project

Venkata Manikanta Prem Sai Potukuchi

December 2025

1 Introduction

YOLO Object Detection

We employ YOLO (You Only Look Once), a single-stage deep convolutional neural network, for real-time object detection in football match videos. Given an input frame, YOLO performs object localization and classification in a single forward pass.

Formally, for each frame I_t, the detector outputs a set of bounding boxes:

$$B_t = \{(x_i, y_i, w_i, h_i, c_i, p_i)\}_{i=1}^{N_t},$$

where (x_i, y_i, w_i, h_i) denote bounding box coordinates, c_i is the class label (player, referee, or ball), and p_i is the confidence score.

YOLO is chosen due to its high inference speed and robustness in detecting multiple objects under dynamic football match conditions.

—

Object Tracking

Object detection alone does not preserve identity across frames. Therefore, a multi-object tracking algorithm is applied to associate detections temporally.

The tracker assigns a unique identity (ID) to each detected player and maintains this ID across consecutive frames by leveraging spatial proximity and motion continuity:

$$Track_k = \{(x_t^k, y_t^k)\}_{t=1}^{T}.$$

This process enables reconstruction of player trajectories over time and allows storage of movement history, which is essential for subsequent motion analysis.

—

Team Assignment Using K-Means Clustering

To automatically identify teams, we use unsupervised clustering based on jersey color information.

For each detected player bounding box, dominant color features are extracted in the RGB or HSV color space. These feature vectors are clustered using K-Means clustering with $k = 2$:

$$\min_{\{\mu_1,\mu_2\}} \sum_{i=1}^{N} \min_{j\in\{1,2\}} \|x_i - \mu_j\|^2,$$

where x_i represents the color feature of the i-th player.

Each cluster corresponds to one team, enabling automatic team assignment without labeled training data.

—

Camera Motion Estimation Using Optical Flow

Football broadcast videos exhibit significant camera motion such as panning and zooming, which introduces errors in raw motion estimation.

To compensate for this, optical flow is computed between consecutive frames to estimate pixel-wise motion:

$$\mathbf{v}(x, y) = (u(x, y), v(x, y)),$$

where $\mathbf{v}$ represents the apparent motion vector.

The estimated camera motion component is subtracted from player trajectories, effectively separating camera-induced motion from true player movement and improving motion accuracy.

—

Perspective Transformation

Pixel distances in image space do not directly correspond to real-world distances. To address this, a perspective transformation (homography) is applied.

A homography matrix H maps image coordinates (x, y) to ground-plane coordinates (X, Y):

$$XY1 = Hxy1.$$

This transformation converts the football field into a top-down metric space, allowing motion measurements to be expressed in real-world units (meters).

—

Analytics Computation

Using the corrected and transformed trajectories, player-level analytics are computed.

Distance Covered The total distance covered by a player is calculated as the cumulative Euclidean distance between consecutive positions:

$$D = \sum_{t=1}^{T-1} \sqrt{(X_{t+1} - X_t)^2 + (Y_{t+1} - Y_t)^2}.$$

Speed Estimation Instantaneous speed is estimated as:

$$v_t = \frac{\Delta d}{\Delta t},$$

where Δt is derived from the video frame rate.

These metrics provide quantitative insights into player workload and performance.

—

Technical Summary

This project integrates deep learning-based object detection, multi-object tracking, unsupervised clustering, optical flow-based camera motion compensation, and geometric perspective transformation to extract real-world football analytics from broadcast video data.