© December 2025 | IJIRT | Volume 12 Issue 7 | ISSN: 2349-6002

Real Time Football Analytics System

Mihir Gahukar[1], Omkar Jagtap[2], Vedant Divate[3], Prajwal Gidde[4], Aarya Gondikar[5], Prof. Sagar Janokar[6]
[1,2,3,4,5] Student, Artificial Intelligence and Data Science Department, Vishwakarma Institute of Technology, Pune, India
[6] Faculty, Artificial Intelligence and Data Science Department, Vishwakarma Institute of Technology, Pune, India

***Abstract*—post-match analytics in association football enables coaches and analysts to identify strengths, weaknesses, and tacti- cal opportunities. However, existing professional tools are costly and often require multi-camera setups or proprietary sensors. We present the Real Time Football Analytics Sys- tem (RTFAS), a low-cost, single-camera, post-match anal- ysis pipeline built entirely with open-source Python tool- ing (OpenCV, Pandas, Matplotlib). RTFAS ingests recorded match video and automatically extracts player trajectories, team assignments, ball events, and tactical structure through:**
(i) camera-to-pitch registration via homography, (ii) motion- and appearance-based multi-object tracking with Kalman filtering and global data association, (iii) possession and pass inference with graph construction, and (iv) report generation comprising player heatmaps, team formation maps, pass- ing networks, shot charts with a distance/angle-based goal- probability proxy, and possession timelines. We describe the complete architecture, algorithms, and engineering trade- offs that allow deployment on commodity hardware without GPUs. Qualitative case studies indicate that RTFAS pro- duces coherent tactical artefacts (heatmaps, pass networks, and shot maps) that align with human analyst expectations. We discuss limitations (occlusiosn, jersey-color overlap, fast ball motion) and outline extensions (multi-camera fusion, learned xG).

***Index Terms*—Computer Vision, Sports Analytics, Player Tracking, Passing Network, Heatmaps, Football (Soccer), OpenCV, Data Visualization**

I. INTRODUCTION

Elite football programs rely on post-match analytics to quan- tify performance and inform tactical planning. Yet many so- lutions require expensive subscriptions, proprietary sensors, or multi-camera infrastructure, placing them out of reach for academies, colleges, and semi-professional clubs. Further, analyst workflows often involve time-intensive manual an- notation.

We propose the Real Time Football Analytics System (RTFAS): a low-cost, single-camera, post-match analytics platform built with open-source tools. RTFAS transforms raw broadcast-style or tripod recordings into coach-ready vi- sual reports—heatmaps, passing networks, shot charts, and possession breakdowns—with minimal configuration.

The main contributions of this work are:

- An end-to-end pipeline for single-camera post-match analytics using only open-source Python libraries.
- A robust camera-to-pitch registration procedure to project trajectories into metric field coordinates for for- mation and heatmap analysis.
- Lightweight, track-by-detection player tracking with Kalman filtering and global data association, plus jersey-color clustering for team assignment.
- Graph-theoretic passing-network construction and re- porting, including centrality and edge-weight mea- sures.
- A reproducible report generator producing heatmaps, pass networks, shot maps, and possession timelines suitable for coaching staff.

II. RELATED WORK

Research on sports video analytics spans object detec- tion/tracking, event recognition, and tactical inference. Practical systems frequently leverage open-source computer vision and visualization stacks to reduce cost and increase portability.

Multi-object tracking traditionally couples a motion model (e.g., Kalman filter) with global data association (e.g., Hungarian assignment), while static-camera scenes bene- fit from background modeling and motion cues. Team- interaction analysis often models play as a graph—players as nodes, passes as weighted edges—enabling centrality- based insights. RTFAS consolidates these ideas into a cohe- sive, deployable pipeline tailored to single-camera record- ings and amenable to low-resource environments.

III. SYSTEM DESIGN / METHODOL- OGY

3.1. Problem Formulation

Let a match video be a sequence of frames $\{I_t\}_{t=1}^{T}$ from a single fixed (or slowly panning) camera. The system out- puts:

- player trajectories in pitch coordinates (meters),
- team labels per track,
- a possession timeline,
- a directed weighted passing graph with edge weights as pass counts,
- player and team heatmaps, and
- shot maps with a probability-of-goal proxy.

3.2. Architecture Overview

RTFAS comprises five stages:

1. Pre-processing and field registration;
2. Player detection and multi-target tracking;
3. Team assignment and identity maintenance;
4. Ball-event and pass inference;
5. Analytics and report generation (heatmaps, pass net- works, shots, possession).

3.3. Core Algorithms / Models

3.3.1 Field Registration

We estimate a homography H between image coordinates and a canonical pitch template (e.g., 105 × 68 m). White line segments are extracted via edge detection and Hough transforms and matched to the template; RANSAC refines H. Registered tracks enable metric heatmaps, formation ex- traction, and shot/location features.

3.3.2. Player Detection and Tracking

Foreground masks from background modeling isolate mov- ing blobs; an optional lightweight person detector stabilizes detections. Tracks are maintained with a constant-velocity Kalman filter and Hungarian assignment on an IoU+position cost.

3.3.3. Team Assignment

Jersey colors are clustered in HSV color space using $k = 2$ k-means over per-track appearance histograms. Temporal smoothing improves stability; clusters are re-estimated to handle lighting drift.

3.3.4. Ball Cueing and Events

The ball candidate is detected as a small, fast-moving blob near active players and along likely pass directions. Pos- session is assigned to the nearest player's team with dis- tance and velocity gating. Passes and shots are inferred with spatial-temporal rules and velocity thresholds.

3.3.5. Passing Network and Metrics

Each successful pass increments A_{ij} for passer i to receiver j. Node and edge metrics include weighted degree, be- tweenness centrality, and edge densities.

3.3.6. Heatmaps, Formations, and Reports

Player coordinates in pitch space accumulate into 2D his- tograms rendered as density maps; per-team formation maps use time-averaged positions with convex-hull overlays. Re- ports are assembled via Matplotlib and NetworkX, exported as PDF/PNG.

3.4. Complexity and Design Choices

Let N_t be detections per frame and K active tracks. De- tection and tracking scale near-linearly with N_t; Hungarian assignment is $O(K^3)$ but K is typically < 50. Homogra- phy estimation is amortized across frames with keyframe re- fresh. Parameters (e.g., velocity gates, pass windows) are selected to favor precision in event detection, trading recall for reviewability in coaching contexts.

IV. EXPERIMENTAL SETUP

4.1. Datasets

RTFAS targets single-camera, broadcast-style or tripod recordings captured from a side-line or midline vantage point. The pipeline assumes a largely static camera with oc- casional pans/zooms and benefits from visible pitch mark- ings. Videos at 720p–1080p and 25–30 fps are supported. When available, manual annotations (player centers, pass events, shots) enable quantitative evaluation; otherwise, out- puts are validated qualitatively by a coach or analyst.

4.2. Baselines and Metrics

Module-level evaluation can use detection precision/recall (players), multi-object tracking metrics (IDF1, MOTA), and event accuracy for passes and shots. Analytics can be cross- checked against manual tags. Team reports are compared against human annotations for formation and zone occu- pancy.

4.3. Implementation Details

The implementation uses Python 3.x, OpenCV 4.x, Pan- das 2.x, Matplotlib 3.x, and NetworkX 3.x. The system runs on commodity laptops without a discrete GPU; optional OpenCV DNN inference is CPU-backed. Configuration is exposed via YAML (thresholds, gating, color clustering) and a CLI for batch processing.

V. RESULTS

5.1. Qualitative Outputs

Player heatmaps highlight positional tendencies and over- loads; passing networks reveal hubs (high weighted degree) and underused channels; shot charts cluster around half- spaces and inside the box; possession timelines correlate with pressing phases.

5.2. Case Study

On a recorded college match (single camera), RTFAS pro- duced trajectory-aligned heatmaps and a coherent 4-3-3 to 4-5-1 defensive transition. The passing graph identified midfield pivots as playmaking hubs, consistent with analyst notes.

5.3. Ablations and Sensitivity

Removing homography (rendering in image pixels) de- grades interpretability of formations and heatmaps. Re- placing background modeling with detector-only tracking improves re-identification but increases compute. Jersey- color clustering is robust in daylight; under night lighting, histogram-based matching with adaptive thresholds stabi- lizes labels.

VI. DISCUSSION

6.1. Practical Deployment

The single-camera assumption enables low-cost adoption but imposes occlusion challenges and depth ambiguity. La- tency is dominated by detection and tracking; batch process- ing post-match is well suited to CPU-only systems.

6.2. Limitations

Player re-identification across long occlusions, ball detec- tion during aerial passes, and kit-color collisions remain challenging. Event inference is heuristic and may miss sub- tle touches.

6.3. Future Extensions

Integrating learned re-identification, multi-camera fusion, and a data-driven expected-goals (xG) model trained on pub- lic event data can enhance accuracy. Semi-automatic anno- tation tools would support coach-in-the-loop correction.

VII. CONCLUSION

We presented RTFAS, a low-cost post-match analytics plat- form for football that converts single-camera recordings into tactical insights and coach-ready visuals. By combining field registration, lightweight tracking, jersey-color cluster- ing, and graph analytics, RTFAS delivers actionable arte- facts (heatmaps, pass networks, shot maps, possession time- lines) using only open-source components. Future work will integrate learned xG, stronger re-identification, and semi- automatic correction tools for analysts.

VIII. COMPLIANCE AND ETHICS

All analyses were performed on recorded matches with con- sent from relevant stakeholders; outputs were anonymized where appropriate. No personally sensitive data are released.

IX. ACKNOWLEDGMENT

The authors thank the coaching staff and peers who provided match recordings and feedback during development.

REFERENCES

[1] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Jour- nal of Software Tools, 2000.

[2] W. McKinney, “Data Structures for Statistical Comput- ing in Python,” in Proceedings of the

Python in Science Conference (SciPy), 2010, pp. 51–56.
[3] J. D. Hunter, “Matplotlib: A 2D Graphics Environ- ment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
[4] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” ASME Journal of Basic En- gineering, vol. 82, no. 1, pp. 35–45, 1960.
[5] H. W. Kuhn, “The Hungarian Method for the Assign- ment Problem,” Naval Research Logistics Quarterly, vol. 2, no. 1–2, pp. 83–97, 1955.
[6] Z. Zivkovic, “Improved Adaptive Gaussian Mixture Model for Background Subtraction,” in Proceedings of the 17th International Conference on Pattern Recogni- tion (ICPR), 2004, pp. 28–31.
[7] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Explor- ing Network Structure, Dynamics, and Function using NetworkX,” in Proceedings of the Python in Science Conference (SciPy), 2008, pp. 11–15.
[8] Kaggle, “DFL – Bundesliga Data Shootout,” competi- tion page, accessed Nov. 1, 2025. [Online]. Available: https://www.kaggle.com/competitions/ dfl-bundesliga-data-shootout
[9] Y. Isaienkov (ihelon), “Bundesliga Data Shootout–EDA,” Kaggle Notebook, ac- cessed Nov. 1, 2025. [Online]. Available: https://www.kaggle.com/code/ihelon/ bundesliga-data-shootout-eda