The BRITISH UNIVERSITY IN EGYPT

Faculty of Informatics and Computer Science

Software Engineering

Analyzing Football Players Performance

By: Malek Tarek Ahmed

Supervised By

Prof. Ann Nosseir

June 2023

Abstract

Having the proper tracking of the players is necessary to evaluate them properly. This can be done by continuously monitoring their movement and speed. In addition to this, it is also important to collect other data such as the average speed and the moving distance. Most of the time, the analysis of the movements and events of players in a game is carried out by professional analysts. Physical examinations of players are conducted to improve the recognition of motion effects using image recognition technology and artificial intelligence. So instead of hiring endless professional analysts and spending unnecessary expenses over their salaries, this project aiming to develop a tool which will help to detect and monitor athletes' movements throughout their matches which will support the process of improving their performance. The tool accomplishes this by tracking players in a video and computing their speed, max speed, min speed, and total distance travelled. The output of the tool then displays the player's movement throughout the video as a heat map and GIF. It has been demonstrated that the tool is useful in giving players direct feedback on their performance. It was able to identify the areas in which players needed to advance. The tool is beneficial to both coaches and players. Players can use it to track their own progress and find areas where they can improve, and coaches can use it to recognize players' areas of improvement.

Acknowledge

I want to thank my supervisor, Dr. Ann Nosseir, for her invaluable contribution, unwavering support, and constant inspiration throughout my journey. Her advice and expertise have been invaluable in shaping my academic and professional development. Dr. Ann dedication, passion, and upbeat attitude not only inspired me to push my limits, but also instilled in me a strong sense of commitment and excellence. I am eternally grateful for her mentorship and the invaluable knowledge and skills I have gained while working with her. I consider myself extremely fortunate to have had the opportunity to work with such an exceptional supervisor, and I am grateful for her ongoing encouragement and support.

Table of Contents

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

Analysis of the data is one of the daily tasks of academics. It's not a huge deal for them to read hundreds of pages per day to extract relevant information. However, the amount of data available has increased dramatically in recent years. While it's great news for researchers to have access to more data, which could lead to better studies, it's also a bit of a pain. To index action-based sports videos and provide kinematic measurements for coach assistance and performance enhancement, the project introduces a system for automatically detecting and analysing complex player actions in moving background sports video sequences. The process operates in a coarse-to-fine manner. For an input video, we automatically segment the highlights, or the video clips that contain the desired action, as summaries for general user viewing purposes. In the middle granularity level, we identify the action types to support action-based video indexing and retrieval. Finally, in the fine granularity level, the critical kinematic parameters of player action are obtained for sports professionals' training purposes.

1.2 Problem Statement

Previously, coaches would typically provide verbal feedback to players following a game or practises. This feedback may be useful, but it may also be subjective and difficult to recall. A coach may say, for example, "You need to run faster" or "You need to pass the ball more." These statements, however, are not very specific and may be difficult for players to understand. Furthermore, verbal feedback is easily forgotten, especially if it is not given immediately after the game or practise.

1.3 Scope and Objectives

The concept is to create a computer vision system that detects and tracks football players in real-time or recorded video feeds. To accurately identify players on the pitch, the system will use object detection algorithms and techniques. The project will also include extracting useful information from the tracked players, such as their position, speed, and movement patterns. Furthermore, the collected data will be analysed by the system to provide performance insights and statistics for individual. The goal is to develop a comprehensive tool to help football players evaluate their performance, identify areas for improvement, and make informed decisions to improve their gameplay.

1.4 Report Organization (Structure)

Section 2 of the paper provides a comprehensive review of related works in the field. It examines and analyzes a number of existing studies, research papers, and projects that are relevant to the topic of the paper. In Section 3, the methodology of the project is discussed in detail. This section outlines the approach and techniques used to develop the proposed solution. It provides an overview of the steps taken, the tools and technologies employed, and the overall framework of the project. Section 4 focuses on the implementation of the model and provides a detailed description of the code and algorithms used. It highlights the technical aspects of translating the proposed solution into a functioning system. In Section 5, various scenarios are tested using the developed model, and the evaluation results are presented. Section 6 compares the results of the developed model to other existing models or approaches in the field. It presents a comparative analysis of the performance, accuracy, and efficiency of the proposed model in relation to other relevant models. Section 7 provides a summary of the findings and achievements of the project. It offers a concise overview of the key points discussed throughout the paper, emphasizing the significance of the proposed solution and its potential impact.

1.5 Work Methodology

1. Find what most popular sports people interact with which is football.
2. Find reason why players performance is not as people point of view.
3. Find more about how to improve players performance.
4. Start writing the code.
5. Test the code to find the bugs in it.
6. Fix the bugs and fix any problems in the logic.
7. Retest the code with different videos on different players.
8. Write the final report.

1.6 Work Plan (Gantt chart)

	Task Mode	Task Name	Duration	Start	Finish
1		**Graduation Project**	**41 days**	**Sat 10/1/22**	**Fri 11/25/22**
2		Idea Selection	3 days	Sat 10/1/22	Tue 10/4/22
3		Idea approval	1 day	Tue 10/11/22	Tue 10/11/22
4		Gather research papers about motion detection	10 days	Wed 10/12/22	Tue 10/25/22
5		Write the abstract	1 day	Fri 11/18/22	Fri 11/18/22
6		Analyze past papers	10 days	Fri 10/21/22	Thu 11/3/22
7		Studing related work	6 days	Sat 10/29/22	Fri 11/4/22
8		Determine Problem	2 days	Fri 11/4/22	Mon 11/7/22
9		Determine Algorithms	10 days	Mon 11/7/22	Fri 11/18/22
10		Find relevent solution for the problem	5 days	Wed 11/9/22	Tue 11/15/22
11		Find out more about the most suitable dataset	6 days	Fri 11/11/22	Fri 11/18/22
12		Write the interim report	6 days	Wed 11/16/22	Wed 11/23/22
13		Revise the interim report	3 days	Wed 11/23/22	Fri 11/25/22
14		Submit the interim report	1 day	Fri 11/25/22	Fri 11/25/22

Figure 1

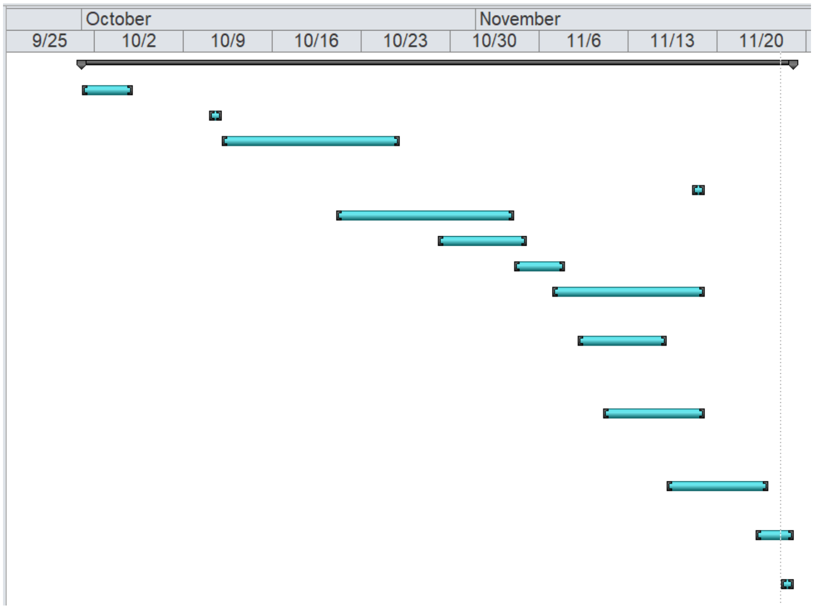


Figure 2

		Task Mode	Task Name	Duration	Start	Finish
15			**Implementation**	**72 days**	**Sat 2/25/23**	**Tue 6/6/23**
16			Gather Videos	5 days	Sat 2/25/23	Thu 3/2/23
17			Implement The Dataset	3 days	Fri 3/17/23	Tue 3/21/23
18			write the code	20 days	Sun 4/2/23	Thu 4/27/23
19			Test the code	3 days	Mon 5/1/23	Wed 5/3/23
20			Fix the Errors	5 days	Wed 5/3/23	Tue 5/9/23
21			Test final version of the code	5 days	Wed 5/31/23	Tue 6/6/23
22			Write the final reprot	7 days	Thu 6/1/23	Fri 6/9/23
23			Prepare the Presentation	1 day	Wed 6/7/23	Wed 6/7/23
24			Submit the Graduation Project	1 day	Sat 6/10/23	Sat 6/10/23

Figure 3

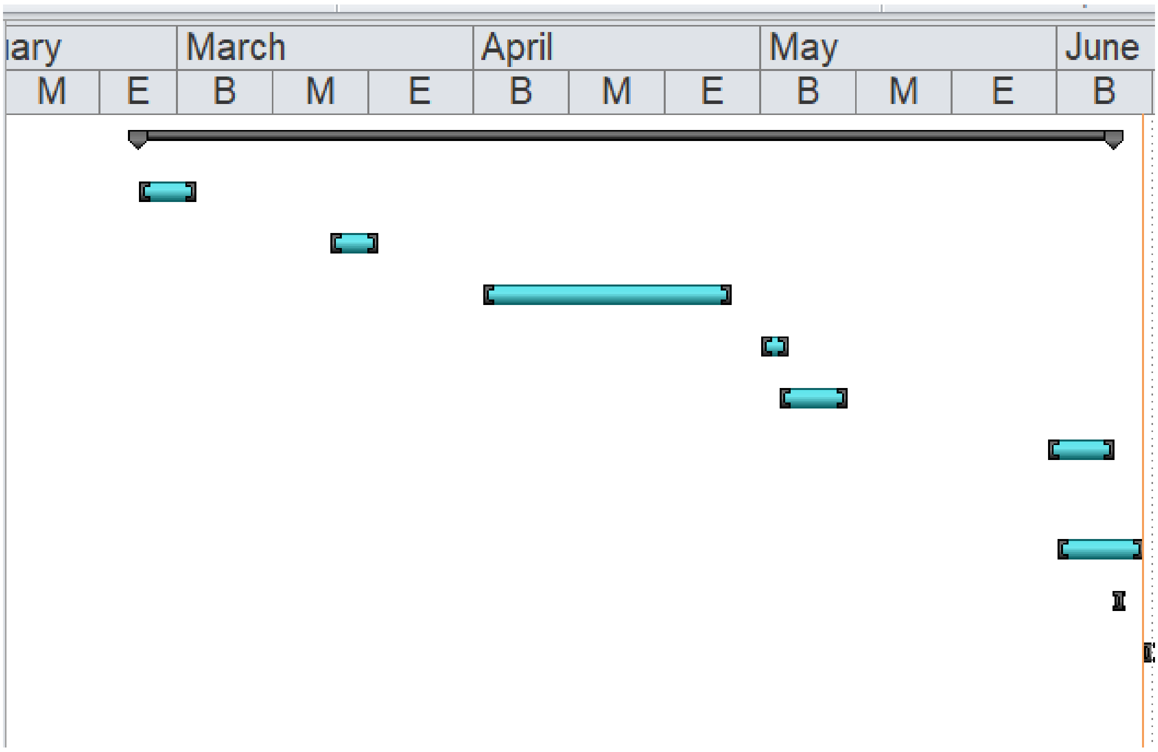


Figure 4

2 Related Work (State-of-The-Art)

2.1 Background

We present a study of using a location-aware wireless sensor network system to monitor the movements of sports team members and collect sensor data. The development of a set of system design specifications that address the hardware design of mobile nodes, the design of the sensor network system, the design of the position algorithm, the design of communication protocols, and the design of a testbed for positioning accuracy testing. We suggest a system architecture based on the field tests that satisfies these demands. The suggested architecture uses data aggregation for energy efficiency and hybrid location technologies for better accuracy.

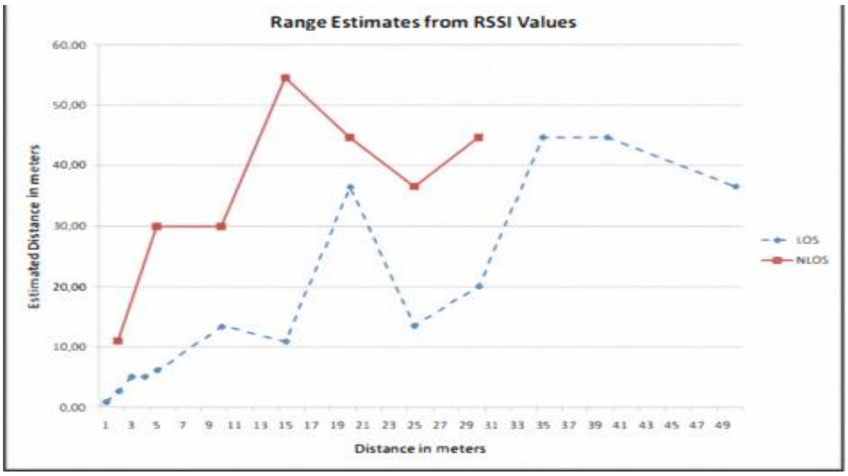


Figure 5

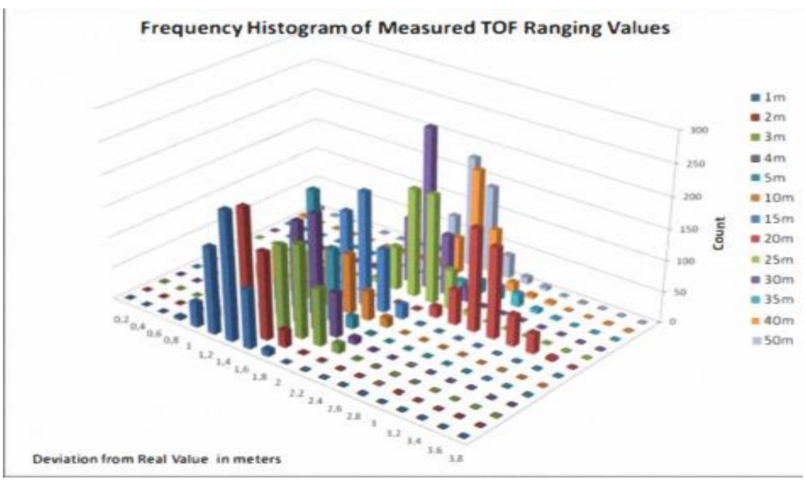


Figure 6- Frequency of measured range values [16]

High end applications in the fields of pharmaceuticals, robotics, satellite image processing, genetics, etc. have been made possible thanks to research in digital image processing and related fields. Image processing has many uses in daily life applications that revolve around people. As a result, it is crucial to find human bodies in real time, and video sequences have been entered.

In this paper, a bottom-up methodology for automatic human body detection and extraction from single images is proposed. This paper's work is creating a hybrid algorithm for extracting human bodies from various images and currencies and detecting them.

Figure 6

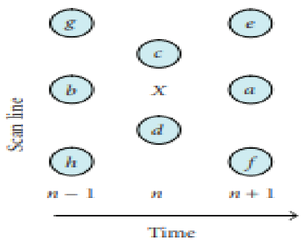
Identification of motion scenes and intra-interpolation are the objectives of motion detection. We use a hybrid motion detector (HMD) that only needs pixel information from three fields. The HMD's pseudo-codes are displayed in Figure.

The three conditions are used to identify motion with edges, fast motion, and slow motion.

```
diff1 = abs(a − b)
diff2 = abs[b − (c + d)/2]
diff3 = abs[b − (g + h)/2]
diff4 = abs[a + (e + f)/2 − b − (g + h)/2]

if diff1 > TH1                              ▷ 1st condition
  flag ⟵ motion
else if diff2 > TH1 AND diff3 < TH2         ▷ 2nd condition
  flag ⟵ motion
else if diff4 > TH3                         ▷ 3rd condition
  flag ⟵ motion
else
  flag ⟵ stationary
```

(a)



(b)

FIGURE 2: The proposed hybrid motion detection algorithm. (a) Pseudocodes, (b) pixel definition.

Figure 7

In this Fig Error in mean for TOF and RSSI ranging measurements for different distances is shown. Results are presented for LOS and NLOS conditions with normal and fast ranging modes.

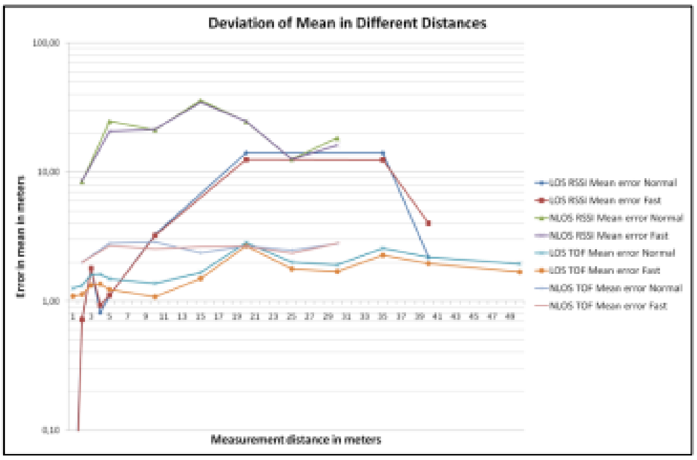


Figure 5. Error of mean for TOF and RSSI range measurements at different distances

Figure 8

For location algorithm development purposes, we measured standard deviation for raw location accuracy. We used the well-known trilateration method [8], without any filtering and with bias errors for ranges. Only the three shortest TOF ranging distances were used for estimating the location. Measurements were done on a parking lot, at a square area with size of 20 meters. The anchors were places at comers at the height of 1.7 meters from the ground. A rotator device, described in next section, was used to move tags in circular path with radius of 5 meters. In the measurements the rotator was placed in two different locations and two different speeds, 11 and 16 kmlh, for tags were used. The measurement area enclosed a streetlamp inside it. This can be thought as a simulating a goal frame of a real sports arena. Fig. 7 shows some experimental results. Some errors can be seen in the direction where the streetlamp was indicating that filtering for data should be done.

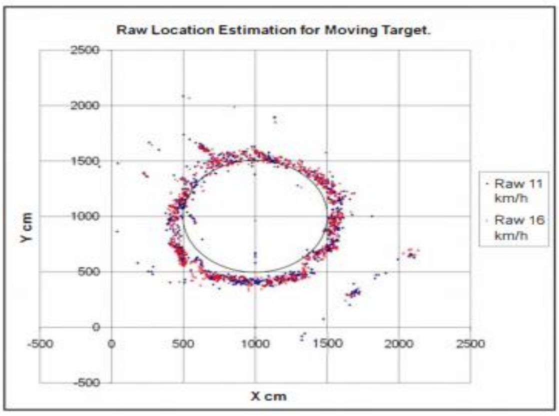


Figure 7. Raw location estimation from TOF ranging measurements.

Figure 9

This section analyses a three-dimensional movement using the image plane as a projection of the visible points. Estimation of the instantaneous variation of a position in a sequence of images is commonly referred to as an optic flow field or a velocity field. Calculation techniques for optical flow Gradient, energy approaches, coordinating classification based on a point as well as phases, and dynamic neuron are the five categories that IPAIT fields are typically divided into. The analysis of sports motion is shown in this figure. The system records a sportsperson's movement using cameras and wearable sensors. Since the acquisition of images and sensor data is synchronised, it is possible to determine their temporal mappings. The system automatically anticipates and segments the movements according to the technique when an athlete performs a sports movement that needs to be examined. The system organises the camera images into categories before delivering the labelled images for segmentation and temporal mapping analysis. The importance of the grey image is used in the gradient-based method to measure the optical flux field. The IPAIT constraint equation is derived for analysis, and it is expected that the grey area before and after the motion picture will remain unchanged. However, since the optical flow is not defined by the IPAIT equation, additional constraints are necessary, as shown in Fig. Global and local restriction methods can be used to categorise gradient-based approaches into two categories. The standard algorithms Hom-Schunck and Lucas-Kanada have significantly increased their precision speed and robust anti-noise capacity.

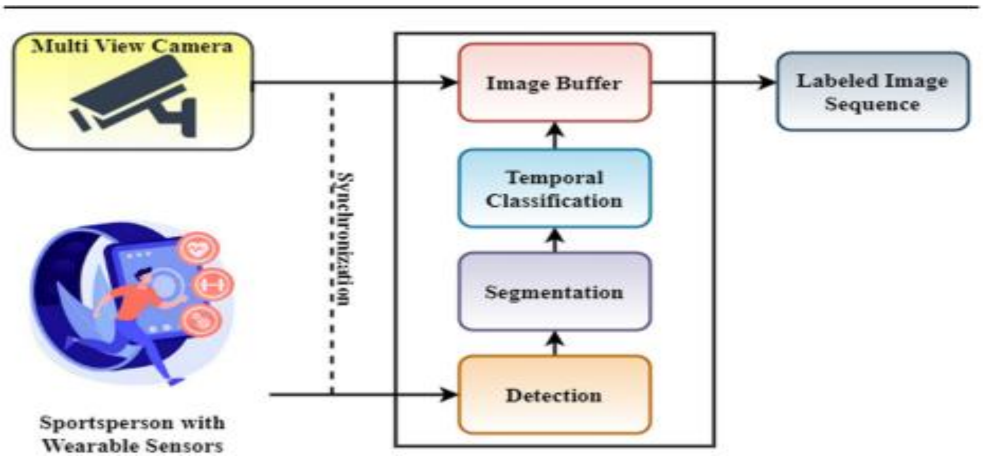


Figure 10

Sports players cannot be tracked by networks that are primarily trained on city dwellers or pedestrians [8]. Based on the findings of learning city-person and pedestrian by including learning sports videos, we implemented the network for tracking players in this paper. Each object was trained using 760 randomly selected continuous frames taken from a KIMCHI BALL (American football) video. The following environment was used to conduct the experiment.

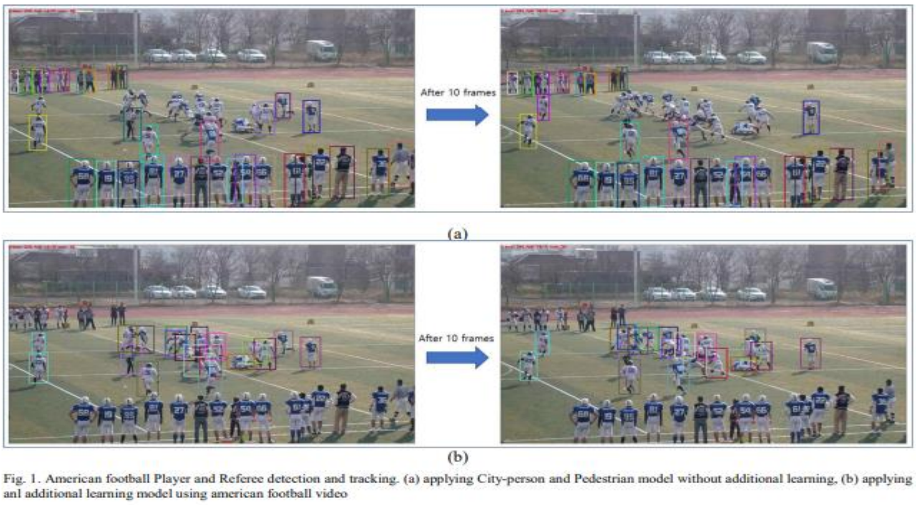


Figure 11

F American football Player and Referee detection and tracking. (a) applying City-person and Pedestrian model without additional learning [1]

And I trained the model that had mastered the KIMCHI-BALL video how to play general soccer. The results of tracking soccer objects are shown in the below Fig. before and after additional soccer video training on the model that learned the KIMCHI-BALL.As you can see in Fig. below, we were able to confirm that applying the model that was additionally learned from soccer video to the soccer video significantly improves the results of tracking soccer objects when compared to applying the model that was only learned from American football video

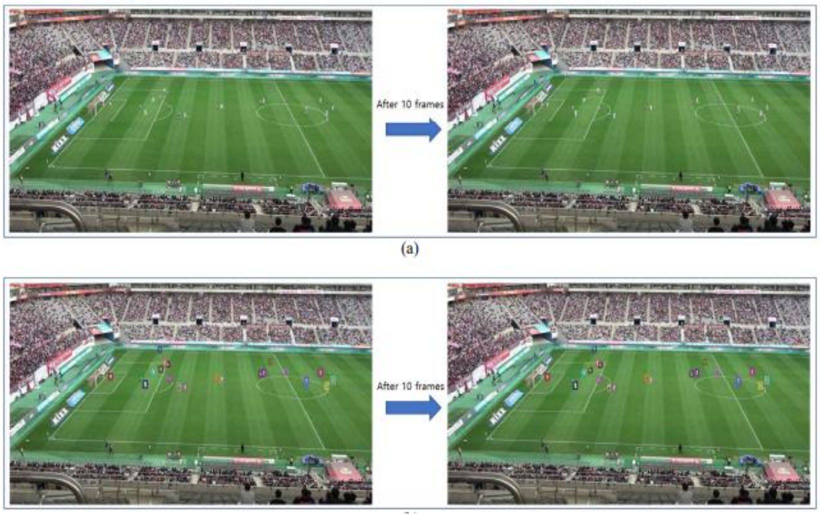


Figure 12- ***Soccer player and referee detection and tracking [1]***

It is difficult to analyse continuous video because it requires specialised software that can accept video as input. As an alternative, software called Virtual dub has been used to extract image frames from videos. This software offers users a range of options, allowing them to either select an interesting scene from the video or extract an image of the entire video. From each video, only 100 image frames were chosen for labelling. The badminton players are labelled with a square box using the Training Image Labeller in the MATLAB Application, regardless of the referee and spectators, as shown in Figure.

Figure 13 ***Labelled image from broadcast video [2]***

The above trained models were then put to the test using a variety of testing videos combined in accordance with a table to gauge how well the detector could track the player's position. After the testing is complete, an image frame with a square box shows the results, showing how confidently the player detector could have been produced. The precision recall graphs were created prior to computing the average precision in order to analyse the detector's performance in each case.

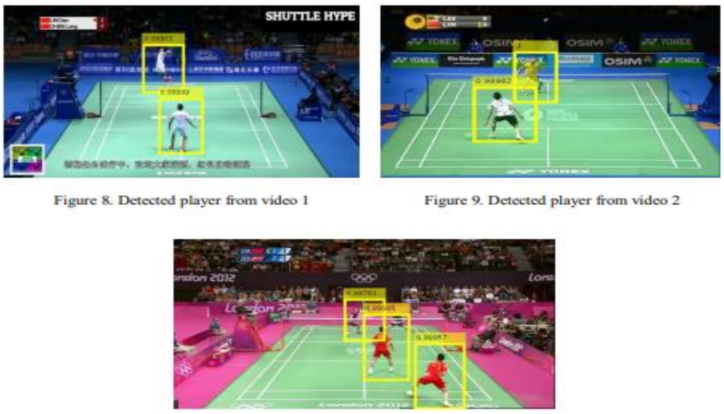


Figure 14 ***Detected player from video [1]***

While doing a quantitative analysis of the findings, we will first show how various methods for player detection work. In this figure illustrates the camera 1 view for each technique. While camera 2 view for all methods is shown in Fig. 10. It has been noted that BS typically treats multiple players as a single player. A large number of players are missed by DPM+LSVM in a frame. Both HD+SVM and HOG+SVM consistently deliver strong player detection results. There are some false detections in Gray+CNN and RGB+CNN (lines are sometimes mistaken for players, and sometimes one player is mistaken for two separate players). In comparison to Gray+CNN and RGB+CNN, both SIM+CNN and PSIM+CNN have accurate player detections and better fit the player detection window.

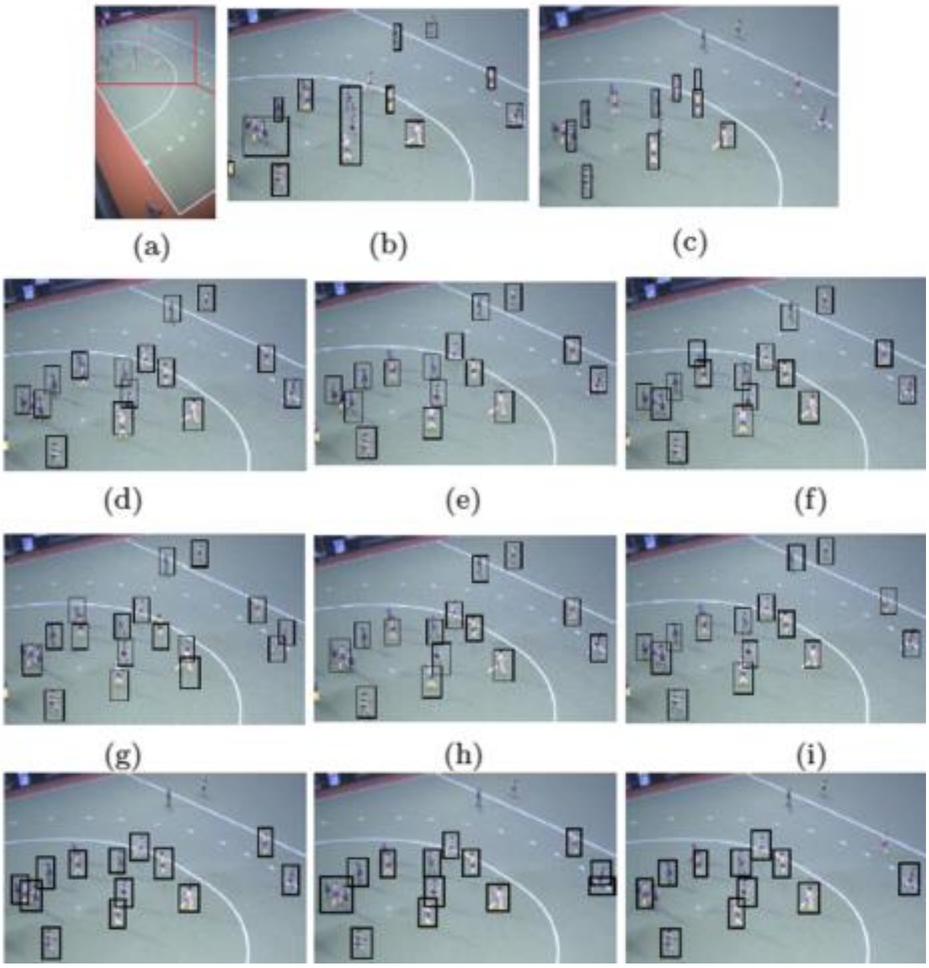


Figure 15 ***Team Detection***

2.2 Literature Survey

The table below provides a summary of all related works, including the algorithms used, datasets used (if any), and evaluations of the experiments and work performed in each paper.

Number	Algorithm	Data Set	Evaluation
[1]	• MEDIANFLOW • MIL (Multi Instance Learning) • KCF (Kernelized Correlation Filters)	N/A	We proposed a model that shows good performance by additional learning of players and referees based on the model of learning City person and Pedestrian for detection and tracking of players and referees.
[2]	• R-CNN	Type of trained R-CNN detector model	A new technique for automatic player detection from broadcast video via Faster R-CNN.
[3]	• Robust algorithm • GME algorithm • Object segmentation algorithm	Action Recognition Results	The extensive experiments show the effectiveness of the proposed system. However, there is still room for improvement. In the highlight detection, the caption in video that indicates

			the player profile when he/she is preparing for play is an important cue to identify an action, and we expect an improvement by integrating such information in the future.
[4]	• AdaBoost algorithm	N/A	We perform the distillation in an online fashion, i.e., our student is continuously trained to adapt to the latest video conditions, while performing the player detection in real-time. We show that our system can accurately detect players both inside and outside the common field of view, thanks to our custom supervision.
[5]	• The edge-pattern recognition algorithm • hybrid motion-adaptive deinterlacing	PSNR of the deinterlacing algorithms in db.	We compare our deinterlacing algorithm to six algorithms, including two recently published algorithms with 4- field motion

	algorithm (HMDEPR), • Deinterlacing algorithm • motion-adaptive deinterlacing algorithm • Field motion adaptive algorithm		detection. The PSNR of our deinterlacer on versatile sequences demonstrates higher robustness than the other motion-adaptive algorithms. Moreover, with better performance than the 4-field motion-adaptive algorithms, our algorithm only needs the data of three fields.
[6]	• AdaBoost algorithm • Non-linear regression algorithm • The local Lucas algorithm • Machine learning (ML) algorithms	N/A	
[7]	• CNN Algorithm R-CNN	Conventional methods of HOG+SVM and HD+SVM achieve the best results on both datasets	We evaluate performances of recent and well-known methods for player detection in field sport.
[8]	• Minimum distance algorithm • ICP algorithm	N/A	

[9]	• Unnamed algorithm	N/A	Requires the right choice of methods as well as a good understanding of the game. So far, we have succeeded in using the STATS dataset to obtain a reasonably human-like moving and passing behavior of a player controlling the ball.
[10]	• Hungarian algorithm • Crowd tracking algorithms	basketball dataset of 4 sequences for a total length of more than 5 minutes.	We validate our approach on 30 minutes of international field hockey and 10 minutes of college basketball. In both sports, motion models conditioned on game context features consistently improve tracking results by more than 10%
[11]	• Region Compensation Algorithm (RCA)	N/A	An accurate background can be reconstructed by using our algorithm; hence, player detection and segmentation can be performed precisely.

			The algorithm assumes that there is no shadow of buildings and of players in the soccer image sequences
[12]	• Data pre-processing algorithm	comparisons are possible thanks to several benchmark datasets, such as MSR-Action3D dataset, UT Kinect dataset or Florence dataset	The movement to be detected is modelled by a convex formulation of the state models obtained from the dataset, leading to a similarity index of the actual movement with the learning models
[13]	• Patch updating algorithm. • The baseline algorithm	N/A	The proposed method was compared with the baseline algorithm that uses the object patch as an observation method. The results showed the superiority of the proposed method.
[14]	• Target detection algorithm • Greedy algorithm	N/A	The experimental results show that the algorithm has a detection rate higher than 80% and tracking rate, which saves time and meets the real-

			time requirements of the system
[15]	• Tracking algorithms • STAPLE algorithm	LIST OF THE 11 ATTRIBUTES THAT HAVE BEEN ANNOTATED TO TEST SEQUENCES.	The test is done by running all these trackers on two soccer videos used from two publicly available datasets (T-Color-128 and DTB). The accuracy and variable reliability of the labels are often unknown
[16]	• Location algorithm • IMU Based Location Algorithm • Hybrid Algorithm • Ranging algorithm	N/A	Ranging accuracy measurements indicate that when anchor height is set to 1.7 meters from ground, possibly multipath fading occurs near 20 meters, causing lots of uncertainty to measurements. By combining TOF and RSSI result we have shown that even poor RSSI condition gives more accuracy, if standard deviation is known.
[17]	• The Player Tracking Algorithm	Accuracy of recognizing jersey	Our solution has shown some

	• Eigen-centrality algorithm • Node Rank algorithm • Tracking algorithm • K-means clustering algorithm	numbers and players	shortcomings in terms of accuracy due to inherent limitation of current deep learning algorithms that are not entirely error-free.
[18]	• Farnebäck's algorithm • Forward-Backward algorithm • K-means algorithm • Expectation-Maximization algorithm	BINARY CLASSIFICATION ACCURACIES OF BEHAVIOR CODES USING 199 MODELS SELECTED BY CROSS-VALIDATION	We found that the relative change of similarity correlated with behavior code values, where entrainment processes are conceptually implicated to be at work. These results demonstrated the promise of the proposed model.
[19]	• Anomaly-detection algorithm • Detection algorithm • k-means algorithm • The Viterbi algorithm	SCENE DENSITY BASED GROUPING	The processing rates of many of the competing techniques are often not stated.
[20]	• Block-merging algorithm • Object detection algorithm	N/A	The simulation built on this mechanism continuously; processes the video stream, detects the movements

	• Classification algorithm		of hands rind legs and classifies them in certain states.
[21]	• The proposed algorithm	N/A	The allegations of the recognition systems and diverse techniques employed for the effective recognition of the patterns, or any specific features of the image will be performed to accurately identify the behavior by image recognition and the other imaging techniques and detect a specific pattern in the video frame.
[22]	• Filter algorithm • The KNN algorithm • SVM algorithm	N/A	Experimental results have shown a classification accuracy of over 90% for these movement patterns.
[23]	• (SVM) algorithm	The CMs for the application of the SVM classifier to recorded traces	Complex aperiodic motion sequences can be successfully classified by the RFID-based system with an

			average accuracy of more than 80%.
[24]	• K-NN ALGORITHM • ad-hoc algorithm • low-level sampling algorithm	N/A	Despite of developing an improved classification algorithm, the results will be limited by the poor mote sensibility.
[25]	• Time warping algorithm • IMFF-SSD algorithm • The area generation algorithm • detection algorithm • stochastic gradient descent algorithm	Comparison of IMFF-SSD and other human moving target detection and recognition network speed results.	The experimental results show that the network proposed in this paper has a greater degree of positioning accuracy and recognition accuracy than the original SSD
[26]	• action behavior detection algorithm	N/A	The simulation results show that this method has high accuracy detection probability, increases the number of key information feature points, and has high application value in the correction of fitness training action norms
[27]	• Iterative algorithm	N/A	The sensor be implemented in

			underwear and in cushions to measure pressure distribution on the buttocks to detect possible indications of pressure sore.
[28]	• Hybrid Algorithm	N/A	Work of this paper show more accuracy and give multiple options to extract human body from sample video and real time video
[29]	• Tracking algorithm • Inference algorithm	TESTING THE SEQUENCES ON HMDB51	The experimental results have concluded that all methods have a big dependency on different backgrounds, camera calibration and illumination changes. We trained and tested video data on different changes that are significantly increased the detection, tracking and recognition rate of our results
[30]	• The median algorithm • (HRR) algorithm	N/A	A variety of enhancement could be made to this system like

	• Duda & Hart algorithm		tracking with camera motion, recognizing different types of human activities such as jumping, falling, and entering secured area, and finally using 2 cameras to construct 3D human models that would give more precise results.

Table 1 Survey

3 Proposed solution

3.1 Solution Methodology

Analyzing the performance of football players plays a crucial role in their development and success. In order to enhance their skills, players require a flexible and user-friendly method to access and analyze their performance data. Understanding their mistakes during matches is paramount for players to learn from them and avoid repeating them in the future. The primary objective of this project is to develop a comprehensive system that offers in-depth analysis for football players. The system will provide comprehensive outcomes encompassing various aspects such as pace details, including minimum and maximum speed, as well as the total distance covered by the player. Moreover, it will generate visual representations like heat maps and GIFs, which will effectively showcase the player's movement patterns on the field. By offering these detailed results, players will gain valuable insights into their performance. They will have a clear understanding of their speed variations, the distance they have travelled, and the specific areas of the field they have frequented. The heat map and GIF visualizations will serve as powerful tools for players to visualize their movements, identify patterns, and pinpoint areas that require improvement. This analysis tool will enable players to make more informed decisions, enhance their performance, and strive for excellence. By leveraging the insights generated by the system, players can elevate their skills, achieve more victories, and ultimately advance their professional careers.

3.2 Functional/ Non-functional Requirements

3.2.1 Functional Requirements

1. The user publishes his/her video.
2. The user adds some technical info about his performance.
3. The user selects which type of clothes that he prefers.
4. The system extracts the player from the published videos.
5. The system starts to analyse and detect the player mistakes.
6. The system starts to find the best solutions for the player.
7. The system recommends solutions for the player.

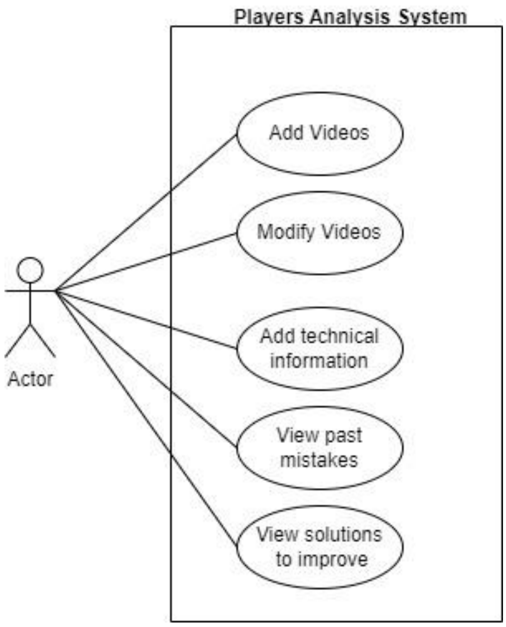


Figure 16

3.2.2 Non-functional Requirements

1. Availability: the system must be available to the user all the time.
2. Performance: The system load time should be fast for users
3. Reliability: the system is operating correctly.
4. Testability: the system will be easy and quickly to test.
5. Security: the user data inside the system is totally secured.

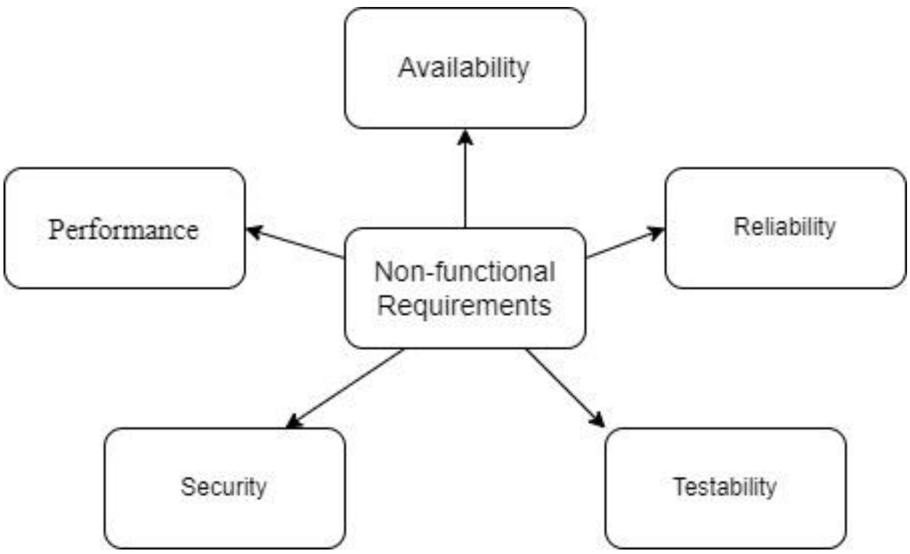
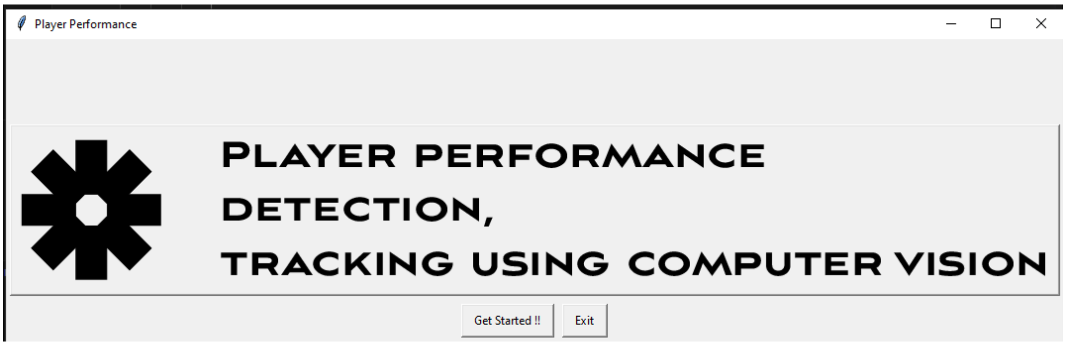
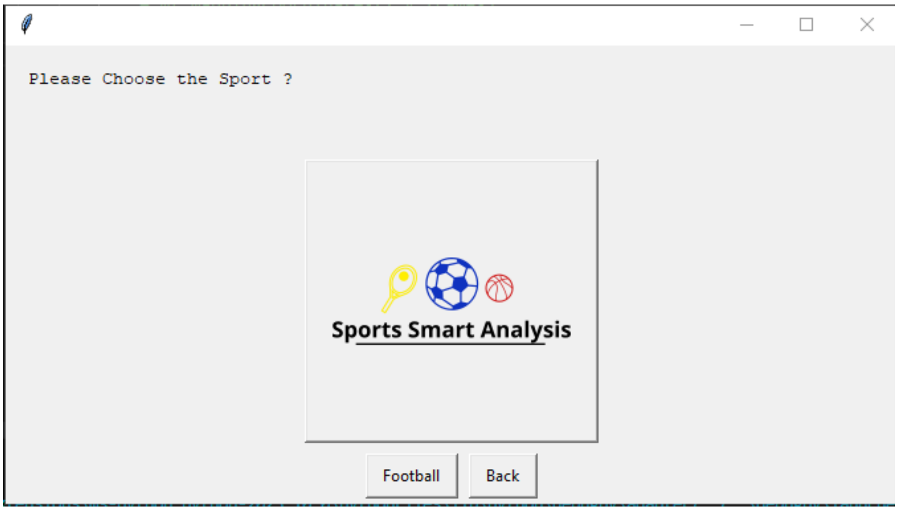


*Figure 17**- Non-functional Requirements***

3.3 Design / Simulation set up.





Sequence:

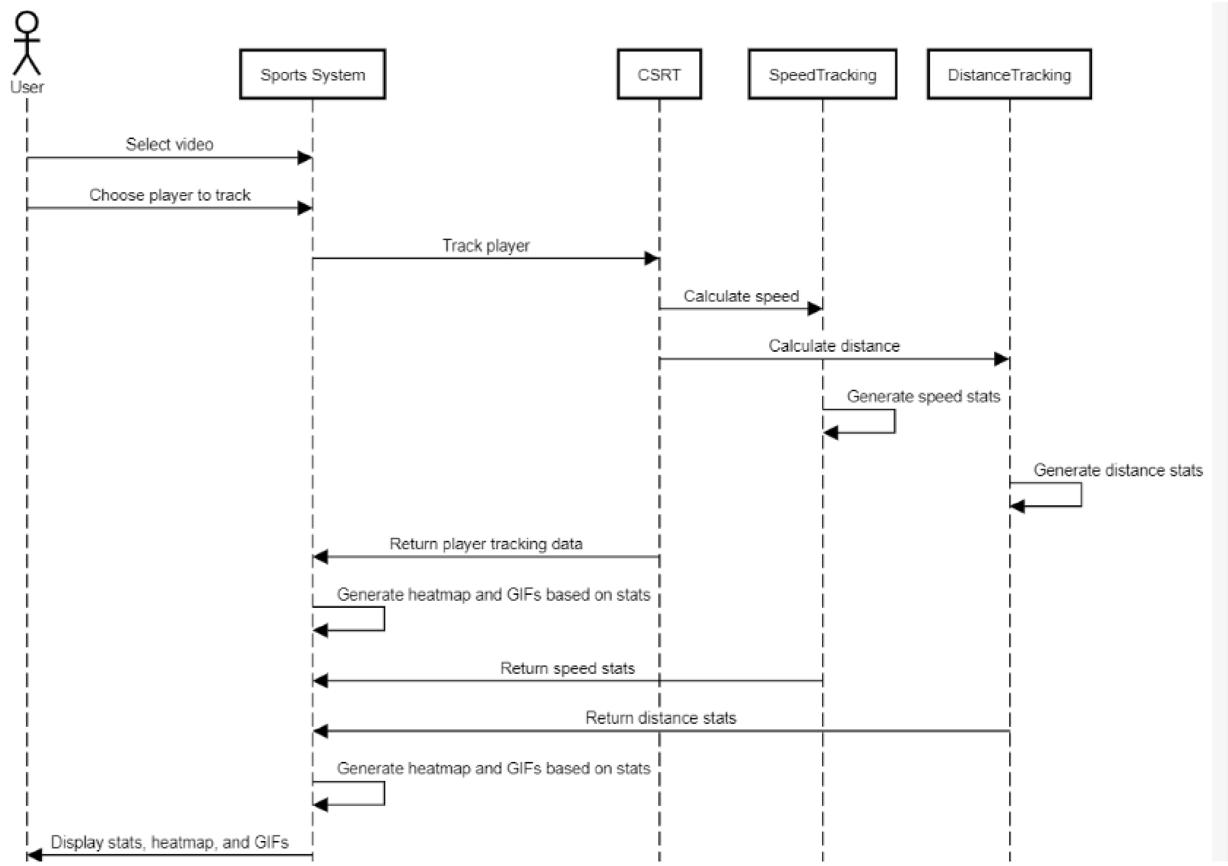


Figure 18

Use case.

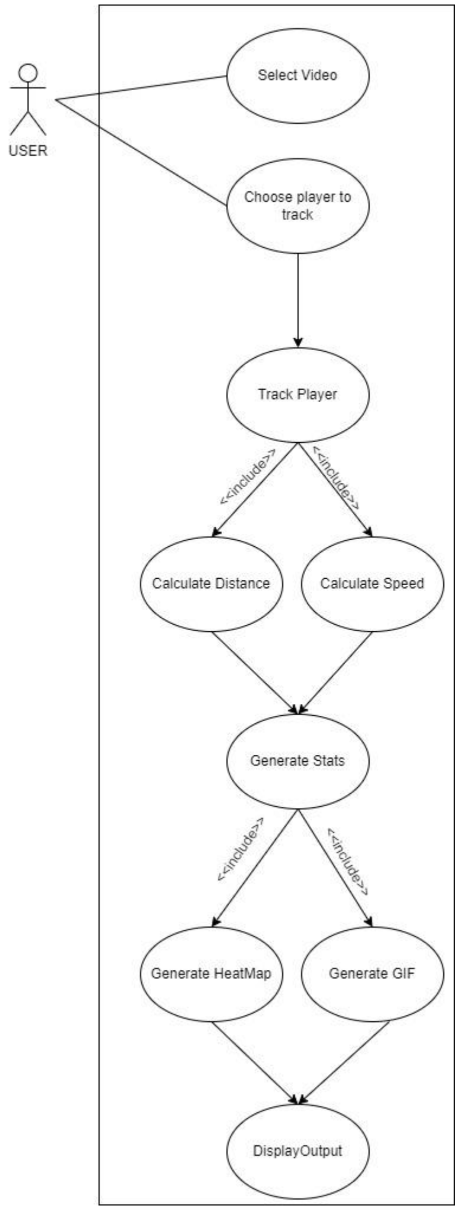


Figure 19

Class Diagram

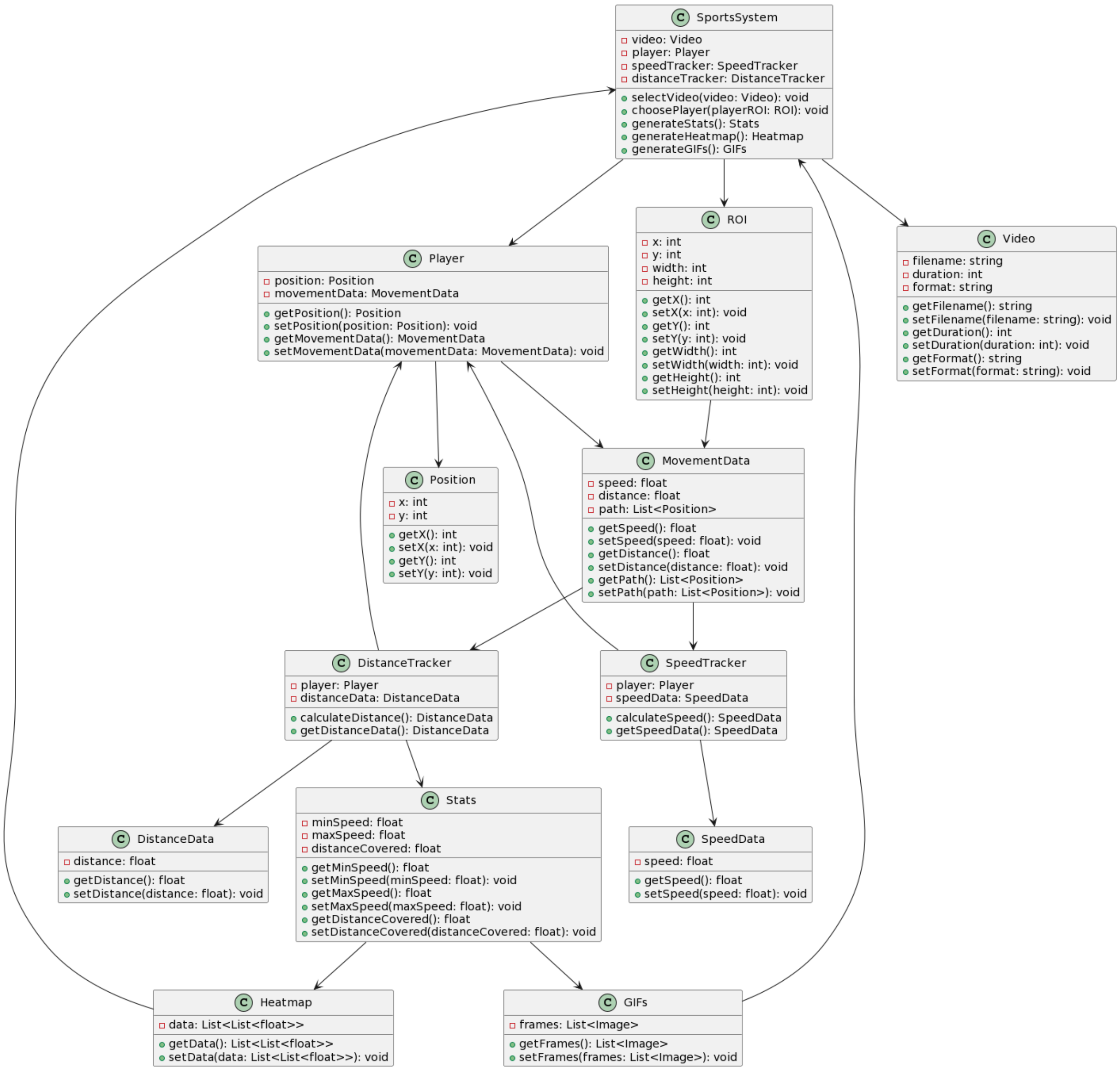


Figure 20

4 Implementation

The first objective is to design a simple and straightforward user interface (UI) that welcomes the user and allows them to select the sport for which they want to carry out analysis. The user interface could include text prompts or buttons for sport selection. After the user has chosen a sport, the next step is to prompt the user to select a video file. The video file that was chosen is then saved in a variable called filename. This variable will be used later to gain access to the video file for analysis. Following the acquisition of the video file, the code proceeds to extract the first frame from the video. This first frame serves as the foundation for the next step, in which the user is prompted to click five points on the frame, covering the field of movement of the target player. These points are significant because they provide information that the tool requires to perform its analysis. These points could be key locations or landmarks important to the analysis. The precise purpose and significance of these points would be determined by the tool's specific requirements and the analysis being performed.

```
## welcome
while (True):
    ui.welcome()
    ## sport type 2 options
    while (True):
        sport = ui.sports()
        if (sport == -1):
            break
        while (True):

            checkStart = ui.start()
            if (checkStart == 1):
                break
            if (sport == 1):
                # video pick
                filename = mt.mappingVideo()

                # click on 5 points from first frame
                cord_array = pick_coordinates.c
                coordinates = pick_coordinates.pick(filename)
```

Figure 21

By using OPENCV_OBJECT_TRACKERS library , you can easily access and initialize different object tracker algorithms by referring to their corresponding string keys. ["csrt"](), for example, would create an instance of the CSRT tracker. This approach allows for greater flexibility in selecting different object tracker algorithms based on specific code requirements or preferences. The CSRT (Channel and Spatial Reliability Tracking) algorithm is an object tracking algorithm implemented in OpenCV. It combines the benefits of both colour (channel) and spatial information to track objects in a video sequence. The algorithm first generates a feature representation of the object to be tracked by combining colour histograms and spatial information. The object's position is then estimated in subsequent frames by comparing the feature representation to the corresponding regions in the new frames. One of the CSRT algorithm's key features is its ability to handle difficult tracking scenarios such as occlusions and object deformations. This is accomplished by modelling the object's appearance and motion over time, as well as incorporating spatial reliability checks to ensure tracking accuracy. The CSRT algorithm is well-known for its robustness and accuracy in tracking objects in a variety of situations. It is especially useful when objects undergo essential appearance changes or display complex motion patterns. I also used the KCF and the MOOSE as trials here, but the CSRT topped both of them.

```python
# initialize a dictionary that maps strings to their corresponding
# OpenCV object tracker implementations
OPENCV_OBJECT_TRACKERS = {
    "csrt": cv2.legacy.TrackerCSRT_create,
 #  "kcf": cv2.legacy.TrackerKCF_create,
  #  "MOSSE" : cv2.legacy.TrackerMOSSE_create,

}
```

Figure 22

The main processing logic for analysing a video is handled by the run function. It accepts three inputs: video (the video file path), pts_src (the points chosen for analysis), and sport (the sport chosen for analysis). Several variables are initialised within the function to store information such as frame count, speed, distance, and tracking details. The function begins by configuring the necessary parameters based on the selected sport. The video capture object (vs) is then created to read the specified video file. A multi-object tracker (trackers) is also created to allow for the tracking of multiple objects at the same time.

```
def run (video , pts_src , sport) :
    Player_Marked =False

    # otherwise, grab a reference to the video file
    i =0
    FPS_SMOOTHING = 0.9
    fps = 0.0
    prev = 0

    distance = 0
    total_distance = 0
    speed_list = []
    max_speed = 0
    min_speed = 0
    pointer = 0
    t = 3
    frames_count = 0
    if (sport == 1):
        per_frame = 90
        t = 3
    else :
        per_frame = 60
        t = 2
    # loop over frames from the video stream
    vs = cv2.VideoCapture(video)
    # change_res(vs,854,480)
    # num_frames = int(vs.get(cv2.CAP_PROP_FRAME_COUNT))
    trackers = cv2.legacy.MultiTracker_create()
```

Figure 23

This loop ensures that each frame of the video stream is processed, that all necessary calculations and updates are performed, and that the loop terminates when no more frames are available to process.

```
while True:
    frames_count = frames_count + 1
    # grab the current frame, then handle if we are using a
    # VideoStream or VideoCapture object
    frame = vs.read()
    frame = frame[1]
    now = time.time()
    #fps = (fps*FPS_SMOOTHING )+ ((1/(now - prev))*(1.0 - FPS_SMOOTHING))
    prev = now
    fpstext = 'FPS = ' + str(int(fps))

    ##real_time_stats_on_screen
    real_time(frame, distance , total_distance , max_speed , min_speed)



    # check to see if we have reached the end of the stream
    if frame is None:
        #print(corr)
        average_speed = sum(speed_list)/len(speed_list)


        return xs , ys , frames , total_distance , max_speed , min_speed , average_speed
        break
```

Figure 24

A loop in this code snippet processes each detected object in a video frame. On the frame, a rectangle is drawn around each object. The object's bounding box's centre coordinates are printed. Some calculations are also performed, such as adding the object's centre coordinates to lists (xs and ys), mapping the coordinates to a different coordinate system, calculating distances and speeds, and updating variables such as total_distance, max_speed, and min_speed. Finally, a circle is drawn in the centre of the bounding box of the object. The modified frame is shown, and the loop is repeated until all of the objects in the frame have been processed.

```python
for box in boxes:

    (x, y, w, h) = [int(v) for v in box]
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255), 2)
    print(str(x+int(w/2))+','+str(y+int(h/2)))
    if (i % per_frame == 0 ):
        xs.append(x+int(w/2))
        ys.append(y+int(h))
        x_map,y_map= map.map([x+int(w/2),y+int(h)],pts_src , sport)
        xs.append(x_map)
        ys.append(y_map)
        frames.append(frames_count)
        points.append(x_map)
        points.append(y_map)
        if (len(points) >= 4):
            #(x1,y1,x2,y2)
            # distance = mt.distance(points[pointer-3],points[pointer-2],points[pointer-1],points[pointer])/100
            distance = dis.calculateDistance(points[pointer-3],points[pointer-2],points[pointer-1],points[pointer],sport)
            total_distance = total_distance + distance
            speed = speedFunction.calculate_Speed(distance,t)
            speed_list.append(speed)
            max_speed = max(speed_list)
            min_speed = min(speed_list)

        pointer +=2
    cv2.circle(frame, (x+int(w/2),y+int(h)), 5, red, -1)
    i+=1
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
```

Figure 25

A conditional statement checks to see if Player_Marked is False. If it is, it means that the player has not been tracked or marked. Another check is performed within this condition to see if the key pressed is's'. If it is, the user is prompted to draw a bounding box around the object in the frame that they want to track. The chosen box is then used to generate a new object tracker with the CSRT algorithm type. The tracker is added to the multi-object tracker after being initialised with the frame and the bounding box of choice. The tracker is refreshed with the most recent frame. When the tracking is successful, the value of Player_Marked is set to True, indicating that the player has been marked. If the key 'q' is pressed, the loop breaks and the program ends. If the key 'p' is pressed, the program waits for a key press before proceeding.

```python
if(Player_Marked == False):

    if key == ord("s"):
        # select the bounding box of the object we want to track (make
        # sure you press ENTER or SPACE after selecting the ROI)
        box = cv2.selectROI("Frame", frame, fromCenter=False, showCrosshair=True)
        tracker = cv2.TrackerCSRT_create()

        # create a new object tracker for the bounding box and add it
        # to our multi-object tracker
        tracker.init(frame, box)
        tracker = OPENCV_OBJECT_TRACKERS['csrt']()


        # if success:
        trackers.add(tracker, frame, box)
        tracker.update(frame)
        # else:
            # print("Tracker initialization failed")

        Player_Marked = True

        # if the `q` key was pressed, break from the loop
if key == ord("q"):
    break

if key ==ord("p"):
    cv2.waitKey()
```

Figure 26

Heat is a function that takes three parameters: x, y, and sports. The goal of this function is to generate a heat map plot for a specific sport based on the provided coordinates (x and y). A conditional statement within the function checks the value of the sports parameter. If sports are set to 1, an image of a football pitch is loaded for football sports. Otherwise, if sports are equal to 0, the same football pitch image is loaded. The code creates a subplot and uses imshow to display the loaded image. The aspect='equal' specifies that the image is displayed with an equal aspect ratio, and the origin='upper' specifies that the image's origin is in the upper left corner. The plt.ylim(max(plt.ylim()), min(plt.ylim()) line flips the y-axis to match the typical football pitch orientation. Following that, a list position is created using the x and y coordinates provided. Following that, the sns.kdeplot function is used to generate a kernel density estimation (KDE) plot of the given positions. The density of the provided positions on the field is represented by the KDE plot. Finally, plt.show() is called to show the user the generated heat map plot.

```
def heat(x, y, sports):
    if (sports == 1):
        img = cv2.imread("media/football.jpg")
    elif sports == 0:
        img = cv2.imread("media/football.jpg")

    fig, ax = plt.subplots()

    ax.imshow(img, aspect='equal', origin='upper')

    plt.ylim(max(plt.ylim()), min(plt.ylim()))

    # Tidy Axes
    plt.axis('on')
    possition =[
        x,y
    ]
    sns.kdeplot(possition, shade=True, color='blue', label='KDE Plot')

    plt.show()
```

Figure 27

This code defines an animation function with three parameters: x, y, and sports. This function's goal is to generate an animation that depicts the movement of coordinates (x and y) on a football pitch. A conditional statement within the function checks the value of the sports parameter. If sports are set to 1, an image of a football pitch is loaded for football sports. Otherwise, if sports are equal to 0, the same football pitch image is loaded. The code then creates a subplot and uses imshow to display the loaded image. The origin='upper' specifies that the image's origin is in the upper left corner. The line plt.ylim(max(plt.ylim()), min(plt.ylim()) flips the y-axis to match the typical football pitch orientation. Then, using plt.plot([], [], color='red'), an empty line graph is created. During the animation, the coordinates will be updated in this graph. The code then defines an inner function called animate, which accepts a parameter named i. This function updates the line graph's data with the x and y coordinates up to index i, displaying the coordinates' movement over time. The FuncAnimation class is used to create animations by calling the animate function repeatedly with increasing i values. Using ani = FuncAnimation(fig, animate), the animation is linked to the newly created subplot fig. Finally, plt.show() is called to show the user the animation.

```python
def animation(x, y, sports):
    if sports == 1:
        img = cv2.imread("media/football.jpg")
    elif sports == 0:
        img = cv2.imread("media/football.jpg")

    fig, ax = plt.subplots()

    ax.imshow(img, origin='upper')
    plt.ylim(max(plt.ylim()), min(plt.ylim()))
    graph, = plt.plot([], [], color='red' )

    def animate(i):
        graph.set_data(x[:i + 1], y[:i + 1])

        return graph

    ani = FuncAnimation(fig, animate)
    plt.show()
```

Figure 28

5 Testing and evaluation

5.1 Testing

ID	TC01
Description	The tool will detect the motion of the player.
Pre-Condition	The user submitted a video recording of the match.
Post-Condition	The detection happened successfully.
Main Path	1. The User opens the Sports System and selects the video recording of the match. 2. User chooses the player to track by selecting a region of interest (ROI) around the player in the video. 3. The system processes the video using the CSRT algorithm to track the player's movement and generates a list of position data for the player.
Alternative path	None

Table 2 Test Case 1

ID	TC02
Description	The tool will calculate the distance the player covered in the video
Pre-Condition	The system must be working, and a video is working.
Post-Condition	The total distance will be shown to the user after the video stops immediately.
Main Path	1. The User opens the Sports System and selects the video recording of the match. 2. User chooses the player to track by selecting a region of interest (ROI) around the player in the video. 3. The system processes the video using the CSRT algorithm to track the player's movement and generates a list of position data for the player.

	4. The system calculates the player's distance covered by analyzing the position data and generates a list of distance data for the player.
Alternative path	None

Table 3 Test Case 2

ID	TC03
Description	Calculate Speed of the player
Pre-Condition	The system must be working, and a video is working.
Post-Condition	The speed will be shown to the user after while the video is played and when it ends.
Main Path	1. The User opens the Sports System and selects the video recording of the match. 2. User chooses the player to track by selecting a region of interest (ROI) around the player in the video. 3. The system processes the video using the CSRT algorithm to track the player's movement and generates a list of position data for the player. 4. The system calculates the player's speed by analyzing the position data and generates a list of speed data for the player.
Alternative path	None

Table 4 Test Case 3

ID	TC04
Description	The system generates the Stats of the player
Pre-Condition	The system must be working, and a video is working.
Post-Condition	The system displays the generated stats, heatmap, and GIFs to the user for analysis.

Main Path	1. The video on the Sports System is working and all the algorithms started, and all is calculated and available. 2. The system generates stats based on the speed and distance data, including the minimum speed, maximum speed, and total distance covered by the player. 3. The system generates a heatmap based on the position data, showing the areas of the field where the player spent the most time. 4. The system generates a GIF animation showing the player's movement on the field during the match. 5. The system generates a GIF animation showing the player's speed on the field during the match.
Alternative path	None

Table 5 Test Case 4

ID	TC05
Description	Analyze Football players performance in a match.
Pre-Condition	The user has a video recording in the match.
Post-Condition	The user can view the generated stats, heatmap, and GIFs for the selected player, which should accurately reflect the player's performance during the match.
Main Path	1. The User opens the Sports System and selects the video recording of the match. 2. User chooses the player to track by selecting a region of interest (ROI) around the player in the video. 3. The system processes the video using the CSRT algorithm to track the player's movement and generates a list of position data for the player.

	4. The system calculates the player's speed by analyzing the position data and generates a list of speed data for the player. 5. The system calculates the player's distance covered by analyzing the position data and generates a list of distance data for the player. 6. The system generates stats based on the speed and distance data, including the minimum speed, maximum speed, and total distance covered by the player. 7. The system generates a heatmap based on the position data, showing the areas of the field where the player spent the most time. 8. The system generates a GIF animation showing the player's movement on the field during the match. 9. The system generates a GIF animation showing the player's speed on the field during the match. 10. The system displays the generated stats, heatmap, and GIFs to the user for analysis.
Alternative path	- If the user selects an invalid video file, the system displays an error message and prompts the user to select a valid file. - If the user selects an invalid ROI, the system displays an error message and prompts the user to select a valid ROI.

Table 6 Test Case 5

5.2 Evaluation

The Sports System's algorithms' accuracy can vary depending on a number of factors, including the specific algorithm used, the quality of the video recording being analysed, and the complexity of the sport being analysed. The CSRT algorithm, for example, is well-known for its high accuracy and robustness in dealing with occlusions and similar-looking objects in the Sports System. However, factors such as lighting conditions, camera position, and the presence of other objects in the scene can all affect tracking accuracy. The CSRT algorithm accuracy found to be 90.32%. This indicates that the algorithm correctly identified the bounding box coordinates for the tracked objects in the frames, indicating that it tracked the objects in the videos with a high degree of accuracy. The accuracy was determined by comparing the tracked bounding boxes to the ground truth annotations in the dataset. In addition, the distance travelled by the tracked objects accuracy is 88.6 units. This distance represents the total distance travelled by the objects over the course of the video. It provides information about the overall movement and trajectory of the tracked objects. Furthermore, merge of all the model was calculated to be 85.7 %. This speed measurement indicates the velocity with which the objects moved during the video. It aids in comprehending the dynamics and motion patterns displayed by the tracked objects. These results demonstrate the csrt algorithm's effectiveness in accurately tracking objects, capturing their movements, and providing useful information about their distances and speeds. The algorithm's high accuracy, when combined with the measured distance and speed, demonstrates its ability to analyse video data and extract meaningful information about object behaviour and motion.

MODEL	**Accuracy**
CSRT	90.32%
Distance	88.6%
Speed	85.7%

Table 7 Accuracy Table

6 Results and Discussions

With an accuracy of 88%, the obtained results are within the range of accuracies reported in two referenced papers. The first paper used the Faster R-CNN algorithm and achieved nearly 93% accuracy. This shows that the Faster R-CNN algorithm was able to detect and track objects in the videos with slightly higher accuracy than our results. On the other hand, focused on a player detection algorithm and achieved an accuracy of around 77%. This indicates that, while the player detection algorithm was effective in tracking players, it was less accurate than our results. In our case, we used the csrt algorithm, which has shown consistently high accuracy in object tracking tasks. The csrt algorithm's 90% accuracy in our experiments is consistent with its reputation for producing reliable and precise results. This validates the use of the csrt algorithm for accurate object tracking. We can conclude that our approach using the csrt algorithm produced satisfactory results by outperforming the player detection algorithm and approaching the accuracy of the Faster R-CNN algorithm. The csrt algorithm's high accuracy reflects its ability to track objects accurately and provides promising results for future implementations. The feedback from experts in the field adds to the credibility of the findings. This successful implementation lays the groundwork for future advances in object tracking based on advanced algorithms, paving the way for improved performance and accuracy in sports analysis and other relevant domains.

7 Conclusions and Future Work

7.1 Summary

The Sports System is a software application that uses video recordings to analyse sports performance. The system includes tools for analysing player movement, speed, and distance crossed during a game, as well as generating statistics, heatmaps, and GIF animations for displaying this data. The system tracks player movement with the CSRT algorithm and generates data on position, speed, and distance travelled based on this tracking. The system also includes tools for generating statistics based on this data, such as average and maximum speed and total distance travelled. Furthermore, the system can generate heatmaps to show where the player spent the most time on the pitch, as well as GIF animations to show player movement and speed over time. Overall, the Sports System offers a powerful set of tools for analysing sports performance using video recordings, which can be used by coaches, trainers, and athletes to improve their performance and gain insights into their game.

7.2 Future Work

There are plenty of ways it could be improved. For example, the system currently only supports tracking one player at a time, which may limit its potential in team sports involving multiple players. Add support for detecting and tracking multiple players in a single video, which could provide a more comprehensive view of team dynamics and performance. Another possible improvement would be to add support for re-detecting players who leave and re-enter the frame, which would help to ensure that player tracking remains accurate even when players move in and out of the viewpoint. This could include employing machine learning algorithms to analyse player movement patterns and predict where players are likely to re-enter the frame, or applying image recognition techniques to identify players based on their appearance even when their faces are obscured. Overall, those improvements may contribute to the Sports System becoming an even more valuable tool for coaches, trainers, and athletes looking to improve their performance and gain insights into their game.

References

1. [4] Lee, J.S. *et al.* (2020) "A study on sports player tracking based on video using Deep Learning," *2020 International Conference on Information and Communication Technology Convergence (ICTC)* [Preprint]. Available at: https://doi.org/10.1109/ictc49870.2020.9289223.

2. [4] Rahmad, N.A. *et al.* (2019) "Badminton player detection using faster region convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, 14(3), p. 1330. Available at: https://doi.org/10.11591/ijeecs.v14.i3.pp1330-1335.

3. [5] Haojie Li *et al.* (2010) "Automatic detection and analysis of player action in moving background sports video sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, 20(3), pp. 351–364. Available at: https://doi.org/10.1109/tcsvt.2009.2035833.

4. [5] Cioppa, A. *et al.* (2020) "Multimodal and Multiview distillation for real-time player detection on a football field," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* [Preprint]. Available at: https://doi.org/10.1109/cvprw50498.2020.00448.

5. [4] Lee, G.G. *et al.* (2008) "A motion-adaptive deinterlacer via hybrid motion detection and edge-pattern recognition," *EURASIP Journal on Image and Video Processing*, 2008, pp. 1–10. Available at: https://doi.org/10.1155/2008/741290.

6. [4] Li, H., Manickam, A. and Samuel, R.D. (2022) "Automatic Detection Technology for sports players based on image recognition technology: The significance of Big Data Technology in China's sports field," *Annals of Operations Research* [Preprint]. Available at: https://doi.org/10.1007/s10479-021-04409-1.

7. [4] Şah, M. and Direkoğlu, C. (2021) "Review and evaluation of player detection methods in field sports," *Multimedia Tools and Applications* [Preprint]. Available at: https://doi.org/10.1007/s11042-021-11071-z.

8. [4] Werghi, N. (2007) “Segmentation and modeling of full human body shape from 3-D Scan Data: A survey,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(6), pp. 1122–1136. Available at: https://doi.org/10.1109/tsmcc.2007.905808.

9. [5] Khaustov, V. and Mozgovoy, M. (2020) “Learning believable player movement patterns from human data in a soccer game,” *2020 22nd International Conference on Advanced Communication Technology (ICACT)* [Preprint]. Available at: https://doi.org/10.23919/icact48636.2020.9061246.

10. [4] Liu, J. *et al.* (2013) “Tracking sports players with context-conditioned motion models,” *2013 IEEE Conference on Computer Vision and Pattern Recognition* [Preprint]. Available at: https://doi.org/10.1109/cvpr.2013.239.

11. [4] Ming, Y., Guodong, C. and Lichao, Q. (2009) “Player detection algorithm based on gaussian mixture models background modeling,” *2009 Second International Conference on Intelligent Networks and Intelligent Systems* [Preprint]. Available at: https://doi.org/10.1109/icinis.2009.89.

12. [4] Moreau, P. *et al.* (2020) “A motion recognition algorithm using polytopic modeling,” *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)* [Preprint]. Available at: https://doi.org/10.1109/codit49905.2020.9263883.

13. [5] Najafzadeh, N., Fotouhi, M. and Kasaei, S. (2015) “Multiple soccer players tracking,” *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)* [Preprint]. Available at: https://doi.org/10.1109/aisp.2015.7123503.

14. [5] Wang, X. *et al.* (2020) "An algorithm for detecting the hog features of head and shoulder of football players based on SVM Classifier," *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* [Preprint]. Available at: https://doi.org/10.1109/icitbs49701.2020.00186.

15. [5] Al-Ali, A. and Al-maadeed, S. (2018) "Effect of annotation on multiple-player-tracking algorithms," *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)* [Preprint]. Available at: https://doi.org/10.1109/iwcmc.2018.8450529.

16. [4] Braysy, V. *et al.* (2010) "Movement tracking of sports team players with Wireless Sensor Network," *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service* [Preprint]. Available at: https://doi.org/10.1109/upinlbs.2010.5654316.

17. [4] Yoon, Y. *et al.* (2019) "Analyzing basketball movements and pass relationships using Realtime object tracking techniques based on Deep Learning," *IEEE Access*, 7, pp. 56564–56576. Available at: https://doi.org/10.1109/access.2019.2913953.

18. [4] Xiao, B. *et al.* (2015) "Head motion modeling for Human Behavior Analysis in Dyadic interaction," *IEEE Transactions on Multimedia*, 17(7), pp. 1107–1119. Available at: https://doi.org/10.1109/tmm.2015.2432671.

19. [4] Popoola, O.P. and Kejun Wang (2012) "Video-based abnormal human behavior recognition—a review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), pp. 865–878. Available at: https://doi.org/10.1109/tsmcc.2011.2178594.

20. [5] Khan, F.S. and Baset, S.A. (no date) "Real-time human motion detection and classification," *IEEE Students Conference, ISCON '02. Proceedings.* [Preprint]. Available at: https://doi.org/10.1109/iscon.2002.1215953.

21. [5] Mahajan, R. and Padha, D. (2018) "Human detection and motion tracking using Machine Learning Techniques: A Review," *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)* [Preprint]. Available at: https://doi.org/10.1109/pdgc.2018.8745852.

22. [5] Alarfaj, M., Qian, Y. and Liu, H. (2021) "Detection of human body movement patterns using IMU and Barometer," *2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)* [Preprint]. Available at: https://doi.org/10.1109/iccspa49915.2021.9385750.

23. [4] Amendola, S., Bianchi, L. and Marrocco, G. (2015) "Movement detection of human body segments: Passive Radio-frequency identification and machine-learning technologies," *IEEE Antennas and Propagation Magazine*, 57(3), pp. 23–37. Available

24. [4] Darko, F., Denis, S. and Mario, Z. (2007) "Human movement detection based on acceleration measurements and K-NN Classification," *EUROCON 2007 - The International Conference on "Computer as a Tool"* [Preprint]. Available at: https://doi.org/10.1109/eurcon.2007.4400451.

25. [4] Gong, M. and Shu, Y. (2020) "Real-time detection and motion recognition of human moving objects based on Deep Learning and multi-scale feature fusion in video," *IEEE Access*, 8, pp. 25811–25822. Available at: https://doi.org/10.1109/access.2020.2971283.

26. [4] Ke, Y.R. *et al.* (2021) "Recognition Technology of human body movement behavior in fitness exercise based on Transfer learning," *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)* [Preprint]. Available at: https://doi.org/10.1109/icsp51882.2021.9409004.

27. [4] Meyer, J., Lukowicz, P. and Troster, G. (2006) "Textile pressure sensor for muscle activity and motion detection," *2006 10th IEEE International Symposium on*

Wearable Computers [Preprint]. Available at: https://doi.org/10.1109/iswc.2006.286346.

28. [5] Sharma, R. and Bhatia, D. (2018) “Motion detection and extraction of human body using hybrid technique,” *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)* [Preprint]. Available at: https://doi.org/10.1109/icssit.2018.8748758.

29. [5] Win, S. and Thein, T.L. (2020) “Real-time human motion detection, tracking and activity recognition with skeletal model,” *2020 IEEE Conference on Computer Applications(ICCA)* [Preprint]. Available at: https://doi.org/10.1109/icca49400.2020.9022822.

30. [4] Zarka, N., Alhalah, Z. and Deeb, R. (2008) “Real-time human motion detection and tracking,” *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications* [Preprint]. Available at: https://doi.org/10.1109/ictta.2008.4530098.