Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Football Matches Analysis

Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Football Matches Analysis

By:

MennaTullah Mohammed Abdelbadea [Computer Science]
Nourelhuda Ashraf Abdelhaleem [Computer Science]
Ismaeel shreif Ahmed [Computer Science]
Ahmed Esmat Ahmed [Computer Science]
Ahmed Hany Fathy [Computer Science]
Abdelrahman Fayez Mohammed [Computer Science]

Under Supervision of:

Dr. Ahmed Salah,
Lecturer,
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Mohammed Essam,
Assistant Lecturer,
Bioinformatics Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Acknowledgement

All appreciation and gratitude are due to Allah, who made it possible for us to finish this task. We hope to accept this work from us.

We are appreciative of our family and parents for their constant assistance and support over our entire academic years of study. We hope to be able to return it to them.

We also offer our sincerest gratitude to our supervisors, Dr. Ahmed Salah, TA. Mohammed Essam who all have supported us throughout our thesis with their patience, knowledge and experience.

Finally, we'd like to thank our friends and everyone else who has helped and supported us.

Abstract

Football is a globally beloved sport, and the ability to analyse matches with precision can provide valuable insights for teams, coaches, and fans. The "Football Matches Analysis using AI" project aims to enhance the understanding of football games by leveraging advanced artificial intelligence techniques. This project is designed to address the need for automated, accurate, and detailed analysis of football matches, which traditionally relies on manual methods that can be time-consuming and subjective.

The system developed in this project incorporates several key features: detection and tracking of players, referees and the ball, team mapping, and event detection. By using sophisticated computer vision algorithms and machine learning models, the system can accurately detect and track players and the ball throughout the match. Team mapping algorithms identify the players' team affiliations, while event detection mechanisms highlight significant events such as goals, fouls, and substitutions.

Furthermore, the system includes defined functions to calculate critical metrics such as ball possession and the number of passes for each team. These metrics provide a comprehensive overview of the match dynamics and can be used to generate detailed performance reports.

The final results demonstrate the effectiveness and accuracy of the developed system in analysing football matches. The AI-based approach not only automates the analysis process but also ensures consistent and objective results, making it a valuable tool for football analysts, coaches, and enthusiasts. The project shows great potential for further enhancements and applications in sports analytics.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

- ACID Atomicity, Consistency, Isolation, Durability
- AI Artificial Intelligence

- API Application Programming Interface
- CNN Convolutional Neural Networks
- COCO Common Objects in Context
- CPUs Central Processing Units
- DOM Document Object Model
- GPUs Graphical Processing Units
- GRU Gated Recurrent Unit
- GSM Gate Shift Modules
- HTTP Hyper Text Transfer Protocol
- IoU Intersection over Union
- LSTM Long Short-Term Memory
- mAP Mean Average Precision
- HSV Hue, Saturation, and Value
- MOT Multi Object Tracking
- RDBMS relational database management system
- RGB Red, Green, Blue
- RNNs Recurrent Neural Networks
- SORT Simple Online and Realtime Tracking
- SPAs single-page applications
- SQL Structured Query Language
- TCNs Temporal Convolutional Networks
- UI User Interface
- YOLO You Only Look Once

1- Introduction

1.1 Motivation

Despite the sport's popularity, there has been limited attention and investment in sports analytics and

Computer Vision technology in Egypt. This project aims to fill this gap by contributing to the advancement of these technologies and achieving acceptable accuracies with limited resources. By doing so, we can enhance the league's performance and make a significant impact on the sport's development. This project is essential because it can help improve the overall quality of the league, attract more fans and investors, and ultimately contribute to the growth of football.

1.2 Problem Definition

The problem at hand is the lack of attention and investment in sports analytics and Computer Vision technology football in Egypt. Despite football being the most popular sport in the world, the research in this area is not given much concern in the open source community. This has resulted in limited resources being allocated to enhance the performance of the leagues. Our aim is to contribute to the advancement of these technologies and achieve acceptable accuracies with the limited resources available. With additional resources.

1.3 Objective

:Our goal is to use Computer Vision achieving the following targets

1. player-Team identification by Mapping each player to his own team.
2. Continuously tracking the ball with player throughout the entire video clip.
3. Detect happened events like (fouls, corners, shots and ball

possession)

4. Generate statistics for each team as passes completed, (successful/missed), shots, yellow/red cards, ball possession, scored goals.

1.4 Time Plan

The figure shows our time plan of our work.

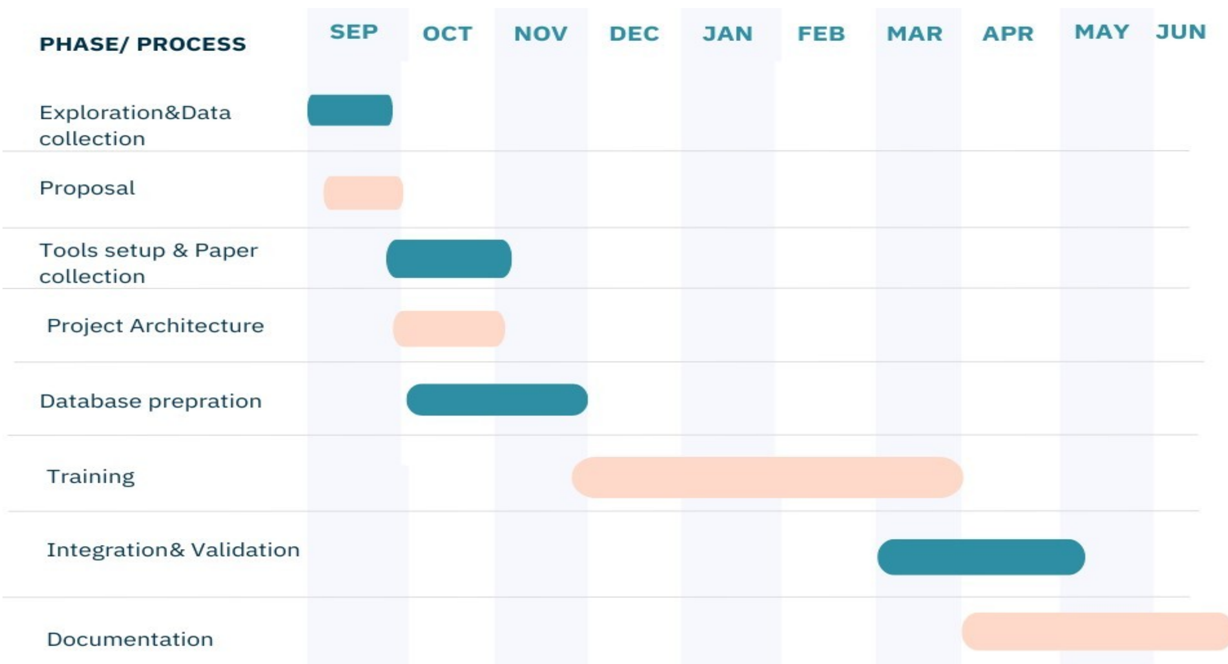

Figure 1.1: Time Plane

1.5 Document Organization

- **Chapter 2:** This chapter discusses the background of the project with respect to its scientific basis and its intended usage field, as well as a brief description of other similar projects

- **Chapter 3:** This chapter will talk about system overview according to system architecture (what architectures are used and what are their designs) , The chapter also includes the system's intended users with their characteristics (basic knowledge required for the user to be able to benefit from the project).

- **Chapter 4:** This chapter explains in detail each function, technique and algorithm implemented in the project, as well as testing procedures done.

- **Chapter 5:** This chapter outlines the detailed user's manual on how to use the system with a step-by-step screenshot guide.

- **Chapter 6:** This Chapter will include a complete summary of the whole project along with the results obtained. What can be done in the future to improve the performance of the project and what additional functions could be added?

2- Background

2.1 Field and scientific background of the project

The field of this project related to computer vision, machine learning and deep learning. Specifically, the focus is on YOLO, a type of deep learning algorithm used for to detect objects within images, CNNs, a type of deep learning architecture particularly well-suited for image recognition tasks.

CNNs used for recognizing and classify objects within images. For example, they can distinguish between different types of ball, player, or any other objects, locate and identify objects within an image or video by drawing bounding boxes around them. YOLO used for real-time object detection in videos and live streams, such as tracking players movements across the field, detect the ball's position.
CNNs can be trained to recognize specific events such as goals, fouls, and saves, while YOLO can locate these events within the video.

To understand YOLO and CNNs, it's important to have some knowledge of neural networks and deep learning which are subsets of AI. Neural networks are a class of machine learning algorithms inspired by the structure and function of the human brain and they are mathematical models designed to recognize patterns within data, while deep learning refers to the use of multiple layers of these networks to improve accuracy and efficiency. In recent years, deep learning has led to significant advancements in computer vision tasks such as object detection, image classification.

2.2 A survey of the work done in the field

The application of AI in sports analytics has garnered significant interest, leading to numerous research efforts and commercial solutions. Football, being a globally popular sport, has been a focal point for many of these advancements. Key areas of research and development include player and ball tracking, event detection, and the generation of actionable insights from video data.

Various studies have employed convolutional neural networks (CNNs) to detect and track players and the ball. For instance, the YOLO (You Only Look Once) algorithm has been widely used for real-time object detection due to its balance of speed and accuracy. Further advancements include integrating tracking algorithms like SORT (Simple Online and Realtime Tracking) to maintain the identity of players and the ball across video frames.

Identifying events such as goals, fouls, and corners is critical for comprehensive match analysis. Techniques involving Temporal Convolutional Networks (TCNs) and Long Short-Term Memory (LSTM) networks have been effective in analysing sequences of frames to detect and classify these events. These models capture temporal dependencies, making them suitable for understanding the progression of a match.

Combining statistical analysis with intuitive visual representations enhances the accessibility of insights. Dashboards and interactive tools have been developed to present data such as player heat maps, ball possession statistics, and event timelines. These tools leverage web technologies to offer real-time updates and user-friendly interfaces.

2.2.1 Related Work

Table 2-2 ***Related Work***

Authors	Dataset	Method(s)	Accuracy
Ragd Alhejaily, Rahaf Alhejaily, Mai Almdahrsh, Shareefah Alessai, and Saleh Albelwi.	Common Objects in Context (COCO)	Team Assignment	
		CAE (Type of CNN) +K-means	92%
		Ball Detection	

		Pretrained Yolov5 +TL	98%
James Hong, Haotian Zhang, Michael Gharbi, Matthew Fisher	Competition split (FS-Comp) Soccer Net	E2E-Spot	74.84%
Ruslan Baikulov	Soccer Net	CNN	86.47%
Anthony Cioppa, Silvio Giancola, Adrien Deliege, Le Kang, Xin Zhou	Soccer Net	ByteTrack	50.257%

Table 2-1 Related Work

2.3 Existing similar systems

After Searching, we found some commercial and open-source platforms provide functionalities similar to those of our project:

- **Opta Sports:** A prominent sports analytics platform that manually collects data to offer detailed statistics and insights into player and team performance. Opta's data is widely used by professional teams, broadcasters, and analysts. Opta Analytics suite of advanced metrics, which measure the quality of passes, shots and individual playing styles, is enabling you to take your analysis of football to the next level by providing underlying insights into the actions of all the key performers on the pitch

- **Playmaker AI:** This platform utilizes AI to provide detailed football data and insights. Playmaker. AI's features include scouting and player development tools, player avatars that describe roles more precisely than traditional positions, and comprehensive individual and team reports. The platform supports football clubs, agents, media companies, and researchers by offering extensive data on player performance and facilitating in-depth analysis to enhance strategic decision-making.

- **Second Spectrum:** Known for its use of AI and machine learning to deliver real-time tracking and analysis, Second Spectrum offers advanced visualizations and insights. It serves both basketball and football, providing detailed analytics for teams and broadcasters.

2.4 Description of technologies used

- **Node.js:**

Node.js is an open-source, server-side JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows you to run JavaScript code outside of a web browser, enabling you to build scalable and efficient network applications.

Here are some key features and aspects of Node.js:

1. Asynchronous and Event-driven: Node.js uses an event-driven, non-blocking I/O model, which makes it well-suited for applications that handle many concurrent connections and require high scalability. It allows developers to write asynchronous code that can handle multiple requests without getting blocked.

2. JavaScript on the Server: Node.js allows you to use JavaScript on the server-side, which means you can use the same language and share code between the client-side and server-side of your applications. This can lead to increased productivity and code reusability.

3. Web Development: Node.js is commonly used for building web applications and APIs. It provides frameworks and libraries such as Express.js, Nest.js, and Koa.js that simplify the process of creating web servers and handling HTTP requests.

4. Real-time Applications: Due to its event-driven and non-blocking nature, Node.js is well-suited for real-time applications like chat applications, collaborative tools, and multiplayer games. It allows for bidirectional communication between the server and clients using technologies like WebSockets.

- **React:**

React is a popular JavaScript library for building user interfaces. It was developed by Facebook and is widely used for creating interactive and dynamic web applications. React follows a component-based architecture, which allows developers to build reusable UI components and efficiently manage the application's state.

Here are some key concepts and features of React:

1. Components: React applications are built using components. A component represents a reusable piece of UI, which can be a small part of a page or the entire page itself. Components are self-contained, meaning they manage their state and can be composed together to create complex UIs. React uses a syntax called JSX (JavaScript XML) that allows you to write HTML-like code within your JavaScript, making it easier to define and render components.

2. Virtual DOM: React utilizes a virtual DOM (Document Object Model) to efficiently update and render the UI. The virtual DOM is an in-memory representation of the actual DOM, and React uses it to track changes and update only the necessary parts of the UI, minimizing performance overhead. This approach allows for fast and efficient rendering of UI components.

3. Unidirectional Data Flow: React follows a unidirectional data flow pattern, also known as one-way binding. Data flows from parent components to child components through props (properties). Child components can't directly modify the data; instead, they receive data from their parent components and communicate changes through callbacks or events.

4. State Management: React allows you to manage and update the application's state using the state object.
   The state represents the data that can change over time and affects the UI rendering. When the state is updated, React automatically re-renders the affected components to reflect the changes.

5. React Hooks: Introduced in React 16.8, hooks are functions that allow you to add state and other React features to

functional components. They provide a more concise and reusable way to manage component state, handle side effects, and access React's lifecycle methods without the need for class components.

- **My SQL:**

  a popular open-source relational database management system (RDBMS) that is widely used for storing and managing structured data. It is known for its reliability, performance, and ease of use, making it a popular choice for various applications ranging from small-scale projects to large-scale enterprise systems.

Here are some key concepts and features of React:

1. Relational Database: MySQL is based on the relational database model, which organizes data into tables with rows and columns. It supports SQL (Structured Query Language) as the standard language for interacting with the database, allowing you to perform operations like creating tables, inserting, updating, and retrieving data, and executing complex queries.

2. Data Integrity and ACID Compliance: MySQL ensures data integrity by supporting various constraints such as primary keys, unique keys, foreign keys, and check constraints. It also follows the ACID (Atomicity, Consistency, Isolation, Durability) properties, which guarantee the reliability and consistency of data even in the presence of failures or concurrent transactions.

3. Scalability and Performance: MySQL offers scalability options to handle increasing amounts of data and traffic. It

supports replication, which allows you to create multiple copies of the database for high availability and load balancing. Additionally, MySQL provides various indexing techniques, query optimization, and caching mechanisms to improve performance and responsiveness.

4. Compatibility and Integration: MySQL is widely supported and integrates with various programming languages, frameworks, and tools. It has official connectors for languages like PHP, Python, Java, and .NET, making it easy to interact with the database from different applications. Additionally, MySQL is compatible with popular web development platforms like PHP, Ruby on Rails, and Node.js.

- **Fast API:**

is a modern, fast, and lightweight web framework for building APIs with Python. It is designed to be easy to use, highly performant, and capable of handling high loads. While FastAPI and the Kaggle API are separate entities, you can use FastAPI to create an API that interacts with the Kaggle API.

1. FastAPI Features:
   - Fast: FastAPI leverages modern Python features like type annotations and asynchronous programming to provide high

performance. It is built on top of Starlette, an asynchronous web framework.

- Easy to Use: FastAPI is designed to be developer-friendly, with automatic generation of interactive API documentation using the OpenAPI standard. It also includes features like input validation, serialization, and dependency injection.
- Standards-Based: FastAPI follows industry standards like OpenAPI and JSON Schema, making it compatible with various tools and libraries in the API ecosystem.

2. Kaggle API Features:
    - The Kaggle API allows you to interact with Kaggle, a platform for exploring and analysing data sets, participating in data science competitions, and sharing insights. The Kaggle API provides programmatic access to datasets, competitions, and kernels (code notebooks).
    - With the Kaggle API, you can programmatically download datasets, submit entries to competitions, retrieve competition information, and perform other actions available through the Kaggle platform.

3. Using FastAPI with the Kaggle API:
    - You can use FastAPI to create an API that interacts with the Kaggle API by defining endpoints that handle requests and make appropriate calls to the Kaggle API.
    - Start by installing FastAPI and importing the necessary modules. You can then define your API endpoints using FastAPI's decorator-based syntax, specifying the HTTP methods, paths, and request/response models.
    - Within your endpoint functions, you can make requests to the Kaggle API using libraries like requests or dedicated Kaggle

API client libraries, such as kaggle-api package. These libraries provide methods for authentication, making requests, and handling responses.

- You can define endpoints to perform actions like retrieving datasets, submitting competition entries, getting competition details, or any other functionality provided by the Kaggle API.
- FastAPI's automatic documentation generation will provide an interactive API documentation that describes your API endpoints, request/response models, and allows users to test the API directly.

3- Analysis and Design

3.1 System Overview

3.1.1 System Architecture

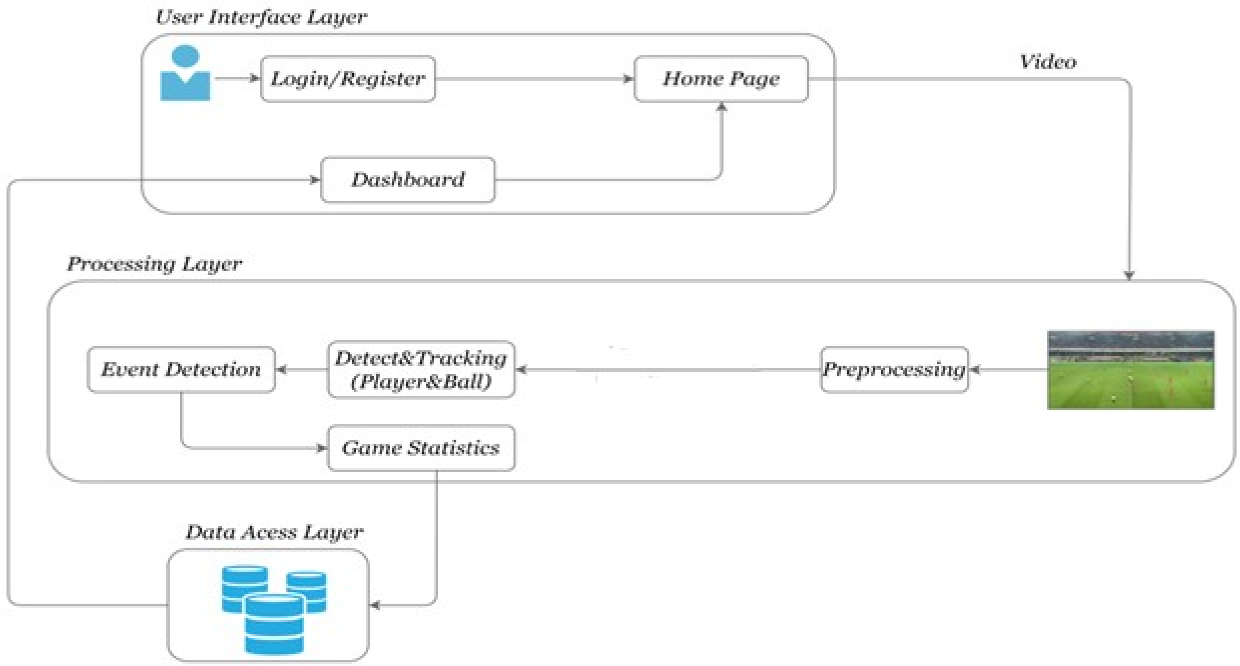

Figure 3.2: System Architecture

- **User Interface Layer**
  Upon opening the website, users are presented with a home screen where they can choose to log in with existing credentials or register for a new account. After successful authentication, users are directed to the home screen where they have the ability to view and edit their personal information. Additionally, the website offers football video Analysis that allows users to upload a video of a Match and give him dashboard of important events.

- **Processing Layer**
  The input video enter the processing stage in in which video frames are read one by one and then start detecting and tracking object in each frame and detect if there is an event in the frame, then we store analytics we extract from the frame to the memory, for the event detection part, we crop the most important event on the match as a match highlights, after repeating the process to the end of the frames then we start saving the analytics of the whole video and present it in the game statistics dashboard

- **Data Access Layer**
  It is the layer in which the website's database is located. This database holds the information of the registered users, including the dashboards and saved highlights of the match.

3.1.2 System Users

A. Intended Users:

Any type of user can use the system, but it's intended to help 3 specific types of users which are:

● clubs who want to know the statistics of the team and the performance of their players

● individual players who would like to assess their performance and statistics

● sports channels who want to automate the process of calculating the statistics in matches.

B. User Characteristics:
The ability to browse the internet to upload videos to the website.
No more experience or skills are needed.

4- Implementation and Testing

4.1 Tools:

- **Visual Studio Code:** is a free source-code editor, It provides developers with a lightweight, yet powerful coding environment that supports multiple programming languages, including JavaScript, Python, Java, C++, and many others.

- **MySQL:** is an open-source relational database management system (RDBMS). It is used to store, retrieve, manage, and manipulate data in databases.

- **Programming languages:** Python, is a high-level programming language that was first released in 1991. Python has a wide range of applications such as web development, data analysis, scientific computing, machine learning, artificial intelligence, and more. The language is highly versatile due to its large library of modules and packages, which allows developers to easily implement complex functionality without having to write extensive amounts of code.

- **React:** a JavaScript library for building user interfaces, particularly single-page applications (SPAs) where you need a fast and interactive user experience.

- **Node.js:** is a runtime environment that allows you to execute JavaScript code on the server side. It is built on Chrome's V8 JavaScript engine

- **FastAPI:** a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.

- **Google Colaboratory (Colab):** is a free, cloud-based platform provided by Google that allows users to write and execute Python code in a Jupyter notebook environment. It is particularly popular among data scientists, machine learning practitioners, and educators for its convenience and powerful features.

- **Kaggle:** It is a popular online platform for data science and machine learning competitions. It provides a platform for data scientists, machine learning engineers, and other experts to compete in solving complex data problems posed by companies, organizations, and research institutions. Kaggle also offers a variety of datasets for users to practice their skills and build their data science portfolio. In addition to competitions and datasets, Kaggle provides a community of data science professionals who can share their knowledge and collaborate on projects.

- **Ultralytics:** is an open-source project known for developing advanced artificial intelligence (AI) and computer vision solutions, particularly in the field of object detection, Ultralytics is renowned for its contributions to the "You Only Look Once" (YOLO) series of real-time object detection models. The YOLO models are popular for their high speed and accuracy.

- **Keras:** An open-source deep learning framework written in Python. It provides a high-level API for building and training deep learning models that can run on both CPUs and GPUs. Keras can be used with various backends, including TensorFlow, Microsoft Cognitive Toolkit, and Theano. It supports a wide range of neural network architectures, including Convolutional Neural Networks (CNNs), RNNs, and transformer models. It also includes a variety of pre-trained models that can be used for tasks such as image classification, object detection, and natural language processing.

- **TensorFlow:** An open-source software library developed by Google for dataflow and differentiable programming across a range of tasks. It is designed to simplify the creation of complex machine learning models, including deep neural networks, and streamline the process of training and deploying these models. It is widely used in industry for a variety of applications, including computer vision, speech recognition, and natural language processing.

- **OpenCV :** short for Open Source Computer Vision Library, is a popular open-source library used for computer vision and image processing tasks, OpenCV provides tools for various image and video processing operations. These include but are not limited to image filtering, edge detection, feature extraction, object detection, and image transformation. It also supports machine learning algorithms and has interfaces for implementing deep learning models, is cross-platform and runs on various operating systems, including Windows, Linux, macOS, Android, and iOS.

4.2 Dataset:

- **Soccernet**
  - The Soccernet dataset is a comprehensive collection of data related to soccer (football) matches. It is often used for research in computer vision, machine learning, and sports analytics.
  - The dataset is composed of 500 complete soccer games from six main European leagues, covering three seasons from 2014 to 2017 and a total duration of 764 hours
  - . A total of 6,637 temporal annotations are automatically parsed from online match reports at a one-minute resolution

for three main classes of events (Goal, Yellow/Red Card, and Substitution).

Figure 4.3 SoccerNet Dataset

- **DFL - Bundesliga Data Shootout**
  - dataset in a Kaggle competition for football
  - it consists of 200 high quality clips of 30 seconds
  - The dataset covers various matches filmed with different cameras

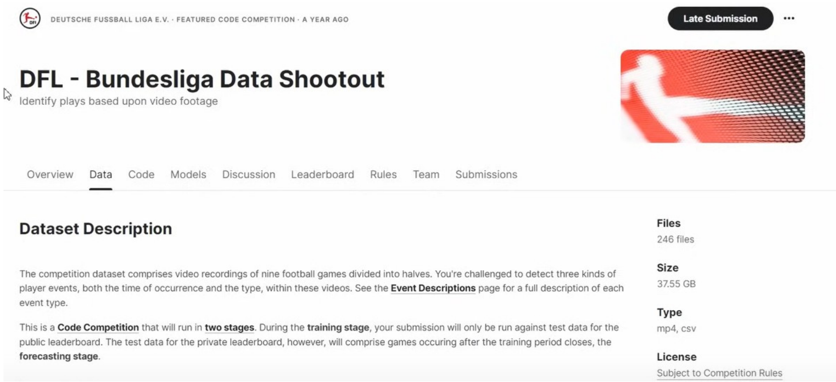

Figure 4.4   DFL Dataset

- **football-players-detection Image Dataset:**
  - consist of 663 total images, 612 for training, 38 for validation and 13 images for testing.
  - We use this dataset to finetune our YOLO model to better identify players, referees and ball.

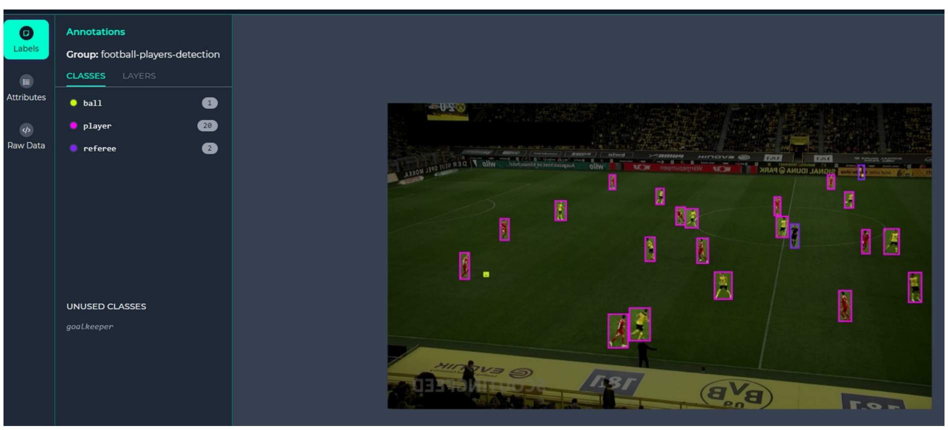

Figure 4.5 Football Player Detection Dataset

4.3 Preprocessing:

- Frame Sampling to reduce computation time for the video.
- Resizing video frames into 448x448 for YOLO.
- Converting frames from BGR format to RGB format.
- Data augmentation: YOLOv5 employs various data augmentation techniques to improve the model's ability to generalize and reduce overfitting. These techniques include:
    - **Mosaic Augmentation**: An image processing technique that combines four training images into one in ways that encourage object detection models to better handle various object scales and translations.
    - **Copy-Paste Augmentation**: An innovative data augmentation method that copies random patches from an image and pastes them onto another randomly chosen image, effectively generating a new training sample.
    - **Random Affine Transformations**: This includes random rotation, scaling, translation, and shearing of the images.
    - **MixUp Augmentation**: A method that creates composite images by taking a linear combination of two images and their associated labels.
    - **Albumentations**: A powerful library for image augmenting that supports a wide variety of augmentation techniques.
    - **HSV Augmentation**: Random changes to the Hue, Saturation, and Value of the images.

- Extract video frames to a common directory to be reusable by multiple models to save resources and reduce running time.
- Stacking frames to create a single, more informative input for a model. This approach helps in capturing temporal information and motion, which can be crucial for tasks that depend on understanding changes over time.

4.4Application functions:

- **Registration:** Anyone can register in our application. By providing an email, password and uploading an image for himself. First, we validate the username and password, and that he uploaded the photo successfully that contains a face of only one person, then we save the information in the database of the application.
- **Login:** Users can login with their email and password. We verify them in the database.
- **Update personal information:** Users can update all their personal information, then the application updates their info in the database.
- **Generate statistics dashboard:** Users can upload video for a match that they want to generate its statistics in the dashboard and then give it a label and save it to the database with the highlights of the most important actions in the match.
- **Play a visualized analysis version of the match:** users can view an annotated video for detecting the ball, players and referees along with real time ball possession.
- **Events transcript**: user can view Transcript showing match events with their timestamps, when clicking on an event the video jump to the time of the event.

4.5 Techniques and Algorithms used:

- **Stream Processing:** we refine the algorithm to process video frame by frame rather than batch processing as its efficient in terms of memory usage, this approach help us process a video of full match at once without any memory crash.
- **Yolov5:**

  We used a pretrained yolov5 model on the COCO dataset (detect 80 different obj) and then we finetuned it with a hundred iteration on football-players-detection Image Dataset to better detect and label objects in the match (players, referees and ball).

  We resizes the input image into 448x448 before going through the convolutional network, a 1x1 convolution is first applied to reduce the number of channels, which is then followed by a 3x3 convolution to generate a cuboidal output, the activation function under the hood is ReLU, except for the final layer, which uses a linear activation function, some additional techniques, such as batch normalization and dropout, respectively regularize the model and prevent it from overfitting.

- **ByteTrack:**

  We used ByteTrack to track multiple object in the frames as its a MOT algorithm, it takes object detections from YOLO (or other detectors) and links them across video frames to create consistent tracks for each object, it handles occlusions, disappearances, and new object appearances.

  It takes a list of detections from YOLO and return a list of tracks, each containing object id (unique identifier for each tracked object) and bounding box information for each frame the object appears in.

  ByteTrack uses a two-step approach to associate detections with tracks, step 1: High-Confidence Matching, uses Kalman filter to predict object positions in the next frame and matches predicted positions with high-confidence detections using IoU (Intersection over Union) for motion similarity, then IoU reflects the overlap between bounding boxes.
  Step 2: Low-Confidence Matching, objects not matched in step 1 (potentially occluded) are considered, These are matched with remaining detections, even those with lower confidence scores, This helps track objects during occlusions where confidence might drop.

- **Team mapping:**
  We implement a technique for team mapping using kmeans clustering, we first check if the that player is assigned to a team to save the time of mapping it to a team, this approach reduce the time of the task in a notable way, if the player is not assigned to a

team then we cluster it using kmeans to one of 2 clusters (team 1 or team 2), the kmeans trained on cropped images for players in the video and cluster them into 2 clusters. Applying this approach enabled precise calculations of possession statistics for each team, including the number of passes executed, as well as individual player possession metrics. These developments mark a substantial enhancement in our ability to extract detailed insights from match footage.

- **Local Spatial-Temporal Feature Extractor (F):**
  We extracts spatial-temporal features from each frame in the video sequence and utilizes a lightweight 2D Convolutional Neural Network (CNN) with Gate Shift Modules (GSM) to capture subtle motion and visual differences between neighbouring frames, then Ensures that each frame's features capture both spatial and temporal information crucial for event recognition.

- **Long-term Temporal Reasoning Module (G):**
  We utilizes a 1-layer bidirectional Gated Recurrent Unit (GRU) network to gather long-term temporal information and Processes dense per-frame features obtained from the feature extractor to capture temporal dependencies and context across the entire video sequence, then Outputs class predictions for each frame, including a 'background' class for frames where no event is detected.
- **Per-frame Cross-Entropy Loss:**
  We optimizes the model's classification performance using per-frame cross-entropy loss and the model's predictions with ground-truth labels for each frame and computes the loss based on the classification error which helps in training the model to accurately spot events in the video sequence.

4.6 Stepup Configuration (hardware):

- **Kaggle and Google Colab GPU:** We used GPU T4x2 or GPU P100 in preprocessing and finetuning processes in order to make them faster.
- **Required inspections for machine:**
  - CPU: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
  - RAM: 8GB
  - Nvidia GTX 1650

4.7 Experimental Results:

YOLO architecture:

Our YOLOv5 model was pretrained on COCO dataset and we use transfer learning on football-players-detection Image Dataset to better detect ant label players, referees and ball, it achieve accuracy of 88% on detection using MOTA metric.
We required a high GPU capability for the transfer learning so we finetuned the model on google colab and uploaded our datasets.

YOLOv5's architecture consists of three main parts:

- Backbone: This is the main body of the network. For YOLOv5, the backbone is designed using the New CSP-Darknet53 structure, a modification of the Darknet architecture used in previous versions.
- Neck: This part connects the backbone and the head. In YOLOv5, SPPF and New CSP-PAN structures are utilized.
- Head: This part is responsible for generating the final output. YOLOv5 uses the YOLOv3 Head for this purpose.

YOLOv5 employs various data augmentation techniques to improve the model's ability to generalize and reduce overfitting. These techniques include: Mosaic Augmentation, Random Horizontal Flip, HSV Augmentation, MixUp Augmentation, Random Affine Transformations and Copy-Paste Augmentation.

YOLOv5 applies several sophisticated training strategies to enhance the model's performance. They include:

- **Multiscale Training**: The input images are randomly rescaled within a range of 0.5 to 1.5 times their original size during the training process.
- **AutoAnchor**: This strategy optimizes the prior anchor boxes to match the statistical characteristics of the ground truth boxes in your custom data.

- **Warmup and Cosine LR Scheduler**: A method to adjust the learning rate to enhance model performance.
- **Exponential Moving Average (EMA)**: A strategy that uses the average of parameters over past steps to stabilize the training process and reduce generalization error.
- **Mixed Precision Training**: A method to perform operations in half-precision format, reducing memory usage and enhancing computational speed.
- **Hyperparameter Evolution**: A strategy to automatically tune hyperparameters to achieve optimal performance.

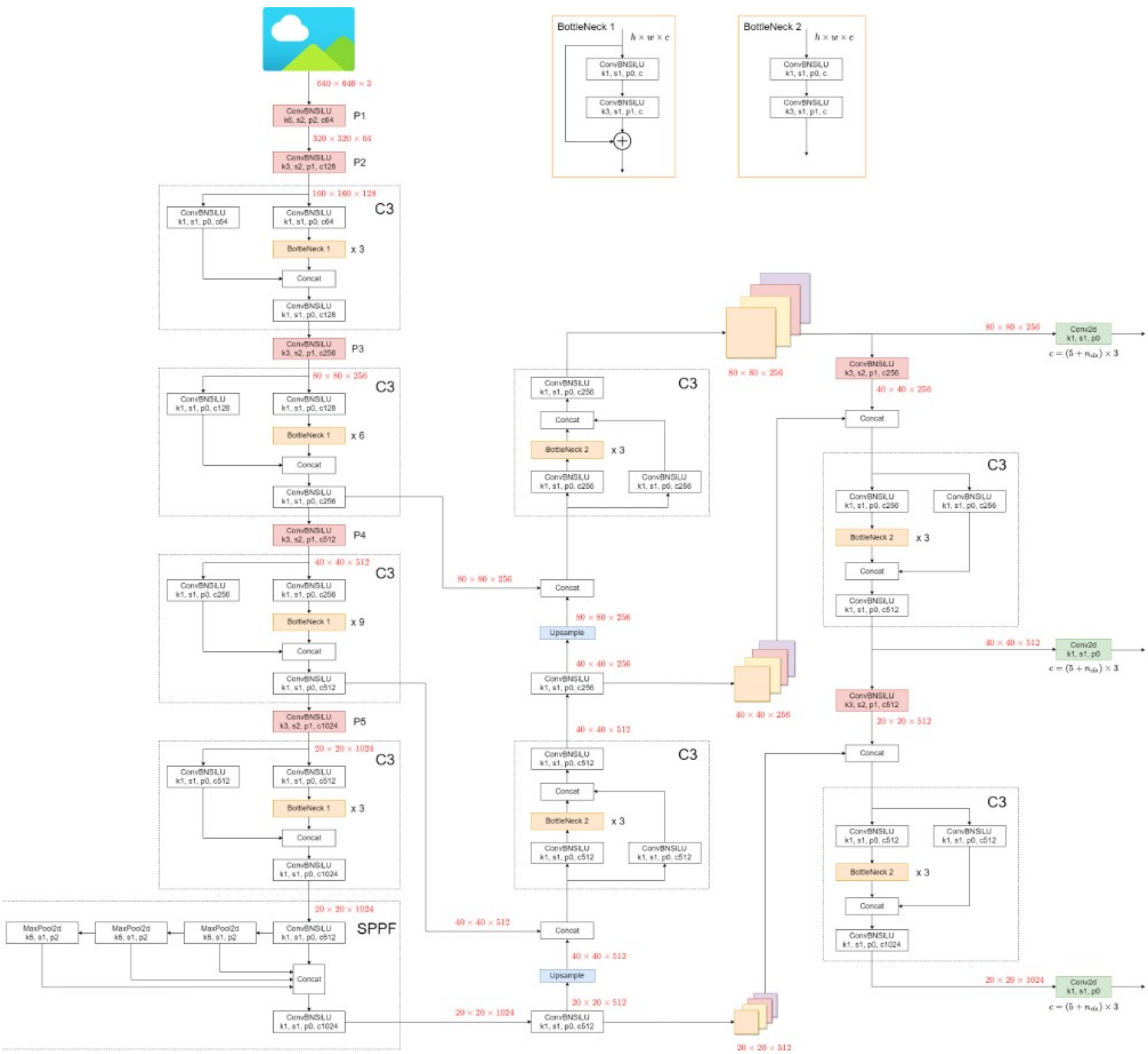

Figure 4.6 YOLO Architecture

Bytetrack Architecture:

After detecting the object using YOLO we use its output as input to Bytetrack.
Its output is a list of tracks, each containing: Object ID (unique identifier for each tracked object) and Bounding box information for each frame the object appears in.

ByteTrack uses a two-step approach to associate detections with tracks.
Step1:

- Uses Kalman filter to predict object positions in the next frame.
- Matches predicted positions with high-confidence detections using IoU (Intersection over Union) for motion similarity.
- IoU reflects the overlap between bounding boxes.

Step 2: Low-Confidence Matching:

- Objects not matched in step 1 (potentially occluded) are considered.
- These are matched with remaining detections, even those with lower confidence scores.
- This helps track objects during occlusions where confidence might drop.

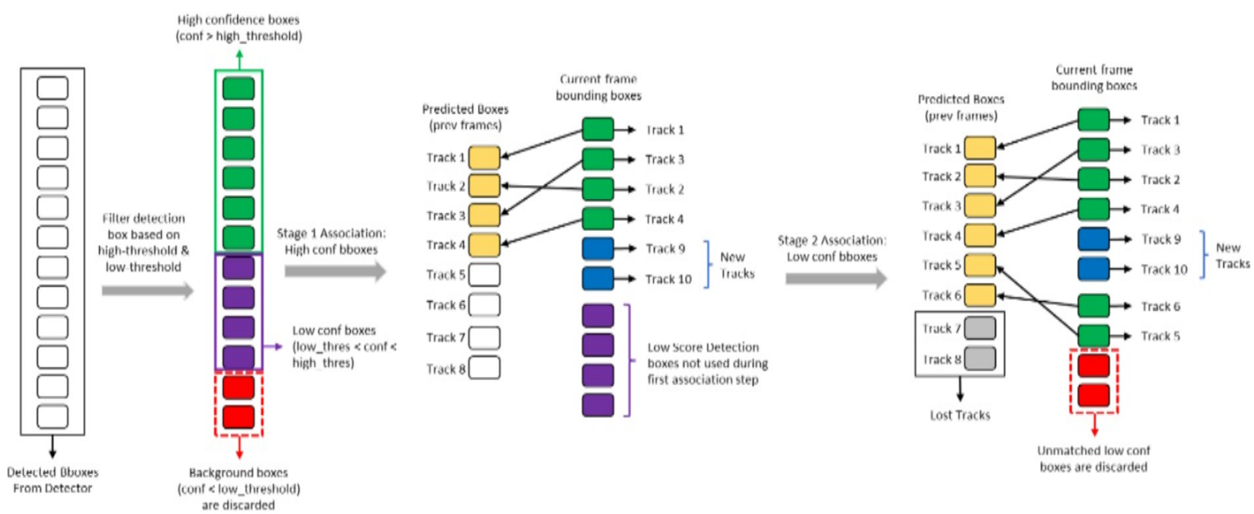

Figure 4.7 Bytetrack Architecture

Evaluation:

Algorithm	Setup	HOTA	DetA	AssA	MOTA
DeepSORT	w/ GT	69.552	82.628	58.668	**94.844**
FairMOT	w/ GT	-	-	-	-
ByteTrack	w/ GT	**71.500**	**84.342**	**60.718**	94.572
DeepSORT	w/o GT	36.663	40.022	33.759	33.913
FairMOT	w/o GT	43.911	46.317	41.778	50.698
ByteTrack	w/o GT	47.225	44.489	50.257	31.741
FairMOT-ft	w/o GT	**57.882**	**66.565**	**50.492**	**83.565**

Figure 4.8 comparison between bytetrack and other open-source models

Result:

After applying tracking and detection the output of each video frame be like:

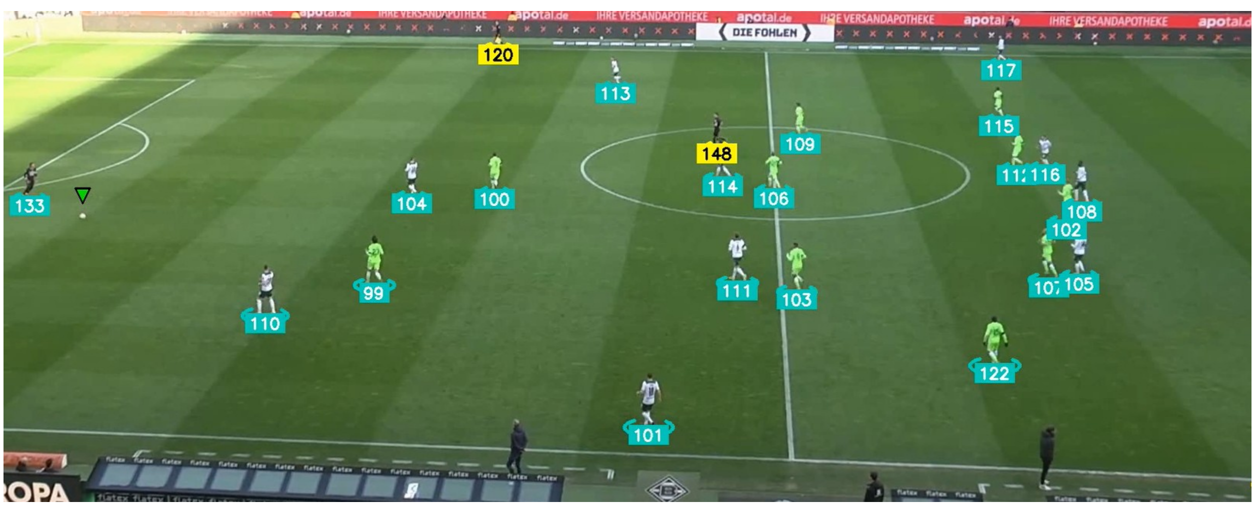

Figure 4.9output of tracking with detection

JHONG Action Spotting Architecture :
It Use 2 Data sets (Soccer Net , Sports Dataset Compilation).

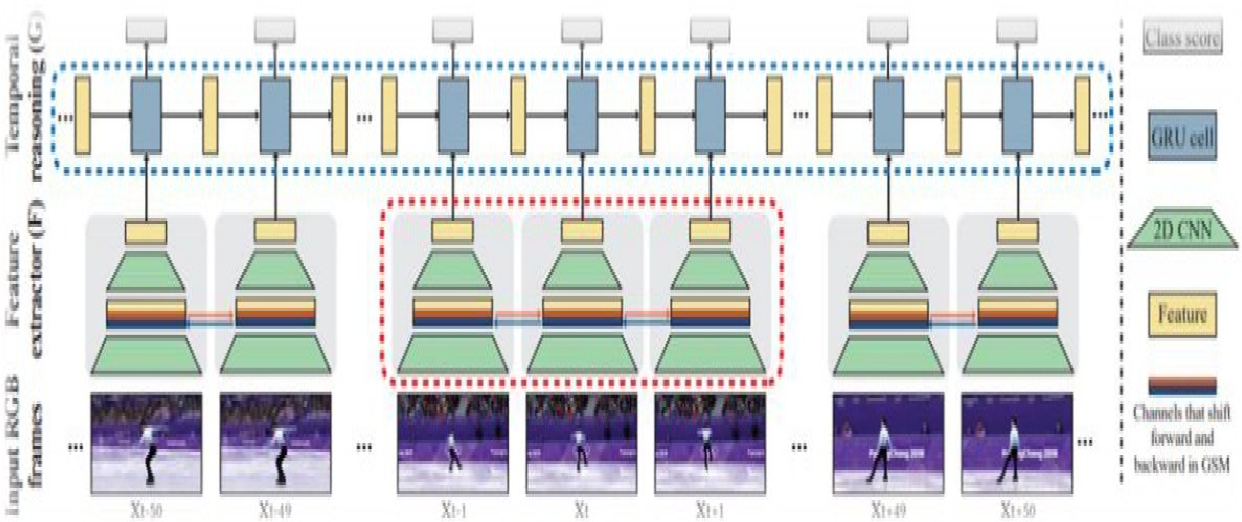

Figure 4.10JHONG Action Spotting Architecture

1-Local Spatial-Temporal Feature Extractor (F):

• Extracts spatial-temporal features from each frame in the video sequence.

• Utilizes a lightweight 2D Convolutional Neural Network (CNN) with Gate Shift Modules (GSM) to capture subtle motion and visual differences between neighboring frames.

• Ensures that each frame's features capture both spatial and temporal information crucial for event recognition.

2. Long-term Temporal Reasoning Module (G):

• Utilizes a 1-layer bidirectional Gated Recurrent Unit (GRU) network to gather long-term temporal information.

• Processes dense per-frame features obtained from the feature extractor to capture temporal dependencies and context across the entire video sequence.

• Outputs class predictions for each frame, including a 'background' class for frames where no event is detected.

3. Per-frame Cross-Entropy Loss:

• Optimizes the model's classification performance using per-frame cross-entropy loss.

• Compares the model's predictions with ground-truth labels for each frame and computes the loss based on the classification error.

• Helps in training the model to accurately spot events in the video sequence.

Evaluation:

Table 4: **Average-mAP @ t for tolerances in seconds.** SOTA in **bold**. We show the top results from the CVPR 2021 and 2022 SoccerNet Action Spotting challenges. ‡ indicates challenge results — trained on the train, validation, and test splits. Shown and unshown refer to whether actions are visible; E2E-Spot is better at detecting the former, but Soares et al. [51] is superior at the latter.

Average-mAP @ tolerances	Test split Tight (1–5 s)	Test split Loose (5–60 s)	Challenge split Tight (1–5 s)	Challenge split Shown	Challenge split Unshown
RMS-Net [57]	28.83	63.49	27.69	-	-
NetVLAD++ [22]	-	-	43.99	-	-
Zhou et al. [71] (2021 challenge; 1st)	47.05	73.77	49.56	54.42	45.42
‡Soares et al. [51] (2022 challenge; 1st)	-	-	‡**67.81**	‡72.84	‡**60.17**
E2E-Spot 200MF	61.19	73.25	63.28	70.41	45.98
E2E-Spot 800MF	61.82	74.05	66.01	72.76	51.65
‡E2E-Spot 800MF (2022 challenge; 2nd)	-	-	‡66.73	‡**74.84**	‡53.21

Figure 4.11 action spotting model compared with other open-source models

Penalty	0.6117246389973663	Ball out of play	0.7597325074474494
Kick-off	0.19144476088758855	Throw-in	0.7390451138504714
Goal	0.747110202356527	Foul	0.33199095178603644
Substitution	0.31609923500818427	Indirect free-kick	0.22296267908741838
Offside	0.29524319380611685	Direct free-kick	0.5463329480875319
Shots on target	0.39556490215213125	Corner	0.8307859501893997
Shots off target	0.4949210089405469	Yellow card	0.6416569714054904
Clearance	0.3591693015295976	Red card	0.0
		Yellow->red card	0.0

Figure 4.12 accuracy per event class

5- User Manual

5.1 Home:

The Home Page is the first screen that you see when you open the website, its purpose is to enable you to upload match video and generate game statistics.

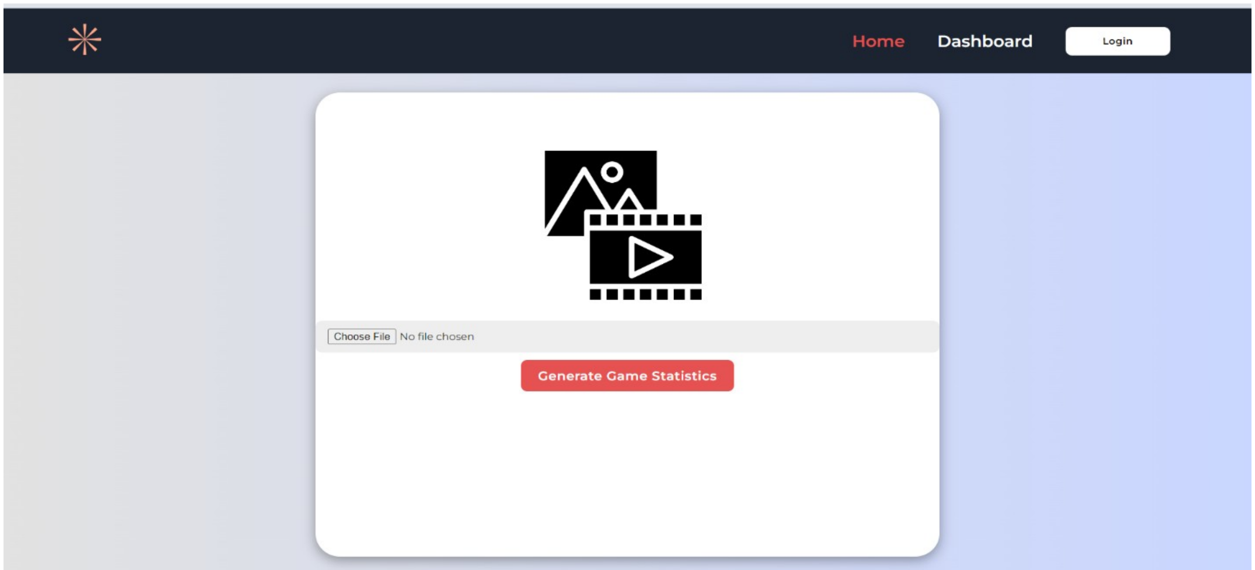

Figure 5.13Home Page

You can also access the home page by clicking on the logo.

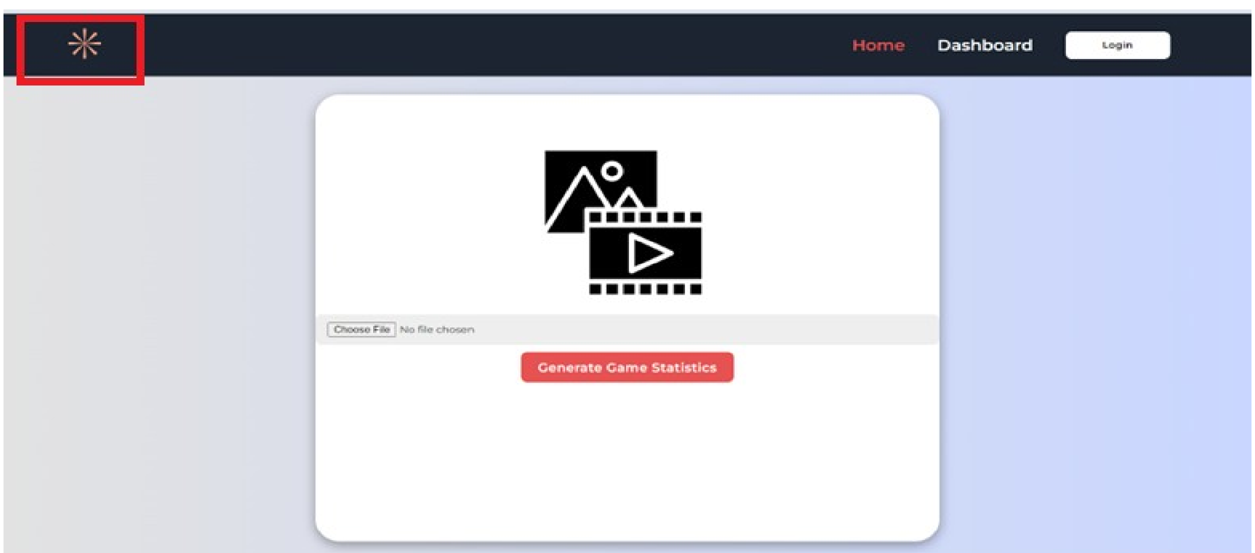

Figure 5.14 Access the home page by clicking on the logo

To upload a match video:

1- Click the "Choose File" button or the image.

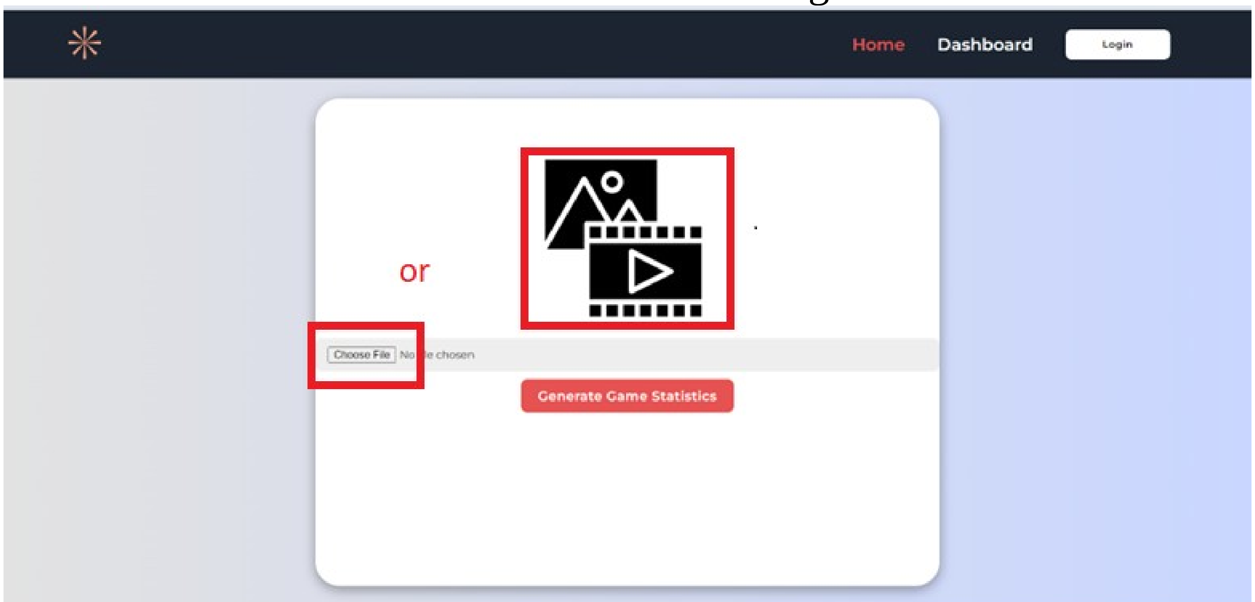

Figure 5.15 Upload Video

2- Select the file you want to upload.

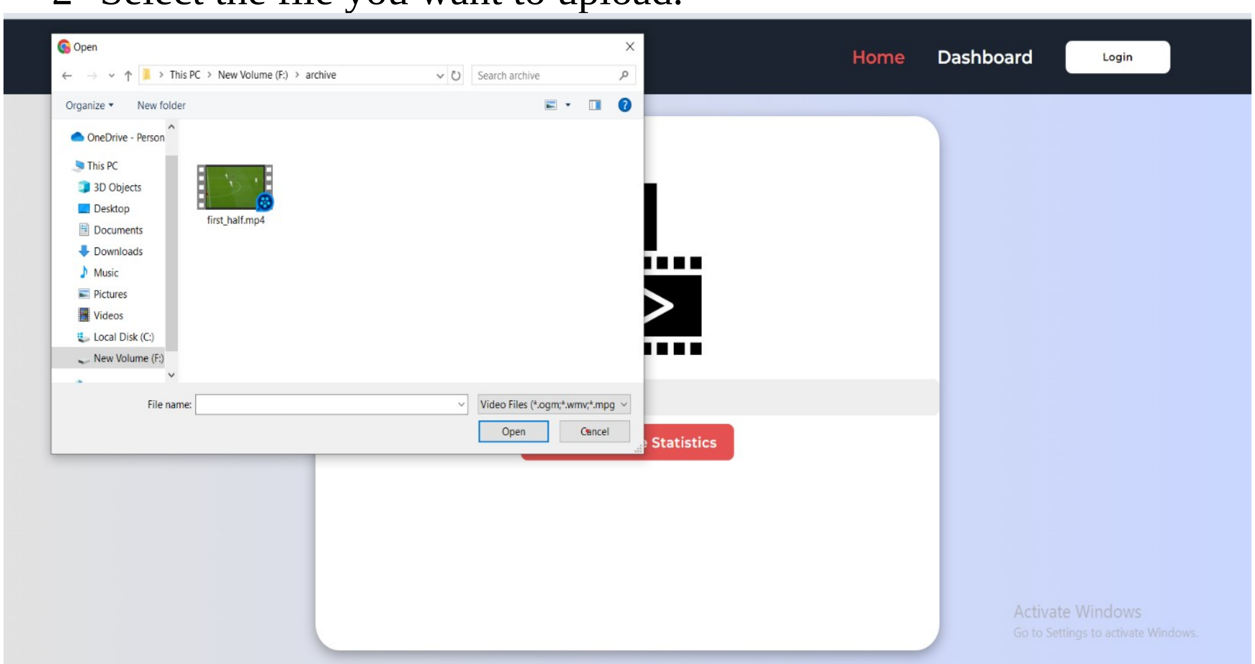

Figure 5.16 Select File

When user clicks “Generate Game Statistics” button, they will be directed to the dashboard page.

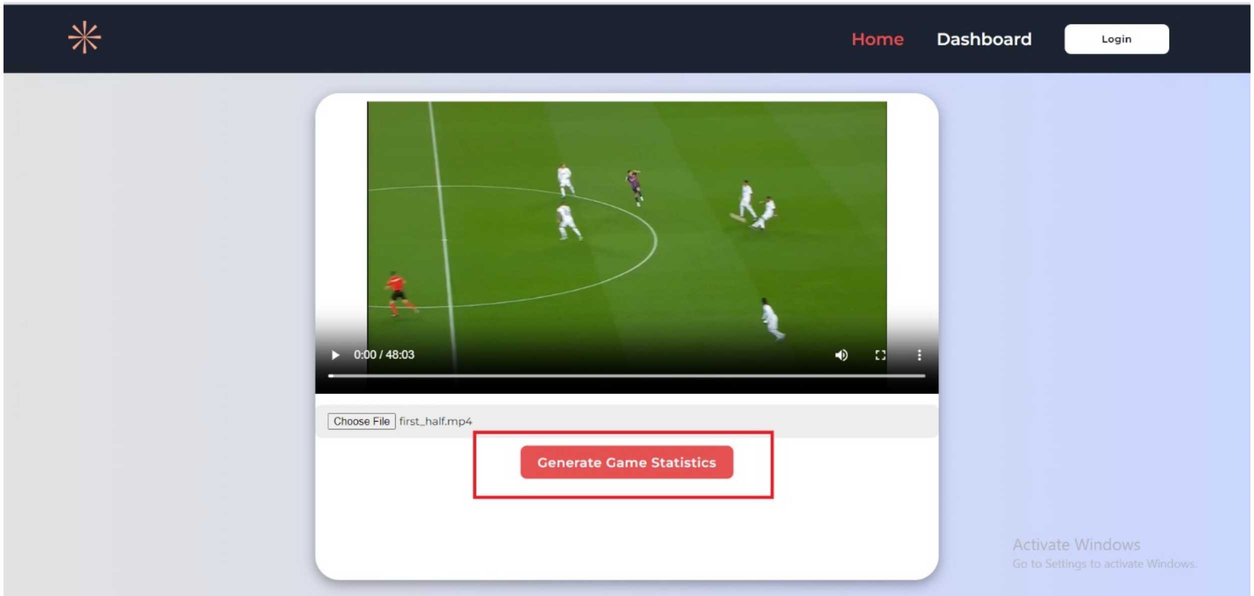

Figure 5.17 Clicking the 'Generate Game Statistics' button directs the user to the dashboard page

5.1.1 Dashboard:

When you click on the dashboard in the navigation bar without uploading video on the home page.

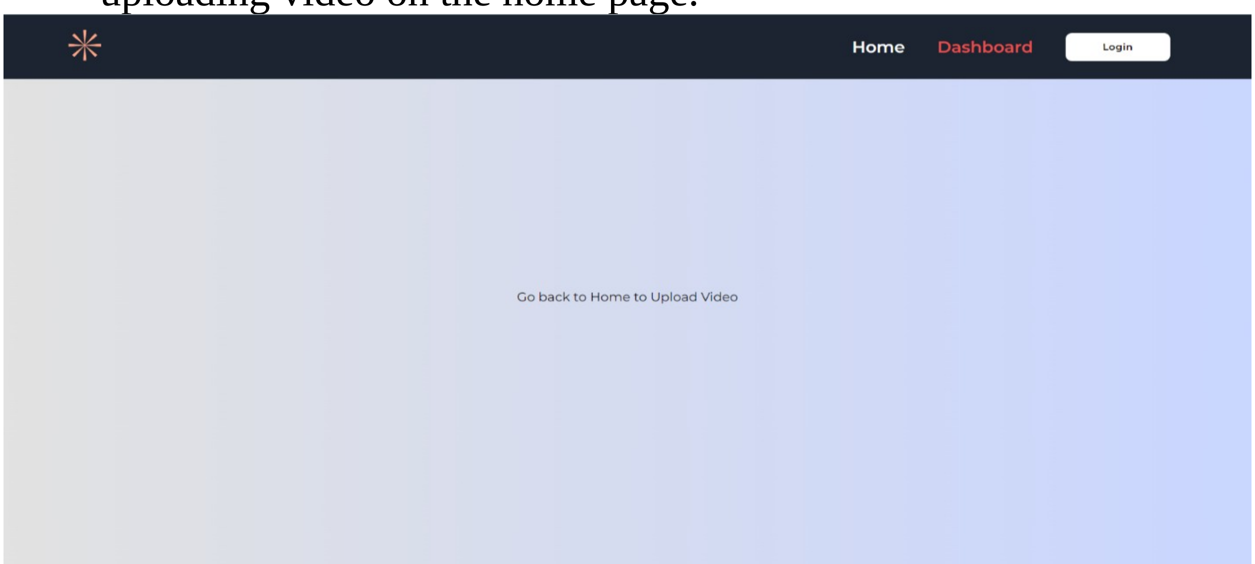

Figure 5.18 Clicking on the dashboard in the navigation bar without uploading a video

When you go back to the home page, upload a video, and click the "Generate Game Statistics" button.

The video illustrates each player mapped to their respective team, and the possession percentage for each team at this moment.

Users can select specific events to jump directly to those moments. For example, at 2 minutes and 28 seconds, the video highlights a player from Team 2 committing a foul, providing a clear reference to the player's team and action.

Figure 5.19 Video displays players mapped to teams, users navigate key events

The bar chart visualizes the possession precentage for each team during the match.

Figure 5.20 Bar Chart

This chart displays the main events that occurred during the match for both teams, including fouls, corners, and other key events.

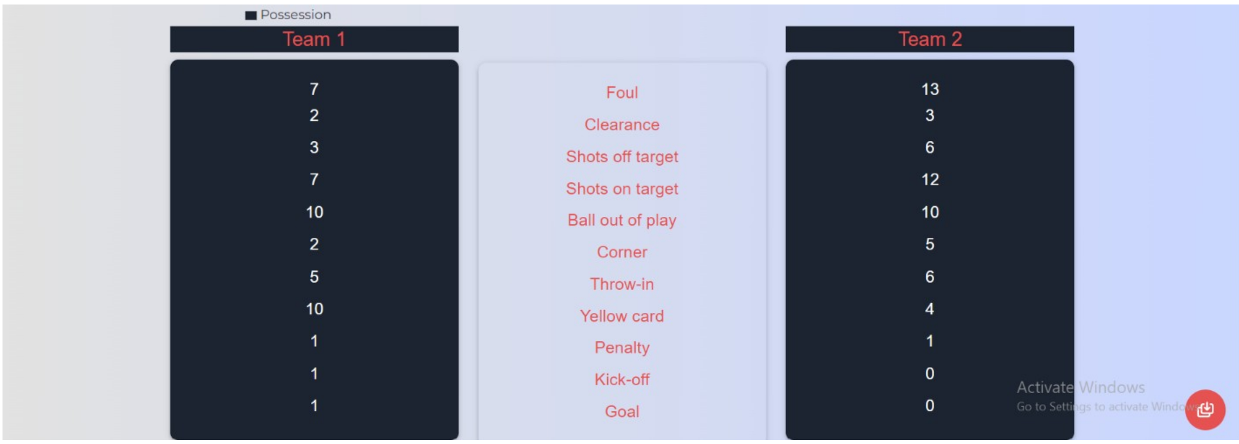

Figure 5.21 Chart displays the main events during the match

To save game statistics:

1- Login.

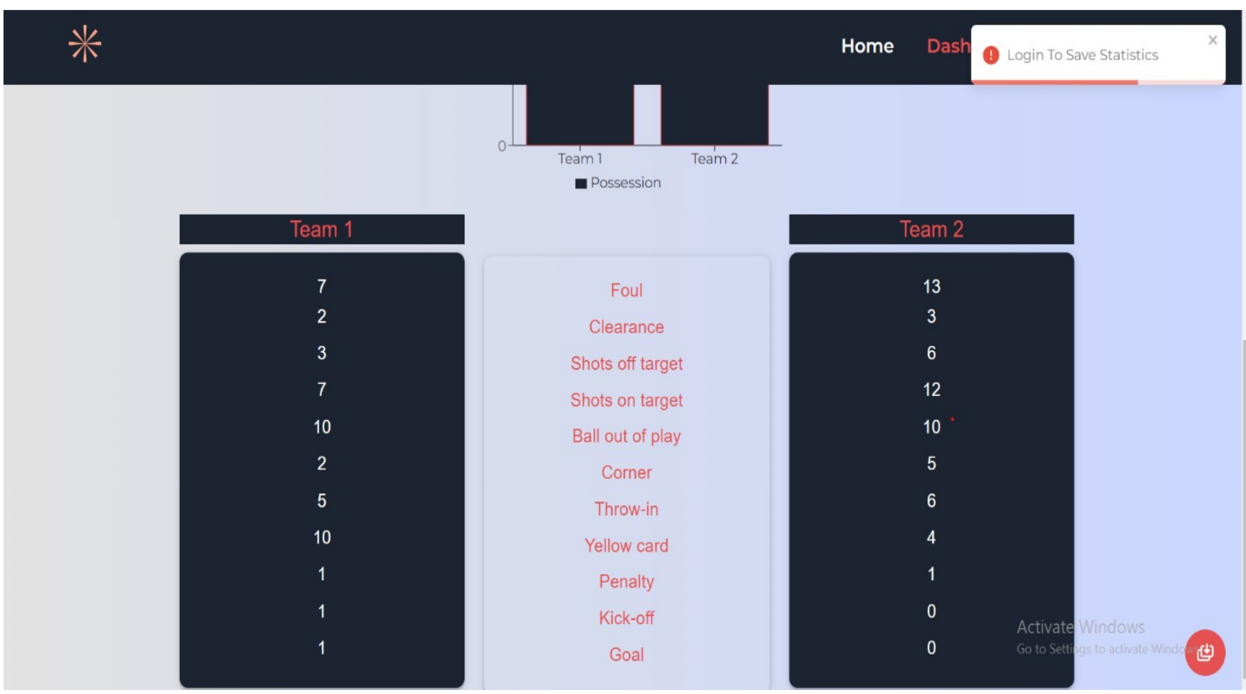

Figure 5.22 Failed to save game statistics

2- Click the save button.

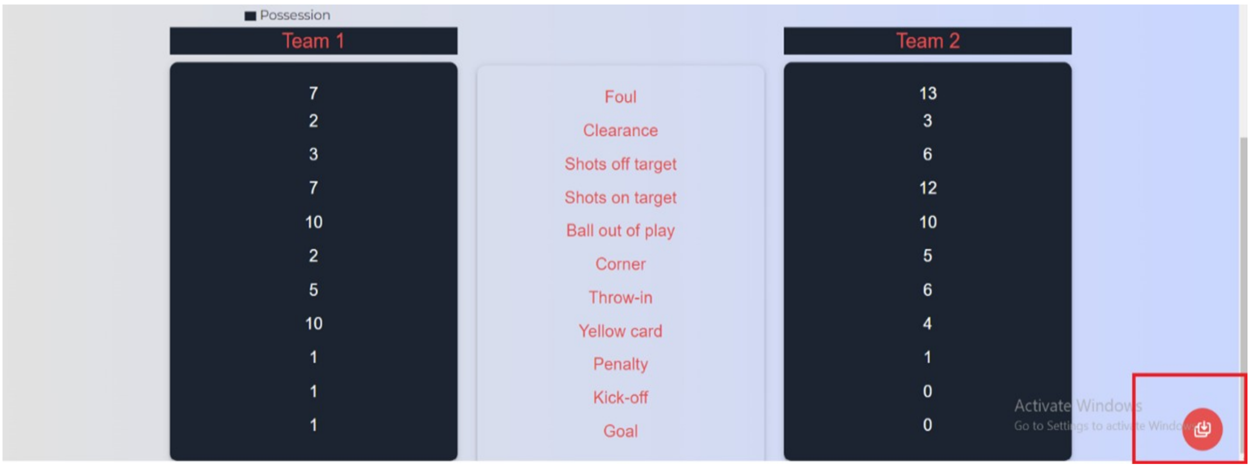

Figure 5.23 Clicking the save button

3- Enter a title.

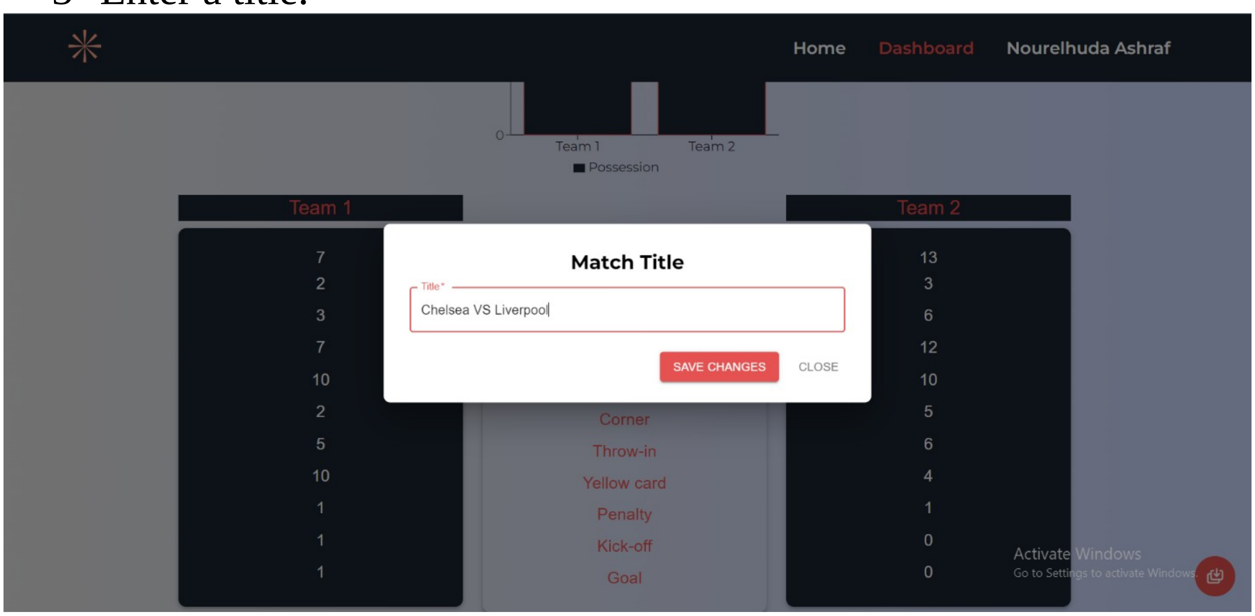

Figure 5.24 Entering a title

4- Click the “Save Change” button.

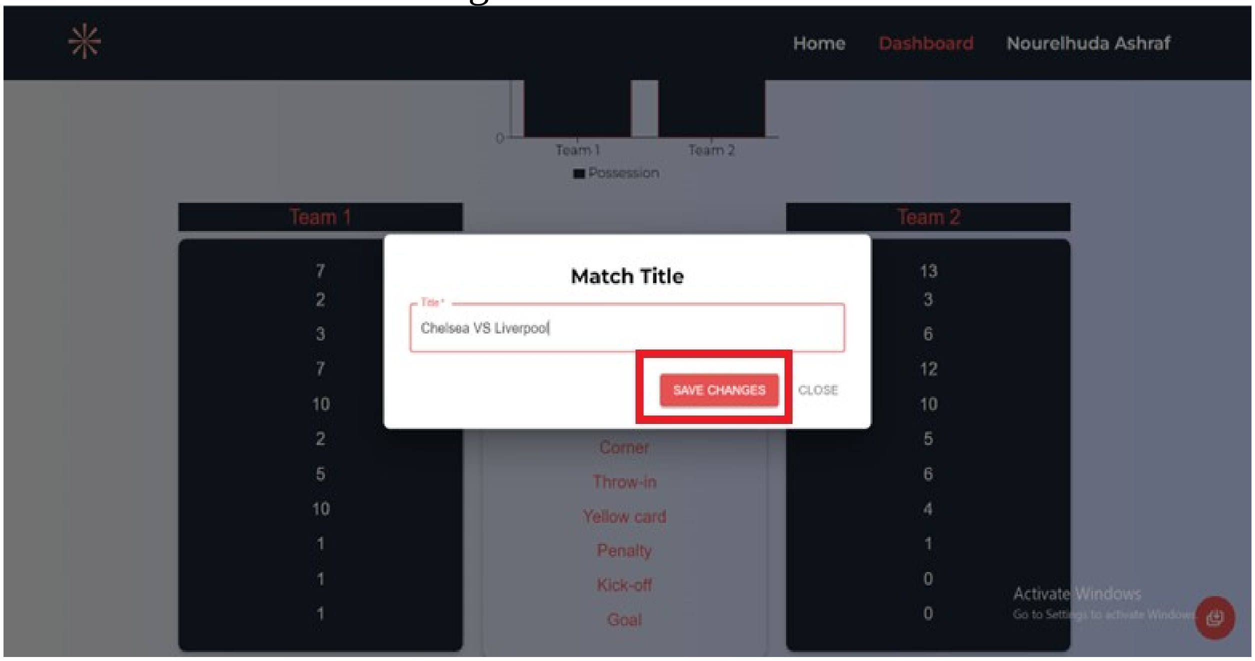

Figure 5.25 Clicking the 'Save Changes' button

5- The statistics will be saved successfully.

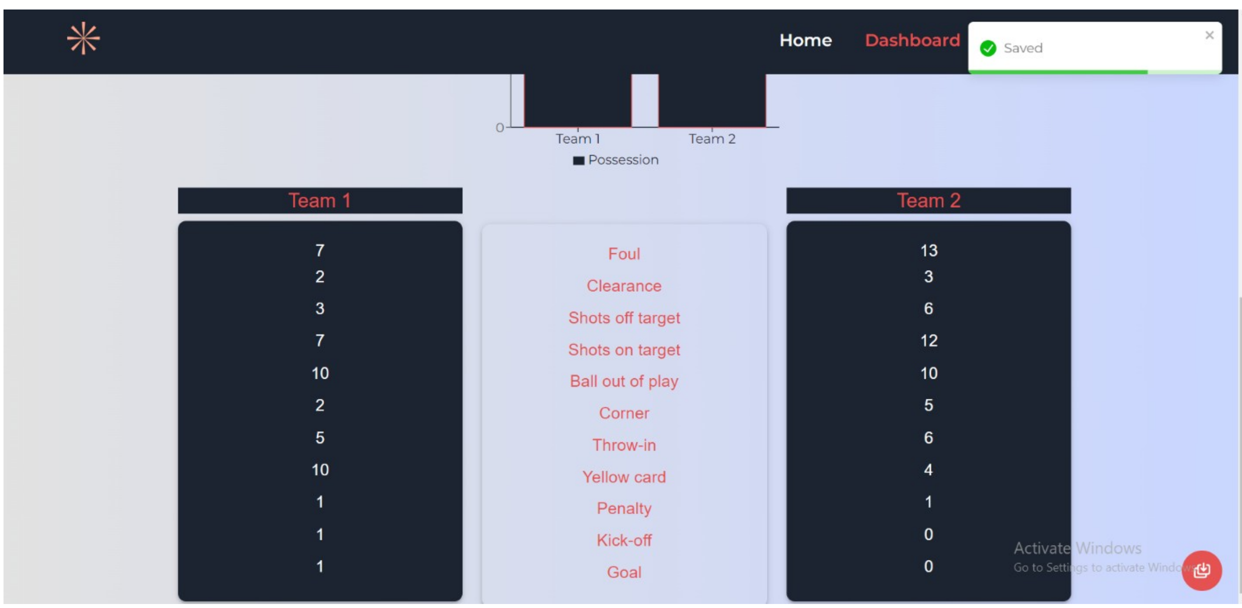

Figure 5.26 Saved Successfully

5.2 Login:

The user should login by entering the correct email and password that were previously entered while signing up.

After clicking on the "Login" button and verifying the user's provided credentials, the user will be directed to the Home page.

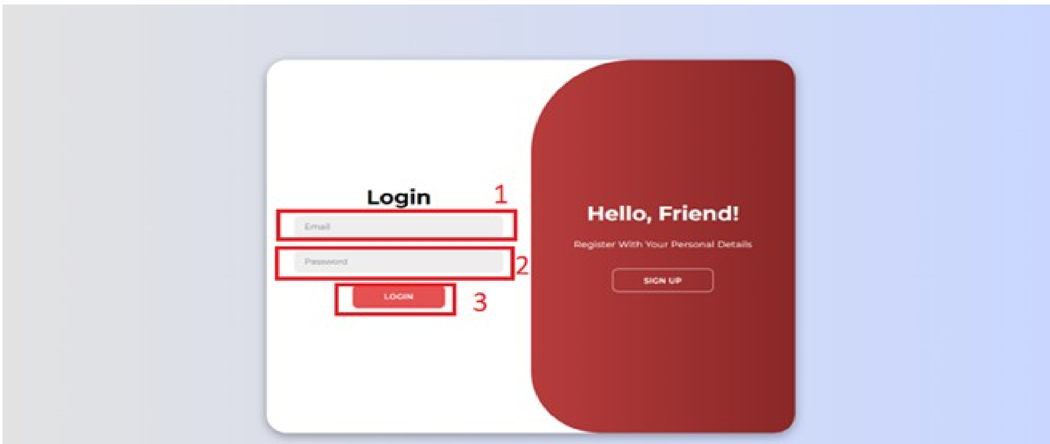

Figure 5.27 Login Interface

5.2.1 Sign Up:

The user registration process requires the user to provide their name, email, password.
Upon clicking the "sign up" button, the user will be directed to the Login page.
Example of valid input: A complete form.

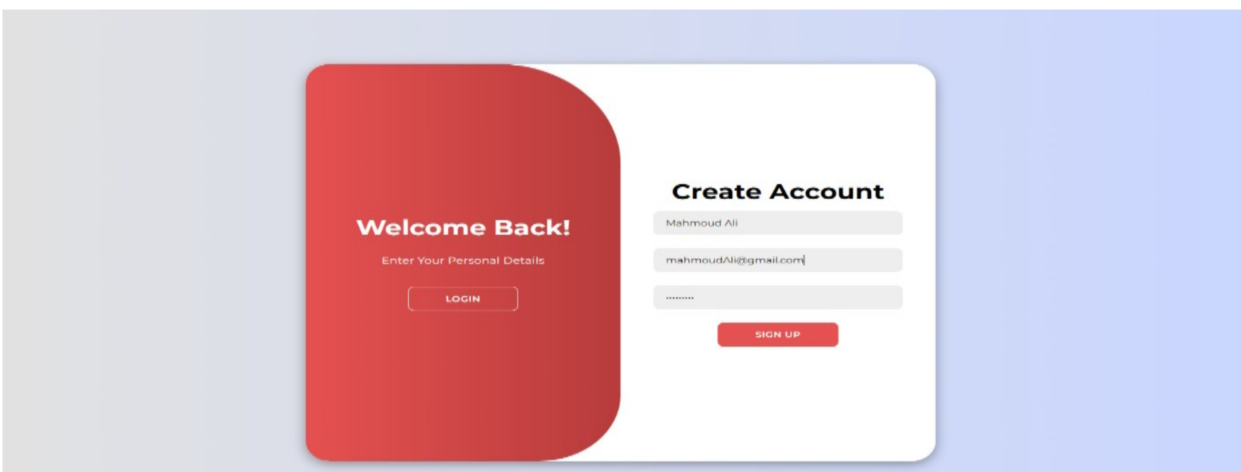

Figure 5.28 Sign Up Interface

Examples of invalid inputs: An incomplete form.

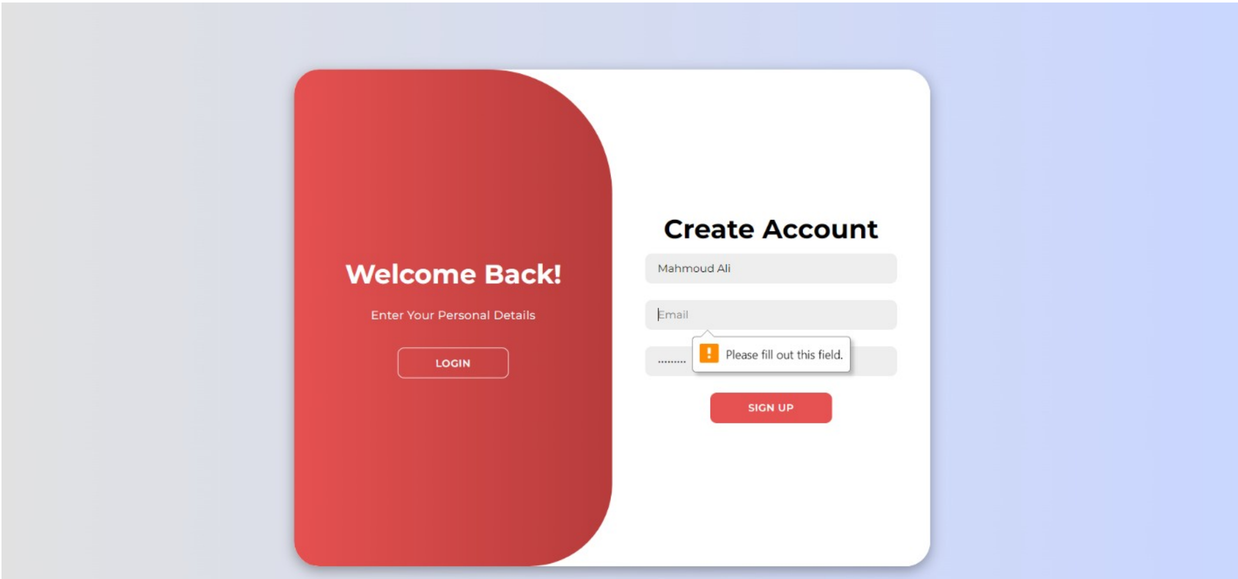

Figure 5.29 No Email

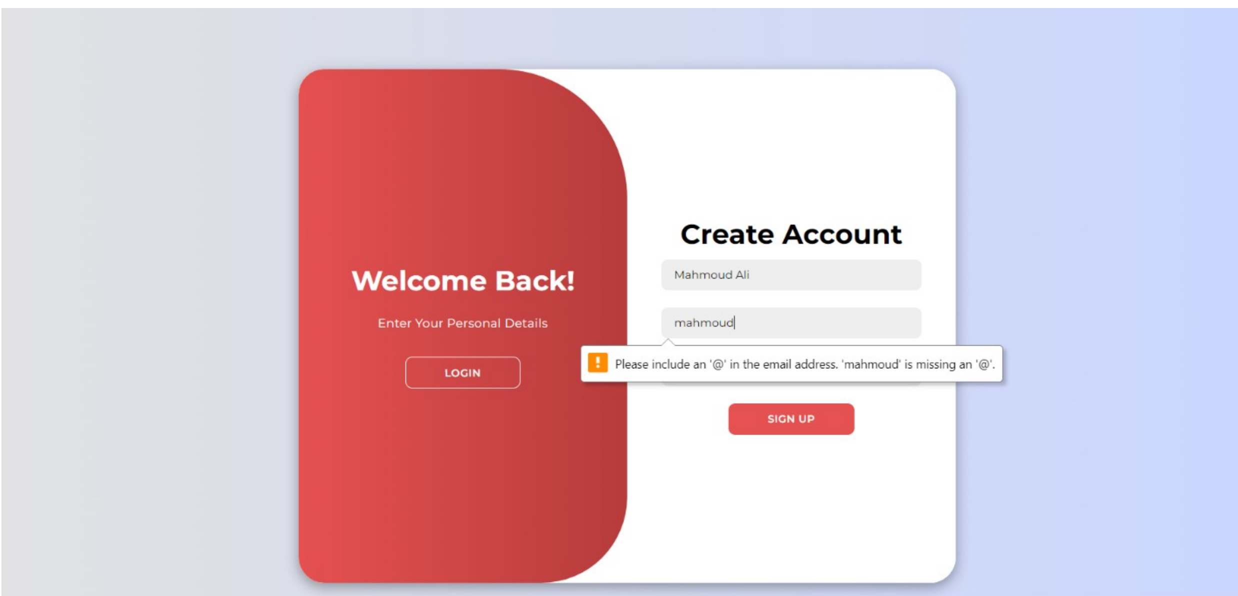

Figure 5.30 Invalid Email without @domain

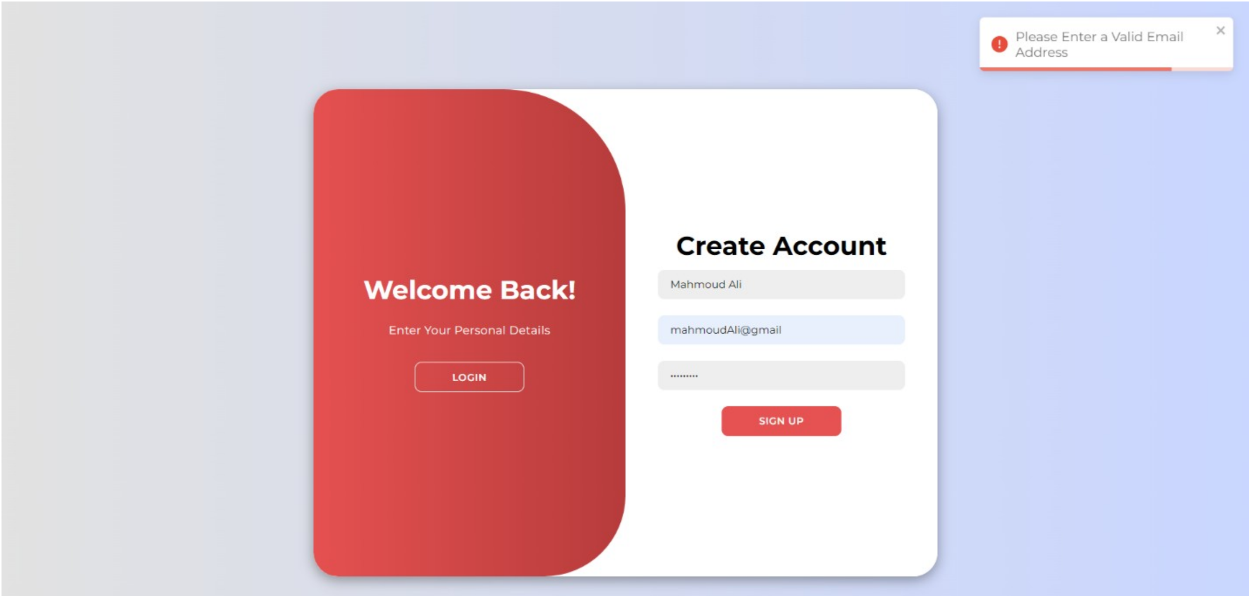

Figure 5.31 Incomplete Email

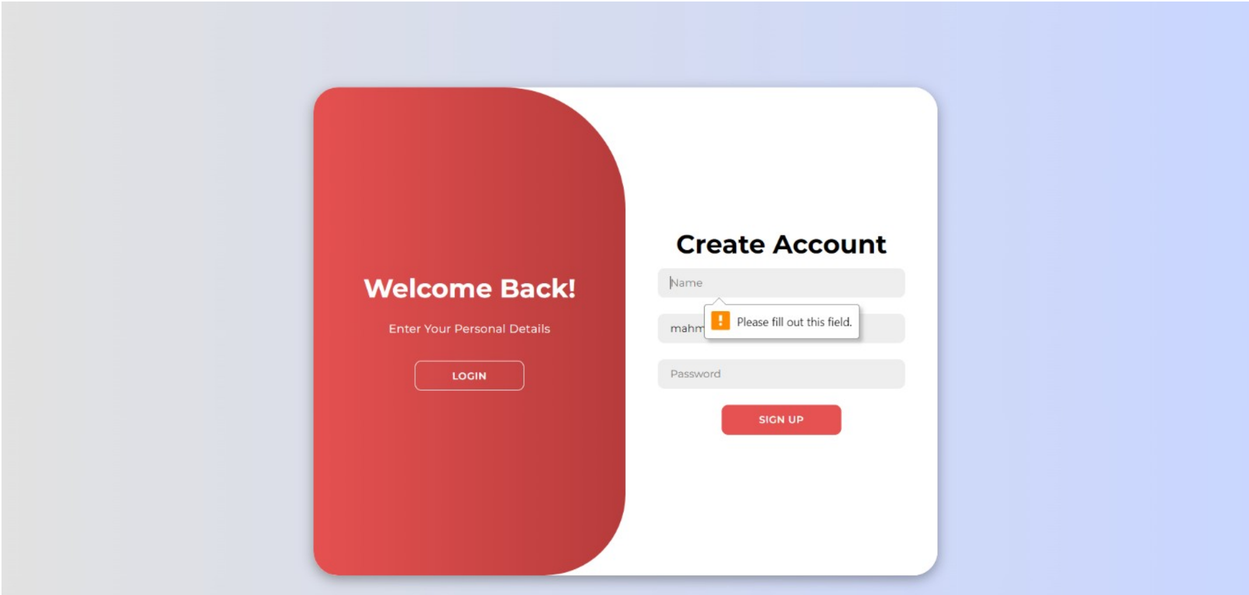

Figure 5.32 No name and password

5.3 My Profile:

Every user has an account where they can store and manage their personal information.
This account allows users to access and edit their data at any time, providing them with control and flexibility.

To access your profile, click on "Your Name" in the navigation bar, then select "My Profile" from the menu.

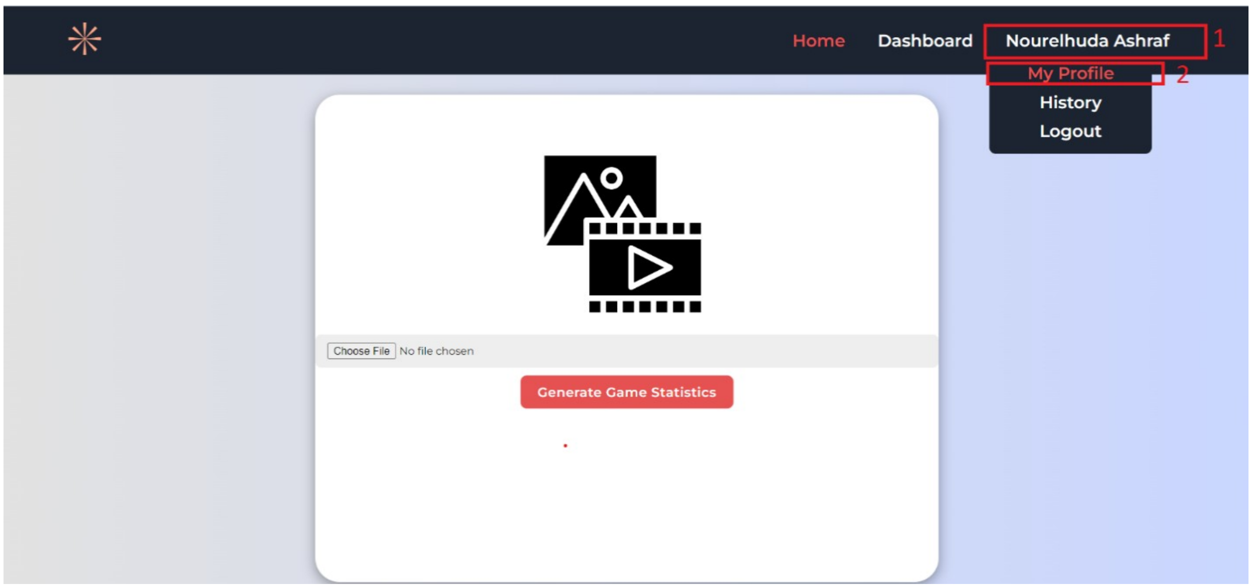

Figure 5.33 Access your profile by clicking on 'Your Name' in the navigation bar and selecting 'My Profile'

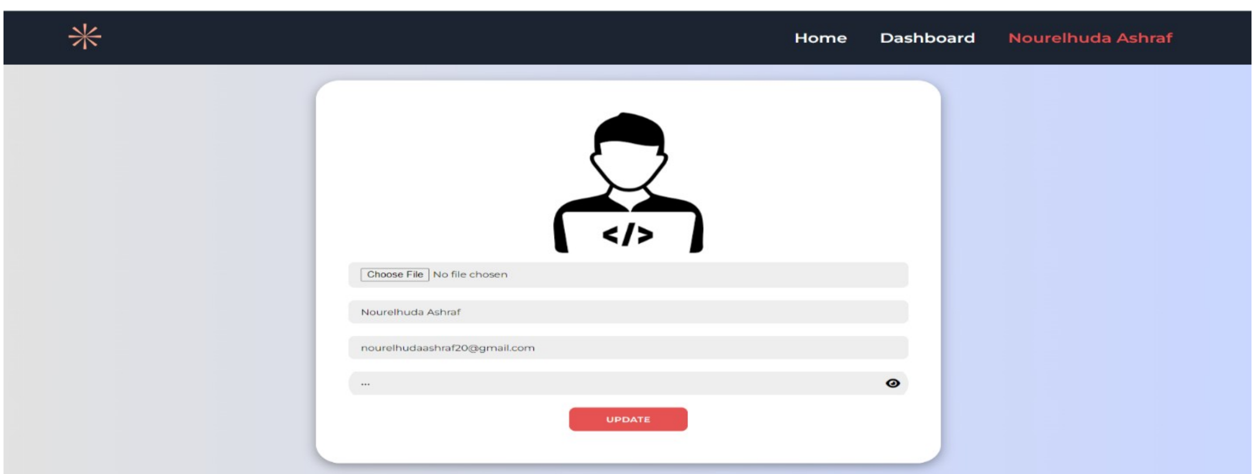

Figure 5.34 My Profile Interface

5.4 History:

Every user has a history where they can store and manage their saved game statistics. These statistics can be accessed and deleted by the user. To view your history, click on "Your Name" in the navigation bar, then select "History" from the menu.

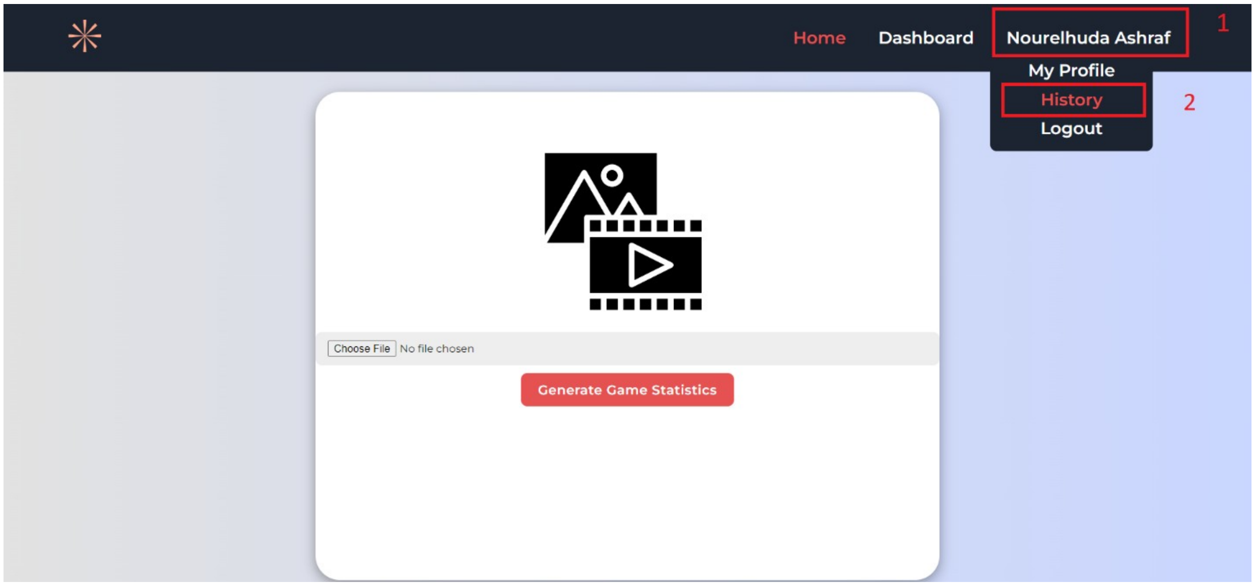

Figure 5.35 View your history by clicking on 'Your Name' in the navigation bar and selecting 'History'

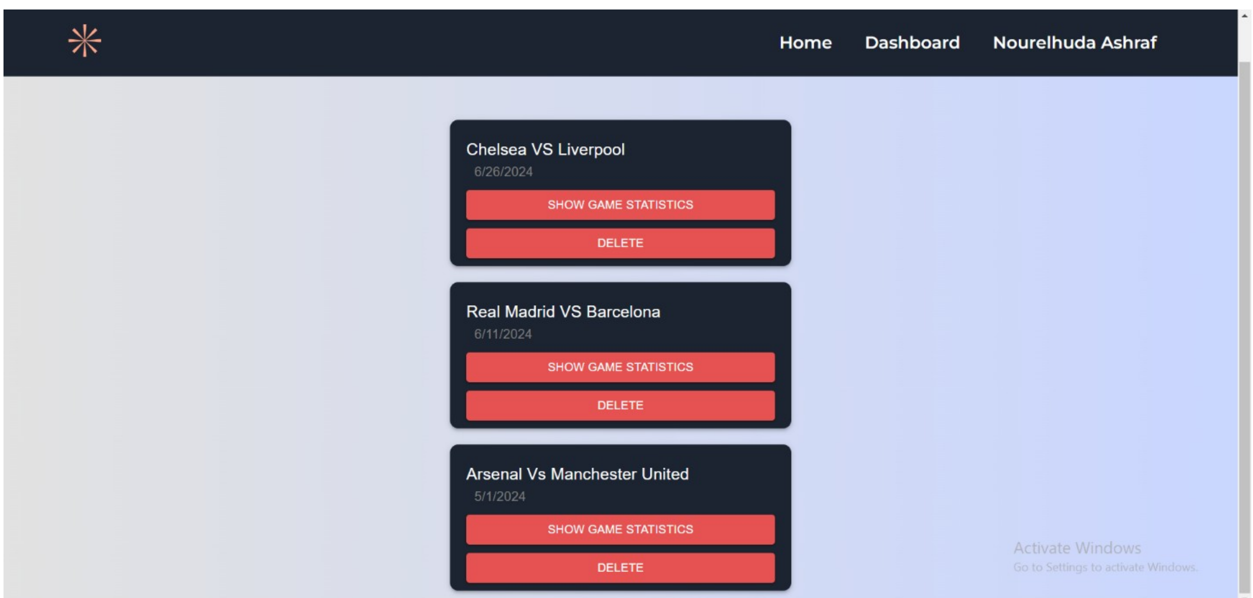

Figure 5.36 History Interface

To show game statistics of a specific match, press the "Show Game Statistics" button.

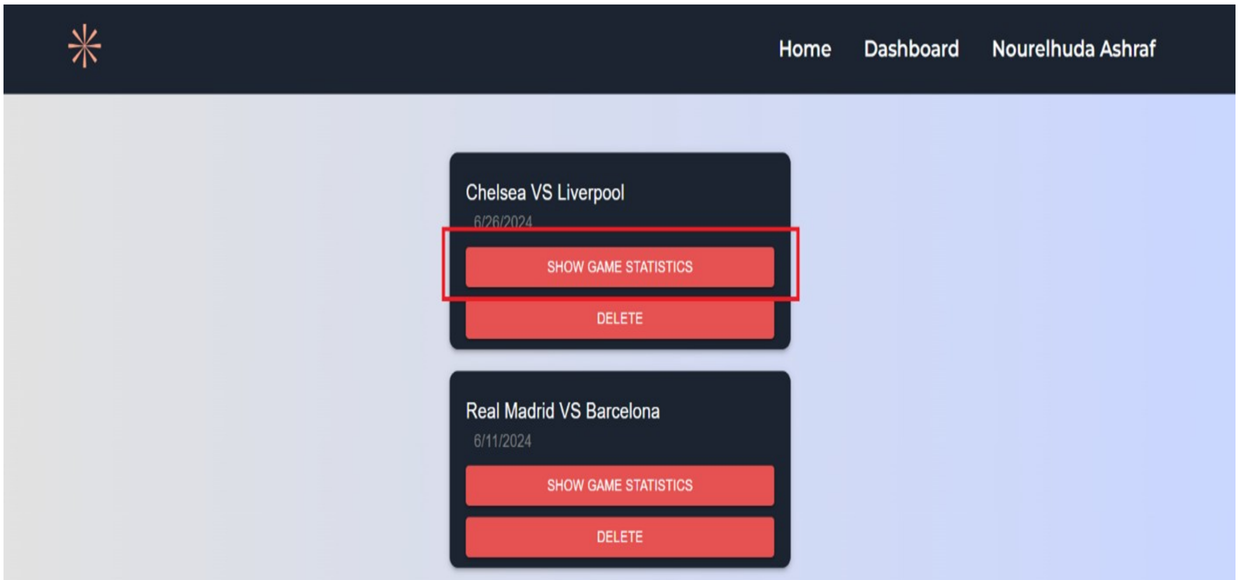

Figure 5.37 Show saved game statistics

To delete the game statistics, press the "Delete" button.

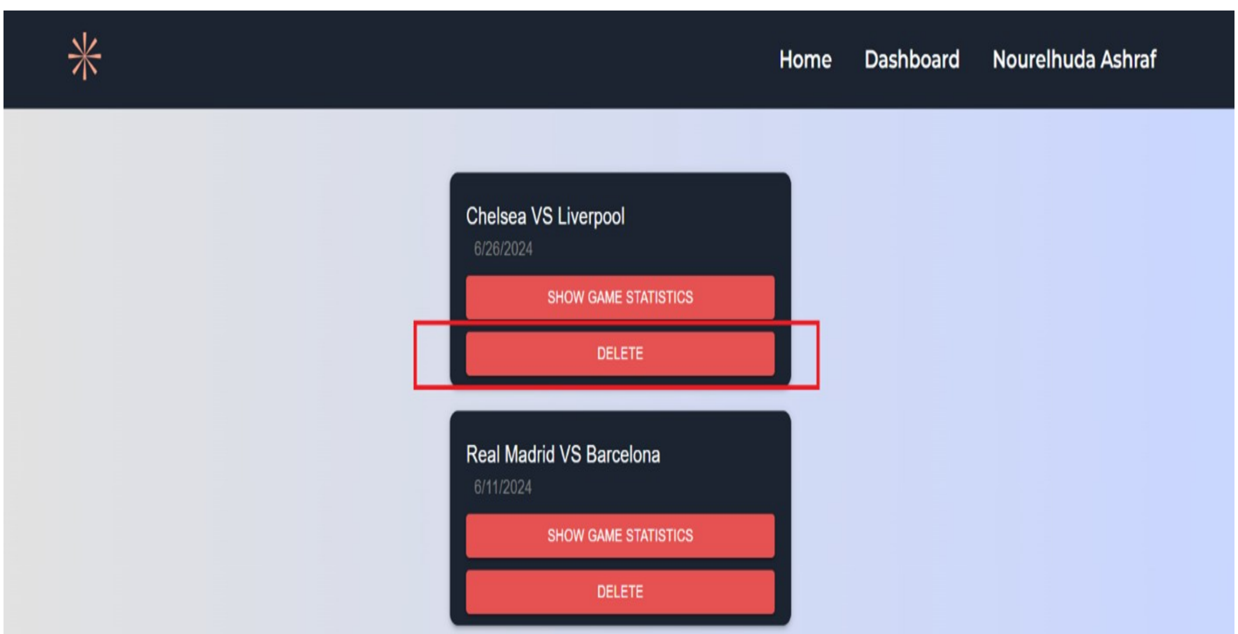

Figure 5.38 Delete game statistics

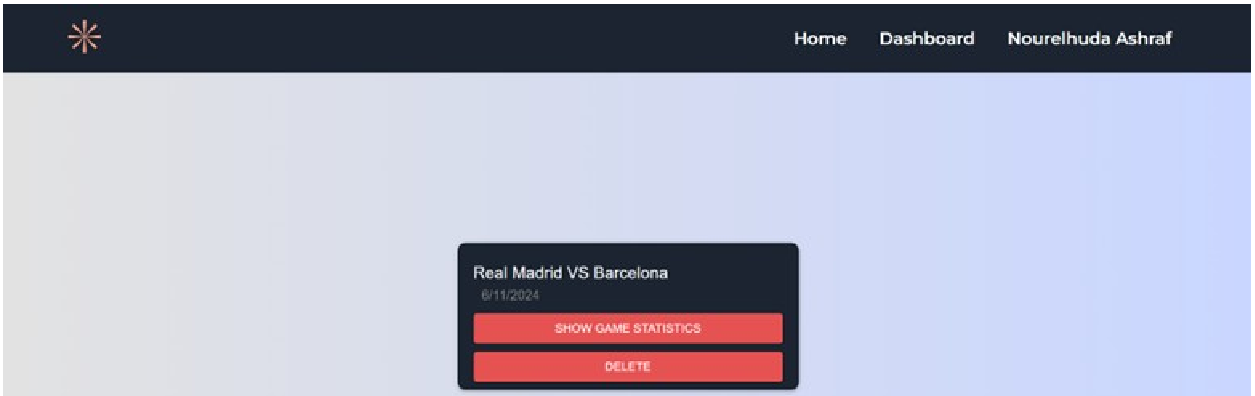

Figure 5.39 Deleted Successfully

5.5 Logout:

To logout, click “Your Name” in the navigation bar, then select “Logout” from the menu, you will be directed to the home page.

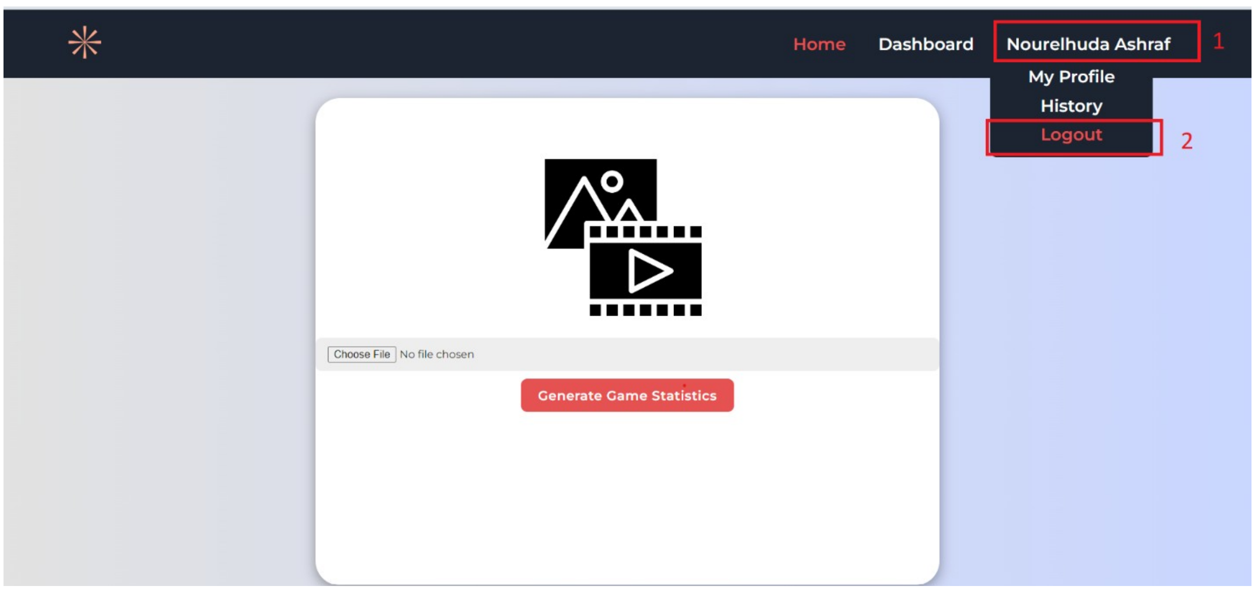

Figure 5.40 Logout by clicking on 'Your Name' in the navigation bar and selecting ‘Logout'.

6- Conclusion and Future Work

6.1 Conclusion

In conclusion, the development of an automated Football Matches Analysis system for calculating match insights represents a promising solution for the football community. The system we proposed automates statistical calculations, which can help individual players, clubs, and leagues improve match quality.

Our proposed system utilizes detection, tracking, and team mapping technology to map players to their teams and calculate statistics. It then employs event action detection, localization, and team mapping to generate highlights of important actions in the match, which are saved in the system's database.

We developed the system using a three-layer architecture and evaluated its accuracy and efficiency in processing and analysing input videos.

However, it is important to note that many features can be added to the system to enhance fan engagement and improve analysis and visualization. Therefore, future work should focus on integrating these features and ensuring the system's sustainability and reliability.

Overall, the development of an automated Football Matches Analysis system for extracting team stats represents a significant step towards addressing the lack of open-source work in football analysis. We believe that this system has the potential to make a positive impact and look forward to its implementation and adoption in the future.

6.2 Future Work

In future work, we plan to further improve the system's accuracy and efficiency. We also plan to integrate additional features such as calculating different statistics for each player, Jersey number recognition, field localization and dense video captioning.

Furthermore, we aim to collaborate with relevant authorities to implement it in the Egyptian Premier League. The project aims to assist in calculating team statistics, which will help improve team performance, automate the task, and speed up the calculation of statistics.

References

- Hong, J., Zhang, H., Gharbi, M., Fisher, M., and Fatahalian, K., "Spotting Temporally Precise, Fine-Grained Events in Video," in Proceedings of the European Conference on Computer Vision (ECCV), 2022.
- Iromul, "Ball Action Spotting," in Proceedings of the European Conference on Computer Vision (ECCV), 2023.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
- Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 1516-1525.
- Wojke, N., Bewley, A., and Paulus, D., "Simple Online and Realtime Tracking with a Deep Association Metric," in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645-3649