieved homography and edge map from a camera pose are not enough accurate some times and this is due to the fact that the estimated camera pose differs greatly from the initial frame's point of view. A main reason for the low similarity is the number of available poses stored in the database and the fact that they come from football matches with specific camera parameters. This has as a result to not be able to produce efficient homographies from match images of different football leagues.

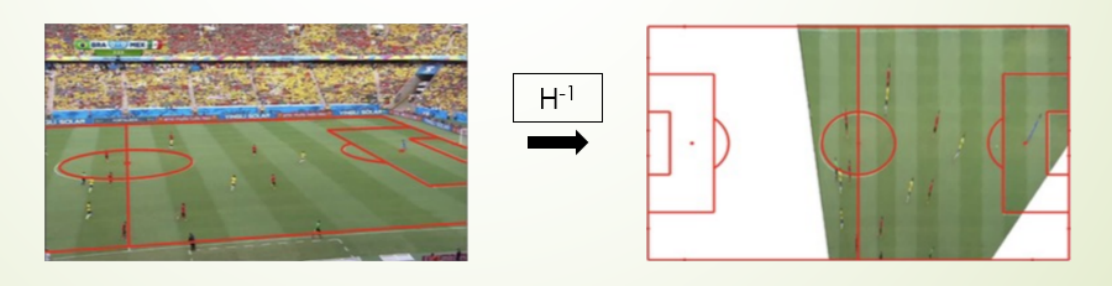


Figure 6: Transformation of football image points on a static top view field with the inverted homography

Features-Camera Poses Database consists of camera poses and their corresponding features vectors. Each of 90,000 poses from sampled camera poses data set is included and is used for its edge features extraction. The pinhole camera model is utilized to create the respective edge maps and each one of them is inserted in the siamese network. The result of the model is a features vector that is stored in the database along with its camera pose. This database is used mainly for a better estimation of a camera pose from a detected edge map. The generation of the database is displayed in the figure 7.

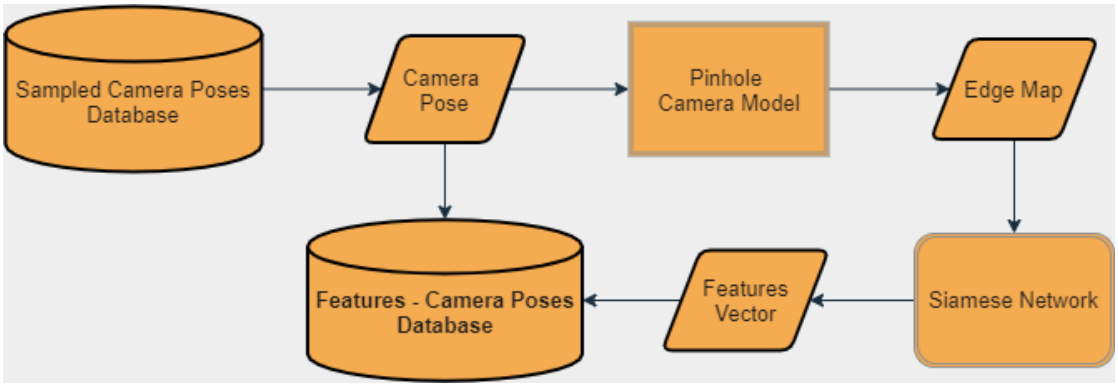


Figure 7: Creation Process of Features - Camera Poses Database

Sports Field Recognition

Before the estimation of camera pose and its related homography, the examined football frame must be processed for the detection of the edge lines from the visible football field. For this task, is implemented a trained two-GAN model that takes as input the broadcast image and returns an edge map. The frame is shaped appropriately before it is inserted in the model and exported as a fixed edge image.

The two-GAN model is composed of two conditional generative adversarial networks (GANs), which are based on the pix2pix network [58]. The first one is a segmentation GAN and its function is to divide the football field from the frame. Then, follows the

detection GAN to recognize the edge lines of the field and any other markings that define the playing area. Aim of this combination is to eliminate the irrelevant information outside the playing surface. Each GAN is correlated also with a discriminative network.

As illustrated in Figure 8, the segmentation GAN returns initially the mask of the field from input image. Then, the football field is applied on the mask, before it moves in the detection GAN. There, the field lines are detected, forming thus the playing area, which is returned as an edge map. For each generative network, a discriminator is utilized to detect its fake outputs.

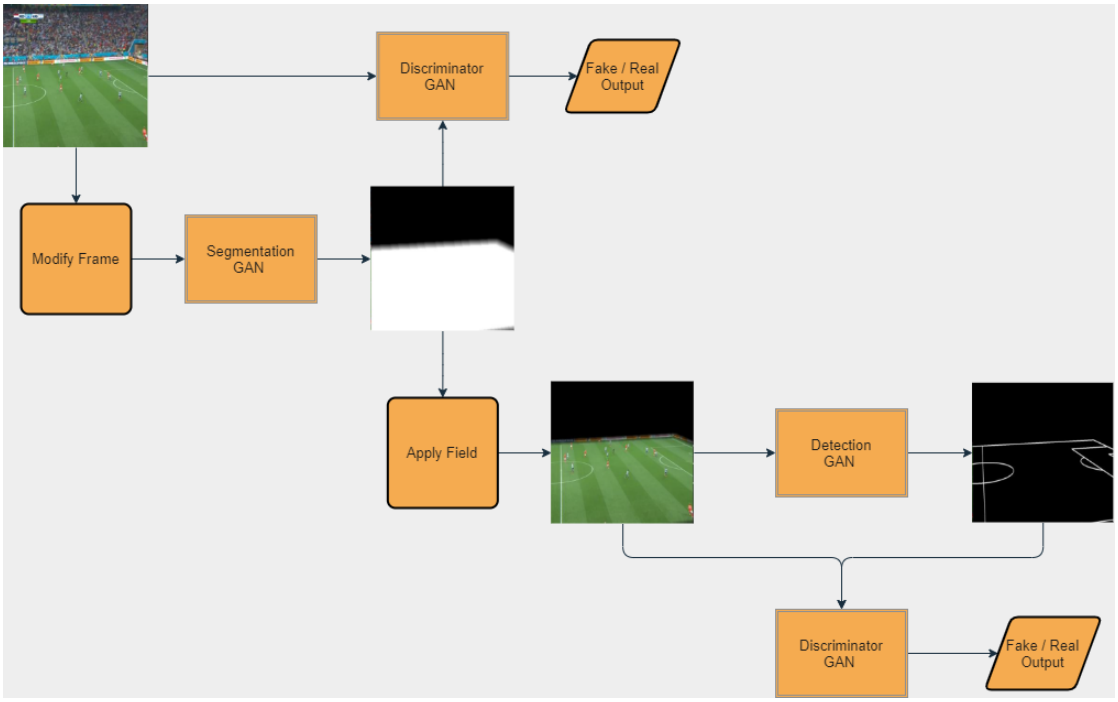


Figure 8: Edge Map Detection with Two-GAN Model

Pix2Pix Network [58] is defined as a conditional GAN, a generative model that learns a mapping from observed image x and random noise vector z to y, $G : x, z \rightarrow y$, while a simple GAN learns a mapping from random noise vector z to output image y, $G : z \rightarrow y$.

More specifically, the GAN structure consists of two components called Generator and Discriminator, or G and D respectively. Aim of the G is to produce new outputs which are as close to the input images, while the D is trained to separate the generated image from the real one. For better results, the conditional GAN adds a label to the corresponding inputs of the G and the D. In a cGAN, the G and D are conditioned during the training on contextual information, controlling this way the generated data [61].

Network Architecture is made up from the generator and discriminator architecture. For the first one, is used an encoder-decoder network, in which the input data have to pass through the entire series of layers until the layer with the fewest nodes, where the down-sample process of data is reversed. The input structure is not connected well with the output one, with result the exchange of poor information between them. This problem is addressed by adapting the general form of a U-Net and adding skip connections across to all n layers and specifically, between each layer k and layer $n - k$.

The discriminator only models high-frequency structure, depending on an L1 term to force low-frequency correctness. This architecture penalizes structure at the scale of patches, distinguishing each N x N patch in an image if it is real or fake. The average of all results is the final output of D. The specific D takes as granted that there is no connection between pixels separated by more than a patch diameter, modelling the image as a Markov random field.

Objective function of a conditional GAN is general defined as

$$L_{cGAN}(G, D) = E_{x,y}[logD(x, y)] + E_{x,z}[log(1 - D(x, G(x, z))], \tag{4}$$

where x is the observed image, z a random noise vector and y the output image. The G tries to minimize this function against an adversarial D that tries to maximize it. Combining the objective with a loss such as L1 distance

$$L_{L1}(G) = E_{x,y,z}[||y - G(x, z)||_1], \tag{5}$$

results in the final objective

$$G* = arg \min_{\mathbf{G}} \max_{\mathbf{D}} L_{cGAN}(G, D) + \lambda L_{L1}(G), \tag{6}$$

Training Phase of the two-GAN model is implemented from scratch. The World Cup dataset published in [15] is used to train the networks. After the training the resulted checkpoints are saved for further experimentation. Each network is trained individually, without to be merged via joint training, as in the original paper.

Each pix2pix model takes as input a pair of images displaying the initial frame and the target, which are resized from 286 x 286 to 256 x 256, before to be normalized and loaded as CUDA tensors in the GPU memory. Values of input tensors vary in range [-0.1, 0.1]. Training epochs are set to 200 with batch size equal to 1. Learning rate is set to 2 $\times$ 10-4 and decays linearly to zero after 100th epoch.

Homography Refinement Process

An edge map along with its homography that have been retrieved from the features-pose database it is possible to differ significantly from the pose of the original frame. Consequently, a procedure is applied to improve the homography matrix for the warping of the initial field to the final top view model.

The detected edge map and the retrieved one are used as input to the homography refinement process. The principal method utilized for optimization according to code[1] of [4] is the Enhanced Correlation Coefficient Maximization (ECC) method, although in the original paper is referred the Lucas-Kanade algorithm as the selected technique.

The two images are firstly transformed into two distance images because the image gradients is likely to be sparse. The distance image computes the distance from each pixel to the nearest zero pixel. Then, both images are fed into the ECC method that is running iteratively to achieve an accurate alignment and produce a more efficient homography. Eventually, the resultant matrix is applied on the retrieved one and a refined homography is created as an output of the refinement process. The final alignment of the two images is highly affected from the retrieved edge image and therefore from the variety of camera poses in the database. The Figure 9 displays the improvement of homography from the optimization of the edge lines detection in the image.

Figure 9: Refinement Process: Detected - Retrieved - Refined Edge Image

Distance Transformation of binary images is a process to enhance the ECC method to produce a better result. A binary edge map is used to locate certain features in an image. It is proven to be more beneficial to specify a cost for a feature at each pixel of an image instead of binary values, so the value for every pixel is replaced by its distance to the nearest zero pixel. For the calculation is used the Euclidean distance and the selected threshold is set to 15.

Enhanced Correlation Coefficient (ECC) method [19] is an image alignment algorithm based on the l2 norm. This method estimates the geometric transformation, or warp, between the input and template frame and returns the warped input frame, which must be close to the template. The correlation coefficient between the template and the warped input frame is maximized by the estimated transformation. Practically, it tries to maximize the Enhanced Correlation Coefficient function, a metric that performs also well against noisy and photometric distortions. Parameters of the ECC method refer to the number of maximum iterations and the desired accuracy by the algorithm to converge. Iterations are set to 50 and accuracy value to 1×10^{-3}.

[1] `https://github.com/lood339/SCCvSD`

3.2.2 Alternative method

A different approach to estimate the homography matrix is examined to be integrated within procedure and is evaluated. The method in [5] that is implemented, tries to minimize the difference between the warped football field template and the broadcast frame. It proposes an optimization process for homography estimation through the reduce of registration error. The overall process is defined from two main components, the initial prediction of homography and the refinement of it.

The first phase includes a trained registration network that is utilized to predict the registered key points to the football field template from a broadcast frame. Then, with the detected key points, it is simple to define the parameters of the homography matrix with the direct linear transformation algorithm [62].

A refinement process comes after the initial prediction of homography with purpose to improve the accuracy of points registration. At first, the top view image of football field is warped based on the first homography estimation to resemble in the point of view of input frame. The resulted edge image from the template transformation is concatenated with the original image and the result is used as an input to a second model, a registration error network. Then, the registration error is predicted and the gradient is derived according to the homography parameters. This gradient is used to update the basic estimate, completing thus the optimization process and returning a final homography that registers much more efficiently the football field and consequently the players positions. The aforementioned process is executed repeatedly until convergence. In Figure 10, it is illustrated the entire process, from the initial estimate until the repetitive optimization.

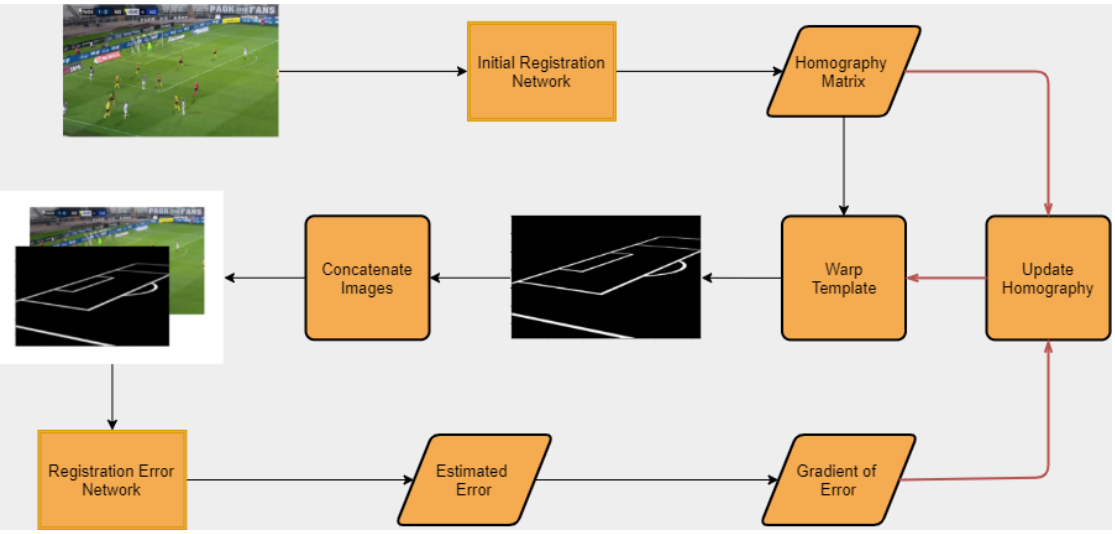


Figure 10: Process for Homography Estimation

Homography Prediction

In the beginning, the football field registration from an input frame is performed with the utilization of a deep neural network. The model provides a first prediction of the homography matrix with the regression of points from the input frame when they are registered on the field template.

This process basically relies on the 4-point parameterization [63] technique for homography estimation. In order to have a better understanding of the entire process, this method is further described.

The input football frame is being normalized to have width and height set to 1, while the centre of the frame is at the origin. The points on the original frame that define the corners of the lower parts of the image are the ones selected to be warped. The coordinates of these points, as they are looked in Figure counterclockwise, are (-0.5, 0.1), (-0.5, 0.5), (0.5, 0.5) and (0.5, 0.1) respectively. These particular points are chosen because the main camera of most football videos looks down the field from a specific angle.

The points from the original image have to be mapped on the top view image of the football field, as it is displayed in Figure 11. If the corresponding points from the input frame and the template are defined as $[\upsilon, \nu]$ and $[\acute{\upsilon}, \acute{\nu}]$, then the offset of the first corner point is $\Delta_{\upsilon_1} = \acute{\upsilon_1}$ - υ_1. The homography now with the 4-point parameterization can be described as

$$H_4points = \begin{pmatrix} \Delta_{\upsilon 1} & \Delta_{\nu 1} \\ \Delta_{\upsilon 2} & \Delta_{\nu 2} \\ \Delta_{\upsilon 3} & \Delta_{\nu 3} \\ \Delta_{\upsilon 4} & \Delta_{\nu 4} \end{pmatrix} \tag{7}$$

The final homography matrix that can warp the initial image to the template is easily computed, once the four points are predicted with the initial registration network. For the computation is used the Direct Linear Transform (DLT) algorithm [62]. DLT method is implemented to formulate the three dimensional points of an image in space using the two dimensional image. The result from this first phase is defined as

$$\mathbf{h} = f_\phi(I) \tag{8}$$

where f_ϕ is the initial registration network and I is the image, while parenthesis indicate the optimization iteration.

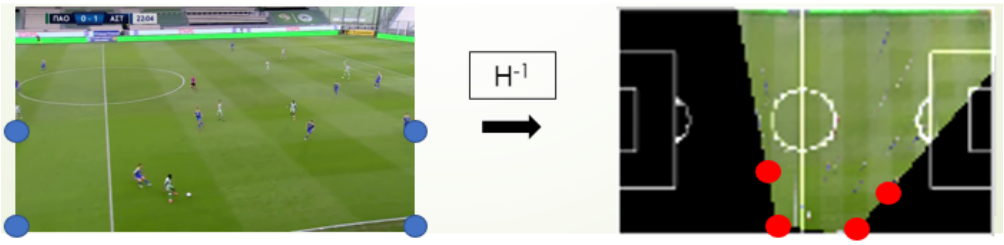


Figure 11: Points Transformation from Image to Template

First Stage Network is an initial registration network with main contribution to the procedure a first estimate for the registration of input frame onto template and consequently for the homography matrix. The model is based on a residual neural network that is 18 layers deep, called ResNet-18. Initial frame is input to this network where it is transformed into a multidimensional feature. The residual network is inspired from VGG nets and consists of convolutional layers which have 3 x 3 filters and the number of filters is the same as the output feature map size [64]. Downsampling is executed from convolutional layers with value for stride equal to 2. In the end, is implemented an average pooling operation and a fully connected layer with linear transformation for the estimation of the final eight numbers that define the homography matrix. The main characteristics that define this neural network as a residual are the skip connections between layers and the batch normalization, which are also help to train deep layers.

Training Phase has been already performed for the network with the fine-tune of the pre-trained weights that are trained on ImageNet [65]. The network has been trained individually. The training is accomplished from known poses and with their ground truth homography, the points are warped from the input frame to the field template and they are computed accurately. The registration network is trained to minimize the loss function

$$L_{init} = ||h_{gt} - \hat{h}^{(0)}||_2^2 = ||h_{gt} - f_\phi(I)||_2^2, \tag{9}$$

where h_{gt} is the ground truth homography and $\hat{h}^{(0)}$ is the first homography estimation, which the network is being trained to retrieve.

Training Dataset is based on the World Cup dataset and it is enlarged with various techniques. The network thus is trained with multiple images and learns to get over situations, in which the final result is affected negatively due to high variance or bias. Every sports frame is randomly cropped and its resulted size is reduced to a maximum of 10 percent. Moreover, it is applied horizontal flip augmentation, where the pixels rows and columns of an image are reversed horizontally, creating the mirror of this image in the opposite direction, like in Figure 12.

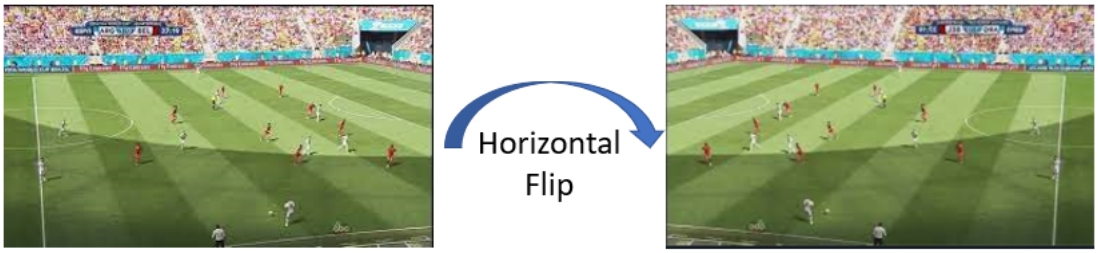


Figure 12: Horizontal Flip of Sports Image

Other technique used is the black transparency over the images by altering the pixels color values with the half opacity. These images are also rotated randomly either clock-

wise or counterclockwise by 0 to 45 degrees and they are translated along the half of x or y direction. The images are further resized either to the half or double width and become blurred with the use of Gaussian distribution on their edges.

Homography Optimization

The result from the first phase often needs an optimization process to reach really high standards in terms of points transformation. The second phase introduces a registration error network for the refinement of the homography parameters.

To begin with, the football field template is warped using the available homography to have a similar point of view as the initial frame. The spatial transformation of the template is executed with differentiable image sampling [66]. A bilinear sampler produces a new feature map from the homography and the input image, sampling from the input all the values identically for each channel with aim to maintain spatial coherence.

The produced edge image is concatenated with the original frame and the result is the input to the second network that is implemented to estimate the registration error. The model prediction is defined as

$$\hat{\epsilon}^{(i)} = g_{\psi}([I; W(m, \hat{h}^{(i)})]), \tag{10}$$

where $\hat{\epsilon}^{(i)}$ is the registration error, I is the input image, $W(m, \hat{h}^{(i)})$ is the transformation process of the template m according to the homography $\hat{h}^{(i)}$ and the symbol [;] indicates the merging of the two images along their channels. The registration error network is symbolized with g_{ψ}.

The refinement process is fulfilled with the gradient provided from the registration error and the use of it for the improvement of the homography parameters. This process is executed until it achieves convergence or reaches a limit for iterations. The update of weights that have derived with the error backpropagation, is applied with the use of Adaptive Moment Estimation (Adam) optimizer [67], in order to converge faster to solutions with better quality and because this optimizer is computationally less expensive than others.

Second Stage Network is a registration error network that takes as input the resulting image of six channels from the union of original image with the transformed template and returns the extent of loss for the initial estimation. Its structure is based on ResNet-18 [64], similar to the network of the first phase. Main difference for this model is that it uses spectral normalization to control the Lipschitz constant by constraining the spectral norm of each layer and normalizing the weight matrix. The final activation function of the second network is the sigmoid as the result of it is always positive and is used for the error metrics of the Intersection over Union (IoU).

Training Procedure for the error network has been executed from start in [5]. The dataset for training consists of images concatenated with football field templates that are transformed according to their ground truth homographies. For various images,

the known homography has been perturbed randomly to create additional inputs. The perturbation of a ground truth homography is accomplished with the addition of a global random translation $\alpha_g \sim U(-\delta_g, \delta_g)$, where $\alpha_g \in R^2$ and a local random translation $\alpha_l \sim U(-\delta_l, \delta_l)$, where $\alpha_l \in R^8$, to every point that determines the final homography. The registration error network is trained to minimize the loss function

$$L_{error} = ||Err(I, W(m, h_{pert})) - g_\psi([I; W(m, h_{pert})])||_2^2, \tag{11}$$

where Err is the error metric and $[I; W(m, h_{pert})]$ is the concatenated image as input to the model.

Training of the network is performed with the Adam optimizer [67] and parameters $\beta 1 = 0.9$ and $\beta 2 = 0.999$. The learning rate is set to 1 x 10^{-4}. The same parameters are applied also for the training of the initial registration network. However, the two networks used in the whole procedure are trained separately as their final outputs are quite different. The process occurs until convergence and is early stopped with the validation dataset. The parameters δ_g and δ_l for the homography perturbation are set to 0.05 and 0.02 respectively. During the inference process, the Adam optimizer is set again with a learning rate equal to 1 x 10^{-3}. The optimization process is set to run for 400 iterations.

The source code[2] implementation of this alternative framework [5] for homography prediction has proved to be efficient for various competitions in the area of football, as it is illustrated in Figure 13.

Figure 13: Homography Refinement Process - Edge Maps

[2] https://github.com/vcg-uvic/sportsfield_release

3.3 Team Identification

The known homography from the previous phase is able to transform points from the field of the original frame to the football field template, but in order to know exactly where the players are in the field, an object detection method is implemented to locate their specific positions. Moreover, after the recognition of players, it is necessary the classification of them in teams, so to become more clear their role during the game. The split of players into teams occurs with the extraction of the primary color from their jerseys, as the teams are obligated to wear uniforms with dissimilar color.

More specifically, an object detector named FootAndBall [3] retrieves bounding boxes of the players in the sports image and the coordinates of their centers are computed. The same method also detects the ball in the frame. The football frame is inserted in the model with a mask, so that only the field to be visible and to avoid any erroneous detections outside the playing surface.

After the players frames are acquired along with their positions, the dominant color feature is extracted from their uniforms, so it is very likely the players from the same team to have close enough feature values. Before the classification of these features in two groups, an Artificial Neural Network (ANN) is generated to predict the color for each one of these features. This model enhances the process for the recognition of a referee or a goalkeeper among the detected players and their subtraction. If the object detection and their primary colors extraction is not completely accurate, the identification of any referees or goalkeeper may fail and then, will be essential the main color of their jerseys to be given as input from the user.

The remaining color features are finally classified in two groups and the labels represent the two teams. The corresponding colors predictions of the features help to find from the first frame that is processed, the top color for each label and identify the primary colors of home and away team. With the knowledge of these colors, it is more simple to match the extracted colors with home and away team in the next frames, even when there is a mislabelling for few players. The results from every color prediction helps to refine the classification and assign the correct label to every player. The whole procedure for team identification is illustrated in Figure 14.

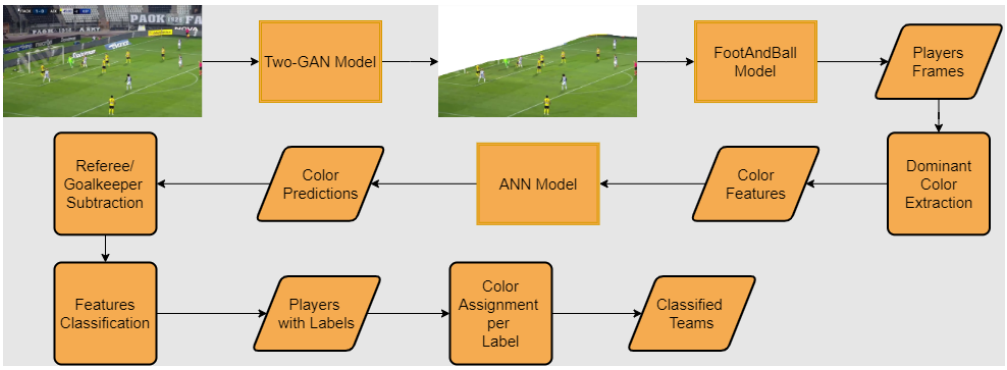


Figure 14: Team Identification Process

3.3.1 Player Detection

The obtainment of players positions in the input frame is relied solely on FootAndBall [3], an object detector based on a deep neural network. This model performs on a single sports frame and has about 199 thousands parameters, much less than other object detection networks. The method uses dense grids for player detection with a size 16 times smaller that the input image, while the size of dense grids for ball detection is cut down 4 times.

The input to the detector is the initial image with a mask, output of the Pix2Pix model, with purpose to cover all the potential figures for detection outside the football field. Outputs of the model are a ball confidence map, a player confidence map and a player bounding box tensor.

The ball confidence map projects how likely it is that there is a ball in each grid cell and the size of it is $w/4 \times h/4$, where the w and h are the width and height of the original image respectively. The location of the ball in the map is the one with the highest confidence and with the precondition that this is above a specified threshold.

The player confidence map displays the probability for every grid cell to represent a player's form and the size of it is $w/16 \times h/16$. The player bounding box tensor locates the coordinates of the potential player's frame in the player confidence map and its size is $w/16 \times h/16$ x 4.

The application of non maximum suppression to the player confidence map locates a player's point (i, j) in the map according to a specific threshold and consequently, the bounding box coordinates (x_{bbox}, y_{bbox}, w_{bbox}, h_{bbox}) are acquired from the bounding box tensor, before they are normalized. The first two values is the center of the bounding box in the confidence map cell, while the other two are its width and height. The coordinates of the bounding box center in the input image are $(\hat{x}_{bbox}, \hat{y}_{bbox}) = (16(i - 0.5) - x_{bbox}w, 16(j - 0.5) - y_{bbox}h)$ and its height and width are presented as ($w_{bbox}w$, $h_{bbox}h$). These values practically locate the players frames in the image and their coordinates that are computed from their max height and their mid width, are the ones wrapped in the field template.

Network Architecture

The FootAndBall network is defined from a top-down architecture that creates multidimensional feature maps at all levels. It is inspired from the Feature Pyramid Network [33] that combines features from all convolutional layers. The first convolutional layers are required to get a specific spatial location of the player, while the next ones increase the classification accuracy with the extra content provided from their wider receptive fields.

Firstly, it consists of five adjacent convolutional blocks and each one of them outputs a feature map with shrinking spatial resolution and increasing number of channels. The produced feature maps are processed from the top to the bottom and they are added with the maps from the lower levels. The addition of these two maps requires for them to have the same number of dimensions and this is accomplished with the use of 1 x 1

convolution blocks.

The final feature maps are used as inputs to three tasks, the player and ball classification and the player bounding box regression. The player classification process produces the player confidence map with size w/16 x h/16, that indicates the players positions, from the input feature map with spatial resolution w/16 x h/16. The same input is used to the player bounding box regression task, where it is produced a player bounding box tensor with size w/16 x h/16 x 4 that points out the player bounding box coordinates for each location in the player confidence map. A point in the player confidence map corresponds to a square in the original image with a side value equal to 16. The ball classification process outputs the ball confidence map with size w/4 x h/4, that displays the potential ball positions, from the input feature map with spatial resolution w/4 x h/4. A location in the ball confidence map is equal to a 4 x 4 square in the input image. The network architecture is illustrated in Figure 15.

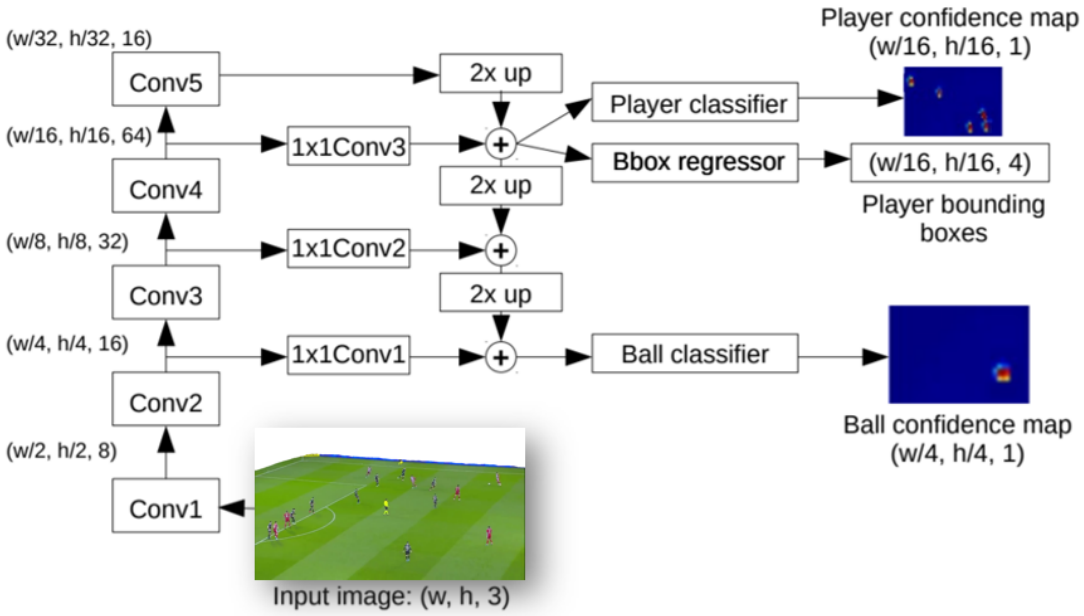


Figure 15: FootAndBall Architecture

Objective function

The deep neural network is trained to minimize a loss function which is based on the training objective function for the SSD detector [31]. The overall loss function is a weighted sum of three components, the ball classification loss, the player classification loss and the player bounding box loss and is defined as

$$L = \frac{1}{N}(\alpha_B L_B + \beta_P L_P + L_{bbox}), \tag{12}$$

where N is the number of training examples and $\alpha_B L_B$, $\beta_P L_P$ are the weights for the L_B and L_P respectively. If N is equal to 0 then the loss is set to 0 as well.
A deeper look at the components gives a better understanding for the loss function. The ball classification loss (L_B) and the player classification loss L_P are both defined from a binary cross-entropy loss over predicted object confidence and the ground truth:

$$L_K = -\sum_{(i,j)\in Pos^K} \log c_{ij}^K - \sum_{(i,j)\in Neg^K} \log(1 - c_{ij}^K), \tag{13}$$

where K symbol is substituted either by B for ball classifier or P for player classifier and c_{ij}^K is the value of the object confidence map at the spatial location (i, j). Pos^K represents the positive object examples, that is the set of locations in the object confidence map corresponding to the ground truth object position, while Neg^K describes the negative examples, that is the set of locations that does not correspond to any ground truth object position. The amount of negative examples is quite larger than the positive examples. In order to have a more balanced training set, a limited number of negative examples is chosen for both ball and players with the highest confidence loss, so for one positive example, there are three negative ones.

The player bounding box loss L_{bbox} is essentially the Smooth L1 loss [27] and computes the error between the predicted and ground truth bounding boxes:

$$L_{bbox} = \sum_{(i,j)\in Pos^P} smooth_{L1}(l_{(i,j)} - g_{(i,j)}), \tag{14}$$

where $l_{(i,j)} \in R^4$ describes the predicted bounding box coordinates in the location (i, j) and $g_{(i,j)} \in R^4$ are the ground truth bounding box coordinates in the location (i, j).

Training Phase

The training of the network has been conducted with a composite dataset. The 8 out of 10 from the whole sports frames are used as a training dataset, while the rest of them are used for inference. A pre-trained model is available at Github[3]. The resolution of one of the training data sets is Full-HD and the original model is trained both in HD and Full-HD frames, so in order to improve the detection results, the resolution of the whole training data set is transformed to HD. The training set is also increased with various data augmentation techniques, such as random affine transformation or changing their horizontal direction. Random cropping was eventually removed from the selected techniques. Image distortions were also conducted by changing the brightness, saturation, contrast and hue with random values. Training of the network is performed with the Adam optimizer [67]. The learning rate is set to 1 x 10^{-3} and the training epochs are set to 20 with a batch size equal to 12.

[3] `https://github.com/jac99/FootAndBall`

Training Dataset used for the training of FootAndBall detector is composed from two other ones, the ISSIA-CNR Soccer and Soccer Player Detection dataset from the corresponding papers [68] and [69].

ISSIA-CNR Soccer data set consists of six image sequences from football videos, which are broadcasted matches of the Italian top football league, Serie A, in the Friuli Stadium. The videos are filmed with six DALSA Pantera SA 2M30 cameras at 25 fps with a Full HD resolution (1920 x 1080 pixels). Every sequence lasts about 60 seconds and for each one of them about 3000 frames are produced. All the frames are manually annotated with bounding boxes on every player and on the ball.

Soccer Player Detection dataset includes images from football video sequences that display major events during a match, like a shot or a goal. The videos are filmed at the same stadium and they are from two professional matches. The filming was conducted with three PTZ cameras at 30 fps with HD resolution (1280 x 720). The number of images is 2019 and they are related with 22,586 annotated player locations.

Figure 16 displays the application of FootAndBall detector in sports frames. The detected players are in bounding boxes and above them, it is indicated the detection confidence from the player classifier. The detected ball in the frame is in a red circle. Next, in Figure 17 are mentioned samples where the players are recognized with complete accuracy, even when they are close enough, while there are cases also where two players are occluded and they are detected as one.

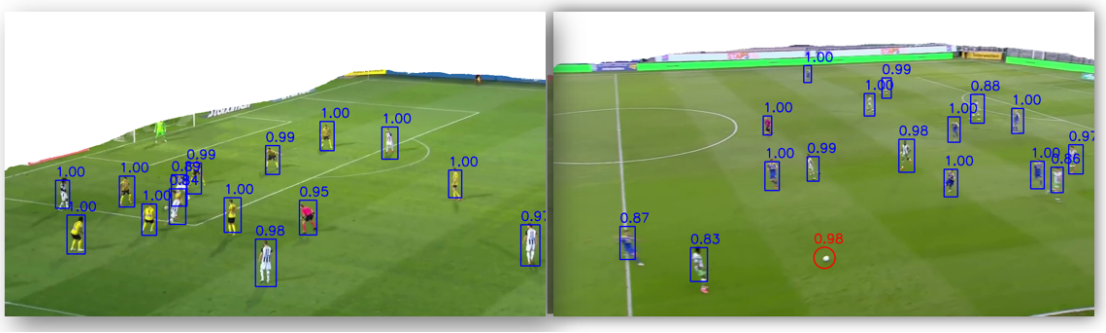


Figure 16: Players Detection with FootAndBall detector

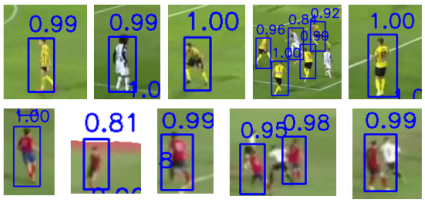


Figure 17: Samples from the Players Detection

3.3.2 Team Classification

The classification of players in two teams follows after their detection in the frame. The two groups are formed with a corresponding color as a label. The division is accomplished with the exploitation of the primary color from the players jerseys. It is common knowledge that the players from the same team wear identical uniforms and it is necessary the two teams and the referees to wear dissimilar colors during a game. The successful implementation of this procedure requires the object detection and the color features extraction to be perfectly executed, avoiding for the user to input any of the necessary colors.

Color Recognition

The predicted bounding boxes coordinates from the detection phase give the ability to separate every player in a new frame from the initial image. Every player's frame is being processed individually to retrieve the most dominant color in it.

More specifically, every frame is converted firstly to HSV color space. The next step is to identify the green color in each image and remove it, as in most bounding boxes it is included area of the grass field. The mask of green color is inverted and applied in the player's frame in order to keep only the major color features. Then, the image is reshaped to a list of pixels and those one that represent the black color are removed. After this process, the image is fitted into a K-means classifier with aim to find the two most dominant color features and their percentage in the image. The features of the image are clustered in three groups and their frequency is computed with the use of histograms. The produced histograms are normalized and they are further sorted according to their percent value along with the clusters centers. The number of clusters are three, because the main colors of a team are the most times two, with one of them to be the top color, as it occupies the most space in the team's jersey. The third cluster accumulates all these features that describe the players skin color or hair. The feature that is the center of the cluster with the highest frequency, describes the primary color in the player's jersey. The Figure 18 illustrates the process for the extraction of top color feature.

A significant issue is possible to emerge in this process from the green color mask. The suppression of green color might not be effective due to lighting or weather conditions and thus the remaining green pixels lead to wrong predictions about the dominant color. Further, when the team wears green jerseys, then the remaining features may describe the players skin and hair or the secondary color.

The above process is implemented almost the same for a second case, where the only difference is that every player's frame is converted in the start to RGB color space. In this way, this process can be examined in two different cases, so to assess which color space is the most suitable for the primary colors extraction and subsequently, for their classification.

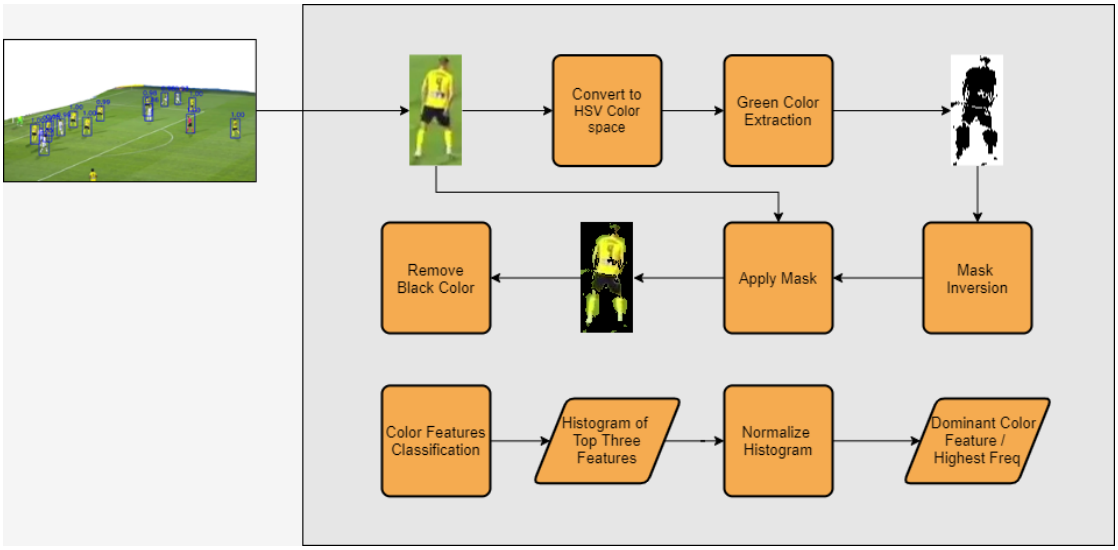


Figure 18: Color Recognition in Players Jerseys

Color Prediction

Prior to the classification of top color features in two groups, it is necessary to create a model that can predict the color for every extracted feature. This model is an HSV color classifier that predicts the color category for a feature based on the HSV. The basic color classes for the classification of features are Red, Green, Blue, Yellow, Orange, Purple, Gray, Black, and White.

The HSV color space is composed from three components, Hue, Saturation and Value. Hue is the color portion of the space and describes the class where a feature belongs. Saturation ranges from 0 to 100 percent and expresses the amount of gray in a specific color. Value describes the brightness of the color and ranges also from 0 to 100 percent, where 0 is totally the black color, while 100 is the brightest version of the color category. The features of every image are converted to HSV with the use of OpenCV, so the components are described in slightly different ranges. Saturation value varies from 0-255, Value varies from 0-255 and Hue value varies from 0-179. Figure 19 illustrates the ranges for every color in OpenCV and how the components are associated.

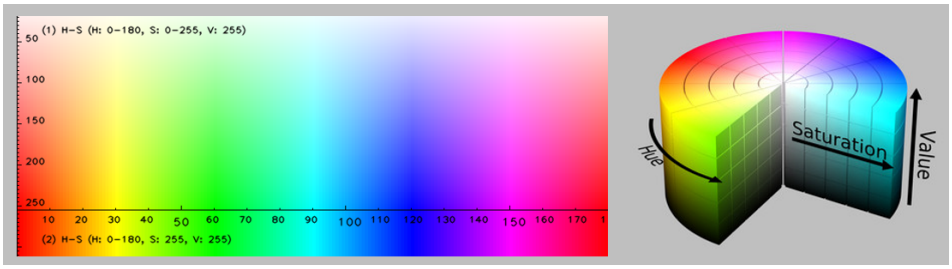


Figure 19: HSV Color Space

HSV Color Classifier is an Artificial Neural Network (ANN) and consists of 5 layers. Every layer is composed from a set of neurons and each neuron is connected with all the neurons of the next layer. The first layer takes as input the HSV color features. All the layers in the model use as an activation function the Rectified Linear Unit (ReLU), which helps a model capture interactions and enhances some level of sparsity in the network. L2 Regularization is also used to prevent excessive fluctuation of the coefficients that will cause overfitting. As a loss function, the Cross-entropy loss is selected to compute the loss between the predicted class and the actual class probability. The Adaptive Moment Estimation (Adam) optimizer is selected to apply the update of weights that have derived with the error backpropagation.

Training Dataset of the ANN model is created from scratch. For every one of the color classes, 1000 color features are generated and they are stored along with their class. Then, for each categorical label, a binary column is created with the use of the one-hot encoding technique and the column with value equal to 1 represents the label of each feature. The ranges of the colors are presented in the Figure 20 along with their encoded labels. The color ranges are inspired from the HSV colormap in Figure 19.

```
{'BLACK': [[180, 255, 75], [0, 0, 0]],
 'BLUE': [[132, 255, 200], [85, 70, 90]],
 'GRAY': [[180, 20, 180], [0, 0, 60]],
 'GREEN': [[80, 255, 180], [35, 50, 90]],
 'ORANGE': [[21, 255, 255], [12, 60, 100]],
 'PURPLE': [[171, 255, 255], [140, 70, 100]],
 'RED1': [[180, 255, 255], [173, 70, 100]],
 'RED2': [[10, 255, 255], [0, 60, 100]],
 'WHITE': [[180, 40, 255], [0, 0, 200]],
 'YELLOW': [[33, 255, 255], [24, 50, 100]]}
```

```
{'BLACK': 0,
 'BLUE': 1,
 'GRAY': 2,
 'GREEN': 3,
 'ORANGE': 4,
 'PURPLE': 5,
 'RED': 6,
 'WHITE': 7,
 'YELLOW': 8}
```

Figure 20: HSV Color Ranges and Labels

Training Phase uses the 80% fraction of the dataset for the training of the model. The other set is going to be used as a test set for the evaluation process. Moreover, the 20% of the training set is separated as a validation dataset and is used to evaluate the model performance during training. The learning rate is set to 1 x 10^{-3} and the training epochs are set to 1000 with a batch size equal to 50.

Referee/Goalkeeper Subtraction

With the use of the HSV color classifier it is now more clear if any of the features belongs to an object unrelated to the players, like the referees, or to a player that wears a special uniform, like the goalkeeper.

The referees are obligated according to their league rules to wear uniforms of specific color that is quite different to the colors of teams. Therefore, if any of the predicted colors from the features does not match with the teams colors, it is very likely to represent the referees main color, so the corresponding object is subtracted. The same process takes place also for the removal of the goalkeeper, who wears a uniform with different type of color than the other players.

The successful implementation of this process requires the object detection and the main color recognition to be executed totally efficient, so to avoid any mislabelling or erroneous subtractions and further, not to be necessary for the user to input the main color of the referees or the goalkeepers.

Players Clustering

After the removal of unrelated objects, the remaining features are fitted into a K-means classifier to be partitioned in two clusters. The labels are predicted and along with the features values are used to predict the primary color for home and away team and correspond them to the correct label.

At the first frame, the most frequent color between the predicted ones of each label defines the main color for the corresponding team. The user then, is asked for each one of these two colors if it is the correct one for home or away team until to match them with both teams. If none of the colors represents the primary color of any of the teams, then the user is asked to stop the procedure or to continue with these random colors.

For every next frame, each label is examined to find the top color for it and then, it is tested if these colors match with the main colors of both teams. If the identified colors are both erroneous, then the user is asked again to stop the procedure or to continue for this frame with the main colors matched to random labels.

The above process is quite helpful as it can match the correct color to each label, even when the color recognition and the classification of players is not completely accurate. This happens because it can find this label that represents certainly one of the teams, so the other label is matched with the second team.

After the classification and the recognition of teams, a sub process takes place to refine the labels. The results from the color prediction by the classifier are used for every player to check if the player belongs to the correct team or to change its label.

3.4 Team Formation Recognition

The final stage of the procedure encompasses all the valuable information from the two former phases to achieve the projection of both teams formations on a top-down view of the football field. Furthermore, the available knowledge gives the ability to find the areas on the football field which a team covers over the entire match and especially, these areas where a team has the control of the ball. It is accomplished thus the primary target, a tool for the coaching staff to have a more general look for the players movements of their team during a game and more significantly, in relation to the opponent's team behavior.

The required data for the visualization of players positions on the template is firstly, an accurate homography estimate for the transformation of original image points and then, the players labels that recognizes them as members of a team. Moreover, the localization of the ball in the frame defines the team that has its control. More specifically, the Euclidean distance between the ball and every player is measured. If a player is within a radius around the ball then, this player has the possession of the ball. If a player from the opponent team is within this radius also, then the phase of the ball possession is considered "Challenging".

Besides the state of the ball possession, it is useful to know if the team that controls the ball is attacking or defending. This information can be found after the distinction of the team sides. It is commonly known that the players of a team are closer to their goalpost than the opponents during the most time in a match, so it is measured the distance of all the players from the left or right side and the team with the most players near this side is matched with the goalpost. In order to choose the correct side from which the distance will be measured, it is tested if the minimum distance of a player from the left side is beyond the half pitch, so to choose the right side as the correct one. The player that has the control of the ball and is in the first quarter of the field from the opponents side, it is considered that is in a "Dangerous Attack", otherwise is just in phase "Attack". The combination of these information leads to present each team's players on the field and the ball possession phase as in Figure 21.

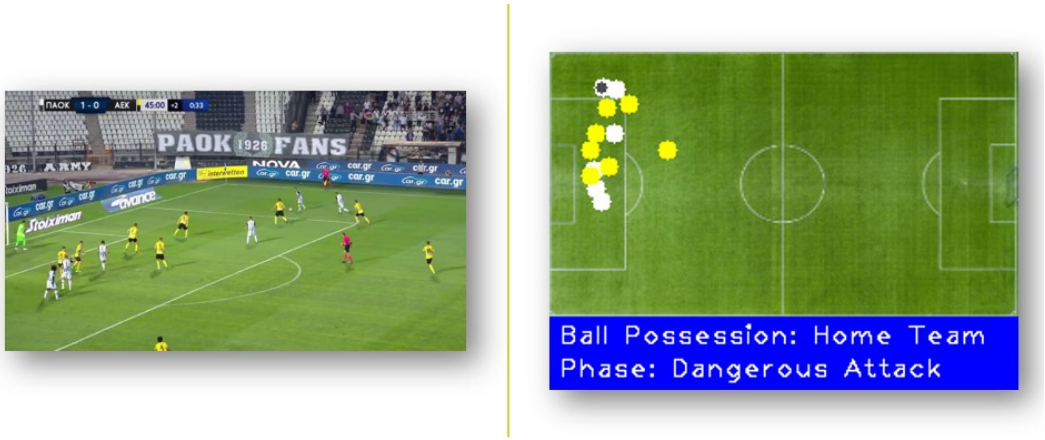


Figure 21: Players Positions and Ball Possession

The projection of players locations and the ball possession yields useful results over a sequence of frames for the escalation of an event in the match. There are times, however, when the coaching staff wants to observe the area that its team or the opponents rules. This raises the need for a more obvious representation of a team's controlling regions during the match. The proposed solution[4] is the construction of a heatmap with size 15 x 15 for every pair of coordinates and the blurring of it with the use of Gaussian filter. As regards the heatmaps colors, for the home team it is selected a brighten color for the heatmap, while the away team is described with a more faded one. The benefits of this process are better displayed in the Figure 22.

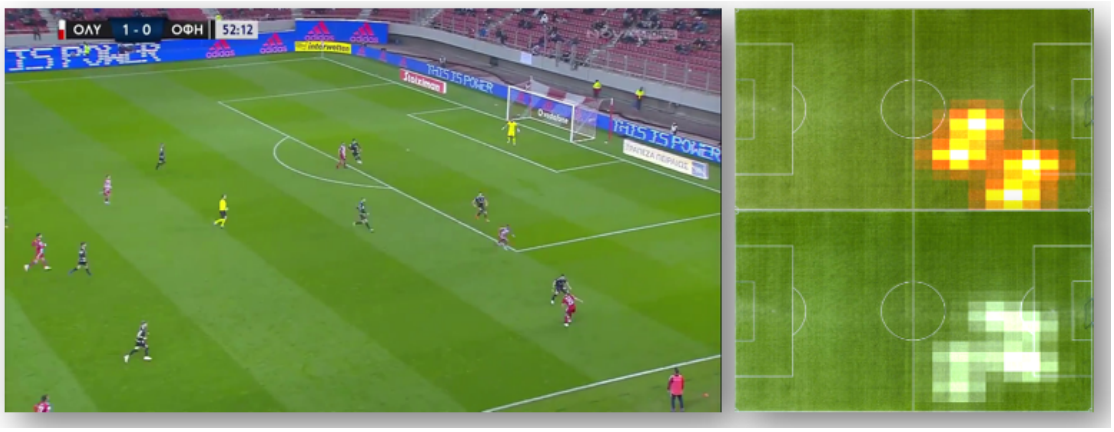


Figure 22: Sequence of Frames and their Respective Heatmaps

A video of a football match can now be transformed to a video that shows the players and states the ball possession or to a video that projects with heatmaps the regions that a team covers and render a different perspective for a team's progress over the game.

The generated data from the whole procedure are exploited to find for each team the coordinates of the players that have the control of the ball over the match. The primary aim is to project on the field the regions where a team has the ball possession and define also the percentage for every region.

First of all, it is essential to gather per team all the coordinates of players that have the possession of the ball. With this data acquired, the regions on the field template must be defined. A grid is applied on the static top view of the field and splits it in identical cells. Each axis of a cell is computed from the division of the template's axis size with the number of selected cells for this axis. For instance, from a grid with size 10 x 6 and a template with size 256 x 144, the respective cell's dimensions are 25.6 x 24. The value of each dimension is increased by 1 x 10^{-4} to avoid to match a point with a cell outside the field, especially in the occasion when this point is the maximum one of the template.

Every pair of the stored coordinates are distributed in the produced cells according to their values, where actually it is counted the percentage of pairs that are corresponding

[4] `https://github.com/TomDecroos/matplotsoccer`

in this cell. A heatmap of the grid is projected eventually and presents the volume of each cell, namely the regions on the football field where a team has the ball possession over the match.

The coordinates of a team are transformed during the time its players attack on the left side, so all the coordinates to be placed with direction to the right. After the completion of the procedure, some refinement steps for the possessions are implemented. One of them is the change of the label between two possessions of the same team where there are a few with the label "Challenging", the assignment of them also in this team and the measurement of their corresponding ball coordinates. Another refinement step for the possessions is the change of a possession label, if it is between two possessions of the same team and it is only one. In Figure 23, it is displayed how the ball possession heatmaps of two teams will look like at the end of the process. Next to the right side of the heatmap there is a colorbar that indicates the percentage of data in each cell.

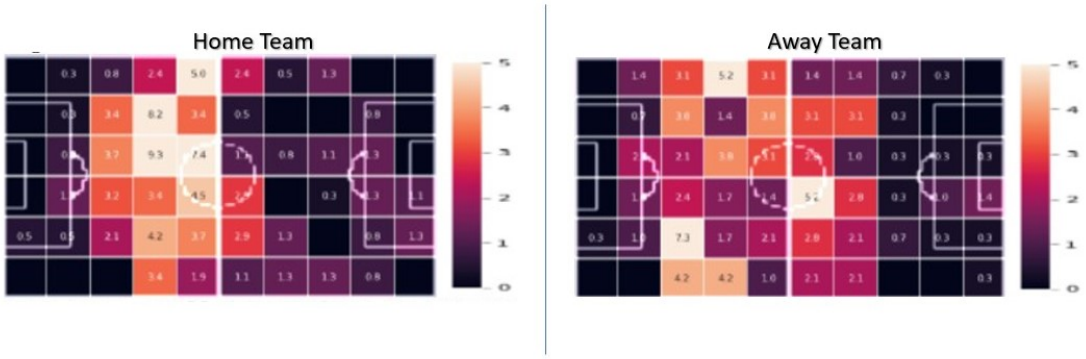


Figure 23: Heatmaps of Ball Possession for Home and Away Team

The results of the ball possession are composed of the percentages of the overall time that each team has the control of the ball and the percentage when the phase of the possession is "Challenging". Moreover, for the time when both of the teams have the control of the ball, it is measured the percentage of ball possession for each one of them to have a more clear look about this statistic.

In addition, the derived data are used to get the most occupied areas on the field for a team and show where it was dominant during the whole match. In the same way as before, a heatmap for each team presents the regions on the football field where the players are detected the most over the match with the coordinates to be placed with direction to the right. The heatmaps for both teams are presented in Figure 24, providing useful insights for the interaction between teams.

The resulted heatmaps from the process of a football video provide significant help to both staff and players for their team's performance during a game. Their ability to present the areas where a team controls the ball along with the most highly occupied regions on the field demonstrates that they are a major key for every team to take advantage over their opponents. Finally, these tools may be useful for further annotations

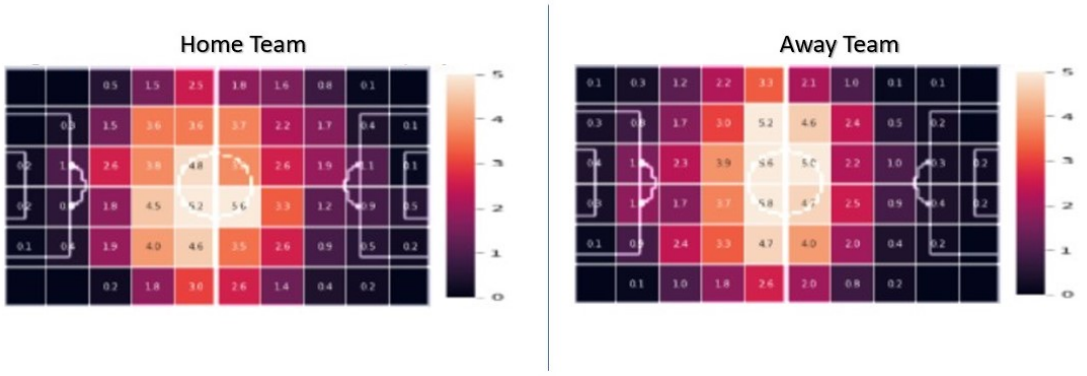


Figure 24: Heatmaps of the most Frequent Positions of Players over a Game

from the football performance analysts after a match or even in real time.

Formation Prediction

The regions on the field where the players of a team are most detected over a football match can be helpful also to predict the team formation and check if the players were keeping their positions during the game according to the rules of the coaching staff.

A model is created and trained to predict the formation of a team from the distribution of the players coordinates on the field template. The classifier is trained to predict one of the following formations: '4-4-2', '4-4-3', '4-3-2-1', '3-5-2', '3-4-3', '5-3-2' and '5-4-1'. Figure 25 projects how the distribution of the coordinates will look like on the football field template for some of the formations. The cells with the highest percentage represent the main positions of the players according to the selected formation.

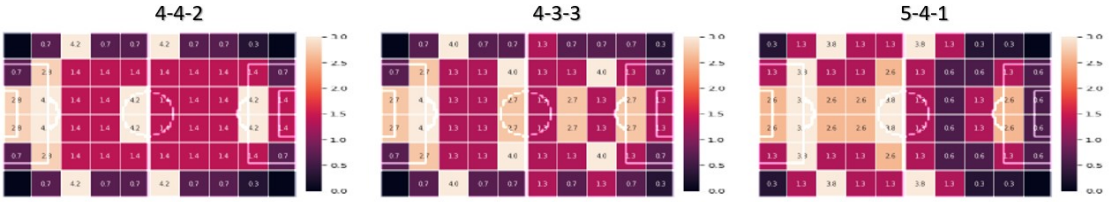


Figure 25: Team Formations

Formation Classifier is an Artificial Neural Network (ANN) with 5 layers, similar to the HSV Color Classifier. The first layer takes as input a vector of the cells values divided with the maximum value from all the cells. The ReLU function is used as an activation function, L2 Regularization tries to avoid overfitting and Adam optimizer is used to update the weights.

Training Dataset of the ANN model is created manually. The lower and upper ranges of every cell for a team formation are initialized and 1000 different cases for every team formation are generated. Every case is reshaped to look similar to a vector, before it is inserted in the model. With the use of one-hot encoding technique, every category is represented from a binary column. The train set is rescaled to 0-1 range before the training process.

Training Phase uses the 80% of the overall dataset for the training of the model and the remaining set is used as a test dataset for the evaluation process. After this split, the 20% of the training set is separated as a validation dataset and is used to evaluate the model performance during training. The learning rate is set to 1 x 10^{-3} and the training epochs are set only to 20 with a batch size equal to 30, because after the 20th epoch the model starts to overfitting.

Chapter 4

Results and Evaluation

The analysis of the procedure to map teams formation over a Greek football game has led to various tasks that needs to be examined. First, two main methods for the prediction of homography, introduced in [4] and [5] respectively, are compared based on the datasets described below. For the second task, namely the split of players into teams after their recognition, the accuracy of classification is measured.

4.1 Datasets

The comparison and the evaluation of different methods, especially for the case of homography estimation, occurs with the World Cup dataset and with sequences of video frames from Greek football games. The datasets from Greek matches have been created also to explore the whole process until the teams formation extraction and more important, for the evaluation of teams classification after players detection in frames.

4.1.1 World Cup Data Set

A set of football images with homography annotation published in [15] composes the Football World Cup dataset. It consists of 395 broadcast frames from 20 different football games of the World Cup in 2014 along with their ground truth fields and the grass segmentations. The training and validation sets are composed of 209 images from 10 games, while the test set includes 186 images from the other 10 games. The games were held in 9 stadiums in various times of the day, so their frames display different grass textures and lightning conditions and some times, their analysis is affected from wide shadows or rain. This dataset is used in [4] to train the Pix2Pix model and to generate the sampled camera poses dataset and consequently, the siamese training dataset. In [5] only 39 images are used for the validation dataset, while the test set remains the same. A diverse set of images from the World Cup dataset is illustrated in Figure 26.

4.1.2 Greek Football Data Sets

The videos from Greek football games were acquired through the platform YouTube and they represent different leagues. The frames used during the tasks display this way the field conditions and the quality of matches in the Greek professional and amateur football.

Figure 26: Frames from the World Cup Data Set

The selected videos are short clips from matches, when the camera shows only the football field and any commercial does not show up. The first three clips are from matches of the Super League 1, the highest professional association football league in Greece and the fourth one is from a match of the third tier of the Greek football league system. The fifth match is a game between national teams and the last one an amateur Greek football match. The football games are the following:

1. Panathinaikos F.C. - Asteras Tripolis F.C.
2. PAOK F.C. - AEK F.C.
3. Olympiacos F.C. - O.F.I. F.C.
4. Panionios F.C. - Charavgiakos F.C.
5. Greece - Montenegro (national)
6. Souli Paramithias F.C. - Naupaktiakos F.C.

The video from the first match is split into 24 frames and is chosen because the home team wears green jerseys, something that may proved to be demanding during the team classification task. From the second clip are extracted 31 images and from the third video 140 frames. Both of them are selected as one of these matches that represent the best possible conditions encountered in Greek football. The fourth game is from an amateur league and is preferred as the edge detection in it can be quite tough because of the field conditions and the broadcast resolution. From this game are extracted 17 images.

The last two matches are selected mainly to evaluate the ball possession and test the team formation classifier, as the videos of them are both quite larger than the others and can provide more remarkable results. From the fifth and the last match are extracted 828 and 833 frames respectively.

4.2 Evaluation Metrics

4.2.1 Intersection Over Union

Two metrics are taken into consideration for evaluation purposes. Both of them are types of Intersection Over Union (IOU) [20] and they are measuring the calibration accuracy of the methods implemented in the homography estimation task.

The first type is called IOU_{part} and computes how close is a produced mask of a field from the original one. The binary masks indicate the field area in a football frame and they are created separately with the use of predicted and ground truth homography. The second type is the IOU_{whole} and is computed similar to the former metric with a major difference that computes the masks of the whole sports field template.

The IOU_{whole} metric compares the whole field, even this that is not visible in the image, in contrast to the IOU_{part} that focuses on the area of the field that is shown in the image. In Figure 27, it is illustrated clearly the area of the field template, with red color, after the use of predicted homography and the ground truth template, pointing out that the result is not the ideal one as regards the invisible area. However, in the case of the invisible field, the registration error can not be computed as it predicts the projection correctness of the visible field and the ground truth homography is derived based on it.

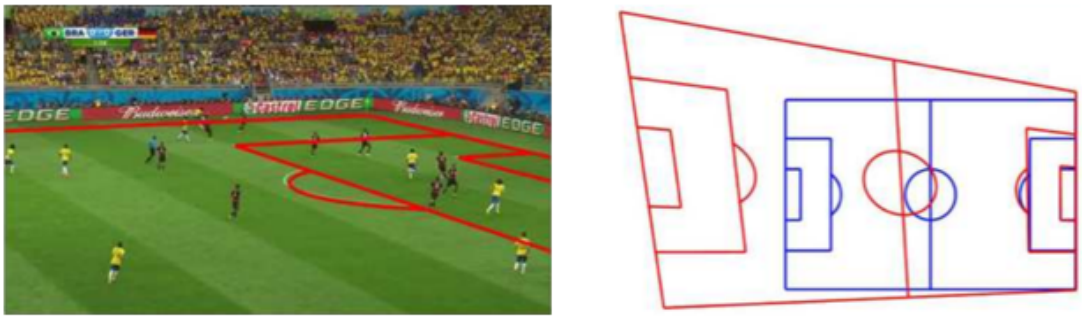


Figure 27: Differences between the Intersection over Union metrics

4.2.2 Completeness Score

The players classification in two teams is evaluated with the use of a metric described in [70], the completeness score. This score measures after a clustering result if all the data points of a given class are assigned to the same cluster. A perfect clustering sets all samples with the same true label to the same cluster. The completeness score is defined as

$$c = 1 - \frac{H(K|C)}{H(K)}, \tag{15}$$

where $H(K|C)$ is the conditional entropy of clusters given class and $H(K)$ is the entropy of clusters.

4.2.3 Confusion Matrix

The confusion matrix is used to evaluate the performance of a classifier. The diagonal elements show the amount of data for which the true label is equal to the predicted label and the higher these values are, the better for the classifier. All the other elements present the number of points that are wrong classified.

The most basic terms for a binary classifier as presented in Figure 28, are True positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

- TP: values which are actually positive and are predicted positive
- TN: values which are actually negative and are predicted negative
- FP: values which are actually negative and are predicted positive
- FN: values which are actually positive and are predicted negative

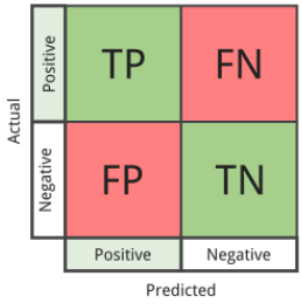


Figure 28: Confusion Matrix

Precision, Recall and F1-Score

From the above parameters, the metrics Precision, Recall and F1-Score can be calculated.

- Precision is the ratio of correctly predicted positive cases to the total predicted positive cases.

$$Precision = \frac{TP}{TP + FP}, \tag{16}$$

- Recall, also known as Sensitivity, is the ratio of correctly predicted positive cases to the whole actual positive cases.

$$Recall = \frac{TP}{TP + FN}, \tag{17}$$

- The F1-score is a combination of the precision and recall metrics and can be defined as the harmonic mean of a model's precision and recall.

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{18}$$

4.3 Homography Prediction

The state of the art methods in [4] and [5] are compared on the World Cup dataset and they are further modified to test their efficiency on the Greek football games. The methods are compared on mean and median IOU scores. It is measured also the average time for the process per frame and the overall time for all frames in seconds. All experiments are performed in Google Colab and the provided GPU is a Tesla P100-PCIE-16GB.

Table 1 shows the results on the World Cup dataset for mean and median of both IOU metrics. The results for the IOU_{whole} of approach [4] are taken from the relative paper.

Table 1: Methods Comparison on World Cup Dataset

Method	IOU_{part}		IOU_{whole}		Avg Time	Overall Time
	Mean	Median	Mean	Median		
[4]	94.7	96.4	89.4	93.8	0.96 sec	189.85 sec
[5]	95.2	96.9	90.0	92.6	1.93 sec	359.46 sec

From the results, it emerges the fact that both methods achieve very high performance, with the first method in the row to show slight better results for mean and median of IOU_{part} than in the respective paper. The first method also completes the process of each frame and consequently the overall procedure in less time than the second. However, it was noticed that the first framework starts to count the process per frame after the extraction of features vector from the siamese network. Specifically, the detected edge maps along with their features vectors had already been computed with result the time counter to start with the search in the features-pose database for the optimal camera pose. The second method was implemented with a batch size of 32 frames for the optimization process and each one of them achieved 6.50 optimization iterations per second.

As regards the Greek football videos, the same approaches are adjusted to process every frame from each game and predict their relevant homography. They are compared on each dataset created from the football videos and based on the mean and median scores of IOU_{part}. For the first case, the time of the procedure is computed in two ways, one starting after features vector extraction for each frame and a second one that counts the whole procedure.

The ground truth homographies have been computed manually between every frame and the football field template. During the computations for the first case, the resolution of a frame is 1280 x 720 and the resolution of the template is 115 x 74. The corresponding resolutions for the second case are 1050 x 680 and 256 x 256.

In Tables 2 and 3, the mean and median scores are presented for the first and second approach respectively. The numbers 1, 2 and 3 in column Match correspond to the video clips from matches Panathinaikos - Asteras Tripolis, Paok - Aek and Panionios -

Charavgiakos. In first table also, specifically in columns for Average Time and Overall Time, the numbers A and B correspond to the different ways for the measurement of time, with A to represent the initial way and B the case when the overall procedure is measured.

Table 2: First Method Results on Greek Football Datasets

Match	IOU_{part}		Avg Time		Overall Time	
	Mean	Median	A	B	A	B
1	34.2	26.9	1.87 sec	10.23 sec	45.06 sec	245.60 sec
2	32.6	39.1	1.33 sec	8.45 sec	41.51 sec	262.02 sec
3	25.4	30.3	1.33 sec	10.11 sec	22.63 sec	171.98 sec

The results in Table 2 and especially the mean and median scores of IOU_{part} are very poor, showing that the process failed to predict an accurate homography in most frames. For the second and third match the median is slightly higher than the mean, while for the first match the median is quite under the mean, indicating that most frames achieved really low scores and only in few of them it is predicted an adequate result for the homography. The average time per frame for both ways enhances the former assumption for the first match and along with the number of failed cases that was 7 out of 24, it is obvious that the process delayed due to completely wrong predictions. For the second match the failed cases were 8 out of 31 and for the third one were 5 out of 17.

From the comparison of different ways for measuring time, valuable insights can not be derived, except that the most time of processing a frame comes from the detection of edge map with the Pix2Pix model and the extraction of features vector with the siamese network.

Table 3: Second Method Results on Greek Football Datasets

Match	IOU_{part}		Avg Time	Overall Time
	Mean	Median		
1	92.1	92.2	14.66 sec	351.88 sec
2	90.1	89.1	14.71 sec	456.31 sec
3	82.0	82.4	14.85 sec	252.47 sec

The results of mean and median metrics in Table 3 are showing a remarkable improvement. The predicted homographies from the two first matches have accomplished adequate results with the third match to have a bit worse scores, but they are still enough high. From these scores it is clear that the computed homographies can project the visible field of the initial frame on the template in a great manner.

The average time for the prediction of homography per frame is extremely high for all cases, indicating that this is a time consuming process. It is necessary to mention

that the optimization process implemented per frame and not with a batch and this is mainly the reason for these results in average time and for the optimization iterations per second which were around 27.5 for all frames. This hypothesis is strengthen with the results in [5], where for the optimization per frame the method achieved 41.76 iterations per second and average time equal to 9.58 seconds, while for a batch with 64 frames, it achieved 4.66 optimization iterations per second and in average 1.36 seconds per frame.

In summary, the integration and execution of both methods in this procedure with aim to find the best way to predict homography from Greek football frames has led to some helpful observations.

The second approach has produced clearly more accurate homographies than the first method and without the limit of failed cases during the refinement process. On the other hand, it takes much longer for the second method to process a frame and extract a homography. However, this duration can be reduced if the frames are processed in batches during optimization task.

The first two frames datasets, that correspond to football matches of top league in Greece, have slight differences on their scores. The third one though shows a noticeable decline in its scores and the reason may be that its frames are from a game of the third division in Greece, where the edge lines in the video are not even visible sometimes because of the poor field conditions. Figure 29 displays some of the produced edge maps from the corresponding homographies on the original frames. These results are from the second approach. Despite the fact that these frames correspond to high IOU scores, it seems that there is still enough room for improvement.

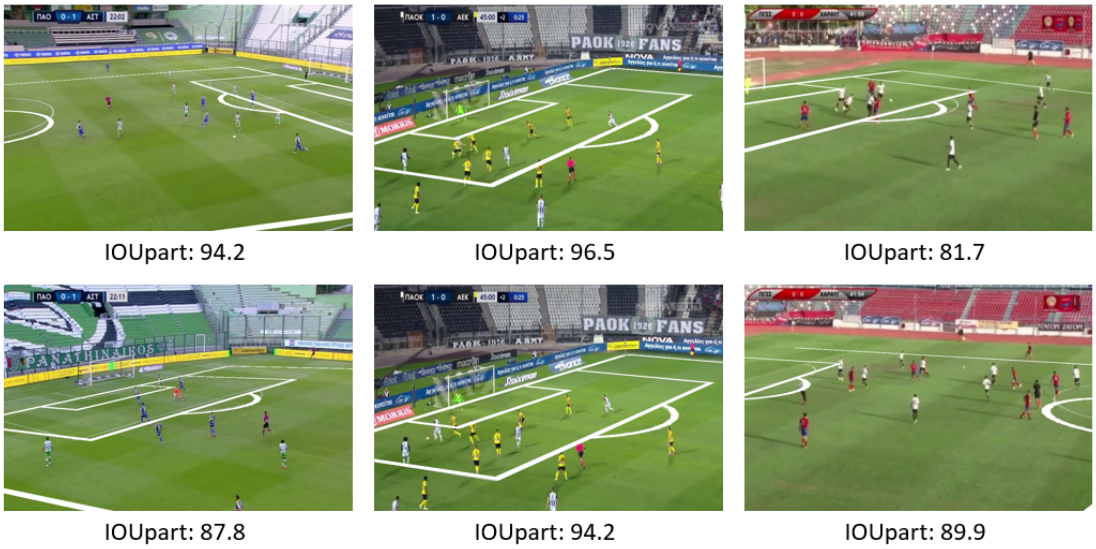


Figure 29: Edge Maps Results from the Second Approach

4.4 Team Recognition

The classification of the top color features from the detected players frames is evaluated on images from the Greek Football data sets. The completeness score is used to assess the results from the clustering process. The features are the remaining ones after the removal of referee or goalkeeper.

More precisely, from the processing of three random images per set and six images from the third set, the extracted color features are classified and their predicted labels are evaluated according to their ground truth. The values of the color features are extracted as RGB and HSV respectively, so to make the comparison between these two color spaces. Moreover, the color predictions of the HSV Classifier are used for the refinement of the classification of the HSV features and its contribution is evaluated. From the whole fifteen images, 179 players frames are detected and their corresponding features are used during the classification process.

In Table 4, the completeness score of the clustering results is computed per image and the average score per dataset is presented. In addition, the average time for each dataset is measured, counting from the color features extraction until the classification of them. The average time for the case that includes the HSV Classifier is not computed as the time for the refinement process is considered negligible. This process is executed for the features in RGB and HSV color space. The numbers 1, 2, 3 and 4 in column Match correspond to the Greek football data sets in the same order as mentioned in 4.1.2.

Table 4: Evaluation Results for Team Classification Method

Match	Avg Completeness Score			Avg Time	
	RGB	HSV	HSV+Classifier	RGB	HSV
1	0.42	0.44	0.88	0.26 sec	0.49 sec
2	0.45	0.88	1.0	0.42 sec	0.73 sec
3	0.27	0.30	0.89	0.27 sec	0.71 sec
4	0.63	0.70	0.80	0.31 sec	0.68 sec

The average completeness score in Table 4 reveals that the clustering with HSV values had better results than the RGB features for all datasets. The classification was not completely accurate most of the times and a main reason is that the quality of clustering is highly affected from the results of object detector. It was noticed that there were a few detected frames where the main player in them was a bit occluded from an other one, especially in the last two datasets, causing the recognition of a wrong color feature and consequently, the erroneous classification. This problem however, is tackled with the use of colors predictions results for every feature and its assignment to the correct label. The average time in the HSV case was a bit higher for all the datasets, but this is probably due to the use of the model for color prediction.

The scores for the first two datasets indicate that the clustering was performed much better with the HSV features. The green color mask is applied more efficient with the

HSV ranges than the fixed RGB color range, causing the elimination of any green values from the image. However, for the first dataset, where the home team has the green and white as primary colors, the main color is predicted to be the second one. The primary colors for the teams in second dataset are the white and yellow respectively, making the clustering with the HSV features of these two colors much more accurate than with the RGB features. The colors predictions in the first set enhances significantly the players labelling as it distinguishes some cases where it was extracted the yellow color as a main feature.

The third dataset presents very poor results for the first two cases, despite the very high quality of broadcasting. The main cause of this mislabelling is that many players were occluded or they were detected from long range. The same situation was noticed also at the fourth dataset, which is from a match of the third division and it had low broadcasting quality. The players in the images from both of these sets are classified quite properly after the refinement process.

As regards the subtraction of the referee and goalkeeper, the use of an ANN model for the prediction of their primary color was proved to be more rational than the fixed color ranges. The detection of the yellow color at the third dataset, where it was the main color for the referee and the one of two goalkeepers, was totally efficient and the detection and subtraction of the other goalkeeper with a uniform of blue color was also implemented successfully. Other images with a detected goalkeeper were noticed at the first dataset and the removal was conducted perfectly. However, the recognition of the referee in the first two datasets was not so clear, as in two cases the predicted color of the referee was purple and red respectively, instead of the purple, mainly because of their close hue values. Figure 30 presents some of the detected players frames. The ground truth and the predicted labels are mentioned under the frames.

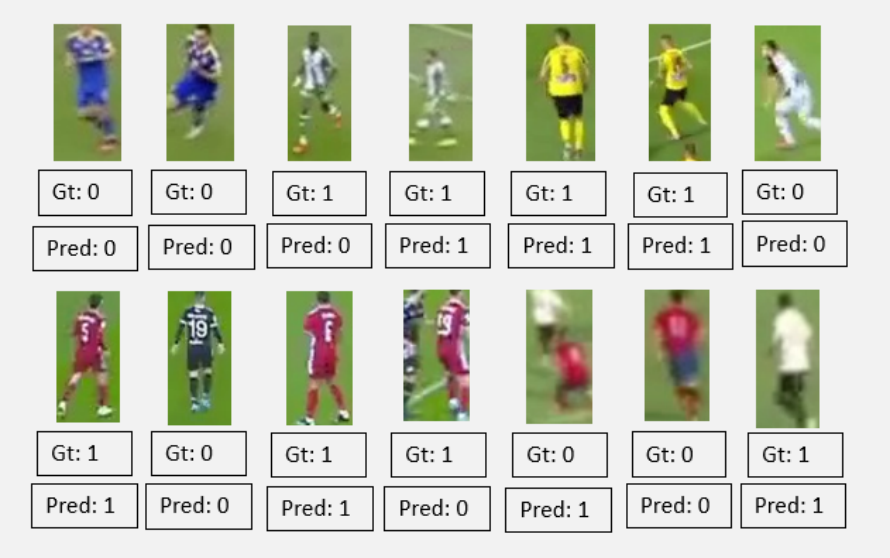
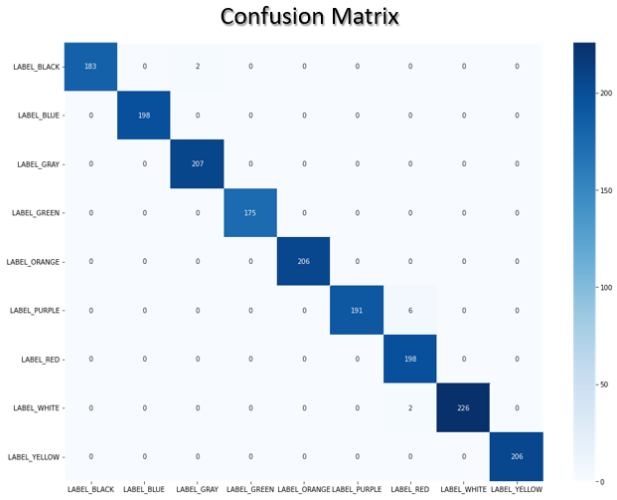


Figure 30: Players Frames and their Ground Truth and Predicted Labels

4.5 Color Prediction

The created ANN model for color prediction is evaluated on the test set, which is the 20% of the train set and was not used for the training of the model.

The model performs with 99.4% accuracy on the test set. A more clear look to the performance of the model gives the confusion matrix and the classification report in Figure 31. The results in the confusion matrix indicate that all of the predicted results were right, with only a negligible amount of cases to have been predicted wrong. From all these cases, only 6 had "PURPLE" as an actual label, but they were predicted as "RED" and only 2 were predicted as "GRAY", when they were actual "BLACK". These wrong predictions are also detected from the results of precision, recall and F1-score for labels "PURPLE" and "RED", that are slightly worse than the others, but they still present very high scores.



Classification Report

	precision	recall	f1-score	support
LABEL_BLACK	1.00	0.99	0.99	185
LABEL_BLUE	1.00	1.00	1.00	198
LABEL_GRAY	0.99	1.00	1.00	207
LABEL_GREEN	1.00	1.00	1.00	175
LABEL_ORANGE	1.00	1.00	1.00	206
LABEL_PURPLE	1.00	0.97	0.98	197
LABEL_RED	0.96	1.00	0.98	198
LABEL_WHITE	1.00	0.99	1.00	228
LABEL_YELLOW	1.00	1.00	1.00	206
accuracy			0.99	1800
macro avg	0.99	0.99	0.99	1800
weighted avg	0.99	0.99	0.99	1800

Figure 31: Confusion Matrix and Classification Report of the HSV Color Classifier

4.6 Formation Prediction

The evaluation of the team formation classifier occurs on the 20% of the initial dataset that has not been used in the training process.

The classifier performs with 92.8% accuracy on the test dataset. According to the confusion matrix and the metrics precision and recall in Figure 32, there are a few cases that were predicted wrong, mainly between the formations '3-4-3' and '3-5-2' or '5-3-2' and '5-4-1'. For example, from the whole 194 cases that have label '5-3-2', the 180 are correctly predicted, which corresponds to the 93%, as the result of the Recall metric. The 87% as a result of the Precision for the same label means that from the total 206 predicted cases as '5-3-2', the 180 were correctly predicted.

In general, an ANN model as a classifier for the prediction of the team formation is performing quite well. The erroneous cases indicate that the network can be trained better, in order for the model to recognize each formation perfectly and a more complex data set for the training process is a solution to refine the model.

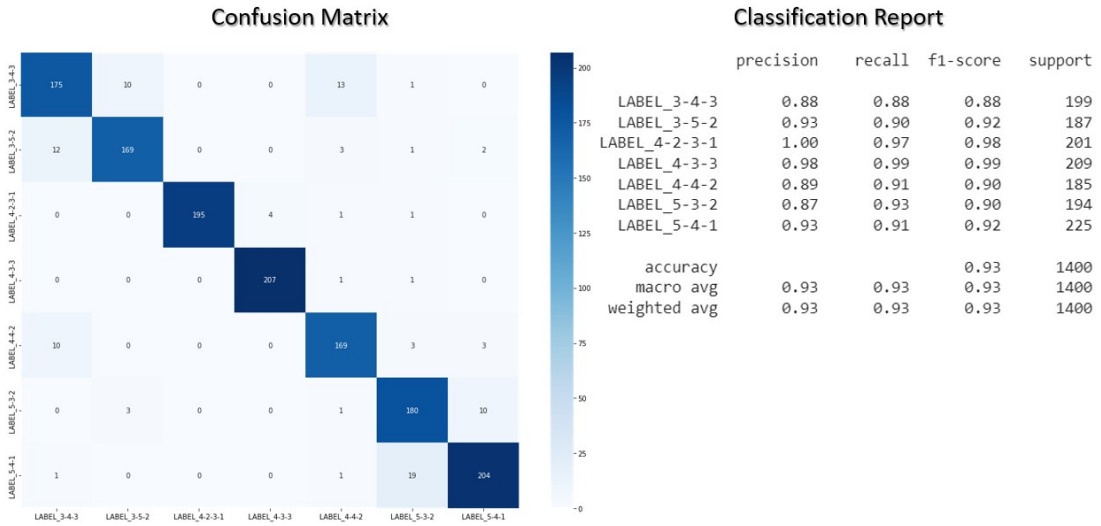


Figure 32: Confusion Matrix and Classification Report of the Team Formation Classifier

4.7 Ball Possession

An evaluation task also takes place to measure the accuracy of the ball possession prediction and examine if the refinement possession steps return more precise results. The difference between the ground truth results and the predicted ones shows how close are the results of ball possession for both teams and also for the "Challenging" phase. The closer the difference is to zero, the better for the computed results.

The ground truth results were manually measured. The ball possession calculation can be tough sometimes, especially when the players of both teams are losing the ball too often in a short time, so the decision to stop or start the time counter for a team is up to the expert who is measuring. Some assumptions that are made during the calculation of ground truth results are that when the ball is in the air is considering challenging and when it is on the possession of the goalkeeper then the phase is also challenging.

In Table 5, it is presented the difference for the three phases of the ball possession, before and after the refinement steps for the possessions. All the Greek football data sets are examined in the same order as mentioned in 4.1.2. Some cases in the data sets 2 and 4 have zero ground truth values, so the error is not calculated.

Table 5: Evaluation Results for the Ball Possession Task

Match	Ball Possession Ground Truth			Ball Possession			Ball Possession after Refinement		
	Home	Away	Challenging	Home	Away	Challenging	Home	Away	Challenging
1	7	3	2	-2	-1	+3	+0.5	-0.5	0
2	12.5	0	3	-3	+2	+1	0	+0.5	-0.5
3	101	11	28	-44	+15	+29	-6	+17	-11
4	7	0	1.5	-4	+0.5	+3.5	-1.5	+0.5	+1
5	401	298	129	-126	-97	+223	-22	-10	+32
6	163	466	204	-19	-315	+334	+109	-216	+107

The difference of the results of ball possession before and after the refinement steps indicate in general that there is a significant improvement.

The first two and the fourth data set are generated from short clips, so the results do not denote with absolute certainty that the refinement is efficient. However, for these three cases the computation of the ball possession has been improved, mainly by reducing the difference for the team that had the highest ball possession.

The other three data sets and especially the last two are from long clips and their results are more reliable. The fifth data set is from a football match of the first division, so the homography prediction and the object detection have implemented effectively. The fact that the prediction of ball possession has quite fine results for all the phases, especially after the refinement, shows that these steps have a good impact on the results. The last dataset is from an amateur football match and for the away team, which had the control for more of the half time, the results also improved.

Chapter 5

Conclusions

A procedure for the recognition of team formation from broadcast video frames of Greek football matches was examined in this thesis. The combination of a diverse set of tasks, with the two major ones to be the homography estimation and the players identification, led to the accomplishment of the final target that was to map the formation of both teams on a static top view of the field. All of the primary methods used in each task were described thoroughly.

For the homography estimation, two main frameworks from the papers [4] and [5] were implemented individually and they were evaluated based on the IoU_{part} metric. For evaluation purposes, the ground truth homographies were created manually, before they were used for each method to measure the accuracy of predictions on Greek football frames. These two methods were further compared with the use of metrics IoU_{part} and IoU_{whole} on the public World Cup data set [15], with the results to reveal that the second method performed slightly better. The framework from [5] also showed way finer results on Greek football data sets, overcoming the limit of the method in [4], where there were failed cases during homography refinement.

This method was selected to be integrated in the main procedure to predict an accurate homography for the transformation of points from the original image to the top view field template. However, it was noticed that homographies with high accuracy but not completely perfect can lead to slightly erroneous predictions about the true positions of players on the field. The Pix2Pix model [58] from the first method was integrated also, so to get the field from the initial frame with a mask.

The detection of players in every frame was achieved with the object detector FootAndBall [3]. This framework provided quite accurate results, but it showed that it needs further optimization, especially for cases where the players are close enough or they have fallen to the ground.There is an issue when players are occluded, because then the wrong feature is extracted and this ends in an erroneous classification. So, it is very important to get completely accurate results from the players detection phase. Furthermore, the detection of the ball performed quite low and needs definitely a refinement process to achieve better scores.

The selected technique for the recognition of players team was to extract the dominant color feature from their jerseys. The features were obtained as HSV and RGB values, with the first one to be the most reliable for the identification of both teams. An ANN model created from scratch helped to predict the color for every feature and subtract the referee and the goalkeeper when they were in the frame.

The remained features were classified in two groups and the classification was evaluated with the completeness score metric for HSV and RGB features. The colors predictions were also used for the refinement of the classification, providing much more accurate labels. The results indicate that the HSV color space is the finest solution for the expression of features values, as the green color mask and the prediction of primary colors for both teams performed certainly in a better way. Nevertheless, the HSV color ranges need to be defined more precisely and cover a wider number of colors, so to create a more composite dataset for the training of the model for color prediction. This will enhance the model to distinguish colors that are represented from values that are too close, like red and orange or purple and red. The differences on colors based on the lighting conditions is also a significant issue that needs to be examined. Moreover, when a team has as primary the green color, then it is likely the dominant color method to fail due to the fixed green color mask and get as main color an other one.

The derived data from all tasks were combined to map on the field template the teams formations for each frame and a new video was composed that showed the players movements as a team over the game. Furthermore, the players and ball positions were used to find the regions on the field where a team had the control of the ball and where the players were most detected over the entire match. The ball possession now can be computed for home and away team and some refinement steps have been introduced in order to improve the results. The created formation classifier uses the players positions to predict the team formation during the football match. These tools enhance the effort of teams staff to gain useful insights for the overall performance of their team in a football game or during specific events, such as a made shot or a goal.

Despite the fact that the overall procedure was finally executed, there were limitations in every phase that indicate the need for further refinement. More specifically, the results from homography prediction showed that the accuracy has to be nearly perfect, so an enlargement of the training data set with frames from different Greek football matches would make the network to adjust in different camera parameters and stadium conditions. In addition, it is necessary the completely accurate detection of the players and the ball in every frame, maybe with the use of videos from multiple cameras in a game or with the incorporation of one more object detector.

An efficient classification requires the correct color label for every player. The technique of extracting the dominant color feature seems helpful but it is not adequate, so the recognition of multiple colors in a jersey should be tested and also with features in different color spaces than RGB or HSV. The further refinement of the classifiers for color and team formation prediction needs also to be examined.

In summary, it has been proved that with the integration of these tasks in a united procedure, it is possible to map the formation of a team during a Greek football match. The provided data from this combination of methods can be used from Greek football teams to derive useful information about their performance. A future task can be the creation of a more complex training set from a huge number of football matches, in order to create a team formation classifier that can make more precise predictions on real data. The further refinement of the aforementioned challenges can be considered also as future work in the area of data analysis in Greek football.

Bibliography

[1] Deloitte. Annual review of football finance 2021, 2021.

[2] The Business Research Company. Sports global market report 2021: Covid 19 impact and recovery to 2030, 2021.

[3] Jacek Komorowski., Grzegorz Kurzejamski., and Grzegorz Sarwas. Footandball: Integrated player and ball detector. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pages 47–56. INSTICC, SciTePress, 2020.

[4] Jianhui Chen and J. Little. Sports camera calibration via synthetic data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2497–2504, 2019.

[5] 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. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 201–210, 2020.

[6] Mahesh Ramachandran, Ashok Veeraraghavan, and Rama Chellappa. Chapter 5 - video stabilization and mosaicing. In Al Bovik, editor, *The Essential Guide to Video Processing*, pages 109–140. Academic Press, Boston, 2009.

[7] I. Skrypnyk and D.G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 110–119, 2004.

[8] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[9] Robin Hess and Alan Fern. Improved video registration using non-distinctive local image features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[11] Kenji Okuma, J. Little, and David G. Lowe. Automatic rectification of long image sequences. 2003.

[12] Carlo Tomasi. Detection and tracking of point features. 1991.

[13] Wei-Lwun Lu, Jo-Anne Ting, James J. Little, and Kevin P. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1704–1716, 2013.

[14] Jianhui Chen, Fangrui Zhu, and J. Little. A two-point method for ptz camera calibration in sports. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 287–295, 2018.

[15] 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.

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

[17] 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, 2020.

[18] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56:221–255, 2004.

[19] Georgios D. Evangelidis and Emmanouil Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[20] 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, 2021.

[21] Ho-Sub Yoon, Younglae J. Bae, and Young-Kyu Yang. A soccer image sequence mosaicking and analysis method using line and advertisement board detection. *Etri Journal*, 24:443–454, 2002.

[22] T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences. In *2002 International Conference on Pattern Recognition*, volume 1, pages 210–213 vol.1, 2002.

[23] Yu Huang, Joan Llach, and Sitaram Bhagavathy. Players and ball detection in soccer videos based on color segmentation and shape analysis. volume 4577, pages 416–425, 2007.

[24] Di Zhong and Shih-Fu Chang. Real-time view recognition and event detection for sports video. *J. Vis. Commun. Image Represent.*, 15:330–347, 2004.

[25] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[26] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[27] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 39(06):1137–1149, 2017.

[29] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 42(02):386–397, 2020.

[30] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[31] W. Liu, Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[32] Jacek Komorowski, Grzegorz Kurzejamski, and Grzegorz Sarwas. Deepball: Deep neural-network ball detector. 2019.

[33] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944. IEEE Computer Society, 2017.

[34] Yang Liu, Dawei Liang, Qingming Huang, and Wen Gao. Extracting 3d information from broadcast soccer video. *Image Vis. Comput.*, 24:1146–1162, 2006.

[35] T. Shimawaki, T. Sakiyama, J. Miura, and Y. Shirai. Estimation of ball route under overlapping with players and lines in soccer video image sequence. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 359–362, 2006.

[36] Chaoke Pei, Shuyuan Yang, Li Gao, and Weipeng Ma. A real time ball detection framework for soccer video. In *2009 16th International Conference on Systems, Signals and Image Processing*, pages 1–4, 2009. doi: 10.1109/IWSSIP.2009.5367707.

[37] Jinchang Ren, James Orwell, Graeme A. Jones, and Ming Xu. Tracking the soccer ball using multiple fixed cameras. *Comput. Vis. Image Underst.*, 113:633–642, 2009.

[38] Jong-Yun Kim and Tae-Yong Kim. Soccer ball tracking using dynamic kalman filter with velocity control. In *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization*, pages 367–374, 2009. doi: 10.1109/CGIV.2009.87.

[39] Arda Senocak, Tae-Hyun Oh, Junsik Kim, and In So Kweon. Part-based player identification using deep convolutional representation and multi-scale pooling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1813–18137, 2018.

[40] Lamberto Ballan, Marco Bertini, A. Bimbo, and Walter Nunziati. Soccer players identification based on visual local features. In *CIVR '07*, 2007.

[41] Matko Saric, Hrvoje Dujmic, Vladan Papić, and Nikola Roić. Player number localization and recognition in soccer video using hsv color space and internal contours. *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 2: 1408–1412, 2008.

[42] Qixiang Ye, Shuqiang Jiang, Yang Liu, and Wen Gao. Jersey number detection in sports video for athlete identification. *Proc SPIE*, 5960:1599–1606, 2005.

[43] 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.

[44] Ruiheng Zhang, Lingxiang Wu, Yukun Yang, Wanneng Wu, Yueqiang Chen, and Min Xu. Multi-camera multi-player tracking with deep player identification in sports video. *Pattern Recognition*, 102:107260, 2020.

[45] Alvin Chan, Martin D. Levine, and Mehrsan Javan. Player identification in hockey broadcast videos. *Expert Syst. Appl.*, 165:113–891, 2021.

[46] Tiziana D'Orazio, Marco Leo, Paolo Spagnolo, Pier Luigi Mazzeo, Nicola Mosca, Massimiliano Nitti, and Arcangelo Distante. An investigation into the feasibility of real-time soccer offside detection from a multiple camera system. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(12):1804–1818, 2009.

[47] Maxime Istasse, Julien Moreau, and Christophe Vleeschouwer. Associative embedding for team discrimination. pages 2477–2486, 2019.

[48] Maria Koshkina, Hemanth Pidaparthy, and James H. Elder. Contrastive learning for sports video: Unsupervised player classification. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4523–4531, 2021.

[49] Patrick Lucey, Dean F. Oliver, Peter Carr, Joerg. Roth, and Iain Matthews. Assessing team strategy using spatiotemporal data. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013.

[50] Laurie Shaw and Mark E. Glickman. Dynamic analysis of team strategy in professional football. 2019.

[51] Cem Direkoglu and Noel E. O'Connor. Team activity recognition in sports. In *ECCV*, 2012.

[52] Rui Marcelino, Jaime Sampaio, Guy Amichay, Bruno Gonçalves, Iain D. Couzin, and Máté Nagy. Collective movement analysis reveals coordination tactics of team players in football matches. *Chaos Solitons & Fractals*, 138:109831, 2020.

[53] Panagiotis Mavrogiannis. Amateur football analytics using computer vision. Master's thesis, University of Piraeus, 2021.

[54] Indriyati Atmosukarto, Bernard Ghanem, Shaunak Ahuja, Karthik Muthuswamy, and Narendra Ahuja. Automatic recognition of offensive team formation in american football plays. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 991–998, 2013.

[55] Floris Goes, Michel Brink, Marije Elferink-Gemser, Matthias Kempe, and Koen A.P.M. Lemmink. The tactics of successful attacks in professional association football: large-scale spatiotemporal analysis of dynamic subgroups using position tracking data. *Journal of Sports Sciences*, 39, 2020.

[56] Alina Bialkowski, Patrick Lucey, Peter Carr, Iain Matthews, Sridha Sridharan, and Clinton Fookes. Discovering team structures in soccer from spatiotemporal data. *IEEE Transactions on Knowledge and Data Engineering*, 28:1–1, 2016.

[57] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *2014 IEEE International Conference on Data Mining*, pages 725–730, 2014.

[58] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[59] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742, 2006.

[60] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[61] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.

[62] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

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

[64] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[65] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[66] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[67] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[68] Tiziana D'Orazio, Marco Leo, Paolo Spagnolo, Pier Luigi Mazzeo, Nicola Mosca, Massimiliano Nitti, and Arcangelo Distante. An investigation into the feasibility of real-time soccer offside detection from a multiple camera system. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(12):1804–1818, 2009. doi: 10.1109/TCSVT.2009.2026817.

[69] Keyu Lu, Jianhui Chen, J. Little, and Hangen He. Light cascaded convolutional neural networks for accurate player detection. *ArXiv*, abs/1709.10230, 2017.

[70] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP*, 2007.