KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Information and Network Engineering

Second cycle, 30 credits

Soccer Data Analysis Based on Computer Vision

Master Thesis at KTH Royal Institute of Technology

RONGFEI PAN

Stockholm, Sweden, 2024

Soccer Data Analysis Based on Computer Vision

Master Thesis at KTH Royal Institute of Technology

RONGFEI PAN

Master's Programme, Information and Network Engineering, 120 credits
Date: February 27, 2024

Supervisors: Ola Lindmark Eriksson, Antoine Honoré
Examiner: Markus Flierl
School of Electrical Engineering and Computer Science
Host company: Football Analytics Sweden AB
Swedish title: Fotbollsdataanalys baserad på datorseende
Swedish subtitle: Masteruppsats vid Kungliga Tekniska Högskolan

Abstract

As the top sport in the world without any doubt, soccer has a wide influence on human society. Since the beginning of modern soccer, soccer tactics have been developed for a long time. Clearly, it requires data for soccer analysis, which includes not only the match results between each team but also performance of players on the pitch. Playmaker.ai, where this degree project has been carried out, is a company that provides soccer analysis services. The major purpose of this project is to create a system that can generate player position by analyzing video data without bird-view information. Besides player position generation, some progress has been made in expected goal calculation and implemented some data preprocessing tools. In this project, the goal is accomplished in following steps:

1. Detect players from camera view images by using YOLO (You Only Look Once) network.
2. Use Strong-Sort method to track the position of players and ball in a long video.
3. Assign the teams to different detected object, methods including K-means are used in this step.
4. Generate bird view position by using perspective transformation method

The result shows that all the machine model successfully converged and achieve good performance in practical usage, despite that there are still existing limitations and problems. By using this system, a 2-D map with player position on this map can be generated. And the data preprocessing tools can also be used for the company. Admittedly, because of several limitation in practical development, there are problems and disadvantage of the system. This system could be considered as a prototype of a complete method for solving multiple issues in soccer data analysis based on machine learning and computer vision. The future developers can iterate this project for further improvement.

Keywords

Soccer, XG, Computer Vision, Object Detection, Perspective Transformation

Sammanfattning

Som den bästa sporten i världen utan tvekan har fotboll ett stort inflytande på det mänskliga samhället. Sedan starten av modern fotboll har fotbollstaktik utvecklats under lång tid. Det kräver helt klart data för fotbollsanalys, som inte bara inkluderar matchresultaten mellan varje lag utan även spelarnas prestation på planen. Playmaker.ai, där jag gjorde det här examensarbetet, är ett företag som tillhandahåller fotbollsanalystjänster. Huvudsyftet med detta projekt är att skapa ett system som kan generera spelarposition genom att analysera videodata utan fågelvyinformation. Förutom spelarpositionsgenerering, gjorde jag också vissa framsteg i xG-beräkning och implementerade några verktyg för förbearbetning av data. I det här projektet uppnådde jag målet i följande steg:

1. Upptäck spelare från kameravisningsbilder genom att använda YOLOv5-nätverket.

2. Använd Strong-Sort-metoden för att spåra spelares och bollens position i en lång video.

3. Tilldela teamen till olika upptäckta objekt, metoder inklusive Kmeans används i detta steg.

4. Generera fågelvyposition genom att använda perspektivomvandlings-metoden.

Resultatet visar att alla maskinmodeller framgångsrikt konvergerade och uppnår bra prestanda i praktisk användning, trots att det fortfarande finns begränsningar och problem. Genom att använda detta system kan vi framgångsrikt generera en 2D-karta med spelarposition på denna karta. Och verktygen för dataförbehandling kan också användas för företaget. Visserligen, på grund av flera begränsningar i praktisk utveckling, finns det problem och nackdelar med systemet. Detta system skulle kunna betraktas som en prototyp av en komplett metod för att lösa flera problem inom fotbollsdataanalys baserad på maskininlärning och datorseende. Den framtida utvecklaren kan upprepa detta projekt för att göra framsteg.

Nyckelord

Fotboll, XG, datorseende, objektdetektering, perspektivomvandling

Acknowledgments

I would like to express my deepest thankfulness to everyone who has contributed to the completion of this Master's thesis. This academic journey has been both challenging and rewarding, and I am sincerely thankful for the support and encouragement I have received along the way.

First and foremost, I extend my heartfelt appreciation to my thesis advisor, Ola Lindmark Eriksson, for his guidance, expertise, and unwavering commitment. Their insights and constructive feedback have been invaluable in monitoring the direction and quality of this research.

Special thanks go to my family and friends for their unwavering encouragement and understanding during the ups and downs of this academic pursuit. Your support has been my pillar of strength. Lastly, I express my gratitude to all those whose work and contributions have been cited and referenced in this thesis. Your research has been instrumental in shaping the theoretical foundation of my study.

This thesis is a significant step in my academic and work journey, and it would not have been possible without the support of all those mentioned above. Thank you for being part of this important milestone in my life.

The two years study in Royal Institute of Technology passed in a flash, and I was about to graduate and embark on a new journey in Shanghai, China. On the occasion of graduation, I sincerely thank the teachers, friends, and family around me for their help and encouragement, and all the workers fighting the epidemic for their efforts to provide us with a safe environment.

Sincerely,

Stockholm, February 2024
Rongfei Pan

Contents

List of Figures

List of acronyms and abbreviations

AFLink	Apperance-Free Link Model
AWS	Amazon Web Service
CNN	Convolutional Neural Network
COCO	Common Objects in Context
Deep-Sort	Deep-Simple Online and Realtime Tracking
Faster-RCNN	Region-based Convolutional Neural Networks
FIFA	International Federation of Association Football
GAN	Generative Adversarial Network
GSI	Gaussian Smooth Interpolation
IDE	Integrated Development Environment
IOU	Intersection over Union
mAP	Mean Average Percision
MLP	Multi-layer Perceptron
ROC	Receiver Operating Characteristic
Sort	Simple Online and Realtime Tracking
Strong-Sort	Strong-Simple Online and Realtime Tracking
xA	Expected Assist
xG	Expected Goal
Yolo	You Only Look Once

Chapter 1

Introduction

The introduction section describes the fundamental information of the project. Firstly, it gives general background information in soccer data analysis field, including concepts that are used in this thesis and the company that can deliver services in this field, which is also the company where this thesis was carried out. Then, the research problems is described by stating the problem when generating and detecting player positions in a soccer game. The purpose and goals sections describe why these methods are developed. Then, the methodology is briefly described, and the delimitations of the study are introduced. Finally, the structure of the thesis section describes how this thesis is organized and written.

1.1 Background

As the top one sport in the world, soccer has a wide influence on human society. The FIFA World Cup is the most valuable and popular sporting event in the world, which even surpasses the Olympic Games in terms of the number of spectators and business income. Clearly, it requires data for soccer analysis, which includes not only the match results between each team but also performance of players on the pitch. And there is a huge amount of data that can be used and generated. For example, in recent years, expected goals (xG) [1] have been more and more popular in soccer data analysis. Other typical evaluation indexes include stress levels on the ball, expected assist(xA), width per sequence and so on. As the foundation of xG and xA, position data of players is also necessary for soccer data analysis.

Playmaker.ai is a company that provides soccer analysis services. They offer the services by transforming soccer data to insight to help clubs and

media companies all over the world. Typical data that the company analyzed about, includeing xG and xA, was mentioned above. To analyze these data, Playmaker.ai collected data from both matches from national teams and regional leagues in Sweden. It includes video and event data which are all tagged manually. From these data, insights can be obtained and provide feedback for players, coaches and analyst.

xG is discussed in the first paragraph, which is defined by the possibility of getting a goal in a certain shot. Obviously, xG could be used in various way. For example, the xG in one game can represent the ability of creating opportunity of team. Also, if xG of a player is much higher than average, people can realize that scoring ability is above average. In modern soccer data analysis, this statistic has become a highly recognizable term, and be used by not only media commentators, but also fans.

In addition, stress level that players can perceive is a significant data during the game. It can be easily observed that some players are very weak under stress level of medium level. On the contrary, top soccer players can maintain top performance even under high level of stress level. In practice, analysts estimate the level of stress level manually. Normally researchers may take many factors into account, including position of the ball, teammates and defenders. Clearly, a quantifying method of measuring stress level can be very useful in soccer analysis.

From the background knowledge above, the research question that uses modern computer science and image processing technique can be brought out to achieve soccer analysis that is more complete and accurate.

1.2 Problem

To analyze the data in soccer games, firstly, it is necessary to get the position of players on the pitch. An ideal solution of this problem is to analyze the image obtained from bird eye view camera, shown in Fig. 1.2, that is put on the top of stadium. It can be done by simply detecting the moving object and recording their corresponding position in images. However, in practical situation, it is very likely that there's no bird view camera, and normally there's only one camera located outside the pitch with a large rotation angle as shown in Fig. 1.1. Therefore, it is necessary to calculate the position of players in 2-dimensional view by analyzing 3D images from cameras view.

Figure 1.1: Camera View Example

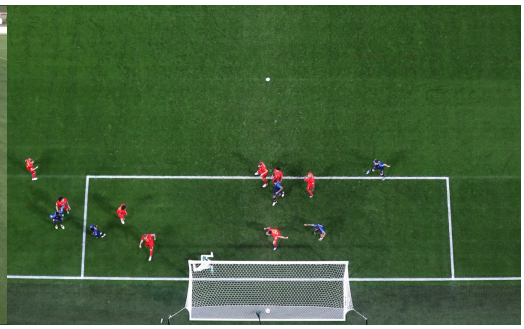

Figure 1.2: Bird View Example

1.3 Purpose

The most significant purpose of this thesis is to create a system that can generate players and ball position by analyzing camera view match video. In the highest level football league in Europe, it is not difficult to identify the position of players, given the fact that they normally have a computer-controlled, stabilized, cable-suspended camera system called Skycam. In the low level football league, e.g., Ettan Football league, which is the third level football league in Sweden, there's no equipment like Skycam. And coaches and players also want to use the relative data to analyze performance and develop tactics. Therefore, such a system is necessary for them. As a company that deliver football analysis service in Sweden, Playmaker.ai is committed to providing data and data analysis services for football matches, including services for Ettan Football league.

1.4 Goals

The goal of this project has been divided into the following three sub-goals:

1. Player and Ball Position Generation

   The first research question concentrate on the generation and detection of players and the ball position in a 2-D bird-view map by analyzing camera view videos. By using computer vision algorithms and machine learning , this study aims to accurately track the positions of players and the ball throughout the game. The resulting bird-view map provides a comprehensive visual representation of the game, enabling in-depth analysis into player and ball movements. During the analysis process, the system should be able to detect the side of each players.

Due to the huge amount of work required to fully develop the entire system, the purpose of this project is to create a prototype for subsequent students. Students in the future degree projects can iterate my project to make some progress and extend the function and improve the robustness of my work.

2. Event searching and data pre-processing

   The second research question focus on developing an efficient part of project to search different kinds of events, e.g., shoots, passes, dribbles and so on. Then find the corresponding time stamp in the video and analyze it automatically. Also, foundational information of each games, including time stamp, color of teams and so on,should be preloaded by analyzing the relative files.

3. xG Calculation

   Based on the generated position data, the final research question focuses on implementing a method to calculate expected goals (xG). By using the calculated players and ball positions, this study aims to develop a robust xG model that accurately quantifies the likelihood of a goal being scored.

1.5 Research Methodology

Object Detection

To find the players from the video data, object detection approach is necessary. *Yolo* [2] is one of the most commonly used approach for general object detection. Although Komorowski et al. [3] and Lu et al. [4] have brought ideas for player detection specifically, general approaches are available for our dataset. As it is shown in the work from Zhang et al. [5], *Yolov*4 performs efficiently on player detection problems.

Object Tracking

Object tracking refers to detect and track the position and dynamic changes of specific targets in videos through computer vision technology. *Strong-Sort* [6] could be used for object tracking. On the basis of object detection, Sort method first predict the position by using Kalman filter [7], and then match the

result with the result of object detection by using matching cascade [8] with global linear assignment and get the optimal bounding box as the output.

Team Color Detection

K-means [9] is one of approach that could be used for color detection. Also, CNN method [10] can be used for this problem. Considering the practical situation, it is also possible to determine the color of team by finding pattern in color distribution of images.

Perspective Transformation

For the perspective transformation, Chen and Little's work [11] brought a complete method that is designed for camera view perspective transformation on soccer pitch. The general steps are as follow:

1. By using the two-GAN network, field and edge images are generated.
2. From camera pose engine, authors generate a database of edge images.
3. After extracting features from edge images, authors compare the text feature and database feature.
4. Use Perspective transform matrix in database to generate 2-D map.

1.6 Delimitations

Within the boundary of this project, several limitations and assumptions need to be clarified.

- Due to the limitation of the validation dataset in this project, which is the data from the Ettan football league, the developer cannot quantify the accuracy performance in the validation dataset.
- The data from Playmaker.ai is not perfect. For example, the time data in Xlsx files and Json files are not completely accurate. Based on my observation, the time data has an error of about 1 second at most. Also, the position data is scaled proportionally to numbers from 0 to 100. It will also cause an error of about 1 meter. All these flawed data will lead to inaccuracy of the result

- In this project, several assumptions have been made. First, we assume that the dataset is correctly filtered, classified, and labeled. There's also an assumption that all manually-labeled data are correct and accurate.
- Due to the large amount of work involved in this project, the purpose of this project may not be fully achieved during the graduation design period. Therefore, this project focuses on creating a prototype for future work. The most important thing is to generate the position of players and the ball.

1.7 Structure of the thesis

This thesis follows a structure to provide a complete introduction to the topic. It begins with an Introduction, presenting the background, research problem, purpose and goals, and some technical considerations of this project. The Related Work section analyzes research and methodologies that are necessary to understand the article. The Methods section outlines the data sources and the data used for development analysis, and this sections also discuss how the methods are designed for problem-solving and the code of the project. The Implementation and Results section presents the implementation details of this project and the major result and outcome, and it also evaluates the performance of the system. The Discussion section discusses the problems that appear during the project. The final section summarizes the result and suggests some potential future working directions. This structured approach ensures a concise and comprehensive understanding of the thesis.

Chapter 2

Related Work

In this chapter, I presented the background study necessary to understand the thesis. This chapter will go through basic theory in computer vision and machine learning, object detection and tracking, image classification and perspective transformation. Some tools and frameworks used in the project are also be introduced in this chapter.

2.1 Machine Learning and Computer Vision

Computer vision and machine learning have made great progress in recent years. Especially, the appearance of the Convolutional Neural Networks (CNN) appeared is important in this domain. CNN [12], initially introduced by LeCun et al., in the 1990s, has become one of the most popular technology in image processing and computer vision area.

2.1.1 Neural Network

Neural networks are composed of a large number of neurons connected to each other. A typical neuron model, shown in Fig. 2.1 consists of the following 3 parts: 1) A set of synapses: represented by weights. 2) Adder (adder): sums the weighted inputs. 3) Activation function: also called squashing function, it acts on the output of neurons and is generally a nonlinear function. After each neuron receives the input of a linear combination, it is initially simply linearly weighted. Later, a nonlinear activation function is added to each neuron to perform a nonlinear transformation and output. The connection between each two neurons represents a weighted value, called weight. Different weights and activation functions will lead to different outputs of the neural network.

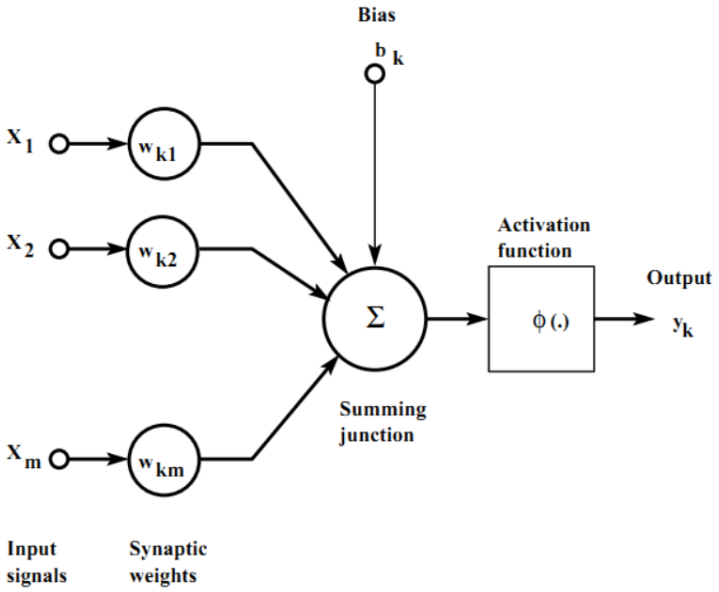

Figure 2.1: A Typical Neuron Structure

2.1.2 Activation Function

Commonly used nonlinear activation functions include sigmoid, Relu [13], etc. The first two, sigmoid, is more common in fully connected layers, and the latter, Relu, are common in convolutional layers. The Relu function is shown in Equation 2.1 and Fig. 2.2. It is not difficult to see from the above figure that the ReLU function is actually a segmented linear function, changing all negative values to 0, while leaving the positive values unchanged. There are many advantages if the model uses ReLu instead of sigmoid as activation functions. First, when using functions such as sigmoid, the amount of calculation is large when calculating the activation function (exponential operation). When backpropagating to find the error gradient, the derivation involves division, which requires a relatively large amount of calculation. However, when using the Relu activation function, the amount of calculation for the entire process is Save a lot. Second, ReLu will cause the output of some neurons to be 0, which causes the sparsity of the network, reduces the interdependence of parameters, and alleviates the occurrence of over-fitting problems.

$$f(x) = max(0, x) \tag{2.1}$$

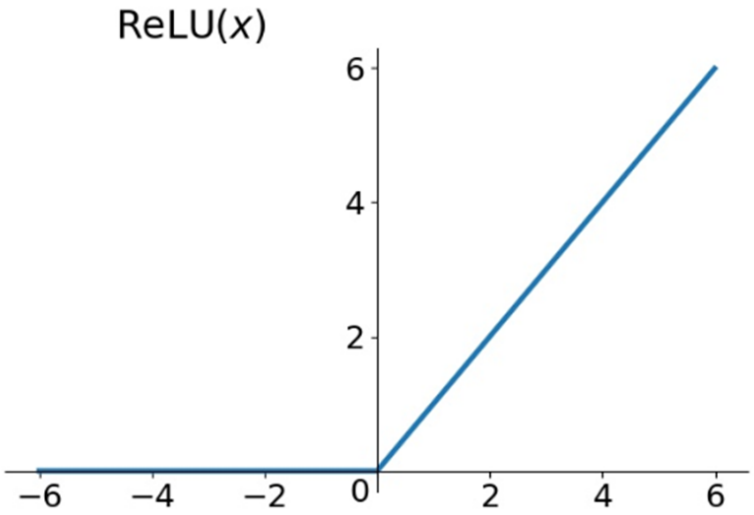

Figure 2.2: Relu Function

2.1.3 Multilayer Perception

Multilayer Perceptron (MLP) is a neural network model composed of an input layer, a hidden layer (one or more layers), and an output layer. It can solve linear inseparable problems that a single-layer perceptron cannot solve. When a multilayer perceptron is used for classification, the number of input neurons is the dimension of the input signal, and the number of output neurons is the number of categories. The equation of MLP are shown in Equation 2.2 and 2.3.

$$z_i^{l+1} = \sum_j W_{ji}^l y_j^l + b_i^l \tag{2.2}$$

$$y_i^{l+1} = f(z_i^{l+1}) \tag{2.3}$$

Among them, y_j^l is the output of the jth neurons in lth layer, z_i^{l+1} is the value of th ith neuron before using activation function in $l$the layer. W_{ji}^l is the weights beteen the jth neuron in lth layer and the ith neuron in $l+1$th layer, b_i^l is the bias, $f(\cdot)$ is the activation function. The loss function is shown in Equation 2.4.

$$J = \frac{1}{2}\sum_i (y_i^L - yi)^2 \tag{2.4}$$

Among them, y_i^L is the output of ith neuron in the last layer of the neural

network. y_i is the truth value of the ith neuron. The purpose of training a neural network is to minimize the value of loss function. The most commonly seen optimization method is stochastic gradient descent method [14].

2.1.4 Convolutional Neural Network

Convolutional Neural Network (CNN), shown in Fig. 2.3 is a neural network whose artificial neurons can respond to surrounding units within a part of the coverage area and have excellent performance in large-scale image processing.

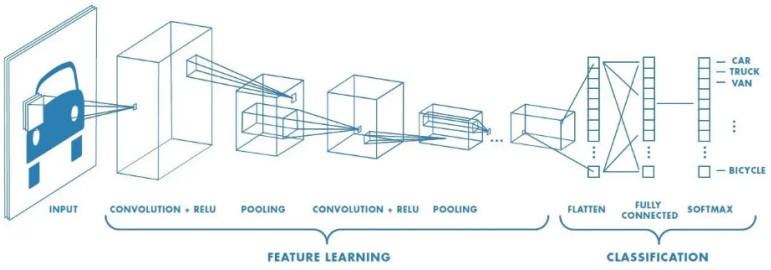

Figure 2.3: CNN Structure

A typical CNN consists of 3 parts: convolution layer, Pooling layer and Fully connected layer. The convolutional layer is responsible for extracting local features in the image; The pooling layer can reduce the data dimension more effectively than the convolution layer. This can not only greatly reduce the amount of calculations, but also effectively avoid overfitting. The fully connected layer is similar to the traditional neural network and is used to output the desired results. A typical CNN normally not just the 3-layer structure mentioned above, but a multi-layer structure. There are several convolutional layers and sampling layers, and the convolutional layer and the sampling layer are alternately set, that is, a convolutional layer is connected to a sampling layer, and then a convolutional layer is connected to the sampling layer, and so on. Since each neuron of the output feature surface in the convolution layer is locally connected to its input, and the corresponding connection weight is weighted and summed with the local input, and then the bias value is added, the input value of the neuron is obtained.

CNN is normally used to analyze and process images in image classification, object detection and image generation tasks. It use a structure based on Multi-layer Perceptron (MLP) structure. In a CNN network, there are many convolution filters traversed into the feature map of the upper layer, and then generate a new feature map as output. These feature maps have

the features of images related to the previous feature maps, and then, the results of the previous layers are stored and used as input to the next layer of the network. Therefore, the features of the image can be extracted and updated in each layers. In a typical CNN, the convolution filter are usually followed by activation function and different types of pooling layers. These features will be updated and spread layer by layer until to the final layer, and then the final layer can generate output of our network. And there are multiple ways to optimize the training process to make it more friendly to engineers, researchers and developers, and some module and techniques in the network can help us to confront problem of machine learning and improve the performance. Firstly, for example, these operations can be trained in batches, which can help developers and researchers to speed up their training process. Sometimes, during the training process, some layers of the model are randomly connected [15], which means that the whole network is not fully connected. Batch normalization [16] is also widely used after the convolutional layer to ensure that it has the ability to overcome the problem of gradient vanishing and exploding. Also, some neural networks use residual connections to learn new features that can't be extracted by normal CNN. Back-propagation algorithm [17] is also used to train these networks.

CNN have achieved amazing results in many different types of computer vision and image processing tasks. One of a significant improvement can be seen in image classification area. Networks like AlexNet [18], VGGNet [19], and ResNet [20] have achieved incredible in ImageNet classification contest. CNN have also been successfully applied to object detection [2], semantic segmentation [21], and image generation [22]. In conclusion, CNN have had a huge influence in computer vision and machine learning.

2.2 Object detection

In object detection area, there are many advancements can be seen in recent years. One of the significant technique is Yolo(You Only Look Once) network.

The Yolo network, developed by Joseph Redmon et al., is an state-of-art machine learning and computer vision that is designed for real-time object detection. The Yolo network detects objects and generates bounding boxes. And then it associates probabilities to each of the detected images using a simple convolutional neural network (CNN). As illustrated in Figure 2.4, the detector usually consists of two parts. One is the backbone network for extracting features, that is, the basic network, which is generally pre-trained on the ImageNet data set. The other is to predict object categories and bounding

box heads. Also, Neck is built between the trunk and the head to bring together different feature map.

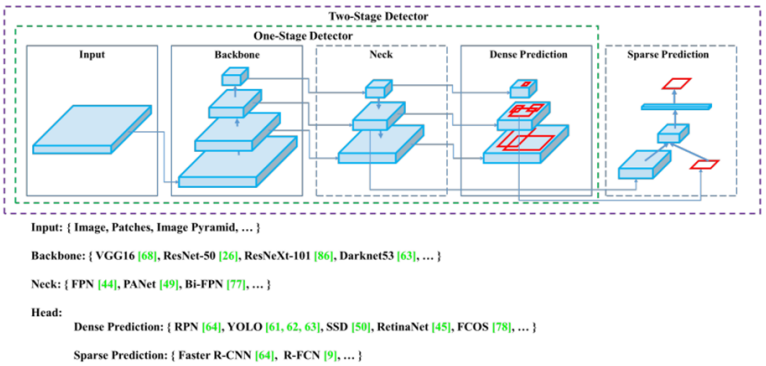

Figure 2.4: Yolo Network Structure

The backbone network of YOLOv4 uses the Cross-Stage Partial Networks (CSP) structure to reduce the amount of calculation and improve accuracy and adopt the CSPDarknet53 [23] architecture, which is an improved version of Darknet53 [24]. The Neck network of YOLOv4 is composed of two modules: Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN). SPP significantly increases the receptive field, isolating important contextual features without slowing down running speed. It can extract feature information at different scales; the PAN module can aggregate multi-layer features to improve detection performance. YOLOv4's Head network is improved from YOLOv3 [24]'s Anchor-based Head and RetinaNet's Focal Loss. Anchor-based Head uses anchor boxes to predict the location and size of the target, while RetinaNet's Focal Loss can alleviate the category imbalance problem and improve detection performance.

YOLOv4[13] uses the Mish activation function [25] in the backbone network, which has the following characteristics: low cost, smoothness, non-monotone, no upper bound, lower bound, etc. The equation of Mish activation function is shown in Fig. 2.5 and the following equation:

$$f(x) = xtanh(ln(1 + e^x)) \tag{2.5}$$

The characteristic that positive values is unlimited, avoids saturation due to maximum value after activation function. The slight allowance for negative values theoretically allows for better gradient flow, rather than hard zero bounds like ReLU, and the smooth activation function allows better information to penetrate deep into the neural network, resulting in better

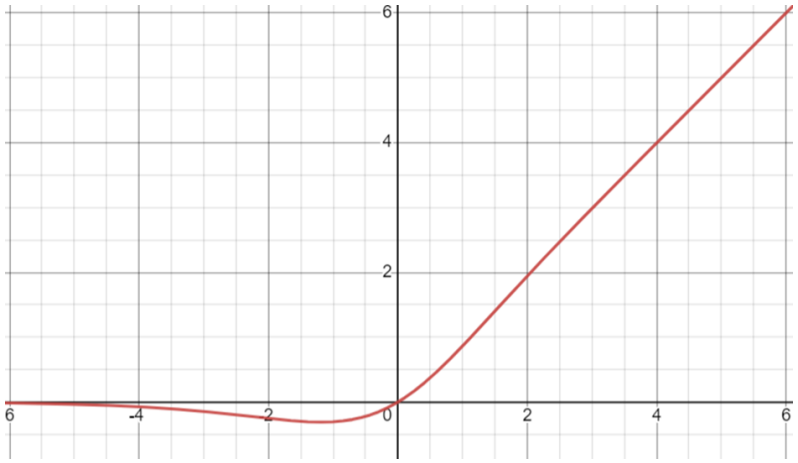

Figure 2.5: Mish Activation Function

accuracy and generalization.

The application of Yolo can be found in many fields. For example, when development autonomous driving algorithm, it is necessary to obtain the information of surrounding environment, including vehicles, pedestrian, road and so on. Obviously, all these objects can be detected by using Yolo network. In addition, in medical image processing area, Yolo network can help doctors to identify the tumors and disease in X-ray images. Based on the good accuracy of Yolo network, it can great improve the diagnosing probability.

2.3 Object Tracking

Object tracking research has made significant progress these years, and a significant algorithm in this field is the Strong-SORT(Strongly Simple Online Real-time Tracking) method. Strong-SORT [6], whose structure is shown in Fig. 2.6 ,proposed by Du et al., gave a solution facing the challenges of real-time object tracking in various scenarios.

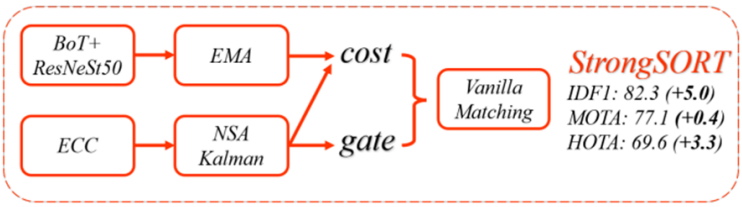

Figure 2.6: Strong-Sort Structure

The structure of Strong-Sort algorithm consist of two parts: Appearance approach and motion approach. And then, it uses a association procedure to combination the information we got in each approach and get the new result.

In the appearance approach, Strong-Sort uses appearance feature extractor, BOT [26], which used Resnet50 [20] to extract discriminative feature. This extractor is used in each frame of the video. Then, it uses Exponential Moving Average (EMA). The EMA strategy retain the information of inter-frame features and update the feature by using the following equation:

$$e_i^t = \alpha e_i^{t-1} + (1 - \alpha) f_i^t \tag{2.6}$$

where f_i^t is the appearance feture in t frame and i tracklet. e_i^{t-1} is the appearance state for the i-th tracklet in $t-1$ frame. And α is a momentum term, which equals to 0.9 in Strong-Sort algorithm. After obtaining every new detection results, the smallest cosine distance between new detected object and the object result in previous frame. And this distance is used as the appearance cost in the association procedure.

In the motion approach, firstly, the correlation coefficient maximization (ECC) [27] is used for used for cammear motion compensation. It can estimate the global rotation and translation between frames. Also, the motion approach, it uses the Noise Scale Adaptive (NSA) Kalman filter [28] from GIAOTracter [29]. The Kalman filter [30] predicts the position by using both state prediction and state update. And the NSA Kalman filter is used to solve the problem of low-quality detection [31]. Different from the Kalman filter, NSA Kalman filter uses the following equation to calculate the noise covariance $\widetilde{R}_k$:

$$\widetilde{R}_k = (1 - c_k) R^k \tag{2.7}$$

where R_k is the preset constant measurement noise covariance and c_k is the detection confidence score in state k. And $\widetilde{R}_k$ is the lower noise covriance. With the motion state obtained from NSA Kalman filter, Mahalanobis distance [32] is used to find the most similar motion state.

To combine the information in both appearance approach and motion approach, it uses the following cost matrix as the total cost:

$$C = \lambda A_a + (1 - \lambda) A_m \tag{2.8}$$

where λ is 0.98, A_a is the appearance cost and the A_m is the motion cost

Strong-SORT algorithm has shown enhanced accuracy, robustness, and efficiency compared to traditional tracking algorithms and previous SORT algorithm.

2.4 Clustering Algorithm

A clustering algorithm, shown in Fig. 2.7, is utilized to help us group similar data points in a given datasets.In recent years, researchers have been developed many clustering algorithms, and one of the most commonly seen algorithm is the K-means algorithm [9]. K-means is a unsupervised learning algorithm that as been widely used in classification tasks. Firstly, it generates random centroid based on number of clusters. Secondly, it divides the given data into clusters by assigning data points to the closest centroid. Normally, the closest color is found by calculating the euclidean distance. Then, it updates the centroids based on the assigned points. The previous steps is repeated time and time again until all the data points are correctly classified. The advantages of K-means algorithm is very obvious, given the fact that it is easily to understand and implement. Also, choosing a proper K value, which is the number of clusters, is very important for ensuring the performance of this algorithm.

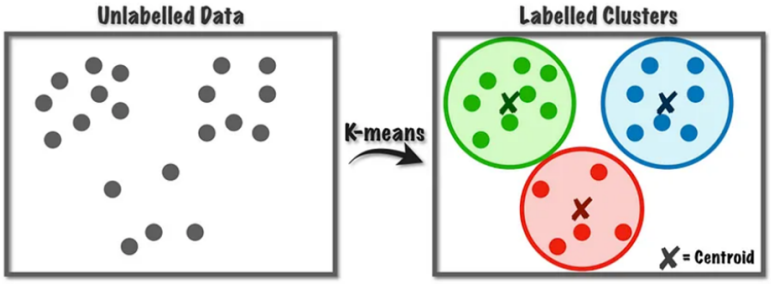

Figure 2.7: K-means Algorithm

The application of K-means algorithm can be found in many aspects. In a classification task, K-means can be used to extract visual features by clustering similar data points. It can also be very useful in segmentation task. For example, it can help marketing company to classify different type of customers based on their shopping habits. Therefore, the company can bring better service for customers.

2.5 Perspective Transformation

Perspective transformation is serves for changing the viewpoint of camera. This process can be achieved by using a perspective transformation matrix. Fig. 2.8 is a typical example that shows the input, which is on the left, and the output, which is on the right, of the perspective transformation process.

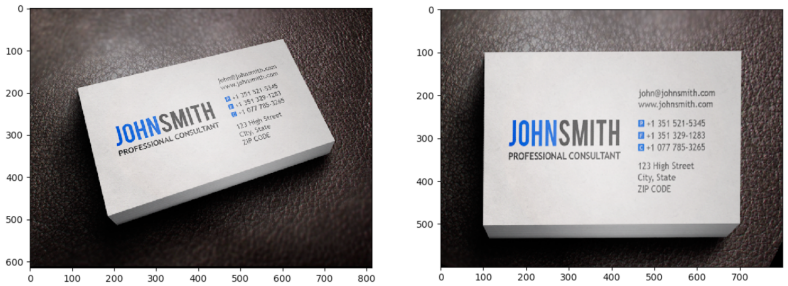

Figure 2.8: Perspective Transformation Example

This process can be achieved by using a perspective transformation matrix. An example of perspective transformation is shown by the following equation:

$$\begin{bmatrix} u \\ v \\ z \end{bmatrix} = \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.9}$$

The matrix on the right side is the input, which x and y are the position in the original image, which is shown in the left side of Fig.2.8 and the top left of Fig. 2.9. The matrix on the left is the output of the process, represents the position in a 2-dimensional template, shown in the right side of Fig.2.8 and the bottom right of Fig. 2.9. The final pixels can be obtained in the following equation:

$$\widetilde{u} = \frac{u}{z} \quad \widetilde{v} = \frac{v}{z} \tag{2.10}$$

where $[\widetilde{u}, \widetilde{v}]$ represents the position in the 2-dimensional template.

The article "Sports Camera Calibration via Synthetic Data" by Chen et al. [11] explores the topic of perspective transformation in the context of sports camera calibration. The authors propose a complete method that can improve camera calibration accuracy. In this article, the perspective transformation method part is used in my thesis.

The general steps is shown in Fig. 2.9. Firstly, the author trained two generative adversarial network (GAN) [33] to detect the edge image of the sports field based on the work of [34]. It includes a segmented GAN, which can segment the grassland area in the input image. And the output of this segmentation GAN is a mask image. Then, it uses a detection GAN to detect the field marking, which is the edge (while lines) of the soccer field. Meanwhile, the author generate a synthetic dataset that simulates sports events which imitates the movement of camera and then calculate their corresponding

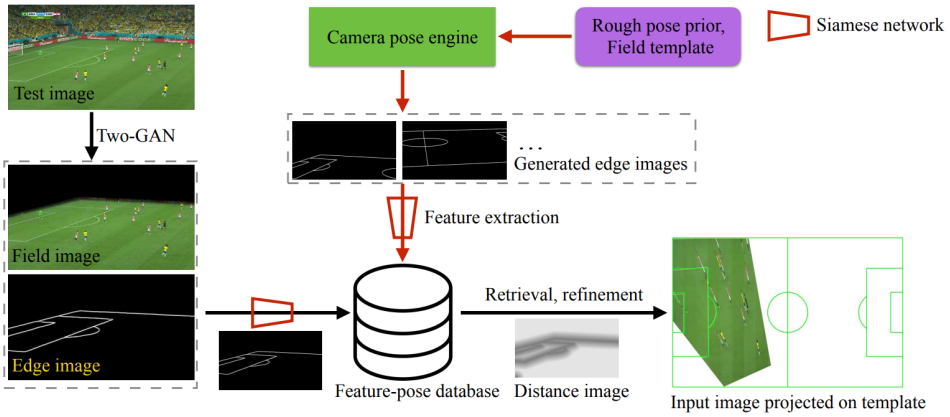

Figure 2.9: Perspective Transformation

edge image and transformation matrix. Then, they stored all these information in a database. When analysis a specific image, the system will firstly use generative adversarial network to generate edge image and use a siamese network [35] to find the most similar edge. The input of this siamese network are two edge images. It has two networks, and both of them are convolutional neural network. The output of the siamese network is the similarity of the two edge images. By using the siamese network, we can find the most similar edge image in the database. As it has been mentioned before, the database collect both the edge image and the corresponding perspective transformation matrix. Then the corresponding perspective transformation matrix can be used to do calculation. The proposed method is evaluated on both synthetic and real sports datasets. And it shows good performance compared to traditional calibration approaches.

2.6 Evaluation methods of machine learning models

In the area of machine learning and computer vision, there are many methods have been developed to evaluate the performance of a model. The most commonly used methods are using accuracy, precision and recall value. These evaluation value can help us to analyze the training process and result. For example, a higher accuracy and precision can definitely guarantee a good training result. There are also information that could be observed during the training process. For instance, the curve of loss during training process

is a good way to see the performance. A successfully training process can definitely generate a converged loss curve. Also, there are some methods can be used when training the machine learning model. For example, k-fold cross-validation method is a way to test the performance of the training process. In conclusion, a good evaluation method is significant for the purpose of ensuring the confidence and reliability of computer vision and machine learning model. In addition, sometimes the practical project lacks of ground truth value. Therefore, a visual evaluation which is done based on researchers experience and knowledge is necessary.

2.7 xG Calculation

Nowadays xG calculating has become a significant topic in soccer analysis. Expected goal means calculate a probability to each shot attempt based on factors such as shot location, angle, and other relevant information. Now, xG models range from statistical approaches to machine learning algorithms. These models require datasets with information from shots and their result to learn features and generate xG values. Factors like player positions, angle, and player information around strikers are considered as important factors in these models. The performance of xG models is evaluated by comparing their predicted value to the actual goal results. The use of xG has greatly influence soccer analysis field. It can provide insights into the quality of shooting opportunities, player performance evaluation, and decision-making strategy.

Chapter 3

Methodology

3.1 Project Process

To accomplish the task in Chapter 1, the project process will be divided into following steps:

1. Collecting data. Collecting a comprehensive dataset is necessary for training a machine learning model. This dataset should include multiple situations in a soccer game, including passes, shots, dribbles and so on to ensure diversity in the dataset. The diversity in dataset can improve the model's ability to learn features in soccer data analysis. In conclusion, collecting well-organized datasets is crucial for this project.
2. Data preprocessing. The data preprocessing in this project consist two parts. Firstly, processing the data should be done for machine learning model training to ensure this quality, consistency and compatibility with the model. It is an important process to optimize the model training process and guarantee the performance. Secondly, it is necessary to process the data comes from playmaker.ai company. These data are stored in xlsx, json, txt and mp4 files. To analysis these resources, preprocessing needs to be done to utilize it in next steps.
3. Position generation.
   (a) The first step in position generation is detecting players and balls in video data by using the pretrained object detection model.
   (b) Then, by using the Strong-Sort algorithm, we will track the players in a video stream.
   (c) In the mean time, after obtaining the clipped player images from the video, K-means method is utilized to detect shirt color of

players. The purpose of this step is to find the side of each players.

(d) By using the perspective transformation method from [11], player and ball position data can be generated in a 2-D bird view map.

4. Position generation method is used for two purpose, which are shown in top and bottom branch in Fig. 3.1. Firstly, in the top branch, image data, which clipped from video files, is used to calculate xG value. Meanwhile. in the bottom branch, the raw video data is used to generate a new video which shows the movement of players on a 2-D map with soccer field as background. By using the new video, coaches and soccer analyst can provide further soccer data analysis service.

3.2 Data Collection

For the purpose of machine learning model training, training dataset is necessary. Also, to implement the methods and algorithms in each steps in position generation, some pretrained models and data are used. In this project, multiple datasets will be used, especially in the position generation part. The object detection part in this project is only used for detecting players and ball in soccer game. Therefore, a soccer analysis dataset with 1000 images will be used in the first step of position generation. The dataset consists of 1000 images with players and ball position as their labels. In the following part, a pretrained model in Strong-Sort algorithm is used. Also, in the perspective transformation part, pretrained siamese network and Two-GAN network is utilized for edge detection and calculated transformation matrix in the database.

Also, the purpose of this project is to do soccer data analysis based on Ettan soccer video data. Therefore, data from playmaker.ai company plays an vital role. The data from playmaker.ai consists of the following four parts:

- video files. The video files are stored on AWS S3 platform, which are videos of each soccer match of Ettan soccer league in the past three years. For each match, there will be four corresponding files, including video file, xlsx file, txt file, and json file.

- xlsx files. For each match, there is a xlsx file that store the specific information of this match. These information includes several events in a soocer match, including passing ball, defense action, losing possession, shot, free kick, dribble, off-side, save from goalkeeper, aerial duel, foul, yellow and red cards, and corner. For each event,

there is a time data and duration data that represent when did this event happened in the video file. The field position data describe where did this event happened on the pitch. In this project, the most important information in xlsx files are name of event, time position and field position. And, there are also other information including results of events, player and team name, and detailed information.

- txt files. For each match, there is a txt file that store the general information of each match. It consist of both home and away team name and their shirt color, which is the information that is necessary for this project. It also stores the name of each player, including lineups and substitutes.

- json files. Also, for each match, there is a json file. The json file stores all the information in xlsx files, and it also shows the expected goal of each shot. These data can help us to test the performance of xG calculation.

3.3 System Overview

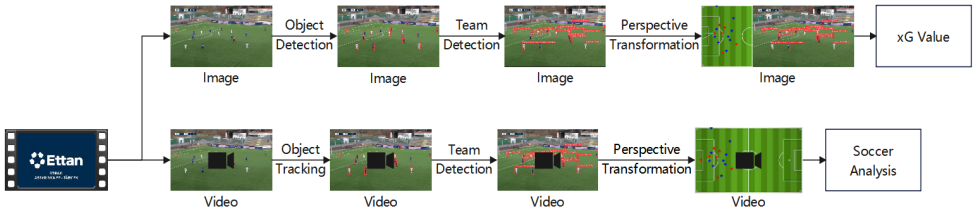

Figure 3.1: System Diagram

3.3.1 Input and Output

As it is shown in Fig. 3.1, there are two kinds of inputs in the project system. The first type is a single images clipped from video file. It correspond to the output of xG calculation and images. The second type is a video, and it corresponds to the output of a new video. This new video shows the movement of players on a 2-D map with soccer field as background. And red and blue dots are used to represent home and away team. So movement of players is shown by the movement of dots.

3.3.2 Data preprocessing

For all the data mentioned in Section 3.2, it is necessary to preprocess it to make it usable for the project. Firstly, the dataset for Yolo network training is preprocessed. I resized the input image to the input size of the neural network. YOLO networks require input images to be 416x416 pixels. Then, I normalize the image data so that it is in the range of 0 to 1 to meet requirements of the input of the network. And finally, I combine the processed images and labels into batches to speed up neural network training. The file structure includes images and labels folder. The images folders store all images for training, and labels folder store txt files that involves information of detected players and balls. For each detected object, the labels are represented with classes, which shows it is player or ball, and position of bounding box, which contour the object.

For the data from playmaker.ai company, each part of data mentioned in Section 3.2 needs to be preprocessed. For xlsx file, codes and programs are written to extract information, including field position, team name and so on, based on time and event name. For video file, codes and programs are written to clip images from video files based on time. For txt file, codes and programs are written by using regulation expression to extract shirt color from it for the team detection part. And xG value can be obtained in json file.

3.3.3 Object Detection and Tracking

To detect players and the ball in a video file for soccer player generation using computer vision and machine learning, the YOLO (You Only Look Once) network is an effective way. It is a popular object detection algorithm. Firstly, individual frames must be extracted from the raw video file. Each image represents a time snapshot that will be fed into the YOLO network for players and ball detection. Then, the YOLO network,trained by using the dataset mentioned in Section 3.2, is applied to each frame. And the network generates bounding boxes around the detected objects, including both players and the ball. The network can also predict the type for each detected object.

Then, the processed frames with the bounding boxes indicating the players and the ball can be used as input of next step. Also, in the Object tracking part of the system, Strong-Sort algorithm is implemented to track players and ball in a long video stream. Strong-Sort algorithm employs techniques including Kalman filtering and data association to fix problems like object disappearances and reappearances in the video stream. It predicts the object's position in each frame and combine it with the detection result from YOLO

network to maintain continuity in the tracking process.

The implementation of Strong-Sort algorithm is shown in Fig. 3.2. In the upper half, which is the appearance approach, I used the pretrained osnet [36] model to extract features. The reason why I chose osnet is that osnet is a relatively light network with less parameters. Therefore, it can speed up the process. Then the exponential moving average method is used to update the appearance status. In the lower half, which is the motion approach part, in the original algorithm, the author used correlation coefficient maximization (ECC) method for camera motion compensation. However, in this project, I disabled this function to speed up the process. Then, the NSA kalman filter is used to predict object motion. Finally, the algorithm combine these two approachs to accomplish tracking.

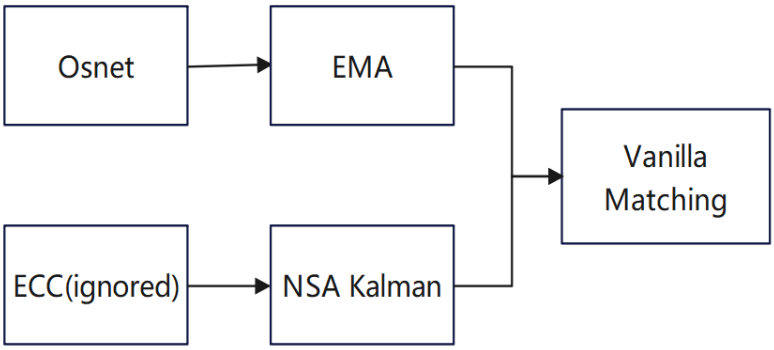

Figure 3.2: StrongSORT Algorithm

3.3.4 Team Detection

To detect the team side of players in the video stream, K-means algorithm is used to identify and analyze the shirt color from images of the players. Initially, the detection result in the previous step is employed as input in team detection. Firstly, for each clipped frame, every pixels are stored in RGB form, which includes value from 0 to 255 in three channels. Then, K-means method is utilized to cluster the pixels in to four clusters. Ideally, these four clusters correspond to four different kinds of information in clipped frame, including colors of the soccer field, skin, shirt and pants. Given the fact that green, which is the color of soccer field is the most commonly seen color, the second large cluster is chosen as the detected color. And I found the closest color as the shirt color in team detection process. In addition, to ensure that the system can detect all colors in the images, I chose nine colors in total as the color

database. These nine colors cover all the shirt color in Ettan football league data, which means these are all the colors that need to be detected.

3.3.5 Perspective Transformation

The perspective transformation process is shown in Fig. 3.3. In previous section, the detection result of players, ball and team, and tracking result of video streams have already been obtained. And it will be used as input in perspective transformation part. Specifically, I extracted the player position in video file. Firstly, the original frame is fed into the Two-GAN network to generate edge image of this frame, this edge image shows the edge of the soccer field. In the feature-pose database, there are many different generated edge images and corresponding perspective transformation matrix. Then, as it is introduced in [11], I used a siamese network to find the most similar edge image. Then I used the edge image's perspective transformation matrix to calculate position of players in 2-D map based on their position in the video frame, which has been generated in the previous section.

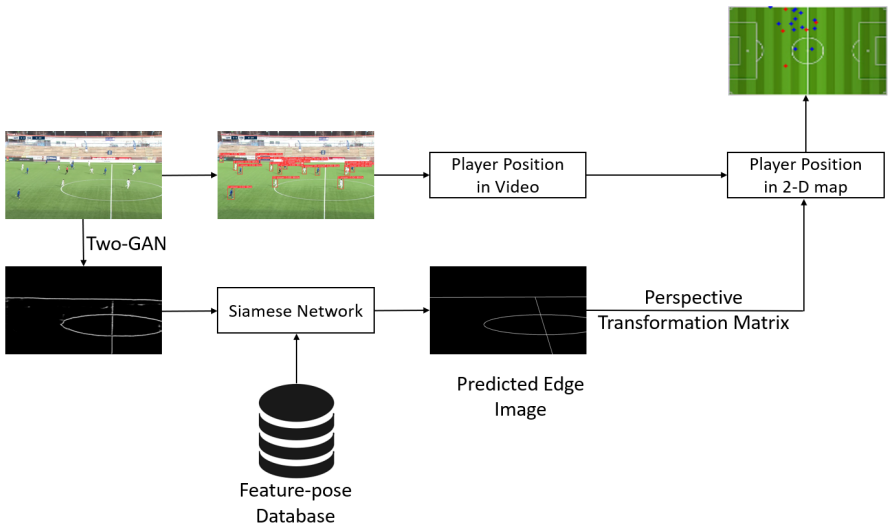

Figure 3.3: Perspective Transformation Diagram

3.3.6 xG Calculation

The xG calculation process is shown in Fig. 3.4. Based on the result in the last section, the 2-D map is used as the input of xG calculation. Firstly, I extracted the necessary data from xlsx files. The purpose of it is to find the shot position

and the current attacking direction. Even though these data are not directly recorded in the xlsx file, I can figure out the attacking direction and target goal according to the initial direction and shot time. Then, with these data, I can draw a triangle with the shooting position and the left and right endpoints of the goal as vertices involves on the 2-D map. Based on all the information above, I wrote program to get various information for xG value calculation, including number of enemy players that is close to the shooting player, the number of players in this triangle, goalkeeper position and so on. The predicted xG will be compared with the ground truth xG value, which is extracted from json file. In this project, presenting a method to accurately calculate the xG value is not the purpose. Therefore, in my project, I didn't make any effort on testing the accuracy of my xG result.

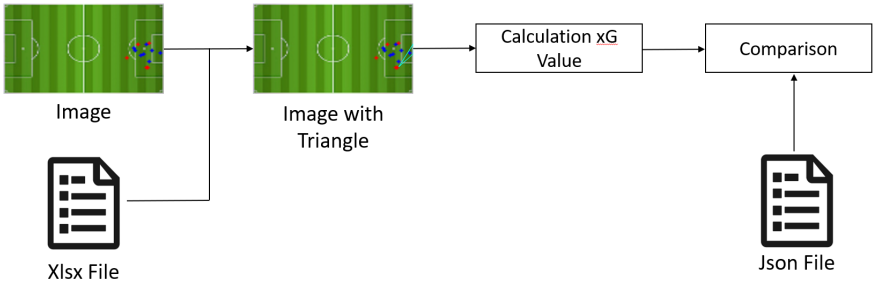

Figure 3.4: xG Calculation

3.4 Hardware and Software tools

In this project, coding is done by using VSCode as IDE. The programming language used in this project is Python. And I used GPU from my personal laptop for machine learning model training, which is RTX3060. The data from playmaker.ai company are stored on AWS platform, S3. While doing image processing and machine learning tasks, OpenCV and Pytorch are significant tools. In my thesis, I used OpenCV in 4.7.0 version and Pytorch in 1.13.0 version.

Chapter 4

Implementation and Results

4.1 Data Preprocessing

In this section, I wrote tools to preprocess the data from playmaker.ai. As it has been discussed above, these data consist of four parts: xlsx files, txt files, video files and json files. And implementation of preprocessing is shown in Fig. 4.1.

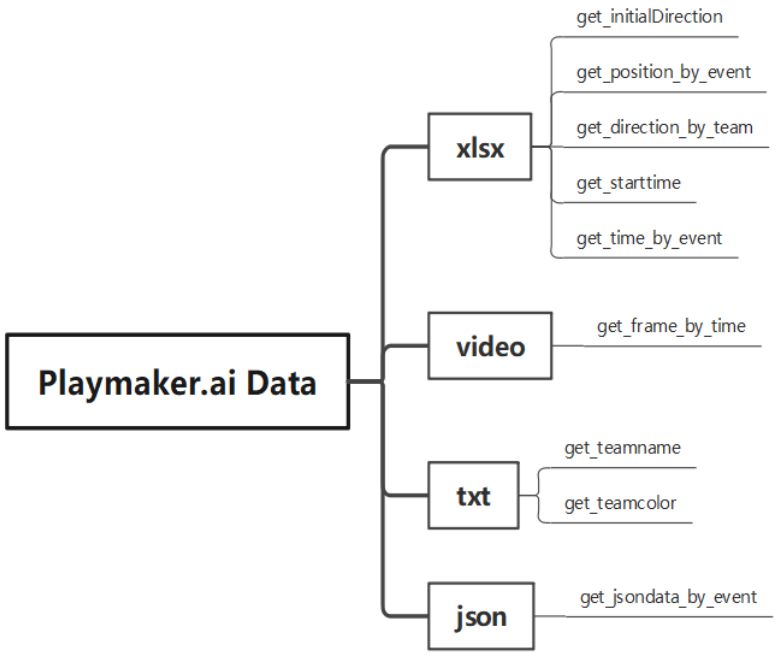

Figure 4.1: Data preprocessing

- xlsx files. A typical xlsx file snapshot is shown in Fig. 4.2

  - get_initialDirection. The initial attacking direction can be found by analyzing the first event in xlsx file. Given the fact that the first event is a soccer match is always a pass with passer and receiver position, this first event is analyzed and the initial attacking direction can be found. In this pass event, the passer always pass the ball the opposite direction of his attacking direction to one of his teammate. Therefore, in this program, I found the initial attacking direction by calculating the differences of passer and receiver position.
  - get_direction_by_team. In the last part, the same pass event also shows the passer's team name.
  - get_starttime. The start time of match in video file is always stored in the same position in xlsx file.
  - get_position_by_event. This program traverse all the information in the file and find all the position data of a single type of event.
  - get_time_by_event. This program traverse all the information in the file and find all the time data of a single type of event.

- txt files. A typical txt file snapshot is shown in Fig. 4.3.To find the teamname and teamcolor from txt file, I used regular expression. Specifically, I used 'team, (.*?) [J,j]erseys[)]' expression to find shirt color of both teams.

- video files. The purpose of this program is to find the exact frame according to time. The video from playmaker.ai has 50 frames for each seconds. Therefore, the frame number can be calculated. Then, I used openCV to extract the target frame as output.

- json files. A typical txt file snapshot is shown in Fig. 4.4. The xG value can be easily found after loading the json file.

4.2 Object Detection and Tracking

4.2.1 Configuration

In object detection part, the training parameters are shown in Fig. 4.5.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	Position	Duration	Result	Type	Pass Outcome	Player Name	Team Name	Field Position	Foot used	Receiver Name	Receiver Position
2	Pass	0:05:45	00:03.0	Successful	Short Pass	Backward Passes	8 Alexander Segor	IF Brommapojkarna	51;51	-	18 Jesper Arvidsson	31;33
3	Pass	0:05:48	00:03.0	Unsuccessful	Long Pass	Forward Pass	18 Jesper Arvidsson	IF Brommapojkarna	35;37	-	-	-
4	Defensive_Action	0:05:50	00:05.0	Successful	-	-	15 Oscar Krusnell	IK Frej Täby	74;88	-	-	-
5	Pass	0:06:00	00:03.0	Successful	Short Pass	Backward Passes	2 Jakob Glasberg	IF Brommapojkarna	77;98	-	8 Alexander Segor	72;84
6	Pass	0:06:00	00:03.0	Unsuccessful	Short Pass	Forward Pass	8 Alexander Segor	IF Brommapojkarna	74;85	-	-	-
7	Defensive_Action	0:06:00	00:05.0	Successful	-	-	7 Olle Edlund	IK Frej Täby	79;92	-	-	-
8	Pass	0:06:01	00:03.0	Successful	Short Pass	Forward Pass	7 Olle Edlund	IK Frej Täby	77;88	-	9 Leon Hien	66;86
9	Defensive_Action	0:06:03	00:05.0	Successful	-	-	6 Gustav Sandberg Magnusson	IF Brommapojkarna	62;85	-	-	-
10	Pass	0:06:04	00:03.0	Unsuccessful	Short Pass	Forward Pass	9 Leon Hien	IK Frej Täby	61;87	-	-	-
11	Pass	0:06:13	00:03.0	Successful	Short Pass	Backward Passes	15 Oscar Krusnell	IK Frej Täby	59;98	-	17 Ouattara Mohammed Aziz	75;87
12	Pass	0:06:17	00:03.0	Successful	Short Pass	Forward Pass	17 Ouattara Mohammed Aziz	IK Frej Täby	80;86	-	8 Almar Sher	79;54
13	Pass	0:06:24	00:03.0	Successful	Short Pass	Forward Pass	8 Almar Sher	IK Frej Täby	73;45	-	7 Olle Edlund	63;86
14	Pass	0:06:25	00:03.0	Successful	Short Pass	Backward Passes	7 Olle Edlund	IK Frej Täby	64;83	-	17 Ouattara Mohammed Aziz	76;72
15	Pass	0:06:28	00:03.0	Successful	Short Pass	Forward Pass	17 Ouattara Mohammed Aziz	IK Frej Täby	72;64	-	8 Almar Sher	70;48
16	Pass	0:06:31	00:03.0	Unsuccessful	Long Pass	Forward Pass	8 Almar Sher	IK Frej Täby	66;39	-	-	-
17	GK_Distribution	0:06:39	00:10.0	Successful	-	-	20 Alexander Lundin	IF Brommapojkarna	3;39	-	18 Jesper Arvidsson	5;27
18	Pass	0:06:45	00:03.0	Successful	Short Pass	Forward Pass	18 Jesper Arvidsson	IF Brommapojkarna	5;26	-	6 Gustav Sandberg Magnusson	23;28
19	Pass	0:06:47	00:03.0	Successful	Short Pass	Forward Pass	6 Gustav Sandberg Magnusson	IF Brommapojkarna	-	-	5 Daniel Svensson	25;4
20	Pass	0:06:49	00:03.0	Successful	Short Pass	Forward Pass	5 Daniel Svensson	IF Brommapojkarna	31;3	-	11 Oskar Fallenius	57;3
21	Defensive_Action	0:06:50	00:05.0	Successful	-	-	4 Kalle Björklund	IK Frej Täby	58;4	-	-	-
22	Pass	0:07:02	00:03.0	Successful	Short Pass	Forward Pass	5 Daniel Svensson	IF Brommapojkarna	58;2	-	11 Oskar Fallenius	66;10
23	Defensive_Action	0:07:02	00:05.0	Unsuccessful	-	-	10 André Alsanati	IK Frej Täby	64;8	-	-	-
24	Pass	0:07:02	00:03.0	Unsuccessful	Short Pass	Backward Passes	11 Oskar Fallenius	IF Brommapojkarna	65;17	-	-	-
25	Pass	0:07:11	00:03.0	Unsuccessful	Short Pass	Forward Pass	5 Daniel Svensson	IF Brommapojkarna	67;2	-	-	-
26	Lost_Possession	0:07:12	00:05.0	-	-	-	4 Kalle Björklund	IK Frej Täby	98;17	-	-	-

Figure 4.2: Xlsx Example

```
Teams:
IK Frej Täby (Home team, Yellow Jerseys)
IF Brommapojkarna(Away team, White jerseys)


Lineups:
IK Frej Täby
Formation: 0
4 Kalle Björklund
6 Axel Sjöberg
7 Olle Edlund 86
8 Aimar Sher
9 Leon Hien 74
10 André Alsanati
15 Oscar Krusnell
17 Ouattara Mohammed Aziz
18 Filston Mawana  74
27 Oliver Dovin
43 Abdul-Halik Hudu (K)
```

Figure 4.3: Txt Example


```
{
"xdest": null,
"xpos": 82,
"header": false,
"start_time": 47,
"game_time": 1,
"ypos": 67,
"xg": 0.043960234960137776,
"ydest": null,
"player": "Linus Pettersson Parnevik",
"penalty": false,
"end_time": 50,
"foot_used": "Left Foot",
"team": "T\u00e4by FK",
"action": "Shot on target",
"xp": null,
"throw_in": false,
"external_id": 28,
"goal_mouth": "-"
},
```


Figure 4.4: Json Example

Parameters	value	Parameters	value
Classes	2	Learning Rate	0.01
Weight Decay	0.0005	Image Size	640x640
Traning Rounds	80	Batch Size	8
Optimizer	SGD	IOU threshold	0.2

Figure 4.5: Object Detection Parameters

4.2.2 Results and Analysis

Firstly, I used the dataset with 1000 images, which 900 training images and 100 validation images. Fig. 4.6 shows the result of object detection model. Firstly, the loss curve provide us an overview of the training process. It can be observed that the loss decreases over epochs, and it shows that the model is well trained. We can claim that the training process is converging and it is reducing the differences between prediction and ground truth detection. We can also observe that percision and recall increases during the training process, which shows that the model is becoming more and more accurate in detection

task.

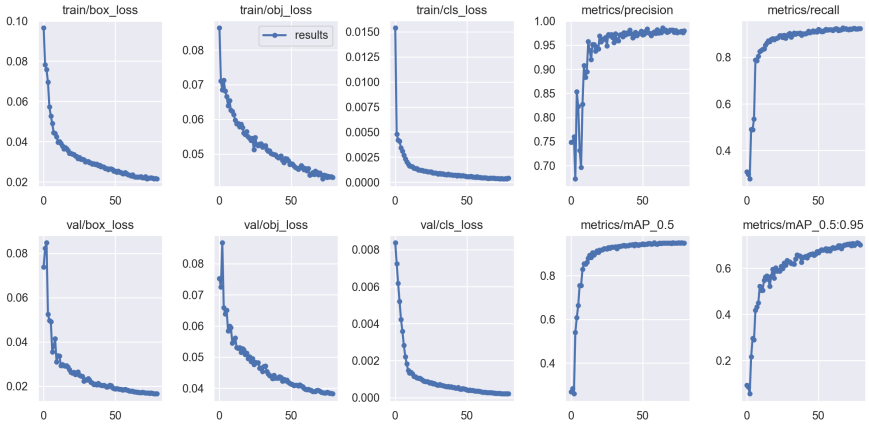

Figure 4.6: Object Detection Result

Finally, I used the trained object detection model onto the playmaker.ai video data, and the result is shown in Fig. 4.7. And it can observed that the model can successfully find the players on soccer field. And the result is visually acceptable for us to do further analysis. The model works correctly in most of the instance, but there are some wrong detection results. First, for example, the referee is also classified as a player. Also, on the top right of the Fig. 4.7, the caddy is detected as a player. I will further discuss related problems in Chapter 5.

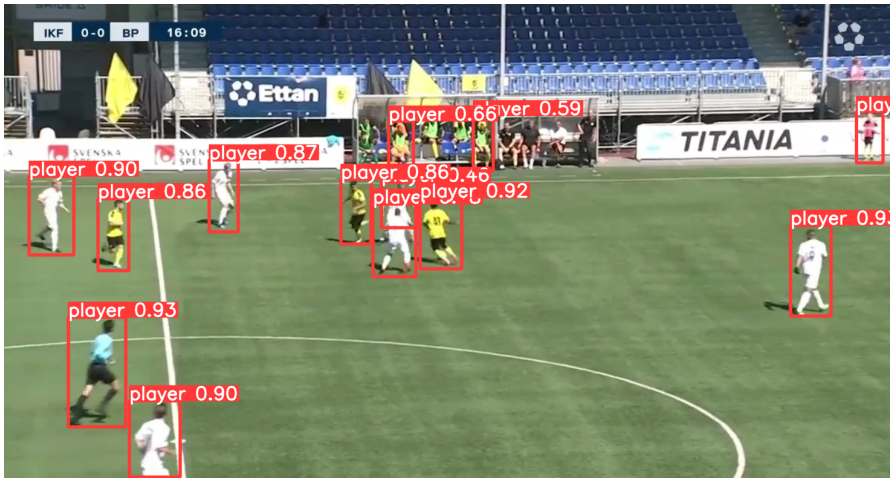

Figure 4.7: Object Detection Example

4.3 Team Detection

A typical team detection example is shown in Fig. 4.8. As it has been introduced in Section 3.3.4, the second largest cluster in K-means classification is chosen as our detected color. Then, based on the data preprocessing tools mentioned in Section 4.1, two color name from txt file can be found. These two colors are the ground truth shirt color of two teams. Then, I calculated to find the more similar color as the predicted color.

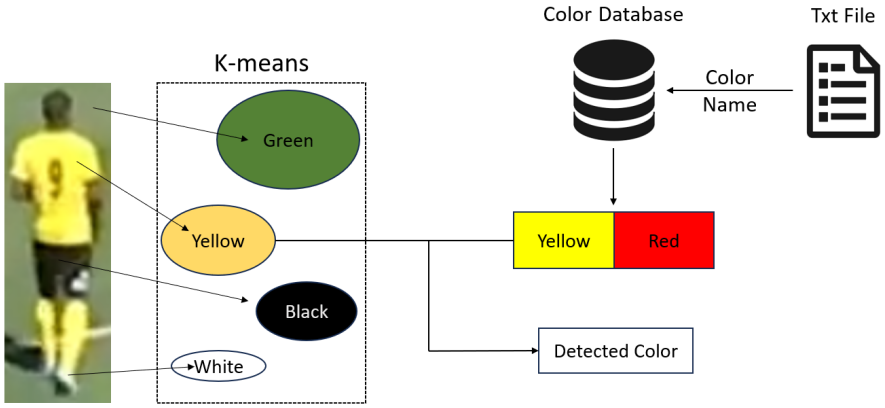

Figure 4.8: Team Detection Example

The color database consist of 9 colors shown in Fig. 4.9. These colors are manually settled to ensure the testing performance.

Color	RGB value	Color	RGB value
Yellow	(195, 195, 10)	White	(235,235,235)
Green	(10, 235, 10)	Blue	(10, 10, 235)
Black	(10, 10, 10)	Red	(245, 30, 30)
Purple	(160, 32, 240)	Orange	(235, 165, 30)
Dark	(85, 85, 85)		

Figure 4.9: Colors

To test the performance of color detection, I manually generate 4 test datasets with 26 test images from 13 matches in each dataset. And I choose 26

images because it includes all colors that appear in the playmaker.ai's dataset. And the average detection accuracy reaches 88.46%. We can claim that the testing performance is convincing enough. I will further discuss the problems in team detection in Chapter 5.

4.4 Perspective Transformation

The perspective transformation process is shown in Fig. 3.3. In the practical implementation, I used the pretrained two-GAN network and siamese network provided in [11]. In the practical implementation, the player position was defined based on the bounding box position. Specifically, the bottom middle point of the bounding box is defined as the player position in video file. In addition, to show the team detection result in this section, I use blue and red to identify the two teams. Two perspective transformation result example is shown in Fig. 4.10 and Fig. 4.11. From the result image you may have notice that all the players can be successfully detected and classified, which means the major goal of this project was met. You may also find that there are points on the edge of the Fig. 4.10. I will discuss these problems in the next chapter.

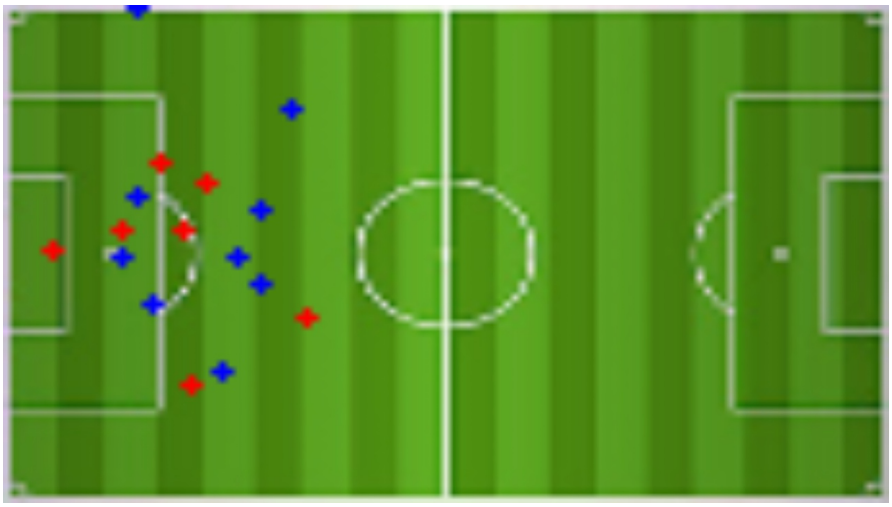

Figure 4.10: Perspective Transformation Result

4.5 xG Calculation

Figs. 4.13 and 4.12 show the result of xG calculation section. From the result image, it can noticed that the target triangle that is necessary for the xG value

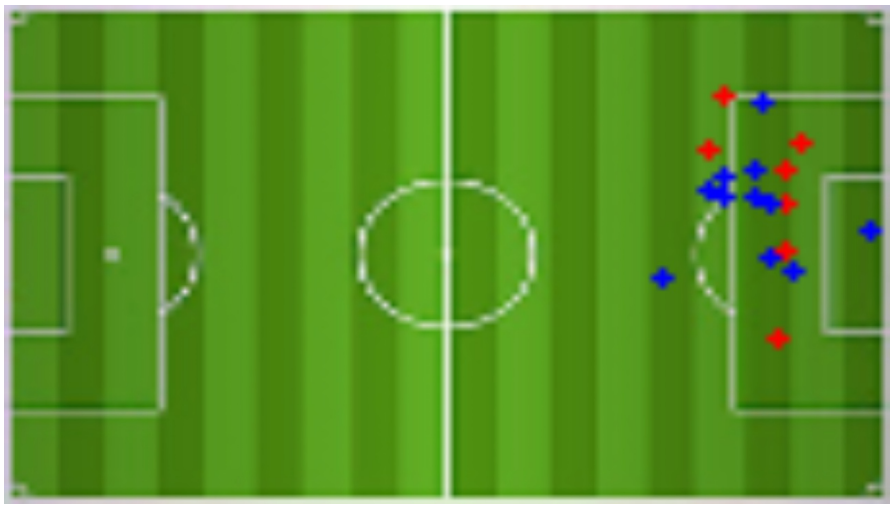

Figure 4.11: Perspective Transformation Result

calculation is correctly shown in the image. The triangle, that is highlighted in light blue color is used to show how many emeny players are located between the striker and the goal. And the three vertices of this triangles are position of striker and two goalposts. Therefore, these result can be used for the possible future work.

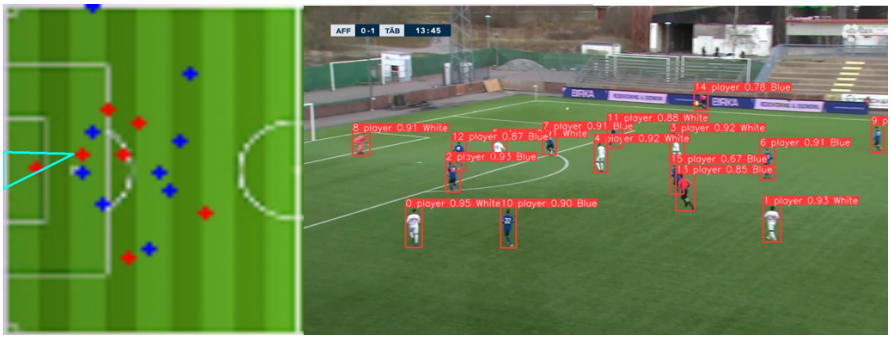

Figure 4.12: xG Calculation Example

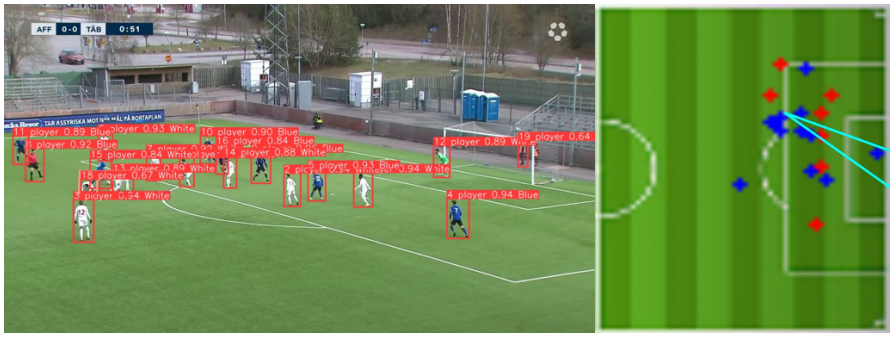

Figure 4.13: xG Calculation Example

Chapter 5

Discussion

In this chapter, I will discuss problems appeared in this project. And I will also give some possible solutions of them.

5.1 Object Detection

In object detection part, there are many existing problems. A typical example is shown in Fig. 5.1, and you may have notice that there are many problems in this example. Firstly, in the middle of Fig. 5.1, the two overlapped players cannot be correctly detected. The overlapping problems in object detection is challenging, however, a few strategies can be employed to mitigate the issue. For example, a different and more powerful dataset may help us to improve the performance facing this problem. Also, changing the training parameters including decreasing the IOU threshold can be a possible solution. In addition, related research including [37] can be implemented. Meanwhile, in Fig. 5.1, the system also detected the players outside the soccer field. It won't influence the final result considering that the player position outside the soccer field won't be included in the final 2-D map. However, the extra detection result will occupy calculation resources and be time-consuming.

Another problem in object detection part would be referee. Ideally, I hope that the referee can be ignored automatically by our system. But currently I can't do this. A possible solution of this problem could be implement in team color detection section. I can design a method that is only used for detecting referee by recognizing the shirt color. And I can implement it on every detected object to identify if he is a referee or not.

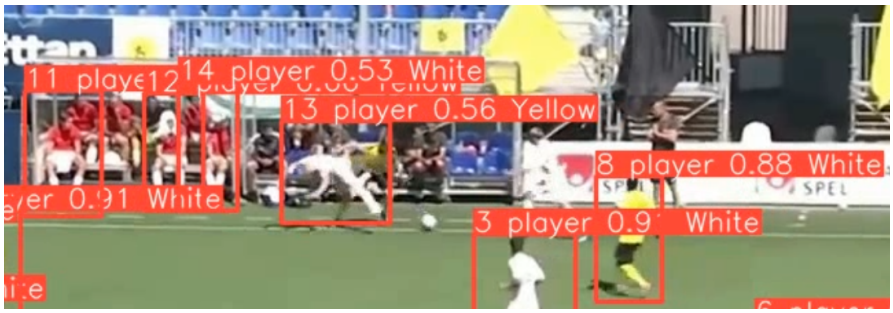

Figure 5.1: Object Detection Result with Problems

5.2 Team Color Detection

In the right side of Fig. 5.1, you may have noticed that the yellow team player is not correctly classified in team color detection process. The reason of this example is that there are plenty of interfering information in the background, which is a white advertising board in this case. The background color could influence the detection result in many different ways, especially when the player is close to edge of the field. Also, as it has been mentioned in the last section, the current system can't detect color of referee. The same problem remains for goalkeepers, who also have different colors.

In addition, problems of color pixels lead to many issues. For instance, some teams own shirt colors with both white and black. Clearly, the current method can't recognize it. Even though there are no green shirts in playmaker.ai company's data, it is possible that there will be green shirt color in other soccer league or in the future. For now, green color can't be identified theoretically. Another big issue relies in the video record. In this project, I used the static value as the ground truth shirt color, which is shown in Fig. 4.9. However, the shirt color in the video would be different from match to match. It could be influenced by various factor, including weather condition, camera angle, match time and so on. It could be even different in the same match given the fact that some soccer fields have ceiling. It means that a more robust detecting method or design specific methods can be very helpful for all these problems.

5.3 Perspective Transformation

An obvious problem in this section comes from company's data. There are many mismatched data in json and xlsx file. For example, the information of

the same event is different or the event in json file can't be found in xlsx file. In addition, the data from the company are not completely accurate, and it caused slight inaccurate result. For example, the player position on x and y axis has been normalized to a range of 0 to 100. However, the standard football field size is 105 meters times 68 meters. Therefore, the final detection result may have error about 1 meter. In this project, I assume that it won't influence the final result.

Chapter 6

Conclusions and Future work

In this chapter, I gave conclusion of the whole degree project. And I also discussed limitation of this project. Then, I introduced some possible directions of work for future researchers. Finally, I presented reflection of my work.

6.1 Conclusions

In this project, I designed a complete system to solve the practical problem in soccer data analysis based on machine learning and computer vision method. Specifically, the system solve the problem of analyzing camera view video. Firstly, I divided the problem to several subproblems. Then I gave solution for all these subproblems. From the result, we can conclude that this project successfully gave a solution of the practical problem. In summary, the purposes and goals of this project are accomplished. Admittedly, there are many existing problems in the system, and it could be done by future researchers.

6.2 Limitations

In this project, there are also limitations that need to be clarified. Firstly, the data provided by playmaker.ai company lacks ground truth value in each necessary steps of the project. For example, in the object detection project, ideally, there should be the detection result, which is the bounding box labels, in video frame data. And in perspective transformation part, ground truth value of the resulting 2-D map should be known. However, currently the

playmaker.ai company can't provide these data. Therefore, in these part of the project, it is hard to quantitatively evaluate the performance of our system. In this project, I evaluate the performance by simply observing the result many times. Secondly, in Section 4.5, I used a simple empirical method to xG calculation method. Clearly, this method is not logically rigorous. However, figuring out a robust method of xG calculation requires not only the method of mathematics but also knowledge in soccer. As a student major in information and network engineering, it is obvious that xG calculation method is not the work in my scope. Finally, all the methods in this project are designed only for playmaker.ai data. For other data structure of soccer videos, the system, especially the data preprocessing part, probably needs to be rewritten.

6.3 Future work

In Chapter 5, I have already mentioned many problems and some possible solutions. Due to the time limitation of this project, this hasn't been implemented. The first work that could be done is to solve these problems. I also mentioned some limitation in the last chapter, and future work can also be found to solve these limitations. For example, future researchers can collect ground truth data of topics in this project. Therefore, there could be quantitative indicators to evaluate the performance of each methods. Some other tools could also be written for the future work in playmaker.ai company. For example, the event detection could be very useful for the company.

6.4 Reflections

As it has been mentioned in Section 1.3, by using this system, the company can solve the practical problem of camera view video and provide services for coaches and soccer players. Admittedly, there are many problems in current system. The result of this degree project can be used as the prototype of a complete robust system of soccer data analysis.

References

[1] A. Rathke, "An examination of expected goals and shot efficiency in soccer," *Journal of Human Sport and Exercise*, vol. 12, no. 2, pp. 514–529, 2017. [Page 1.]

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. [Pages 4 and 11.]

[3] J. Komorowski, G. Kurzejamski, and G. Sarwas, "Footandball: Integrated player and ball detector," *arXiv preprint arXiv:1912.05445*, 2019. [Page 4.]

[4] K. Lu, J. Chen, J. J. Little, and H. He, "Light cascaded convolutional neural networks for accurate player detection," *arXiv preprint arXiv:1709.10230*, 2017. [Page 4.]

[5] Y. Zhang, Z. Chen, and B. Wei, "A sport athlete object tracking based on deep sort and yolo v4 in case of camera movement," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020. doi: 10.1109/ICCC51575.2020.9345010 pp. 1312–1316. [Page 4.]

[6] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, 2023. [Pages 4 and 13.]

[7] G. F. Welch, "Kalman filter," *Computer Vision: A Reference Guide*, pp. 1–3, 2020. [Page 4.]

[8] N. M. Al-Shakarji, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Multi-object tracking cascade with multi-step data association and occlusion handling," in *2018 15th IEEE International Conference on*

Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2018, pp. 1–6. [Page 5.]

[9] H.-H. Bock, "Clustering methods: a history of k-means algorithms," *Selected contributions in data analysis and classification*, pp. 161–172, 2007. [Pages 5 and 15.]

[10] M. Istasse, J. Moreau, and C. De Vleeschouwer, "Associative embedding for team discrimination," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. [Page 5.]

[11] J. Chen and J. J. Little, "Sports camera calibration via synthetic data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019. [Pages 5, 16, 20, 24, and 32.]

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. [Page 7.]

[13] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323. [Online]. Available: https://proceedings.mlr.press/v15/glorot11a.html [Page 8.]

[14] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016. [Online]. Available: http://arxiv.org/abs/1609.04747 [Page 10.]

[15] S. Xie, A. Kirillov, R. Girshick, and K. He, "Exploring randomly wired neural networks for image recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. [Page 11.]

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html [Page 11.]

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986. [Online]. Available: https://api.semanticscholar.org/CorpusID:205001834 [Page 11.]

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012. [Page 11.]

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. [Page 11.]

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. [Pages 11 and 14.]

[21] A. Briot, P. Viswanath, and S. Yogamani, "Analysis of efficient cnn design techniques for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 663–672. [Page 11.]

[22] S. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot… for now," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2020. doi: 10.1109/CVPR42600.2020.00872 pp. 8692–8701. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00872 [Page 11.]

[23] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934 [Page 12.]

[24] K. J. Oguine, O. C. Oguine, and H. I. Bisallah, "Yolo v3: Visual and real-time object detection model for smart surveillance systems(3s)," 2022. [Page 12.]

[25] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *CoRR*, vol. abs/1908.08681, 2019. [Online]. Available: http://arxiv.org/abs/1908.08681 [Page 12.]

[26] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu, "A strong baseline and batch normalization neck for deep person re-identification," *CoRR*, vol. abs/1906.08332, 2019. [Online]. Available: http://arxiv.org/abs/1906.08332 [Page 14.]

[27] G. D. Evangelidis and E. Z. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008. doi: 10.1109/TPAMI.2008.113 [Page 14.]

[28] Z. Duan, C. Han, and H. Dang, "An adaptive kalman filter with dynamic rescaling of process noise," in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 2, 2003, pp. 1310–1315. [Page 14.]

[29] Y. Du, J.-J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021," *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 2809–2819, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244531248 [Page 14.]

[30] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995. [Page 14.]

[31] D. Stadler and J. Beyerer, "Modelling ambiguous assignments for multi-person tracking in crowds," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 133–142. [Page 14.]

[32] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999. [Page 14.]

[33] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018. [Page 16.]

[34] R. A. Sharma, B. Bhat, V. Gandhi, and C. V. Jawahar, "Automated top view registration of broadcast football videos," *CoRR*, vol. abs/1703.01437, 2017. [Online]. Available: http://arxiv.org/abs/1703.01437 [Page 16.]

[35] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006. doi: 10.1109/CVPR.2006.100 pp. 1735–1742. [Page 17.]

[36] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3702–3712. [Page 23.]

[37] H. Xu, X. Wang, D. Wang, B. Duan, and T. Rui, "Object detection in crowded scenes via joint prediction," *Defence Technology*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:240263788 [Page 35.]

TRITA-EECS-EX-2024:117