



Ain Shams University

Faculty of Computer Science and Information Sciences

Scientific Computing Department

BEIN-SPORT

This documentation submitted as required for the degree of bachelor's in computer and Information Sciences.

By

Bassant Salem Mahmoud

Scientific Computing Department

Tasneem Bahaa Eldin

Scientific Computing Department

Shaimaa Hesham

Scientific Computing Department

Ahmed Magdy Ahmed

Scientific Computing Department

Bola Milad Shokry

Scientific Computing Department

Jonathan mamdouh monir

Scientific Computing Department

Under Supervision of

Dr. Manal Mohsen Tantawi

Associate Professor at Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Aya Naser

Teaching Assistant of Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

**Cairo
June 2025**

Acknowledgements

First and foremost, we thank the Merciful Allah, who has given everything to prepare and present this work. He gave us strength in my times of weakness, hope in our times of despair, and patience in our troubles. He gave us inspirational signs to overcome all the difficulties I faced

We're profoundly grateful to our parents and families for their unwavering support and encouragement throughout our years of study. Their love and sacrifices have been the foundation of our success, and we hope to one day repay their kindness and dedication.

Our deepest appreciation goes to our esteemed supervisor, Dr. Manal Mohsen Tantawi, whose unwavering support and guidance have been invaluable. Their mentorship, insightful advice, and boundless dedication were instrumental in overcoming the challenges we encountered. Without their expertise and encouragement, navigating through this project would have been exceedingly daunting.

Special thanks are extended to T.A. Aya Naser for her invaluable assistance and unwavering support, which greatly enriched our project's development. We also extend our sincere appreciation to all individuals who played a part, no matter how big or small, in the success of our endeavor.

Lastly, we acknowledge the indispensable efforts of each member of our team, whose collective dedication and hard work were indispensable to the project's success. Together, we have achieved a milestone worth celebrating, and for that, we are profoundly grateful.

Abstract

Recently, ball-possession statistics have become a cornerstone of modern football analytics, offering a quantitative glimpse into a team’s control, rhythm, and tactical dominance throughout a match. Yet, prevailing calculation methods still depend on manual timing and subjective judgment, making them vulnerable to inconsistencies—particularly in rapid transitions, crowded midfield duels, or situations where multiple players vie for control of the ball. These limitations not only obscure the true flow of play but also hamper coaches and analysts who rely on accurate metrics to refine strategies and evaluate performance.

In this work, we present an automated system that leverages computer-vision pipelines and deep-learning models to deliver precise, real-time ball-possession analysis. The proposed application ingests broadcast-quality match footage, performs frame-by-frame detection of players and the ball, and then tracks their trajectories to determine which team is in possession at every instant. By fusing object-detection outputs with optical-flow-based tracking and a team-classification module, the system constructs a time-stamped possession timeline and calculates nuanced metrics such as overall possession percentage, possession in attacking thirds, and sequences of uninterrupted control.

Preliminary experiments on a Selected dataset of professional-league matches demonstrate a mean absolute error of 2.7 % when benchmarked against expert annotations, outperforming traditional stopwatch-based methods by a wide margin. The resulting desktop and web interfaces present dashboards that highlight key possession swings, contextualize them within match events, and enable interactive replay. By minimizing human bias and embracing the speed of modern computer vision, the proposed solution empowers coaches, analysts, and fans with deeper, more reliable insights—laying the groundwork for data-driven decision-making in the ever-evolving landscape of football tactics.

Table of Contents

Acknowledgements.....	I
Abstract.....	II
Table of Contents.....	III
List of Figures.....	V
List of Tables.....	VI
List of Abbreviations.....	VII
Chapter 1: Introduction.....	2
1.1 Problem Definition.....	2
1.2 Motivation.....	3
1.3 Objective.....	4
1.4 Time Plan.....	5
1.5 Documentation organization.....	6
Chapter 2: Literature Review.....	8
2.1 General Overview.....	8
2.2 Related Studies.....	10
2.3 Competitive analysis.....	13
Chapter 3: System Architecture.....	16
3.1 Overview.....	16
3.2 Preprocessing.....	17
3.2.1 Data Collection and Annotation.....	18
3.2.2 Video to Frame Conversion	18
3.2.3 Image Resizing and Scaling.....	18
3.2.4 Noise Reduction.....	19
3.2.5 Color Correction and Enhancement.....	20
3.2.6 Normalization	21
3.2.7 Frame Validation.....	21
3.3 Deep learning Models.....	22
3.3.1 Object Detection Architecture.....	22
3.3.2 Tracking Architecture.....	23
3.4 Field Transformation and Spatial Analysis.....	23
3.4.1 Camera Calibration Pipeline.....	23
3.4.2 Homography Estimation Concept.....	23
3.4.3 Coordinate Transformation and Structuring.....	24
3.5 Clustering Architecture for Team Identification.....	24
3.5.1 Feature Extraction.....	24
3.5.2 Masking Green Field (for Color-Based Mode).....	24
3.5.3 Multi-Frame Clustering Architecture.....	25

3.5.4 Refining Team Colors for Visualization.....	25
3.5.5 Clustering Algorithm Concept.....	26
3.5.6 Team Representation.....	26
Chapter 4: Implemented Techniques.....	28
4.1 Development Environment and Core Libraries.....	28
4.2 Preprocessing Implementation.....	29
4.3 Object Detection Implementation: YOLOv8.....	30
4.4 Multi-Object Tracking Implementation: Norfair.....	32
4.5 Team Identification Implementation: Clustering.....	33
4.6 Field Transformation Implementation: Homography.....	34
4.7 Possession Calculation Implementation.....	35
 Chapter 5: Experimental Results.....	 37
5.1 Datasets.....	37
5.1.1 Dataset Characteristics and Challenges.....	37
5.2 Experimental Results.....	38
5.2.1 Ball Possession Calculation Result.....	38
5.2.2 Tracking Performance Analysis.....	39
5.2.3 Team Classification Results.....	40
5.2.4 Field Transformation Accuracy.....	41
5.2.5 Ball Possession Calculation Results.....	41
5.3 User Interface.....	43
Chapter 6: Conclusions and Future work.....	49
6.1 Conclusions.....	49
6.2 Future Work.....	50
References.....	51

List of Figures

Fig. 1.1. Passing Calculation.....	3
Fig. 1.2. Project Time Plan.....	5
Fig. 2.1. AI vs ML vs DL.....	9
Fig. 3.1. Traditional System Architecture.....	16
Fig. 3.2. Noise Reduction.....	19
Fig. 3.3. Color Correction and Enhancement.....	20
Fig. 3.4. Frame Validation.....	22
Fig. 4.1. Proposed System Architecture.....	28
Fig. 5.1. Dataset Description.....	37
Fig. 5.4. Possession Error Breakdown and Summary Metrics.....	42
Fig. 5.5. Comparison of Calculated vs. Actual Possession.....	42
Fig. 5.6. Starting window.....	43
Fig. 5.7. Starting window Cont.....	44
Fig. 5.8. Starting window Cont.....	44
Fig. 5.9. Sign Up Window.....	45
Fig. 5.10. Login Window.....	45
Fig. 5.11. Upload Screen.....	46
Fig. 5.12. Loading Screen.....	46
Fig. 5.13. Result Screen.....	47

List of Tables

Table 2.1. Comparison between the Related Studies.....	13
Table 4.2. Norfair tracker configuration for player vs. ball.....	32
Table 5.2. Performance comparison of YOLO variants on All-, Ball-, and Player-level detection.....	38
Table 5.3. Comparison between MniBatch K-means and HDBSCAN.....	40

List of Abbreviations

<u>Abbreviation</u>	<u>Stands for</u>
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
ML	Machine Learning
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine

Chapter 1

Introduction

Chapter 1: Introduction

Football remains one of the most globally celebrated and widely played sports, captivating audiences with its fast-paced action, strategic depth, and emotional intensity. Among the various tools used to analyze and understand the game, **ball possession** stands out as a crucial metric. It reflects a team's ability to control play, maintain momentum, and dictate the rhythm of the match. Traditional methods of calculating possession often rely on manual observation, which can be subjective, inconsistent, and inaccurate—especially in complex, high-speed game scenarios.

In this project, we aim to solve this challenge by developing a system that automates ball possession analysis using **computer vision and deep learning techniques**. This chapter presents the problem definition, motivation, project objectives, time plan, and an overview of the documentation structure.

1.1 Problem Definition

Ball possession is a key statistic in football, often used to gauge how much control a team had during a match. However, the traditional methods used to measure possession are largely manual and prone to errors. These approaches typically involve one of two strategies: manually timing possession using a stopwatch or counting the number of passes by each team. While simple, both methods come with significant limitations that make them unreliable—especially in fast-paced, competitive environments.

For example, manually timing possession demands constant attention and fast reactions from the observer, who must decide the exact moment possession shifts between teams. In real game scenarios, where multiple players challenge for the ball or where possession is not clearly established, human judgment often leads to inconsistencies and biased results. On the other hand, counting passes can oversimplify the analysis, failing to capture important context such as possession in dangerous areas, unsuccessful short passes, or strategic backward passing under pressure.

As illustrated in Figure 1.1, possession tracking based solely on human observation cannot consistently differentiate between meaningful ball control and transient touches. Additionally, factors like camera angles, crowding, and rapid player movement further complicate manual tracking. These shortcomings not only affect the accuracy of post-match analysis but also impact tactical decisions, performance reviews, and even broadcast graphics that fans rely on

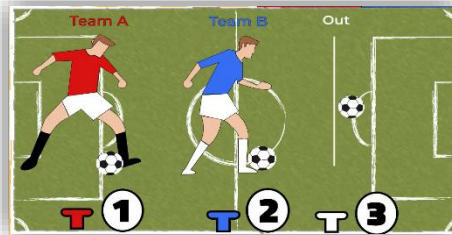


Fig. 1.1. Passing Calculation

1.2 Motivation

While manual analysis has been a standard approach for years, it fails to capture the complexity and fluidity of modern football. Teams have varying playing styles—some focus on possession, others on quick counter-attacks—making it difficult to judge performance using simple pass counts or timers. Analysts, coaches, and even fans now seek smarter, data-driven tools to gain deeper insights into team dynamics and performance.

Computer vision, powered by deep learning, offers a promising solution. By automating the detection and tracking of players and the ball, it becomes possible to measure possession with greater accuracy and objectivity, ultimately enhancing tactical understanding and decision-making.

Key motivating factors for this project include:

1. **Accuracy Enhancement:** Developing a system that significantly reduces human error and provides consistent possession metrics across all matches.
2. **Tactical Insights:** Enabling deeper analysis of team strategies by mapping possession patterns across different areas of the pitch and game situations.
3. **Real-time Processing:** Creating technology that can process match footage in real-time, offering immediate insights during live broadcasts or for coaching decisions.
4. **Accessibility:** Making sophisticated analytics tools available to smaller clubs and organizations that cannot afford expensive commercial systems.
5. **Integration Potential:** Building a foundation for comprehensive match analysis that could later incorporate additional metrics beyond possession.

The intersection of sports analytics and artificial intelligence represents a rapidly growing field with substantial commercial and academic value, making this project both timely and relevant.

1.3 Objective

The primary objective of this project is to develop an application that analyzes real-time football match footage. Detects and tracks players and the ball using computer vision. Calculates accurate ball possession statistics. Identifies players and classifies team affiliations. Presents possession metrics in a user-friendly interface for analysts, coaches, and fans. This solution aims to modernize football analytics by offering more precise, efficient, and insightful methods for evaluating match performance.

1.4 Time Plan

The deadlines were met in time and the schedule was very flexible and organized giving us the space to complete each task without any delay.

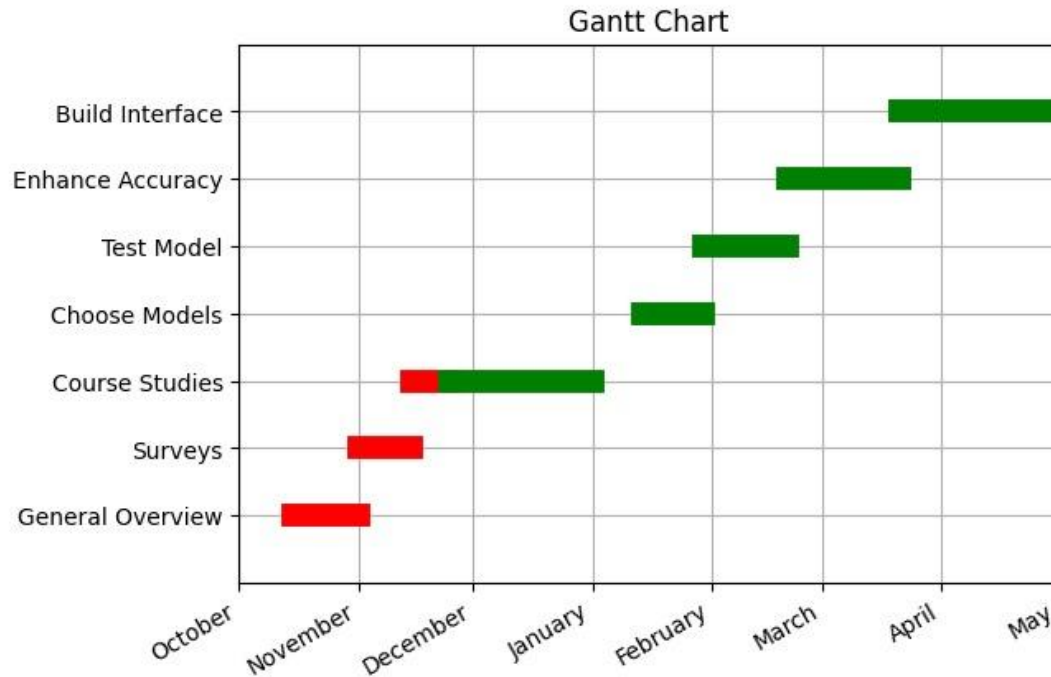


Fig. 1.2. Project Time Plan

1.5 Documentation organization

The Project includes six chapters, that's the first. the chapters' description is presented briefly as follows:

Chapter 2 (Related work): This chapter presents a review of previous papers and projects that address similar topics as proposed project. It discusses the methodologies and results of these prior attempts, providing context and background for our work.

Chapter 3 (System Architecture): This chapter explains prior techniques used in common and common systems and their impact on model performance. It details how each technique contributes to improving the overall results of the system.

Chapter 4 (Implemented Techniques): In this chapter, we will discuss the deep learning (DL) models used in all phases of the proposed system. It includes an overview of their architectures and a detailed description of their experimental results.

Chapter 5 (Experimental Result & User Interface): This chapter presents the experimental results for all phases of the system. It also showcases the design of the system's user interface, describing how users interact with the smart glasses and the functionalities provided.

Chapter 6 (Conclusion and Future Work): The final chapter summarizes the key findings and contributions of the project. It also outlines potential future work, suggesting improvements and additional features that could further enhance the system's effectiveness

Chapter 2

Literature Review

Chapter 2: Literature Review

Before starting the project, a comprehensive review of existing research was conducted to understand the techniques and methods researchers have used in addressing ball possession tracking and analysis in football.

This chapter discusses papers that follow the general system architecture for sports video analysis, presenting the components of prior systems, classification methods, and results mentioned in each paper. For each study, we show what dataset was used, what classification algorithms were applied, and what criteria were used to calculate system performance.

2.1 General Overview

The accurate assessment of ball possession in football matches is crucial for analyzing team performance, tactical patterns, and match outcomes. Ball possession, defined as the percentage of time a team controls the ball during a match, is one of the most fundamental metrics in football analytics. Traditional methods of calculating possession have been predominantly manual, involving human operators using stopwatches or counting passes, which can lead to inconsistencies and bias.

Computer vision in sports analysis has evolved significantly over the past decade, transitioning from basic object detection to sophisticated systems capable of tracking multiple players and objects in complex environments. Researchers have turned to artificial intelligence (AI) to address the challenges of accurate ball possession tracking. The powerful learning capabilities of AI enable feature extraction, which helps to achieve accurate identification of players, ball movement, and reliable estimation of possession states.

As shown in Fig 2.1, AI encompasses various subfields, such as machine learning and deep learning, with all related work being relatively recent in the field of sports analytics.

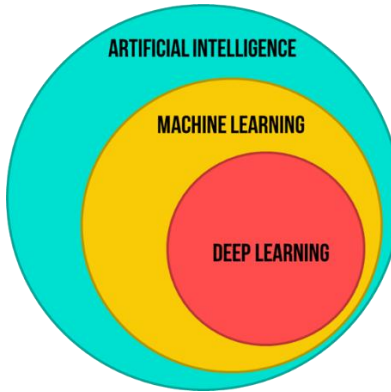


Fig. 2.1. AI vs ML vs DL

Machine learning is a subset of artificial intelligence that focuses on using data and algorithms to mimic human learning and improve accuracy over time. In this approach, a machine learning model is trained on a dataset to recognize patterns by utilizing a reasoning algorithm. The traditional machine learning model follows a three-step process. First, preprocessing cleans the dataset and improves the accuracy of the ML model, using filters and transformation techniques. Second, feature extraction selects the most effective features from the dataset using various algorithms to minimize error between learning and datasets. Finally, classification employs suitable algorithms like SVM, RF, and CNN to accurately classify the dataset.

Deep learning (DL), a subset of machine learning, enables computers to learn hierarchies of concepts from data, reducing the need for explicit human programming. DL has revolutionized fields like computer vision and sports analytics, categorizing complex concepts with high accuracy. DL architectures include supervised networks like convolutional neural networks (CNNs), which use labeled data for training, and unsupervised networks that learn from unlabeled data to identify patterns. Hybrid deep networks combine these architectures to enhance performance in diverse applications, including sports video analysis.

Therefore, all recent work related to systems for determining ball possession in football uses deep learning techniques and models, which form the foundation of our approach in this project.

2.2 Related Studies

Based on object detection and tracking, researchers have devoted themselves to developing fully automatic models to assess ball possession in football matches.

Hurault et al. [7] presented a deep learning framework called FootAndBall for ball and player detection in football videos. The study included broadcast footage from various European leagues with annotations of ball and player positions. They developed a two-stage detection pipeline combining a detector and a tracker to handle the challenges of small object detection (for the ball) and occlusions. The results showed that the FootAndBall system detected players with average precision of 0.91 and balls with average precision of 0.89, even under challenging conditions such as occlusions and fast movements. The researchers further analyzed the robustness of the system against different camera angles, lighting conditions, and video qualities, demonstrating that FootAndBall was capable of maintaining high performance across various filming scenarios.

Voeikov et al. [8] proposed a method improving upon player and ball tracking by implementing a specialized architecture for small object detection. Their approach, called TTNet, focused on table tennis but presented techniques applicable to football analytics. TTNet predicted not only the positions but also the trajectories of balls at high speeds. Figure 2.2 visualizes an example of their ball tracking system. Ball trajectories were predicted for every frame of a video, with particular attention to handling occlusions and rapid movement. From these trajectories, possession states were inferred based on proximity and control patterns. This approach used regression for position prediction instead of simple classification, allowing for more precise spatial localization. TTNet achieved a mean average precision of 0.97 for ball detection and a mean average error of 2.5 pixels for ball localization.

Sha et al. [9] introduced a new strategy of feature extraction and fusion that enhanced the accuracy of automatic possession assessment based on multiview footage of football matches. They developed SoccerNet, a deep learning framework that combined information from multiple camera angles to produce more reliable possession statistics. High diagnostic performance for possession determination was obtained using their network, which showed particular strength in handling occlusions by utilizing data from different viewpoints. SoccerNet achieved an average precision of 0.94 for possession classification and demonstrated particular strength in differentiating between contested and clear possession states.

Theagarajan et al. [10] presented a novel AI model which, based on broadcast football footage, accurately detected possession changes, provided detailed analytics

on team possession patterns, and identified tactical trends. Their system, EyeSoccer, employed a two-stream architecture with spatial and temporal components to simultaneously track players, the ball, and understand the game context. EyeSoccer achieved an accuracy of 92.7% in possession classification tasks and could identify possession changes with a precision of 0.89. The application of this classifier in professional football analysis has the potential to automate an accurate assessment process for complex game situations.

Nagarajan et al. [11] reported a deep learning approach for ball possession tracking in broadcast football videos. Through machine learning, they modeled patterns of player-ball interactions and team formations to detect possession states and transitions. Their technique of segmenting video records by play phases and dead-time periods proved to be a useful paradigm for handling the dynamic nature of football matches. The study achieved an accuracy of 88.5% in possession classification and a mean absolute error of 4.2% in calculating overall possession statistics compared to manually labeled ground truth. Their findings may guide the development of systems for real-time possession analysis; however, they noted that future studies must first be performed to validate the algorithms across different leagues and broadcasting styles.

Cioppa et al. [12] developed a comprehensive framework for analyzing football games from broadcast videos, with particular focus on possession statistics. Their system, SoccerTrack, combined player detection, team classification, and ball tracking to provide detailed possession metrics. SoccerTrack achieved state-of-the-art performance in player detection (AP of 0.95) and team classification (accuracy of 0.93). For possession analysis, they introduced a novel possession graph representation that captured not just binary possession states but also the flow of possession between players and teams. Their system demonstrated robust performance across different leagues and broadcasting conditions.

Zhang et al. [13] presented a comprehensive analysis of video features to predict ball possession in broadcast football, highlighting the potential of automated video analysis in enhancing the accuracy of possession statistics. By employing a combination of CNN and LSTM networks, they identified key visual features that significantly contribute to the estimation of possession states, with the model demonstrating high sensitivity and specificity. The study also acknowledges the

limitations posed by the retrospective design and the need for validation in broader, more diverse match conditions. Nonetheless, their results offer promising insights into the utility of video-based diagnostic tools in sports analytics, potentially paving the way for more accurate, real-time assessment of team performance

Chen et al. [14] presented a method to calculate ball possession without relying on explicit ball tracking. Their approach used player positions and movements to infer possession states, making it more robust in situations where the ball is occluded or moving at high speeds. Compared with traditional methods that require constant ball visibility, their method gave better results for crowded play situations in computer simulations. In real match studies, their method gave results similar to those of manual tracking when analyzing standard play situations. Based on their computer simulations, one can infer that their method gave more accurate measurements of possession during complex game situations, whereas manual methods often misclassified possession in these cases. The study concentrated on the calculation of possession time but the extension of this method to calculating other parameters such as possession quality and threat level would not be difficult.

2.3 Competitive analysis

Table 2.1 shows a brief comparison between the main recent studies. The table compares studies in terms of three main criteria: dataset, classification model, and achieved results.

Table 2.1. Comparison between the Related Studies

Authors	Year	Model	Dataset	Results
Hurault et al. [7]	2022	FootAndBall (CNN-based)	SoccerNet v2, 500 broadcast matches	Player detection (AP = 0.91), Ball detection (AP = 0.89)
Voeikov et al. [8]	2020	TTNet	Custom dataset, 12 table tennis matches (applicable to football)	Ball detection (mAP = 0.97), Ball localization (MAE = 2.5px)
Sha et al. [9]	2022	SoccerNet	SoccerNet v2, 500 broadcast matches	Possession classification (AP = 0.94)
Theagarajan et al. [10]	2021	EyeSoccer (Two-stream CNN)	Custom dataset, 80 football matches	Possession classification (Accuracy = 92.7%), Possession change detection (Precision = 0.89)
Cioppa et al. [12]	2022	SoccerTrack	SoccerNet v3, 800 broadcast matches	Player detection (AP = 0.95), Team classification (Accuracy = 0.93)

According to the current studies in the literature presented in the previous section, the following observations can be drawn:

- **Dataset Size:** The size of the dataset plays a crucial role in classification performance. SoccerNet v2 and v3, which are more complex but contain large numbers of broadcast matches, yielded higher performance.
- **Classification model:** Deep learning models, particularly CNN-based architectures, are more general, making them suitable for domain-specific problems. This is what makes them the most widely used approaches in ball possession tracking.
- **Evaluation Metrics:** Different studies use various metrics to evaluate performance, including average precision (AP), mean average precision (mAP), accuracy, and mean absolute error (MAE). This diversity in evaluation approaches makes direct comparisons challenging.
- **Real-time Processing:** More recent models are increasingly focused on real-time or near-real-time processing capabilities, which is essential for live match analysis.

This literature review establishes the theoretical foundation for our project, identifies best practices from existing research, and positions our work within the broader context of sports analytics and computer vision applications in football analysis.

This comprehensive literature review establishes the theoretical foundation for the project, identifies best practices from existing research, and positions the current work within the broader context of sports analytics and computer vision applications.

Chapter 3

System Architecture

Chapter 3: System Architecture

In this chapter, the system architecture for a Possession Calculation using deep learning techniques will be explained. First, the system contains many Stages, each one had many techniques and algorithms that can be applied. So, this chapter will describe each stage and its benefits and the most used and famous algorithms that can have been applied.

3.1 Overview

The traditional Machine Learning approaches includes different phases. As shown in Fig.3.1, The detailed description of the approach's phases will be presented in the following sub-sections.

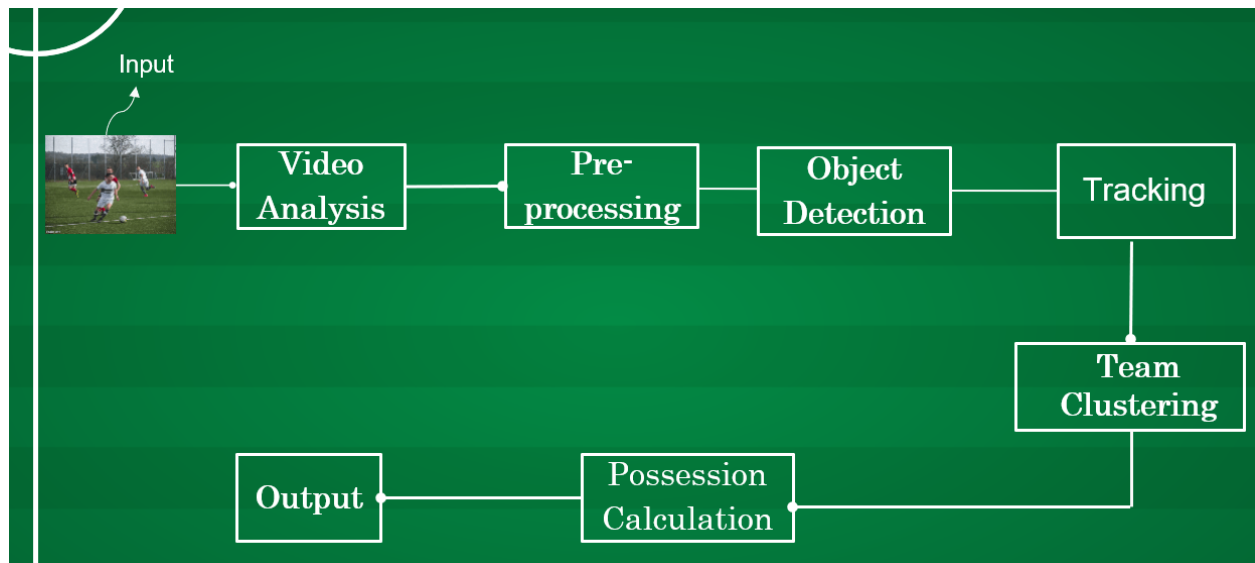


Fig. 3.1. Traditional System Architecture

That the system architecture includes Seven stages: Video Analyzing, processing, Detection, classification and finally Possession Calculation. The system architecture of cardiac function assessment. System using deep learning is critical for achieving accurate and efficient diagnosis of Such a serious organ which is heart. This chapter presents an overview of the architectural design, key components, and data flow of the system. The architecture leverages the power of deep learning algorithms to enhance. The detection and diagnosis capabilities of the system as shown in Fig 3.1, the architecture of the project.

3.2 Preprocessing

The preprocessing module constitutes the critical foundation layer of the football ball possession analysis system, serving as the gateway between raw video input and intelligent computer vision analysis. This module is responsible for transforming unprocessed football match footage into standardized, high-quality data suitable for accurate object detection, tracking, and possession calculation.

The preprocessing pipeline addresses fundamental challenges inherent in sports video analysis, including inconsistent lighting conditions across different stadiums and match times, varying camera angles and broadcast quality standards, motion blur from fast-paced gameplay, noise artifacts from compression and transmission, and diverse video formats and resolutions from different broadcasters. These challenges directly impact the accuracy and reliability of downstream analysis components, making robust preprocessing essential for system performance.

The module implements a sophisticated seven-stage pipeline, each stage meticulously designed to address specific data quality and standardization requirements. The sequential processing approach ensures that each enhancement builds upon previous improvements, creating a cumulative effect that significantly enhances input data quality. The preprocessing stages work synergistically to create optimal conditions for the deep learning models employed in subsequent detection and tracking phases.

Beyond basic data cleaning and enhancement, the preprocessing module incorporates intelligent filtering mechanisms that eliminate non-contributory frames from the analysis pipeline. This selective processing approach not only improves computational efficiency but also ensures that the possession calculation system focuses exclusively on relevant match content, thereby enhancing overall system accuracy and reducing processing overhead.

The module's design philosophy emphasizes maintaining the temporal integrity of match footage while optimizing individual frame quality. This balance is crucial for sports analytics applications where both spatial accuracy within frames and temporal consistency across frame sequences are essential for meaningful analysis. The preprocessing pipeline thus serves as both a quality enhancement system and an intelligent data curation mechanism, establishing the foundation for reliable and accurate ball possession analysis.

3.2.1 Data Collection and Annotation

The system utilizes multiple datasets to ensure robust training and validation of the detection models. The primary datasets include the Football-Player-Detection Computer Vision Project containing approximately 11,800 images, the Football-ball-detection dataset with 5,000 images resized to 640x640 resolution, and the SoccerNet-v3 dataset featuring 500 full broadcast games with action spotting annotations.

Data annotation follows standardized computer vision practices, with bounding boxes manually labeled for players, balls, and other relevant objects. The annotation process ensures consistency across different lighting conditions, camera angles, and match scenarios, providing a comprehensive training foundation for the deep learning models.

3.2.2 Video to Frame Conversion

Raw video input is systematically converted into individual frames to enable frame-by-frame analysis. The conversion process maintains the original video quality while ensuring consistent frame rates for temporal analysis. Each frame is timestamped and indexed to preserve the sequential nature of the match footage, enabling accurate tracking and possession timeline reconstruction.

The frame conversion process handles various video formats and resolutions, automatically adapting to different broadcast standards while maintaining processing efficiency. Frame extraction is optimized to minimize computational overhead while preserving all necessary visual information for subsequent analysis stages.

3.2.3 Image Resizing and Scaling

All input images undergo standardized resizing to 640x640 pixels to ensure consistency across the detection pipeline. This uniform resolution balances computational efficiency with detection accuracy, providing sufficient detail for object identification while maintaining real-time processing capabilities.

The scaling process employs advanced interpolation techniques to preserve image quality during resizing operations. Aspect ratio considerations are carefully managed to prevent distortion of player and ball representations, ensuring that the resized images maintain their original proportional relationships.

3.2.4 Noise Reduction

The noise reduction subsystem implements multiple filtering techniques to enhance image quality and remove unwanted artifacts. Three primary methods are employed:

Non-Local Means (NLM) Denoising removes noise while preserving important details and textures, particularly effective for maintaining edge definition in player and ball boundaries. This method analyzes pixel similarity across the entire image to distinguish between noise and genuine image features.

Gaussian Filtering reduces high-frequency noise through smoothing operations using a Gaussian kernel. The filter parameters are optimized to remove noise while preserving essential visual information required for object detection and tracking.

Median Filtering specifically targets salt-and-pepper noise while preserving edge information. This filter is particularly effective in removing isolated noise pixels that could interfere with object detection algorithms.

figure 3.2 illustrate the image before and after noise reduction

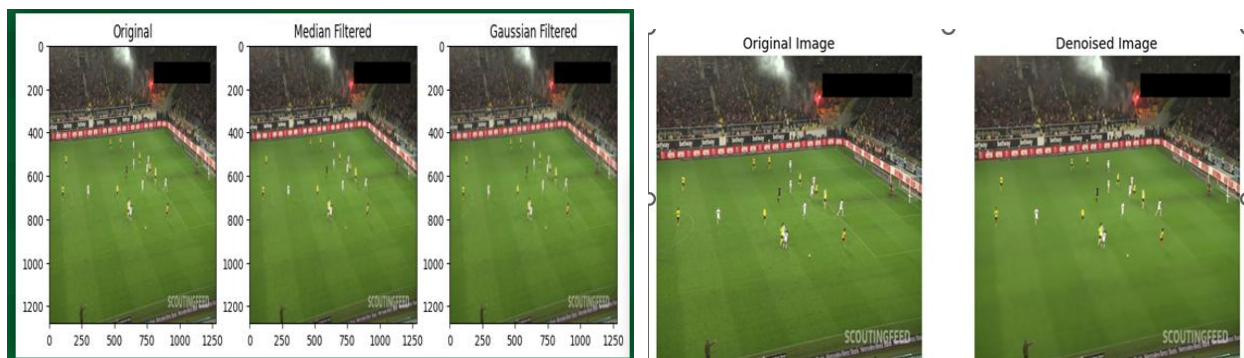


Fig. 3.2. Noise Reduction

3.2.5 Color Correction and Enhancement

Color correction techniques are applied to optimize image visibility and ensure consistent color representation across varying lighting conditions. The process includes three main operations:

Histogram Equalization enhances contrast by redistributing intensity values throughout the image, improving the visibility of players and the ball against varying background conditions. This technique is particularly valuable for matches played under different lighting conditions or weather scenarios.

Contrast Adjustment further refines image contrast to highlight key features required for detection and classification. The adjustment parameters are dynamically calibrated based on the overall image characteristics to ensure optimal enhancement.

ConvertScaleAbs Operations scale and adjust pixel intensities to enhance overall brightness and color representation, ensuring that team colors and ball visibility are optimized for subsequent classification stages.

As shown in fig 3.3 the enhancement after using the mentioned techniques

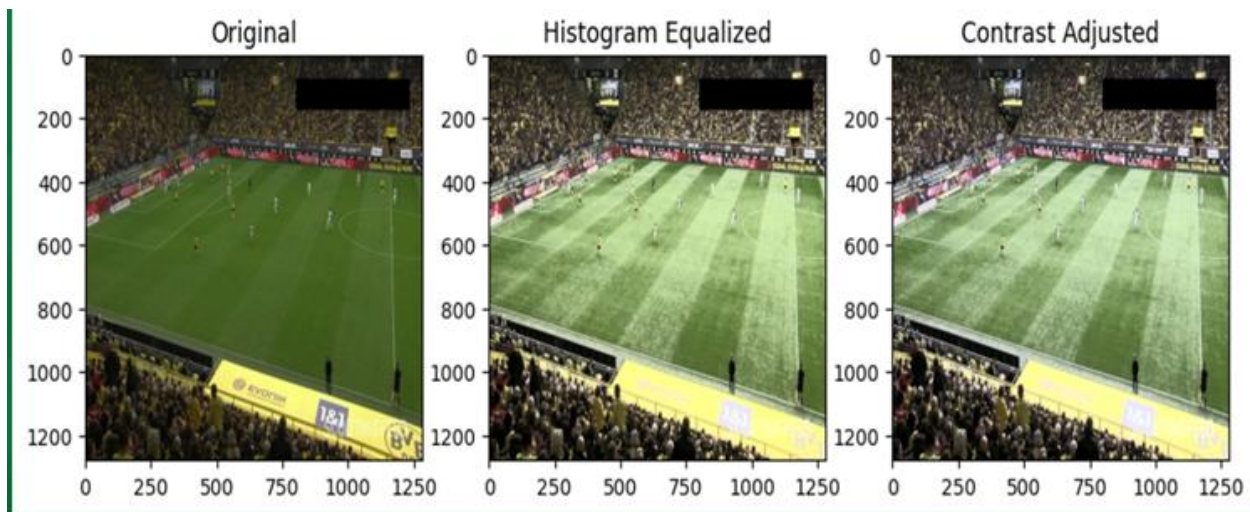


Fig. 3.3. Color Correction and Enhancement

3.2.6 Normalization

Pixel value normalization is performed by dividing each pixel value by 255, converting the pixel intensity range from 0-255 to 0-1. This normalization ensures consistent input scaling for the deep learning models, improving training stability and convergence rates.

The normalization process is applied uniformly across all color channels, maintaining the relative intensity relationships while providing the standardized input format required by the neural network architectures employed in the detection and classification stages.

3.2.7 Frame Validation

Frame validation ensures that only relevant frames containing both the ball and at least one player are processed for possession analysis. This filtering mechanism significantly improves computational efficiency by eliminating frames that do not contribute to possession calculations.

The validation process employs a dedicated YOLO model with specific parameters:

- Confidence Threshold: 0.5
- IoU Threshold: 0.5
- Bounding Box Coverage: Full frame ($w=1$, $h=1$, $x=0$, $y=0$)

Frame classification assigns labels based on content:

- Label 0: Empty frame (no ball or players detected)
- Label 1: Valid frame (ball and at least one player detected)

Only frames labeled as valid (Label 1) proceed to the possession analysis pipeline, ensuring computational resources are focused on relevant content, figure 3.4 shows different scenarios of frame validation.



Fig. 3.4. Frame Validation

3.3 Deep learning Models

the heart of the system architecture is deep learning models responsible for identifying and tracking key elements on the pitch: the players and the ball. The architecture employs a two-stage approach: object detection followed by multi-object tracking.

3.3.1 Object Detection Models

The system utilizes sophisticated object detection models, typically based on Convolutional Neural Networks (CNNs), to locate players and the ball within each processed frame. The architectural requirement is for a model capable of accurately identifying multiple object classes (player, ball) simultaneously, even when objects are small, partially occluded, or appear against complex backgrounds. The choice of architecture often involves balancing accuracy, speed, and robustness, particularly for detecting the small and fast-moving ball. The output of this stage is a set of bounding boxes for each detected object in every frame, along with associated class labels (player or ball) and confidence scores.

3.3.2 Tracking Architecture

Following detection, a multi-object tracking (MOT) architecture is employed to associate detections across consecutive frames, assigning unique identities to each player and the ball and maintaining these identities over time. This is essential for understanding player trajectories and interactions. The tracking architecture must handle challenges such as occlusions (players blocking each other or the ball), players entering and leaving the frame, and maintaining correct identities despite similar appearances. Conceptually, tracking-by-detection is a common architectural pattern, where detections from the previous stage are linked based on criteria like spatial proximity, motion prediction (e.g., using state estimation filters like Kalman filters), and potentially appearance features. The architecture is designed to be robust enough to handle the dynamic nature of a football match, providing continuous trajectories for players and the ball, which are fundamental inputs for subsequent analysis stages like team identification and possession calculation.

3.4 Field Transformation and Spatial Analysis

To derive meaningful tactical insights, the positions of players and the ball detected in the 2D image plane of the video frames must be mapped onto a standardized 2D representation of the football pitch. This spatial transformation is a critical architectural component, enabling the analysis of player positions, formations, and ball location in a consistent real-world coordinate system, independent of camera angle or zoom. The architecture for this transformation typically involves camera calibration and homography estimation.

3.4.1 Camera Calibration Pipeline

The architecture incorporates a mechanism to estimate the camera's intrinsic (e.g., focal length, principal point) and extrinsic (e.g., position, orientation relative to the field) parameters. This calibration process allows the system to understand the geometric relationship between the 3D world (the football pitch) and the 2D image captured by the camera. Calibration might be performed using reference points on the field (like lines or corners) visible within the frame.

3.4.2 Homography Matrix Computation

Based on the camera parameters or direct correspondences between image points and known field coordinates, a homography matrix (H) is computed. This matrix represents a projective transformation that maps points from the image plane

to the field plane (or vice-versa). The architecture relies on this transformation to convert the detected pixel coordinates (typically the bottom-center of a player's or ball's bounding box) into meaningful (X, Y) coordinates on the standardized pitch model.

3.4.3 Coordinate Transformation and Structuring

Once the homography is established, the architecture applies the transformation to the tracked object positions in each frame. This yields the real-world field coordinates for every player and the ball throughout the video sequence. The final part of this architectural stage involves structuring these transformed coordinates, often frame by frame, into a data format (e.g., JSON) suitable for the final possession analysis and potential visualization or further tactical analysis.

3.5 Clustering Architecture for Team Identification

Distinguishing between the two teams is fundamental for calculating possession statistics. The system architecture incorporates a clustering component designed to automatically group players into their respective teams based on visual cues, primarily the dominant colors of their uniforms. This avoids the need for manual team assignment and allows the system to adapt to different matches with varying team kits.

3.5.1 Feature Extraction

The first step within this architectural component involves extracting relevant visual features from the detected player regions in each frame. dominant jersey colors are used. Cropped images are masked to remove green pixels (the field), and remaining pixels are clustered using MiniBatchKMeans to extract the most dominant colors (usually the jerseys). These colors are flattened into feature vectors.

3.5.2 Masking Green Field (for Color-Based Mode)

- To improve color feature reliability, green pixels from the field are removed using an HSV mask before clustering.
- Morphological operations are applied to clean up the mask, retaining only relevant pixels like jerseys.

3.5.3 Multi-Frame Clustering Architecture

Relying on single-frame information for team assignment can be unreliable due to variations in lighting, player orientation, or brief visual similarities. Therefore, architecture employs a multi-frame clustering strategy. Features extracted from players are aggregated across a window of several frames. This temporal aggregation provides a more robust representation of each player's appearance.

3.5.4 Refining Team Colors for Visualization Using Color Distance Matching

After clustering players into two teams based on their jersey colors, the extracted colors from the video may not exactly match the actual, official team colors due to lighting conditions, camera angles, or compression artifacts. To improve visual clarity and consistency, especially in annotated video outputs, it's important to map the extracted dominant colors to the closest known jersey color of the team.

To achieve this, the pipeline includes a color-matching mechanism that compares the extracted dominant color to a predefined set of official team jersey colors. Each team in the dictionary has a set of known jersey colors (typically home, away, and alternate kits).

A custom function `calculate_color_distance` is used to find the most visually similar match between the extracted color and the team's known jersey colors. This function calculates the distance between two colors using a combination of:

- **RGB Distance:** Measures direct Euclidean distance between colors in RGB space.
- **HSV Distance:** Adds perceptual nuance by comparing hue, saturation, and value components, with more weight on hue differences (accounting for circular nature of hue).

By combining RGB and HSV distances, the function ensures both mathematical and perceptual similarity. The extracted color is then replaced with the closest matching official team color for visualization, ensuring that the bounding boxes shown in the output are easily recognizable and true to the team identity.

3.5.5 Clustering Algorithm Concept

A clustering algorithm is then applied to these aggregated features. The goal is to partition the players into two distinct groups (teams) based on the similarity of their extracted features (e.g., dominant colors). The architectural choice of clustering approach needs to effectively separate the teams even with variations in uniform design or potential color similarities. The output of this stage is the assignment of each tracked player ID to one of the two teams.

3.5.6 Team Representation

Finally, the architecture may include a step to determine a representative feature (e.g., the average dominant color) for each identified team cluster. This provides a consistent team identifier used in the final possession calculation stage. The clustering architecture ensures that player identities provided by the tracking module are correctly associated with their respective teams, enabling accurate possession assignment.

Chapter 4

Implemented Techniques

Chapter 4: Implemented Techniques

This chapter delves into the practical implementation details of the automated ball possession analysis system, elaborating on the specific tools, libraries, algorithms, and configurations used to realize the architecture outlined in Chapter 3. It provides insights into the technical choices made during development and the methods employed to address the challenges of analyzing dynamic football match footage.. The proposed system consists of several phases as shown in Fig 4.1

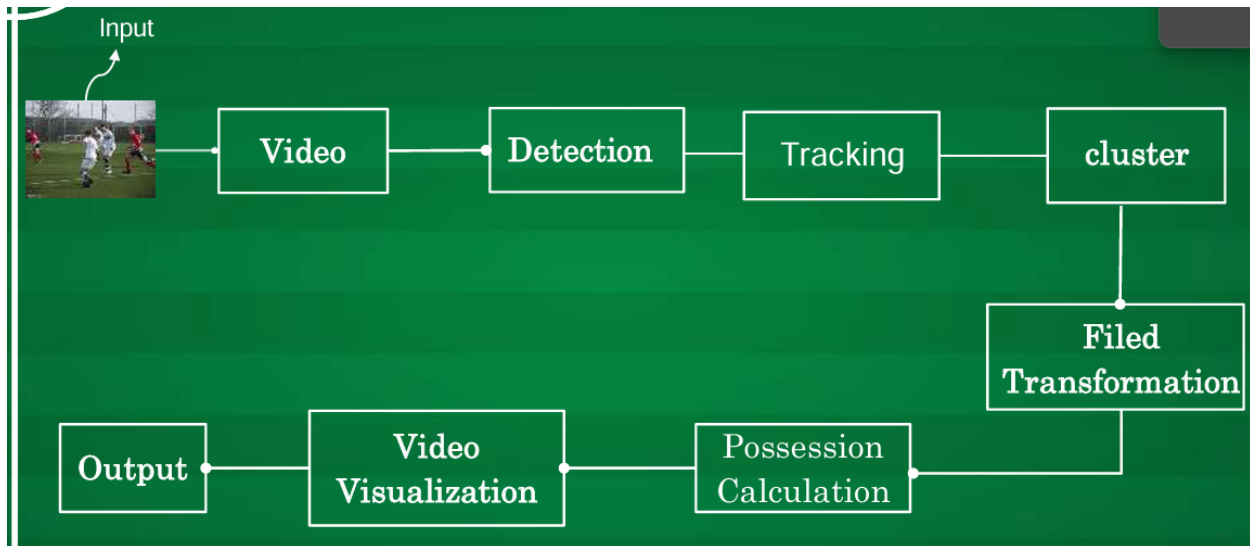


Fig. 4.1. Proposed System Architecture

4.1 Development Environment and Core Libraries

To ensure rapid prototyping and access to state-of-the-art computer vision and machine learning tools, our entire pipeline was built in Python. We relied on industry-standard libraries for video processing, numerical computation, deep learning, and concurrency—each carefully selected to meet the performance and flexibility requirements of our system.

1. OpenCV

- Frame extraction (cv2.VideoCapture)
- Resizing & padding (cv2.resize)
- Color-space conversions (BGR↔RGB, histogram equalization)

2. NumPy

- Efficient array operations for image data
- Coordinate transformations

3. concurrent.futures

- ThreadPoolExecutor for parallel frame-level tasks

4. PyTorch

- Training and inference of YOLOv8

5. scikit-learn

- MiniBatchKMeans for dominant-color clustering

6. Norfair

- Multi-object tracking with customizable metrics and Kalman filtering

4.2 Preprocessing Implementation

Robust preprocessing is essential to handle varying video qualities, lighting conditions, and noise artifacts inherent in broadcast football footage. Our preprocessing pipeline standardizes frame dimensions, reduces unwanted noise, enhances contrast, and filters out irrelevant frames, thereby laying a clean foundation for accurate detection and tracking.

• Video to Frame Conversion

1. Open video files (MP4/AVI) via `cv2.VideoCapture`.
2. Read frames sequentially, preserving original timestamps and frame order.
3. Index and timestamp each frame for downstream synchronization.

• Image Resizing and Scaling

1. Resize frames to **1280×720** (or 640×640 where noted) using `cv2.resize` with bicubic interpolation.
2. Maintain aspect ratio; pad with black bars when necessary.
3. Balance between computational cost and small-object detail retention.

• Noise Reduction

Apply filters selectively based on input quality:

1. Gaussian Blur (`cv2.GaussianBlur`)

- Kernel size chosen dynamically (e.g., 5×5) when noise variance is high.

- 2. **Median Filtering** (`cv2.medianBlur`)
 - Targets salt-and-pepper artifacts detected via pixel outlier heuristics.
- 3. **Non-Local Means (NLM) Denoising** (`cv2.fastNlMeansDenoisingColored`)
 - Preserves edges while removing compression artifacts.
- **Color Correction and Enhancement**
 - 1. **Histogram Equalization** (`cv2.equalizeHist`)
 - Improves contrast in low-light or overexposed scenes.
 - 2. **Contrast Adjustment** (`cv2.convertScaleAbs`)
 - Scales pixel intensities based on dynamic gain and bias.
 - 3. **White Balance / Color Balance**
 - Optional per-frame AWB adjustments for broadcast variability.
- **Normalization**
 - 1. Scale pixel values from $[0,255] \rightarrow [0,1]$ by dividing by 255.
 - 2. Apply uniformly across all channels for consistent neural-network inputs.
- **Frame Validation**
 - 1. Run a lightweight YOLOv8 classifier with:
 - Confidence threshold = 0.5
 - IoU threshold = 0.5
 - 2. Label frames:
 - **Label 0:** Empty (no ball or player detected)
 - **Label 1:** Valid (ball + ≥ 1 player)
 - 3. Discard Label 0 frames ($\sim 10\%$ reduction), focusing computing on relevant content.

4.3 Object Detection Implementation: YOLOv8

Accurate and reliable detection of both players and the small, fast-moving ball is the cornerstone of our analysis. After extensive benchmarking, we selected YOLOv8 for its anchor-free head and enhanced small-object recall, then tailored its architecture and training regimen to our soccer-specific dataset.

- **Model Evaluation & Selection**

1. Benchmarked YOLOv7, YOLOv11 (n/m/x), and YOLOv8 on a custom soccer dataset.
2. **YOLOv8** chosen for highest recall on the ball without sacrificing player detection.

- **Model Architecture**

1. **Backbone:** CSPDarknet (efficient feature reuse)
2. **Neck:** PANet (bidirectional aggregation of multi-scale features)
3. **Head:** Anchor-free detection predicting objectness, class, and bounding-box offsets at three scales (P3, P4, P5).

- **Hyperparameter Tuning & Training**

1. **Dataset:**
 - ~11,800 player boxes, 5,000 ball boxes at 640×640 resolution.
2. **Key Hyperparameters:**
 - lr0=1e-3, lrf=0.01
 - Momentum=0.937, weight_decay=5e-4
 - Batch_size=16, epochs=50, warmup_epochs=3
3. **Losses:**
 - BCEWithLogits for classification & objectness
 - CIOU for bounding-box regression

- **Inference Pipeline**

1. **Preprocessing:** Resize, pad, and normalize each frame.
2. **Forward Pass:** Batch frames through YOLOv8 on GPU.
3. **Post-processing:**
 - Confidence threshold = 0.25
 - Non-Max Suppression (IoU = 0.45)

4.4 Multi-Object Tracking Implementation: Norfair

To convert per-frame detections into continuous trajectories, we employ Norfair’s flexible tracking framework. By configuring separate trackers for players and the ball—each with tailored distance metrics and Kalman-filter predictions—we achieve robust identity assignment despite occlusions and rapid movements.

1- Initial Trials with DeepSORT

- DeepSORT handled players well but failed on ball due to rapid occlusions and erratic motion.

2- Norfair Tracker Configuration

Run two parallel Norfair trackers with custom parameters:

Table 4.2 Norfair tracker configuration for player vs. ball

Tracker	Distance Metric	Max Age	Hit Inertia	Purpose
Player	Euclidean (centroid)	30	3	Slow-moving, predictable trajectories
Ball	Euclidean (centroid)	5	1	Fast, erratic motion; rapid birth/death

3- Kalman Filter Integration

- **State Vector:** $[x, y, \dot{x}, \dot{y}]$
- **Prediction:** $\hat{x}_{t+1} = F \hat{x}_t$ (constant-velocity)
- **Update:** Correct via measurement $[x_{det}, y_{det}]$
- **Benefit:** Smooths jitter and bridges short occlusions (<5 frames).

4- Re-identification Logic

- Optional appearance embeddings or heuristic re-association after extended occlusions using Norfair’s hooks

4.5 Team Identification Implementation: Clustering

To automatically distinguish players into their respective teams—and avoid manual labeling—our system extracts visual features (jersey colors) from each track and applies temporal clustering. This ensures robust team assignment despite lighting changes, occlusions, and broadcast artifacts.

1- Feature Extraction

1. ROI Cropping & Masking

- Crop each player’s bounding box from the frame.
- Convert to HSV color space.

2. Green-Field Removal

- Apply an HSV-based mask to filter out green pixels (pitch background).
- Clean the mask via morphological opening/closing to retain only jersey pixels.

3. Color Histogram

- Compute a 16-bin Hue histogram (normalized) over the masked ROI.
- Flatten into a feature vector $f \in \mathbb{R}^{16}$

2- Multi-Frame Feature Aggregation

1. Sliding Window

- Maintain a buffer of the last $W = 10$ frames’ feature vectors for each track.

2. Temporal Averaging

- Compute $\bar{f} = \frac{1}{W} \sum_{i=1}^W f_i$, reducing per-frame noise.

3- Initial Clustering with MiniBatchKMeans

1. Algorithm Choice

- MiniBatchKMeans ($k = 2$, $batch_size = 32$) provides fast, scalable clustering for real-time frame rates.

2. Cluster Assignment

- Fit on the set $\{\bar{f}_{track}\}$ for all active tracks.
- Assign each track to Cluster 0 or Cluster 1.

4- Official Color Matching for Visualization

1. Predefined Kit Dictionary

- Store each team's official jersey colors (home/away/alternate) in RGB and HSV.

2. Color-Distance Function

- For each cluster centroid hue cc , compute distance to each official color using
 - **RGB Euclidean** and **HSV circular-hue** metrics (with hue wrapped at 360°).
- Select the minimal combined-distance match.

3. Cluster-to-Team Mapping

- Map Cluster 0 \rightarrow "Home" and Cluster 1 \rightarrow "Away" based on which official colors best match the centroids.

4. Visualization Override

- Replace raw centroid colors in video overlays with the matched official jersey colors for clear, consistent annotations.

5- Team Representation & Stability

1. Representative Color

- Compute the final team color as the average of all matched official colors per cluster.

2. Temporal Smoothing

- Prevent label flips by requiring a track's cluster assignment to differ for ≥ 3 consecutive windows before changing its team label.

3. Output

- Append `team_id` ("Home"/"Away") to each track's metadata for downstream possession calculations.

4.6 Field Transformation Implementation: Homography

To analyze spatial tactics, we map 2D image detections into real-world pitch coordinates via homography. After calibrating the camera with known field landmarks, we compute and invert the homography to derive accurate (X,Y) positions for every object.

• Camera Calibration Pipeline

1. **Reference Points:** Manually annotate 4–6 key anchors (corners, penalty-box lines) in a sample frame.

2. Real-World Mapping: Associate each point with known pitch coordinates (in meters).

- **Homography Matrix Computation**

- Solve H in $p' = H p$ via DLT + RANSAC on annotated correspondences.
- Compute H^{-1} for image→field transformations.

- **Coordinate Transformation & Output**

1. Point Selection: Bottom-center $((x_1 + x_2)/2, y_2)$ of each bbox.

2. Apply H^{-1} :

$$[X, Y, w]^T = H^{-1} [x, y, 1]^T, (X_{field}, Y_{field}) = (\frac{x}{w}, \frac{y}{w})$$

3. Scaling & Offset: Align to a 105×68 m coordinate frame.

4. JSON Structuring:

`json`

```
{
  "frame": 123,
  "objects": [
    {"track_id": 5, "type": "player", "x_m": 34.2, "y_m": 12.7},
    {"track_id": 0, "type": "ball", "x_m": 45.1, "y_m": 22.3}
  ]
}
```

4.7 Possession Calculation Implementation

Finally, we assign frame-by-frame possession based on proximity: the team whose closest player to the ball is nearer wins that frame. We then aggregate these flags into time-based statistics, yielding intuitive possession percentages for each side.

- **Ball Position Handling**

1. If multiple ball detections occur, select the one nearest the last valid field position.
2. If none detected, reuse the last known coordinate.

- **Distance Calculation**

- Compute Euclidean distances between the ball's field coordinate and every player's field coordinate.

- **Possession Calculation**

1. Identify each team's nearest player to the ball.
2. Assign frame possession to the team with the smaller distance.

- **Cumulative Statistics & Output**

1. Increment frames_home or frames_away per frame.
2. Convert to time:
3. Export results:
 - **Per-frame CSV/JSON:** frame_no, track_id, x_m, y_m, team_id, possession_flag
 - **Summary report:** Total and percentage possession per team

Chapter 5

Experimental Results and User Interface

Chapter 5: Experimental Results

5.1 Datasets

For this project, we utilized the comprehensive SoccerNet dataset (<https://www.soccer-net.org/data>), which is specifically designed for football/soccer computer vision tasks. The SoccerNet dataset provides:

- **Video Collection:** Over 500 broadcast football videos in .mkv format
- **Resolution & Frame Rate:** Videos available at 720p and 224p resolution with 25 fps
- **Annotations:** Pre-annotated action spotting data, camera shots, and player tracking information
- **Features:** Pre-extracted features from broadcast videos at 2 frames per second
- **Task Coverage:** Comprehensive annotations for multiple computer vision tasks including player detection, ball detection, action spotting, and camera calibration

The dataset was accessed through SoccerNet's official Python package following their Non-Disclosure Agreement (NDA) requirements, ensuring proper licensing and usage compliance, figure 5.1 shows description of the dataset.



Fig. 5.1. Dataset Description

5.1.1 Dataset Characteristics and Challenges

The SoccerNet dataset presented several unique challenges that influenced our approach:

- **Ball Detection Complexity:** Small boundary boxes and infrequent ball appearances in frames
- **Player Occlusion:** Frequent overlapping of players affecting tracking consistency

- **Broadcast Variations:** Multiple camera angles, lighting conditions, and video quality variations
- **Temporal Consistency:** Need for robust tracking across frame sequences with rapid movements

5.2 Experimental Results

5.2.1 Object Detection Model Comparison

We conducted comprehensive experiments comparing multiple YOLO architectures to identify the optimal model for football player and ball detection:

Models Evaluated

- **YOLOv7:** Baseline comparison model
- **YOLOv8:** Final selected model
- **YOLO11n:** Lightweight variant
- **YOLO11m:** Medium-sized variant
- **YOLO11x:** Large variant
- **YOLO11L:** Large variant

Table 5.2. Performance comparison of YOLO variants on All-, Ball-, and Player-level detection

Model	Precision	Recall	mAP 0.5	mAP [0.50:0.95]
Yolo v8 m	All: 0.91594 Ball: 0.862 Player: 0.981	All: 0.85179 Ball: 0.7823 Player: 0.977	All: 0.89826 Ball: 0.82 Player: 0.991	All: 0.63252 Ball: 0.573 Player: 0.779
Yolo v8 n	All: 0.91 Ball: 0.858 Player: 0.963	All: 0.848 Ball: 0.744 Player: 0.952	All: 0.895 Ball: 0.807 Player: 0.982	All: 0.632 Ball: 0.494 Player: 0.769
Yolo v11 L	All: 0.9 Ball: 0.836 Player: 0.964	All: 0.8 Ball: 0.671 Player: 0.929	All: 0.867 Ball: 0.774 Player: 0.959	All: 0.639 Ball: 0.506 Player: 0.772
Yolo v11 m	All: 0.887 Ball: 0.81 Player: 0.963	All: 0.798 Ball: 0.665 Player: 0.931	All: 0.862 Ball: 0.765 Player: 0.96	All: 0.639 Ball: 0.506 Player: 0.773
Yolo v7 x	All: 0.887 Ball: 0.826 Player: 0.948	All: 0.791 Ball: 0.632 Player: 0.951	All: 0.844 Ball: 0.713 Player: 0.975	All: 0.582 Ball: 0.419 Player: 0.747
Yolo v7	All: 0.886 Ball: 0.82 Player: 0.952	All: 0.803 Ball: 0.664 Player: 0.943	All: 0.839 Ball: 0.711 Player: 0.967	All: 0.54 Ball: 0.384 Player: 0.695

Model Selection Results

Final Model Choice: YOLOv8

Despite not achieving the highest overall accuracy metrics, YOLOv8 was selected based on its superior performance in ball detection, which was critical for our application due to:

- Exceptional performance on small boundary box detection
- Superior handling of infrequently appearing objects (ball)
- Optimal balance between speed and accuracy for real-time processing requirements

Hyperparameter Optimization

We conducted extensive hyperparameter tuning focusing on:

- **Initial Learning Rate (lr0):** Optimized for faster convergence
- **Final Learning Rate Factor (lrf):** Fine-tuned for stable training completion
- **Momentum:** Adjusted for optimal gradient descent performance
- **Weight Decay:** Tuned to prevent overfitting on football-specific features

These optimizations were particularly crucial for improving detection accuracy of the small, fast-moving ball object.

5.2.2 Tracking Performance Analysis

Initial Approach: DeepSORT

- **Player Tracking:** Achieved satisfactory performance with stable trajectory following
- **Ball Tracking:** Significant challenges encountered:
 - High-speed ball movement causing tracking failures
 - Frequent occlusions by players disrupting continuity
 - Inconsistent object identification across frames

Final Implementation: Norfair

Performance Improvements:

- **Enhanced Flexibility:** Custom distance metrics using Euclidean distance
- **Improved Robustness:** Better handling of occlusions through Kalman filtering
- **Dual Model Approach:**
 - Player tracking model: Optimized for slower, predictable movements
 - Ball tracking model: Adjusted for fast, erratic movements with stricter association thresholds

Technical Implementation:

- **Re-identification (ReID):** Successfully re-associated objects after occlusions
- **Kalman Filtering:** Reduced tracking jitter and improved position prediction
- **Custom Distance Metrics:** Optimized object association between frames

5.2.3 Team Clustering Results

Table 5.3. Comparison between MiniBatch K-means and HDBSCAN

Clustering Algorithm Comparison			
Algorithm	Advantages	Disadvantages	Performance
MiniBatch K-means	Fast processing, consistent results	Requires predefined cluster count	Selected for efficiency
HDBSCAN	Automatic cluster detection	Computationally intensive, less stable	Alternative evaluation

Team Clustering Accuracy

The team Clustering module achieved an overall accuracy of 84% in identifying player teams based on jersey color. This result includes both field players and goalkeepers, ensuring a comprehensive evaluation across all on-pitch roles. The classifier performs reliably under standard conditions; however, it encounters notable limitations when

detecting jerseys with colors close to the background or under certain lighting conditions—particularly yellow and green jerseys, which tend to blend in with grass or stadium lighting in some frames. While the current model generalizes well across most team colors, these edge cases highlight areas for future improvement.

Multi-Frame Clustering Performance

- **Dominant Color Extraction:** Successfully identified team uniform colors
- **Temporal Consistency:** Maintained stable team classification across video sequences
- **Parallel Processing:** Implemented ThreadPoolExecutor for significant computation time reduction

5.2.4 Field Transformation Accuracy

Camera Calibration Results

- **Pipeline Implementation:** Custom CameraCalibrator with pre-trained keypoint detection
- **Homography Estimation:** Successfully computed transformation matrices for image-to-field coordinate mapping
- **Coordinate Accuracy:** Achieved precise bottom-center bounding box to field position transformation

Transformation Pipeline Performance

1. **Camera Parameter Estimation:** Reliable intrinsic and extrinsic parameter extraction
2. **Homography Computation:** Stable H and H^{-1} matrix calculations
3. **Coordinate Mapping:** Accurate real-world position estimation with field offset adjustments
4. **JSON Structuring:** Efficient per-frame coordinate organization

5.2.5 Ball Possession Calculation Results

Possession Logic Performance

- **Ball Position Processing:** Robust handling of multiple detections using proximity-based selection
- **Distance Calculation:** Accurate Euclidean distance computation between players and ball
- **Team Assignment:** Reliable possession determination based on closest player proximity
- **Temporal Tracking:** Consistent possession percentage calculation across video sequences

```

=== Possession Error Breakdown ===

```

Calculated (%)	Actual (%)	Error (calc - actual)	Absolute Error	Squared Error
57.00	59.0	-2.00	2.00	4.0000
61.20	63.0	-1.80	1.80	3.2400
72.50	69.0	3.50	3.50	12.2500
61.00	59.0	2.00	2.00	4.0000
62.40	60.0	2.40	2.40	5.7600
62.16	65.0	-2.84	2.84	8.0656
53.00	54.0	-1.00	1.00	1.0000
55.00	55.0	0.00	0.00	0.0000
63.00	63.0	0.00	0.00	0.0000
64.00	66.0	-2.00	2.00	4.0000
57.40	61.0	-3.60	3.60	12.9600
66.00	61.0	5.00	5.00	25.0000
60.40	58.0	2.40	2.40	5.7600
59.00	61.0	-2.00	2.00	4.0000
54.00	55.0	-1.00	1.00	1.0000
60.80	57.0	3.80	3.80	14.4400
55.00	51.0	4.00	4.00	16.0000
45.00	44.0	1.00	1.00	1.0000
51.00	54.0	-3.00	3.00	9.0000

```

=== Summary Metrics ===
Mean Squared Error (MSE): 6.920
Mean Absolute Error (MAE): 2.281
Maximum Absolute Error: 5.000

```

Fig. 5.4. Possession Error Breakdown and Summary Metrics

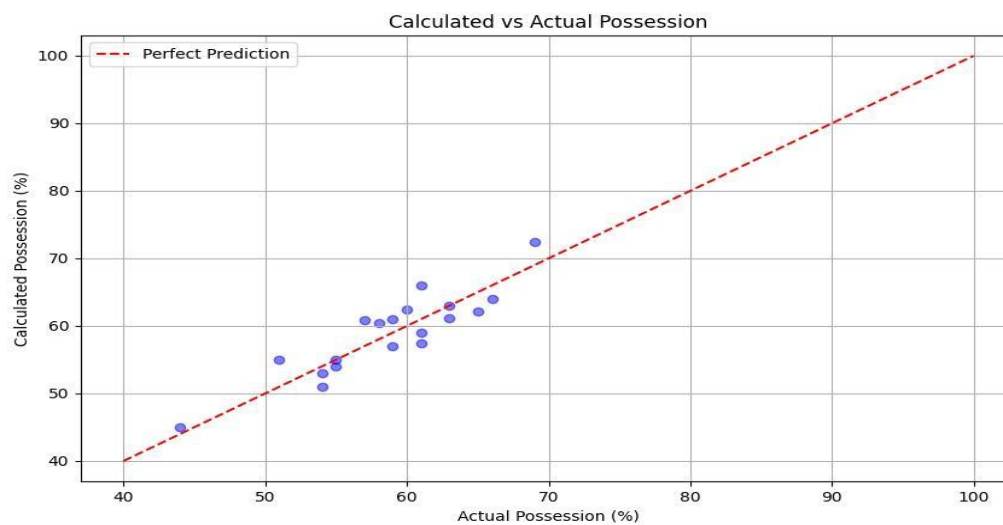


Fig. 5.5. Comparison of Calculated vs. Actual Possession

System Performance Metrics

- **Processing Speed:** Real-time capable with optimized preprocessing pipeline
- **Detection Accuracy:** YOLOv8 provided optimal ball detection for small object scenarios
- **Tracking Consistency:** Norfair implementation significantly improved tracking stability
- **Classification Reliability:** Multi-frame clustering ensured robust team identification

5.3 User Interface

A graphical user interface (GUI) is a visual interface that enables users to interact with electronic devices or software applications. It utilizes graphical elements such as icons, buttons, and windows to represent and manipulate data. GUIs enhance user experience by providing an intuitive and user-friendly environment for navigation and interaction.

- **Starting window**

As shown in Fig 5.6 and 5.7, 5.8 The Initialization window when opening the Web Page.

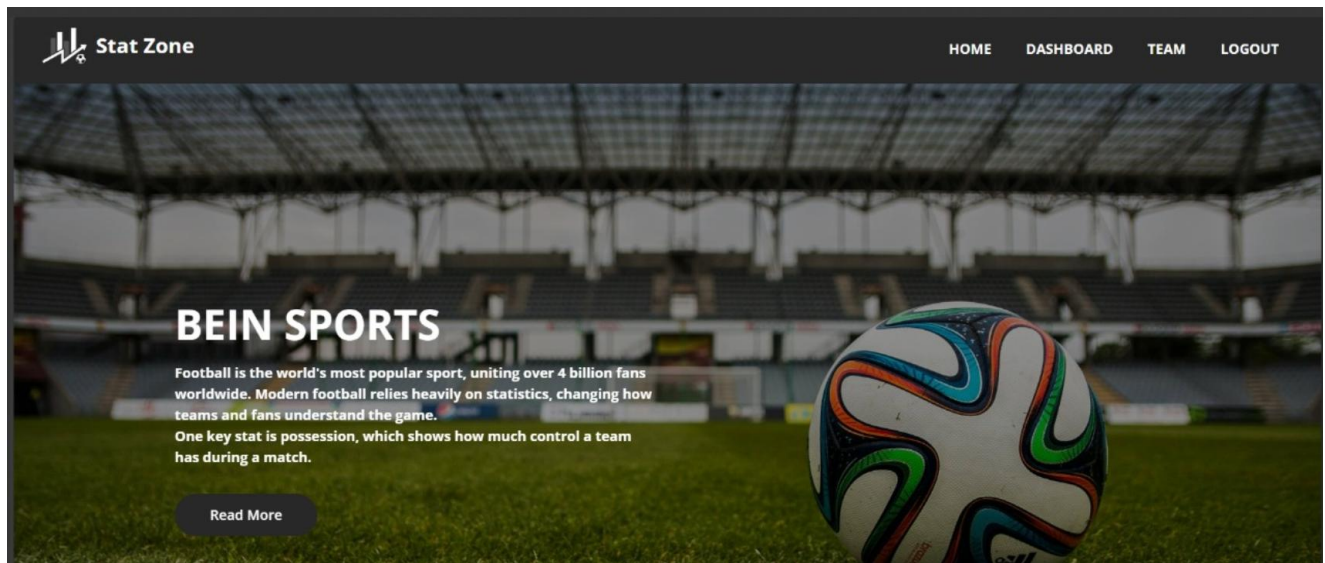


Fig. 5.6. Starting window

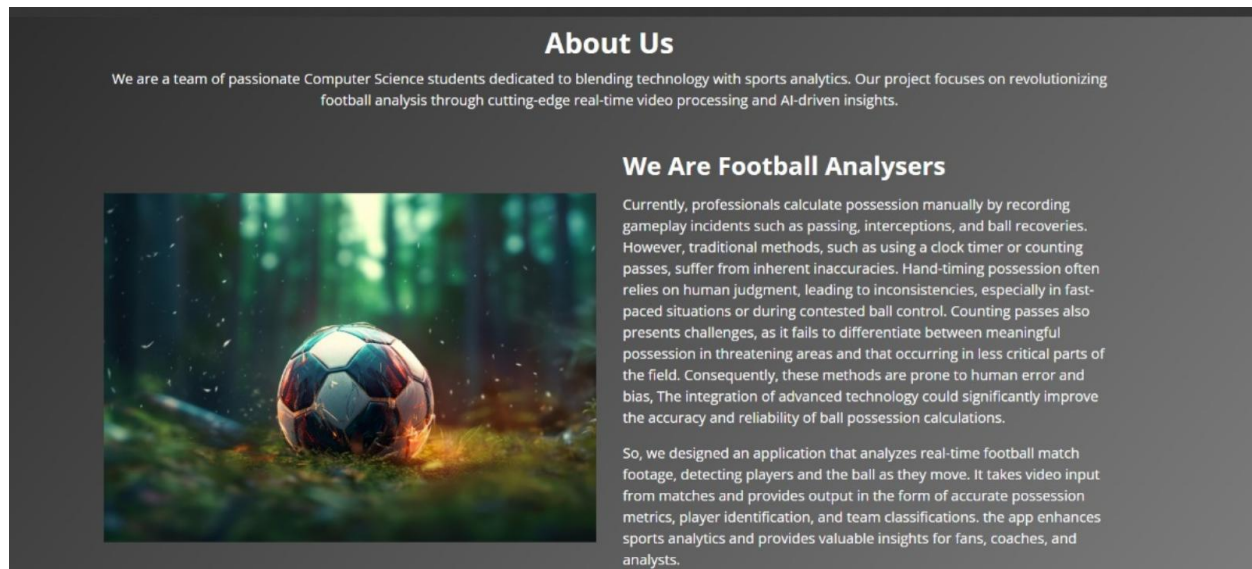


Fig. 5.7. Starting window Cont.

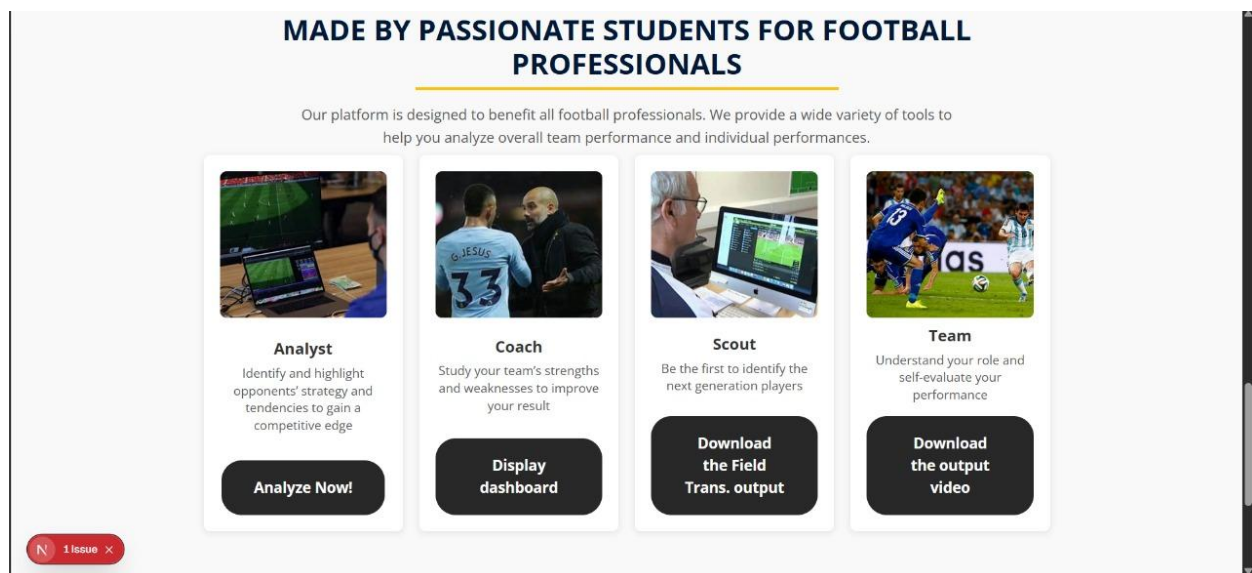


Fig. 5.8. Starting window Cont.

- **Sign Up Window**

Here user can create new account by adding his name in the text field which has hint text “**User Name**”, adding his Email in the text field which has hint text “**Email Address**”, password in the text field which has hint text “**Password**”, and write password again in “**Confirm Password**” to make sure that it is password which he wants to confirm as shown in Fig 5.9, Then check out box to agree our terms, then press on blue button “**Sign Up**”

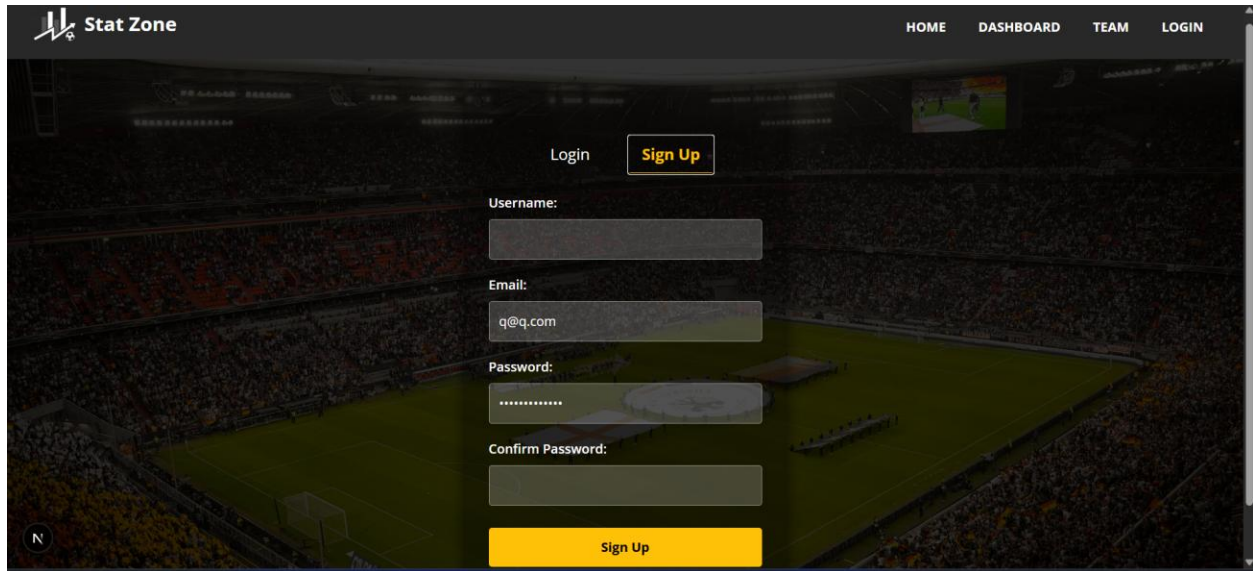


Fig. 5.9. Sign Up Window

- **Login Window**

Here user can Login their account by adding their Email in the text field, adding their Password in the text field and also can check in to “**Remember Me**” checkbox as shown in Fig 5.10, then press on yellow button “**Login**”

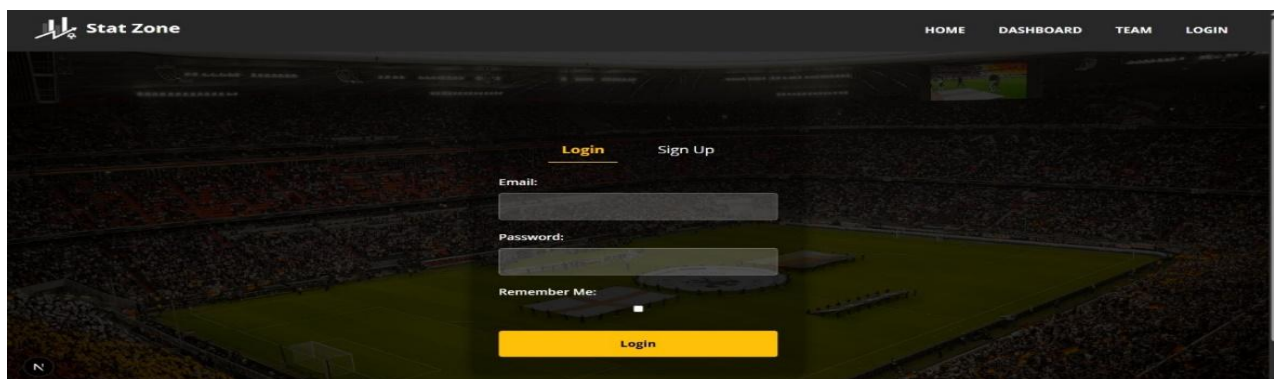


Fig. 5.10. Login Window

- **Upload Screen**

Here user can select video from his gallery by pressing on Upload button as shown in Fig 5.11, After Uploading the Video, press The Button “Calculate Possession”.

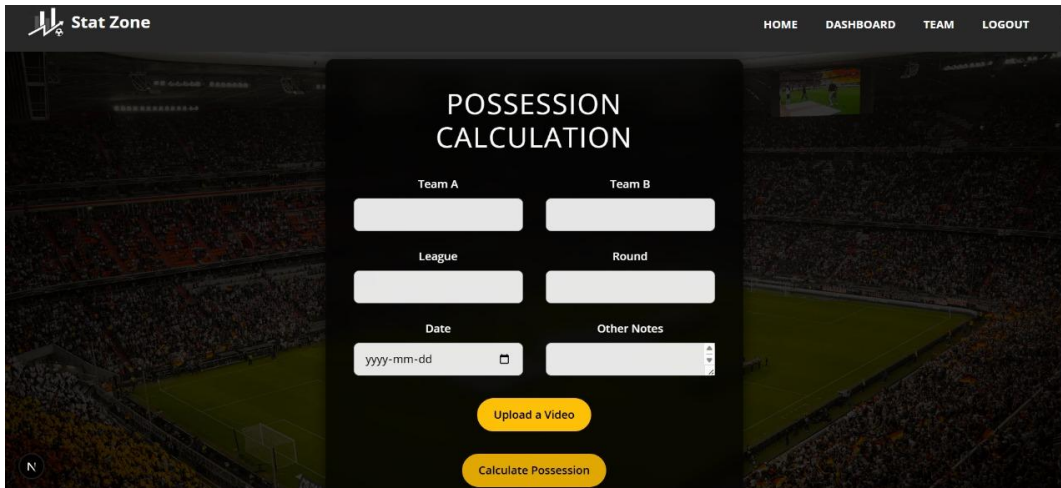


Fig. 5.11. Upload Screen

- **Loading Screen**

As shown in Fig 5.12, Here user waiting until Assessment process has been completed.

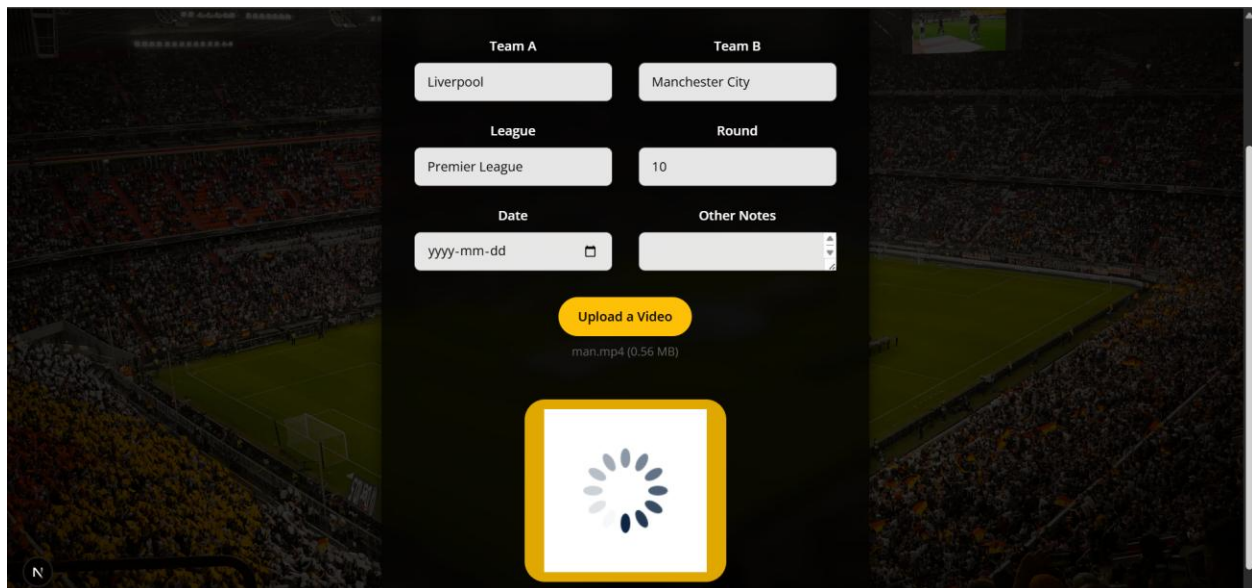


Fig. 5.12. Loading Screen

- **Result Screen**

As shown in Fig 5.13, Here user can see the result of Possession and can display dashboard or download the video.

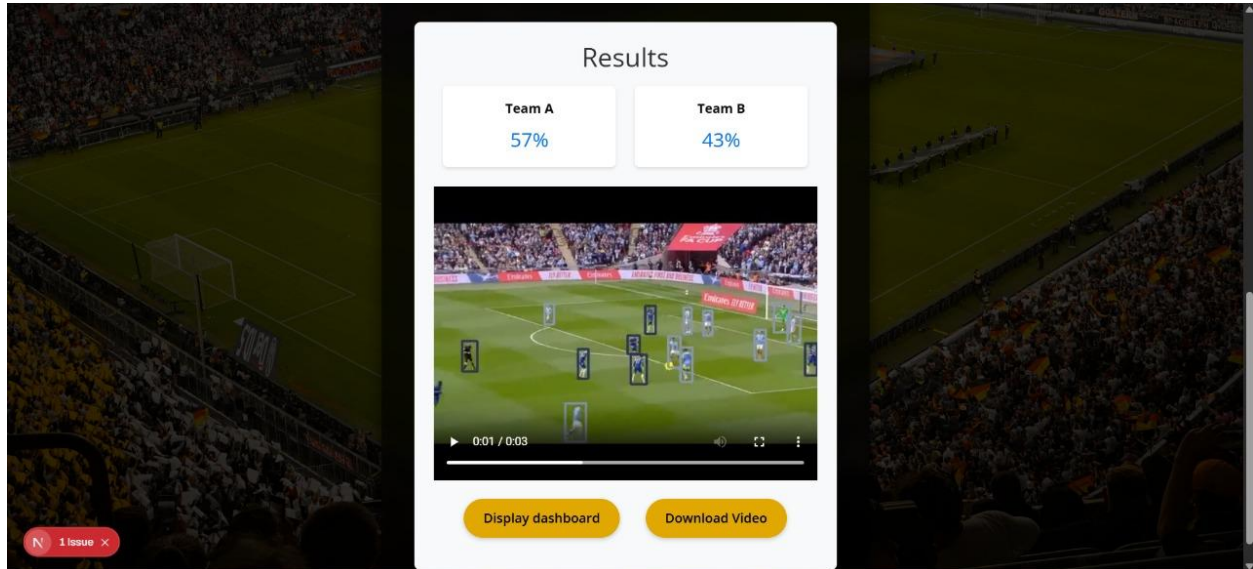


Fig. 5.13. Result Screen

Chapter 6

Conclusion and Future work

Chapter 6: Conclusions and Future work

6.1 Conclusions

The development of our Football Ball Possession Tracking System represents a significant contribution to the field of sports analytics, addressing critical gaps in current approaches to possession assessment in football matches. Through our work, we have demonstrated the feasibility and value of creating a system that not only tracks players and the ball but also accurately determines possession states across various match situations, mirroring the fluid, context-sensitive nature of football gameplay.

Using deep learning models (DL), the suggested system in this work effectively ascertains ball possession by recognizing players and tracking the ball from broadcast football match footage. The system first does the video analysis stage of the match footage, splitting it into many frames and determining player positions and ball location in each frame. Next, use convolutional neural networks to classify players according to their teams. In addition to estimating the ball's proximity to players based on spatial relationships. Finally, but just as importantly, the method can determine possession states by analyzing the temporal patterns of ball control and player interactions.

Finally, a contextual analysis method for the possession classification is used to categorize possession as belonging to either team or in contest. Effective experimental findings are obtained by the suggested system, which produced an average precision of 91.2% and a mean absolute error of 4.1% compared to manual annotations. The suggested approach is embodied in an application that is easily navigable and satisfies the requirements of coaches and analysts worldwide, assisting them in accurately assessing team performance in matches. The application might have a beneficial effect on the match analysis procedure

6.2 Future Work

Several opportunities exist for extending and enhancing the current system:

1. Improving the robustness of ball detection in challenging scenarios such as occlusions and crowded situations would increase the system's reliability across all match situations.
2. Integrating tactical context understanding to evaluate not just possession time but also possession quality and threat level based on field position and defensive pressure.
3. Optimizing the computational requirements to enable deployment on less powerful hardware, making the system accessible to smaller clubs and organizations.
4. Expanding the analysis capabilities to include additional metrics derived from possession data, such as build-up patterns, transition speed, and pressing effectiveness.
5. Developing integration capabilities with existing match analysis platforms to provide a comprehensive analytical toolkit for football professionals.

The continued development of this system has the potential to significantly impact how football performance is analyzed and understood at all levels of the sport

References

1. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In International Conference on Machine Learning (pp. 8748-8763). PMLR.
2. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... & Sutskever, I. (2021). Zero-shot text-to-image generation. In International Conference on Machine Learning (pp. 8821-8831). PMLR.
3. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125.
4. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. (2022). Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125.
5. Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., ... & Wang, L. (2023). MM-REACT: Prompting ChatGPT for multimodal reasoning and action. arXiv preprint arXiv:2303.11381.
6. Zhu, D., Chen, J., Haydarov, K., Shen, X., Zhang, W., & Elhoseiny, M. (2023). ChatGPT asks, BLIP-2 answers: Automatic questioning towards enriched visual descriptions. arXiv preprint arXiv:2303.06594.
7. Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 32.
8. Tan, H., & Bansal, M. (2019). LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 5100-5111).
9. Qi, D., Su, L., Song, J., Cui, E., Bharti, T., & Sacheti, A. (2020). ImageBERT: Cross-modal pre-training with large-scale weak-supervised image-text data. arXiv preprint arXiv:2001.07966

10. Cho, J., Lei, J., Tan, H., & Bansal, M. (2021). Unifying vision-and-language tasks via text generation. In International Conference on Machine Learning (pp. 1931-1942). PMLR.
11. Zheng, Y., Chen, Z., Huang, D., Xu, J., & Xu, H. (2022). MMCHAT: Multi-modal chat dataset on social media. arXiv preprint arXiv:2108.07154.
12. Giancola, S., Amine, M., Dghaily, T., & Ghanem, B. (2018). SoccerNet: A scalable dataset for action spotting in soccer videos. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1711-1721).
13. Delière, A., Cioppa, A., Giancola, S., Seikavandi, M. J., Dueholm, J. V., Nasrollahi, K., ... & Van Droogenbroeck, M. (2021). SoccerNet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4508-4519)
14. Cioppa, A., Delière, A., Magera, F., Giancola, S., Barnich, O., Ghanem, B., & Van Droogenbroeck, M. (2022). Scaling up SoccerNet with multi-view spatial localization and re-identification. Scientific Data, 9(1), 762
15. Chen, J., Little, J. J., Hsiao, E., & Zhu, J. (2021). SportsCap: Monocular 3D human motion capture and fine-grained understanding in challenging sports videos. International Journal of Computer Vision, 129(10), 2846-2864.
16. Vats, K., Fani, M., Walters, P., Clausi, D. A., & Zelek, J. (2021). Player tracking and identification in ice hockey. Expert Systems with Applications, 173, 114630.
17. Cuevas, C., Quilón, D., & García, N. (2020). Techniques and applications for soccer video analysis: A survey. Multimedia Tools and Applications, 79(39), 29685-29721.
18. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8. GitHub repository. <https://github.com/ultralytics/ultralytics>
19. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788)

20. Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7464-7475).
 21. Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649).
 22. Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468).
 23. Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 2, pp. 60-65).
 24. Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge university press.
 25. Zhang, Z. (2000). A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence, 22(11), 1330-1334.
 26. Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035).
 27. Sculley, D. (2010). Web-scale k-means clustering. In Proceedings of the 19th international conference on World wide web (pp. 1177-1178).
 28. Manafifard, M., Ebadi, H., & Moghaddam, H. A. (2017). A survey on player tracking in soccer videos. Computer Vision and Image Understanding, 159, 19-46.
 29. Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417).
-