Football Live Match Analysis Using YOLO

Ahmed
El-Manylawi

Ali
Ahmed

Ahmed
Reda

Hazem
Ashraf

Ahmed
Ali

Ahmed
Said

ABOUT PROJECT

- **The goal of this project is to detect and track players, referees, and footballs in a video using YOLO, one of the best AI object detection models available. We will also train the model to improve its performance. Additionally, we will assign players to teams based on the colors of their t-shirts using Kmeans for pixel segmentation and clustering. With this information, we can measure a team's ball acquisition percentage in a match. We will also use optical flow to measure camera movement between frames, enabling us to accurately measure a player's movement. Furthermore, we will implement perspective transformation to represent the scene's depth and perspective, allowing us to measure a player's movement in meters rather than pixels. Finally, we will calculate a player's speed and the distance covered. This project covers various concepts and addresses real-world problems, making it suitable for both beginners and experienced machine learning engineers.**

Problem Statement

In football analytics, extracting meaningful insights such as player movement, ball possession, team control, and individual performance traditionally requires manual video annotation—an extremely time-consuming and error-prone process.

Current automated systems are either expensive, closed-source, or require advanced hardware. There is a need for a fully automated, low-cost, and accurate computer-vision pipeline that can:

- Detect and track players, referees, and the football in real time
- Assign players to their respective teams
- Measure movement, speed, and distance covered
- Estimate ball possession over time
- Correctly account for camera motion in broadcast footage

This project addresses these challenges using modern computer vision, deep learning, and geometric analysis techniques.

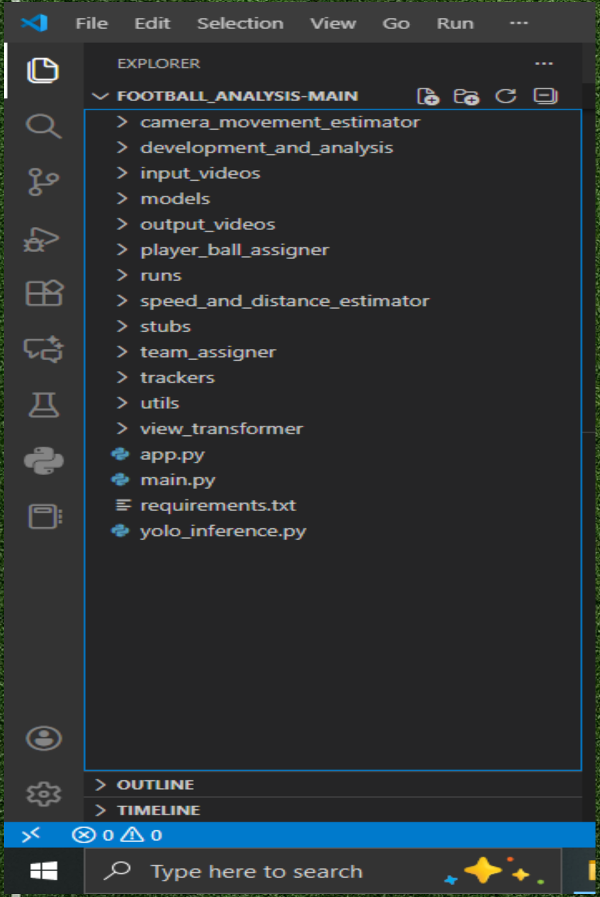
Objective

1. Detecting players, referees, and the football using a trained YOLO-based object detection model.
2. Tracking players across frames to maintain consistent identities.
3. Classifying players into teams based on t-shirt colors using K-means clustering.
4. Calculating ball possession statistics for each team across the match timeline.
5. Measuring camera movement using optical flow to ensure accurate movement tracking.
6. Applying perspective transformation to convert pixel movement into real-world distance (meters).
7. Calculating player speeds and total distance covered throughout the match.

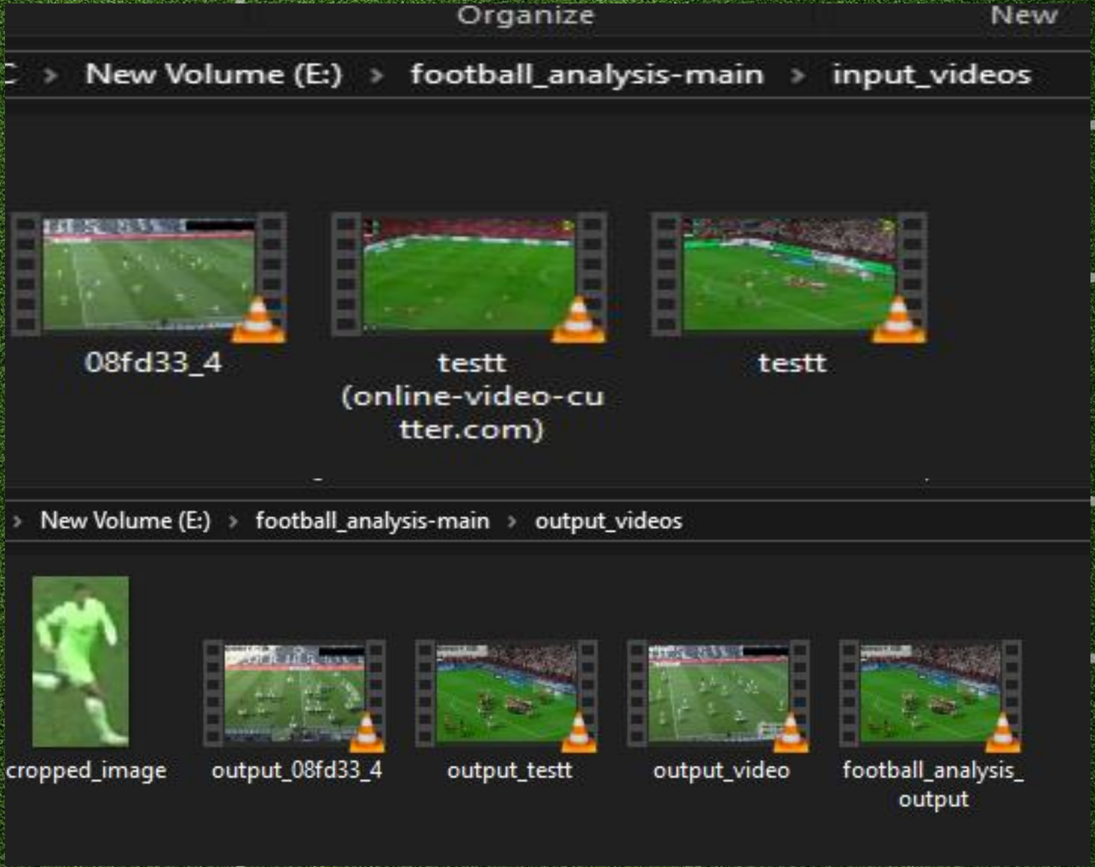
Modules / Components

Component / Module	Used For
OpenCV	Measures camera movement to correct player motion.
YOLO (Ultralytics)	Detects players, referees, and the football in every frame.
SORT / Custom Tracking Logic (inside Tracker)	Tracks players and ball across frames and assigns IDs.
K-Means (scikit-learn / custom implementation)	Classifies players into teams based on dominant t-shirt colors.
Streamlit	Web application UI (file upload, progress bars, video display).

DATA AND FILES



FOOTBALL_ANALYSIS-MAIN
- camera_movement_estimator
- development_and_analysis
- input_videos
- models
- output_videos
- player_ball_assigner
- runs
- speed_and_distance_estimator
- stubs
- team_assigner
- trackers
- utils
- view_transformer
- app.py
- main.py
- requirements.txt
- yolo_inference.py



New Volume (E:) › football_analysis-main › input_videos

- 08fd33_4
- testt (online-video-cutter.com)
- testt

New Volume (E:) › football_analysis-main › output_videos

- cropped_image
- output_08fd33_4
- output_testt
- output_video
- football_analysis_output

1. Object Detection

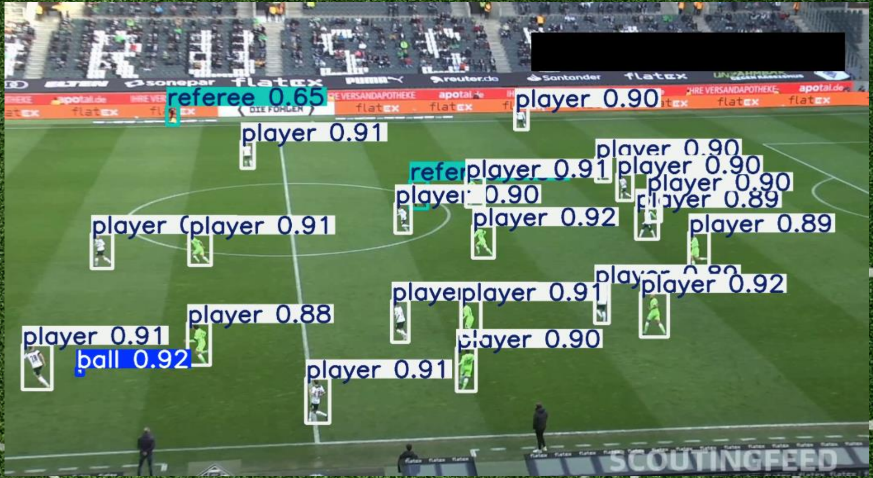
Model: Yolo5

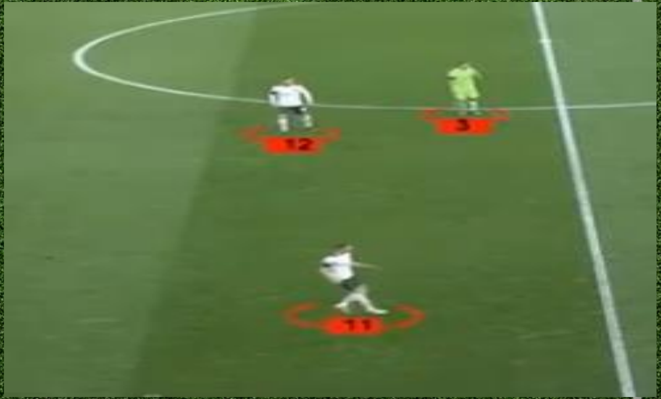
Use: **Detects players, referees, and the football in every frame.**

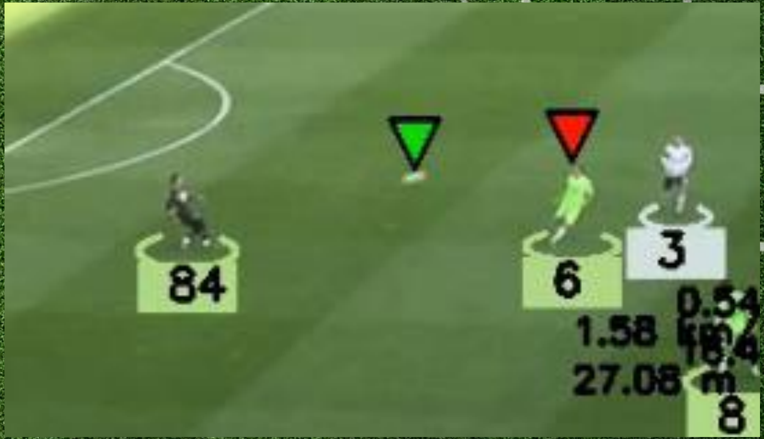
2-Trackers

Model: SORT

Use: Tracks players and ball across frames and assigns IDs.


[redacted]

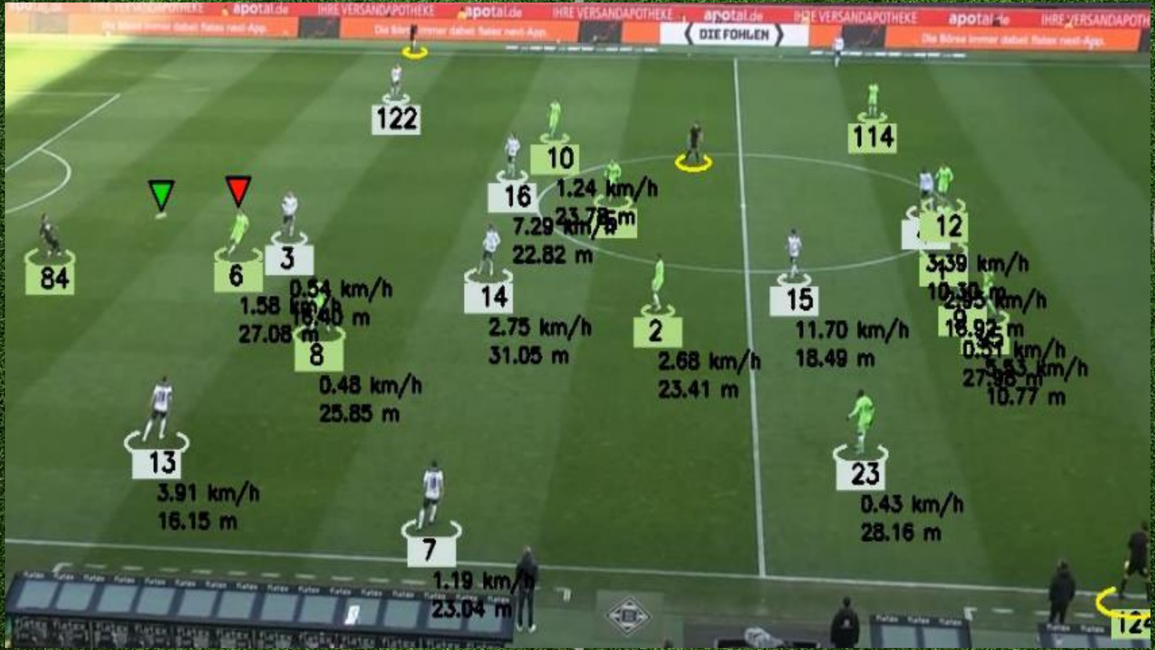




3-Speed & Distance

Model: cv2,sys

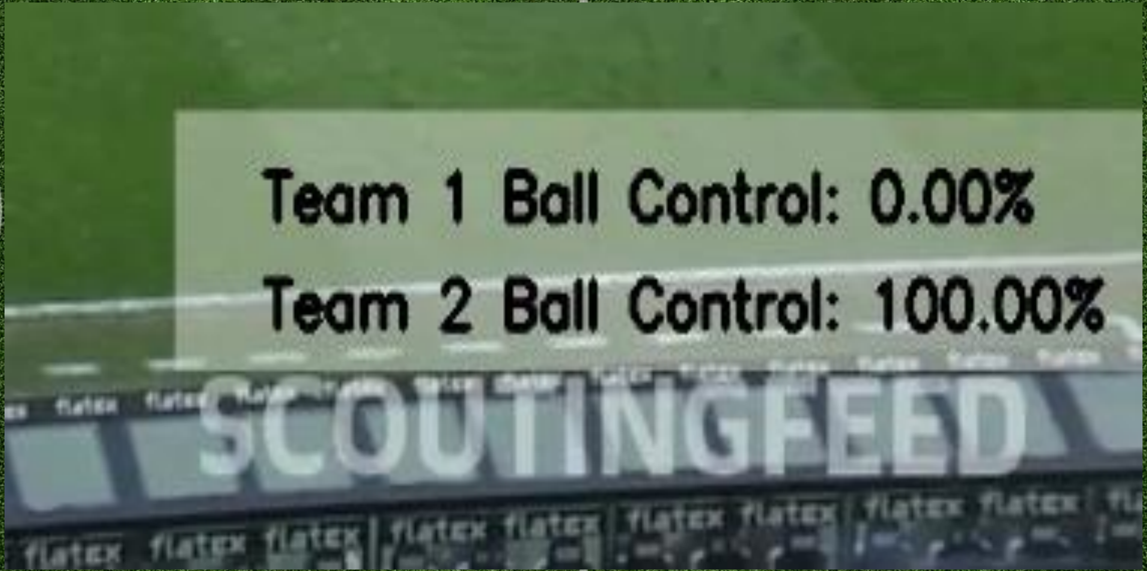
Use: Predicts speed & distance



3- Ball Possesion

Model: utils

Use: possession of teams

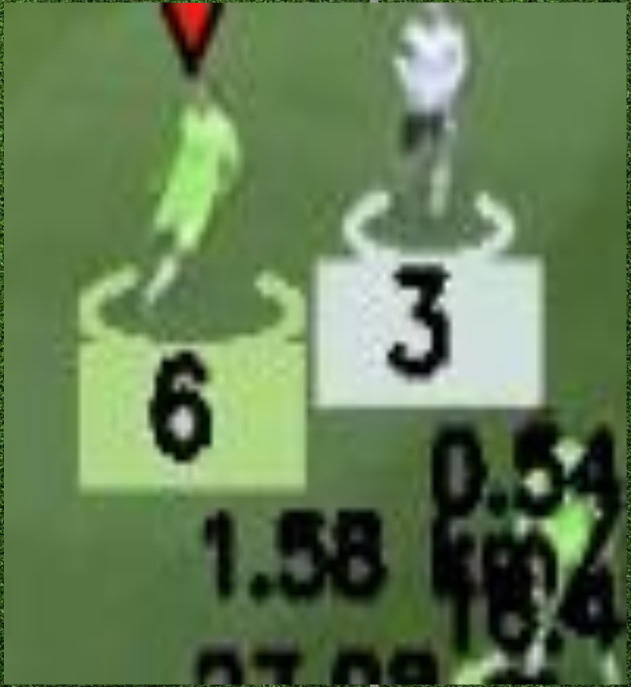


```
Team 1 Ball Control: 0.00%
Team 2 Ball Control: 100.00%
```

4- color assignement

Model: cv2,matplotlib

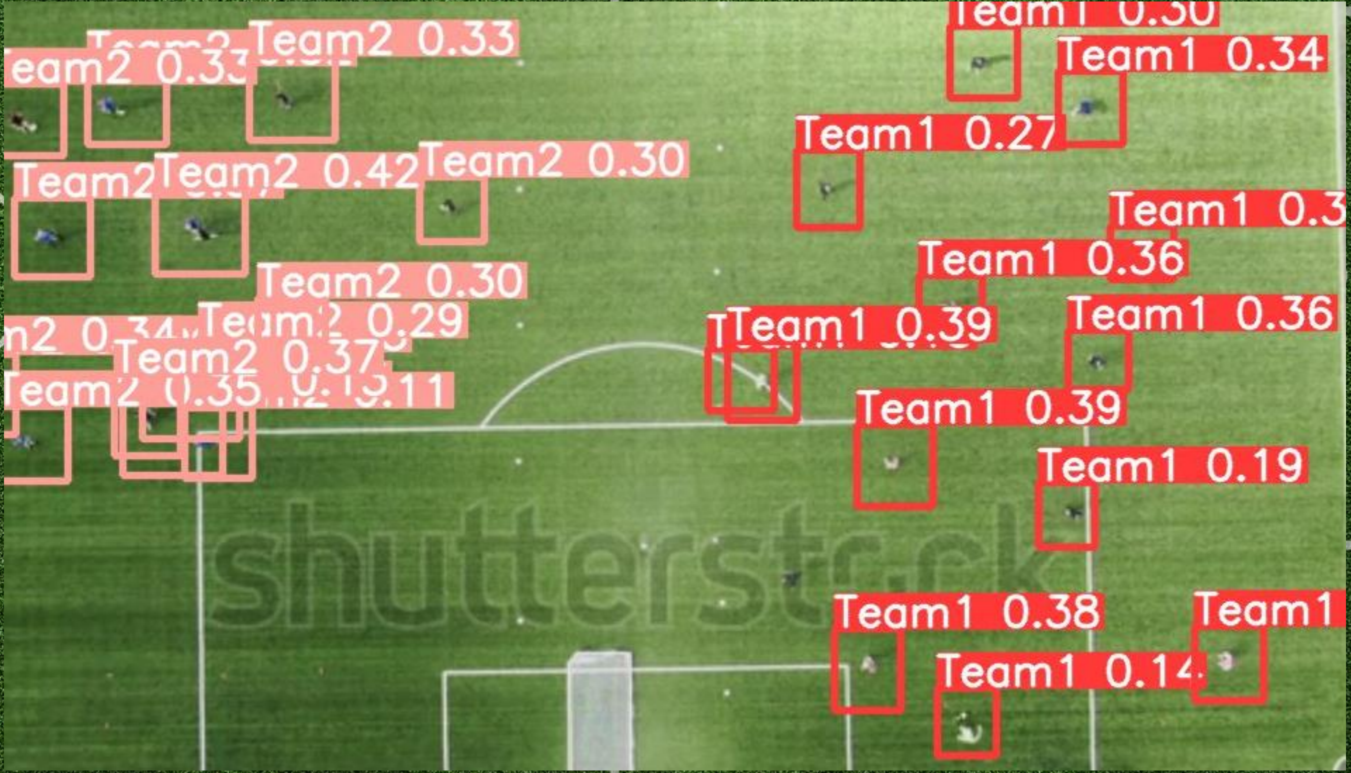
Use: color the circle around the players of each team



5- Team assigner

Model: kmeans

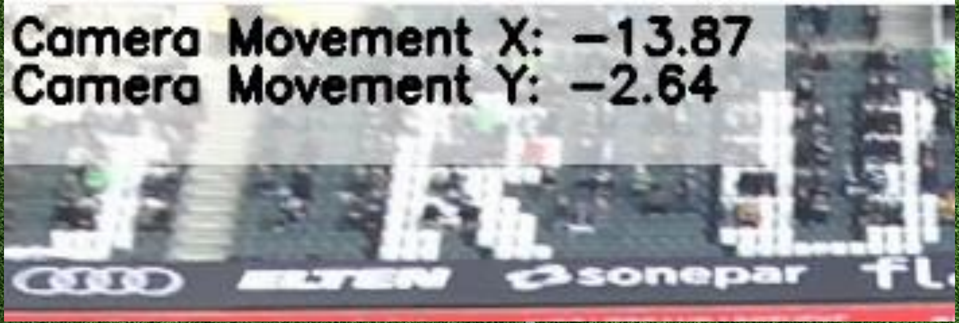
Use: detect which player belong to team 1 or 2



6- Camera Movemnet

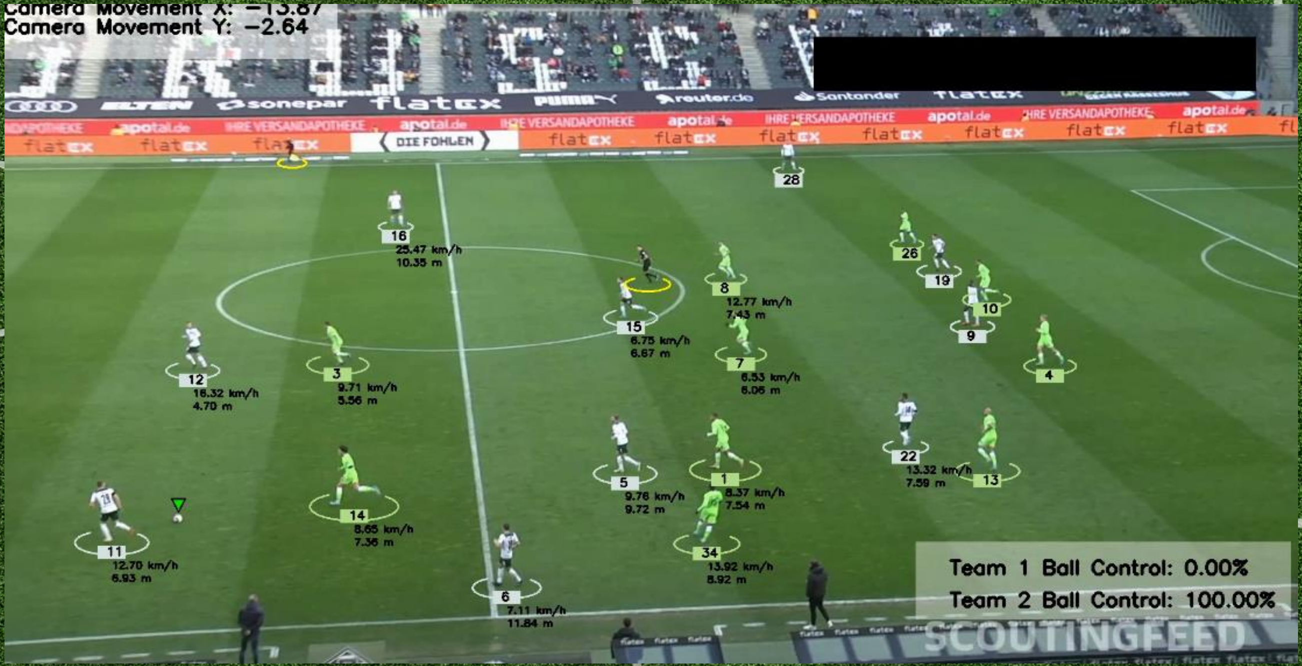
Model: picle,cv2

Use: Camera adjustment for speed and distance



Camera Movement X: -13.87
Camera Movement Y: -2.64

FINAL RESULTS

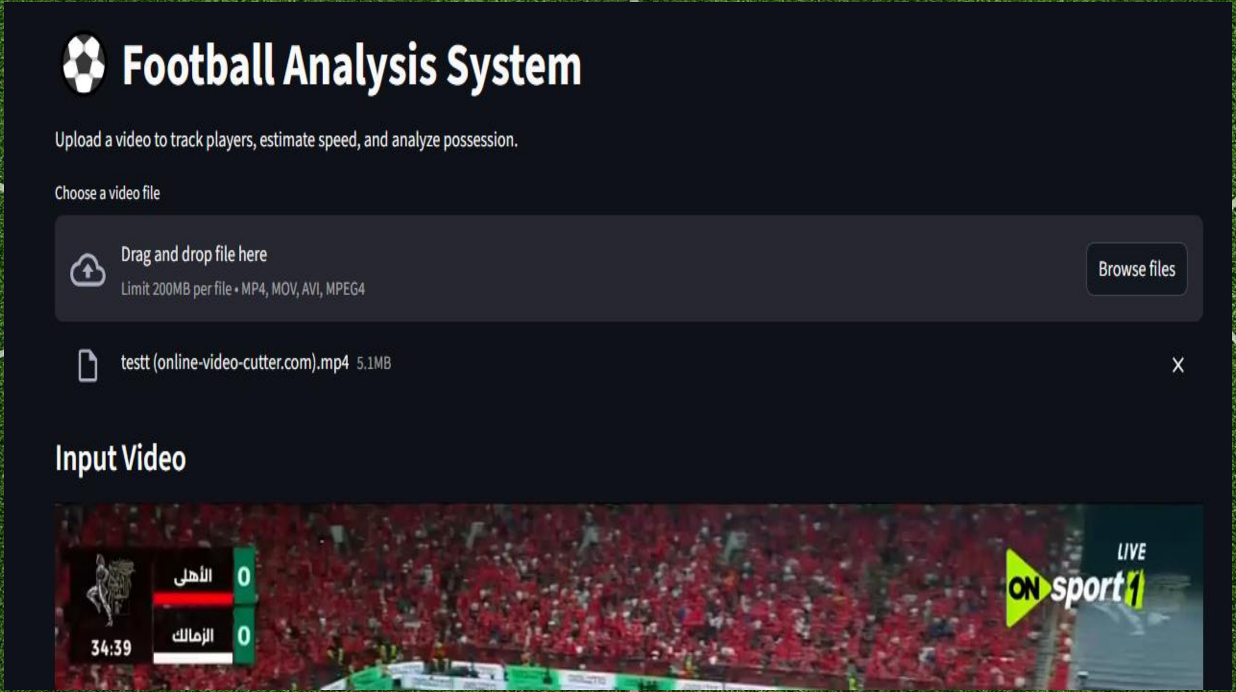


```
Camera Movement X: -13.87
Camera Movement Y: -2.64
```

[redacted]

```
Team 1 Ball Control: 0.00%
Team 2 Ball Control: 100.00%
```

FINAL RESULTS



⚽ Football Analysis System

Upload a video to track players, estimate speed, and analyze possession.

Choose a video file

Drag and drop file here

Limit 200MB per file • MP4, MOV, AVI, MPEG4

Browse files

testt (online-video-cutter.com).mp4 5.1MB

Input Video