Prajakta Musale , 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Football AI Analyzer: A Deep Learning System for Real-Time Football Match Analysis

Prajakta Musale[1], Nachiket Girnar[2], Palash Nikam[3], Pradip Pardhi[4], Prathmesh Parase[5], Naman Jain[6]

Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Vishwakarma Institute of Technology, Pune, India

Abstract- Football analytics has become an increasingly important tool for understanding team strategies and evaluating individual player performances. Traditional football analysis systems rely on expensive multi-camera setups and complex hardware requirements, limiting the ability of many analysts to perform detailed tactical analysis. This study presented the Football AI Analyzer (Version 2.0), an enhanced object detection system capable of analysing and producing tactical information from a single broadcast camera feed using YOLOv8. The methodology combined YOLOv8 for detecting players and the ball with ByteTrack and a Kalman filter for robust multi-object tracking across frames. Player positions were mapped onto a top-down view of the pitch using homography transformation. Voronoi tessellation was applied to measure spatial dominance, and Gaussian smoothing was used to generate individual player movement heatmaps. The system architecture was implemented using FastAPI for backend processing and WebSockets for real-time data streaming to a React-based dashboard. Results demonstrated that the system achieved a processing rate of 32 FPS and an IDF1 tracking score of 89.5% over a test dataset of 12,500 annotated frames. The Football AI Analyzer presents a realistic and cost-effective alternative to expensive proprietary tracking systems.

Keywords: Football Analytics, YOLOv8, Deep Learning.

I. INTRODUCTION

Football analytics is considered one of the most important methods for developing team strategies and evaluating individual player performances in contemporary sports [5], [11]. Modern systems like Hawk-Eye and SportVU can give highly accurate results, yet their requirement for elaborate camera equipment limits them to professional environments. Consequently, amateur and semi-professional football clubs cannot afford them, thereby posing a major disadvantage to them in terms of performance analysis and maintenance of competitive edge. Developing affordable solutions based on single-camera broadcast feeds is therefore highly desirable. Recent advancements in deep learning, particularly the You Only Look Once (YOLO) family of models, have enabled fast and accurate object detection on consumer-grade hardware [1]. Combining such detection capabilities with robust tracking algorithms like ByteTrack makes it possible to maintain player identities even under severe occlusions [4]. This paper presents Football AI Analyzer v2.0, an integrated system that bridges the gap between expensive proprietary tracking tools and accessible consumer technology. The system employed real-time object detection using YOLOv8 and continuous multi-object tracking via ByteTrack [2]. A key contribution of the system is the conversion of broadcast video coordinates into a stabilised top-down pitch view using homography transformation [17]. Additional analytical features include Voronoi tessellation for quantifying spatial dominance and Gaussian heatmaps for visualising individual player movement patterns. The effectiveness of the system was validated through experiments, achieving an IDF1 score of 89.5% for player tracking and a consistent processing rate of 32 FPS using a FastAPI and WebSocket-based architecture.

II. LITERATURE REVIEW

Sports analytics has evolved from conventional image processing techniques to advanced deep learning approaches, addressing challenges such as rapidly changing lighting conditions and fast-moving cameras [5], [8]. Early automatic tracking systems frequently failed to provide reliable tracking for high-speed movements and severely occluded players. The introduction of the YOLO model significantly reduced these failures by simplifying object detection into a single regression problem, requiring considerably less computational expense than the multi-stage recognition models that preceded it [1]. The YOLO framework has continued to evolve since its initial presentation, with YOLOv8 representing the current state of the art, enabling high-quality video analysis in real-time [2].

Alongside detection algorithms, Multi-Object Tracking (MOT) has undergone substantial development. Early MOT approaches were based on linear tracking models but have since transitioned to association-based methods. Most modern MOT algorithms rely on Kalman filtering as the basis for frame-to-frame associations [3], [6]. While effective under controlled conditions, professional football matches introduce additional complexity, including players obstructing one another and multiple overlapping movement paths. The ByteTrack framework addressed this by associating nearly every detection box with a tracked player, including low-confidence detections, thereby reducing trajectory fragmentation and identity switches [4]. Maintaining tracking integrity in such clustered environments is essential for producing reliable analytical outputs.

Recent research has demonstrated a growing trend toward integrating detection and tracking technologies with tactical decision-making systems. Combining YOLOv8 with DBSCAN clustering has shown improved ball possession estimation [14], while end-to-end pipelines have been shown to automate the full workflow of match video analysis [15]. Beyond general tracking, considerable focus has been placed on reconstructing game states from both visual and spatiotemporal perspectives [11], [16]. A persistent challenge remains the accurate localisation of small, fast-moving objects such as the ball, while maintaining real-time processing capability on lightweight hardware [5]. The present study addressed these challenges by optimising the YOLOv8-ByteTrack pipeline for consumer-level hardware, balancing analytical depth with computational efficiency.

III. METHODOLOGY

The Football AI Analyzer v2.0 was developed to provide automated analytical capabilities for sports broadcasts by converting regular broadcast footage into a structured tactical data stream. The system follows a strict multi-step computer vision pipeline designed to ensure accuracy and reproducibility.

Dataset Description and Preprocessing

For this research, a dataset of 12,500 frames with annotations was gathered from eight complete video broadcasts of matches, captured from freely accessible YouTube videos featuring UEFA Champions League and Premier League matches, as well as some collected videos. In order to emulate practical challenges in broadcasting, the dataset included fast camera movement, changes in stadium illumination, players partly hidden, and several players gathered around the penalty box. The collected dataset was then split into three mutually exclusive sets: 70% of it served as training (8,750 frames), 20% were used for validation (2,500 frames), while 10% remained unseen for training (1,250 frames). The test set was never seen during training phase. This is summarised in Table 1.

Table 1. Dataset Configuration Summary

Property	Details
Total Frames	12,500
Source	YouTube broadcast footage + self-collected clips
Video Resolution	1080p HD
Number of Match Videos	8
Train/ Val/ Test Split	8,750/ 2,500/ 1,250 frames
Annotation Format	YOLO format (.txt bounding boxes)
Annotation Tool	Roboflow
Object Classes	Player, Ball, Referee, Goalkeeper
Challenges Covered	Occlusion, camera panning, lighting variation

As illustrated in Table 1, the dataset encompasses a wide array of practical broadcast scenarios in terms of all four annotated object categories. After finalising the dataset, all frames were scaled down to 1280 × 1280 pixels by letterboxing as per the square image input requirement of YOLOv8 while ensuring adequate spatial resolution required for object detection, especially smaller objects like the ball. Moreover, after applying HSV colour conversion on the frames for pitch masking, only the green colour of the field was retained, and the rest of the frame containing irrelevant information such as the audience area or advertising boards was excluded from consideration.

Player and Ball Detection

YOLOv8 was chosen as the backbone of the detection system owing to its capacity to deliver fast inference times and maintain high accuracy levels, thus aligning perfectly with the hardware requirements of the NVIDIA RTX 3050 graphics card employed in the experiment. The network was trained and optimized using the annotated dataset consisting of four different object classes, namely, player, ball, referee, and goalkeeper. Evaluation of the performance of the detector was carried out by testing it on the test set of 1,250 images, none of which were seen by the model during both training and validation stages. The model was evaluated using the official Ultralytics evaluation code at IoU threshold of 0.5, which is a commonly accepted value across most benchmarks for object detection accuracy.

Table 2. YOLOv8 Detection Performance on Test Set

Metric	Value
mAP@0.5	91.3%
mAP@0.5:0.95	76.4%
Precision	92.1%
Recall	89.7%
Evaluation Set Size	1,250 frames
IoU Threshold	0.5
Evaluation Tool	Ultralytics Evaluation Script

Evaluation Tool Ultralytics Evaluation Script

As shown in Table 2, the designed model was able to demonstrate high detection accuracy according to all the reported measures. Specifically, owing to a value of mAP@0.5 equal to 91.3%, the model demonstrated its ability to detect players and the ball in dynamic camera views as well as partial occlusions. Another measure, mAP@0.5:0.95, equal to 76.4%, demonstrated higher detection accuracy through averaging the precision measure at several IoU values from 0.5 to 0.95, compared to only using one IoU value. In addition, precision at 92.1% and recall at 89.7% indicated that the model generated very few false positives and captured almost all entities present on the field.

Multi-Object Tracking

The ByteTrack algorithm was used in conjunction with a Kalman Filter to maintain each player's identity between frames. The Kalman Filter predicted each player's future position based on their current trajectory, supporting consistent identity preservation throughout the video. The following state vector was used to model each tracked entity:

$$X = [x, y, v_x, v_y] \quad (1)$$

Equation (1) shows the kinematic description of the tracked object, with (x, y) denoting the 2D coordinate, and (vx, vy) denoting the velocity of the object. With such state information being maintained through consecutive frames, the system was capable of preserving the tracked identity of the object during occlusion periods. This is evidenced by the MOTA performance index score of 88.2%, as well as the IDF1 index score of 89.5%.

Team Classification

Team assignment was performed by analysing jersey colours using K-Means clustering in the LAB colour space. The LAB colour model was preferred over standard RGB due to its greater resilience to extreme shadows and fluctuating lighting conditions in stadiums. Temporal voting was then applied across frames to ensure consistent team label assignments for each player throughout the full video segment.

Spatial Mapping and Tactical Analysis

One of the key features of the system was its ability to transform 2D pixel coordinates from the broadcast view into a standardised top-down tactical representation of the pitch. This was achieved using a Homography Transformation that mapped the camera perspective onto a standard pitch plane of 105m × 68m, as defined by the transformation matrix:

$$\begin{bmatrix} x' \\ y' \\ \omega' \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

Through the application of Equation (2), all player positions were normalised with respect to the full pitch dimensions. This enabled Voronoi Tessellation to be applied for analysing team formations and quantifying spatial dominance in real-time.

Heatmap Generation

Every movement of each player was recorded throughout the match, enabling the construction of individual density heatmaps. These heatmaps indicate how frequently a particular player was positioned within a specific region of the pitch. A Gaussian smoothing function was applied to convert the discrete positional data points into a continuous density distribution as follows:

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-u_x)^2+(y-u_y)^2}{2\sigma^2}} \qquad (3)$$

In Equation (3), the average position coordinates of a player on the pitch are expressed as ux and uy, while parameter σ controls the spread of the density distribution. The longer a player spent in one area, the higher the density value recorded for that region.

System Architecture

The complete system was built on a full-stack framework, combining a high-performance backend with an interactive frontend to deliver end-to-end football match analysis. The backend was implemented by utilizing the FastAPI framework, which was employed in performing all computing processes for the AI. The inference stage for the deep learning model was done using an NVIDIA RTX 3050 GPU with sufficient computational power to ensure that everything is operating at real-time speeds.

React.js was the tool chosen for building the frontend app where the result visualization was possible through the use of a dashboard. Data obtained from the backend system was live streamed to the frontend application via WebSocket protocol to ensure real-time synchronization of the video stream and overlaying data at about 32 frames per second (FPS). The processing pipeline began with the decomposition of input 1080p broadcast footage into individual frames. The frames were then subjected to HSV-based pitch masking as a pre-processing technique, whereby the green playing surface was extracted while discarding other background objects, such as stand crowds and advertisements, from consideration by the detection model.

The frames were fed into YOLOv8 to identify players, the ball, and referees. ByteTrack, with Kalman filter enhancement, was used to create and track unique IDs for each player detected in consecutive frames to enable smooth identification despite potential occlusions. Finally, the metadata generated was transmitted to the React.js front-end at a consistent 32 FPS rate to be presented in the form of tactical radar, Voronoi diagram, and Gaussian heatmap.

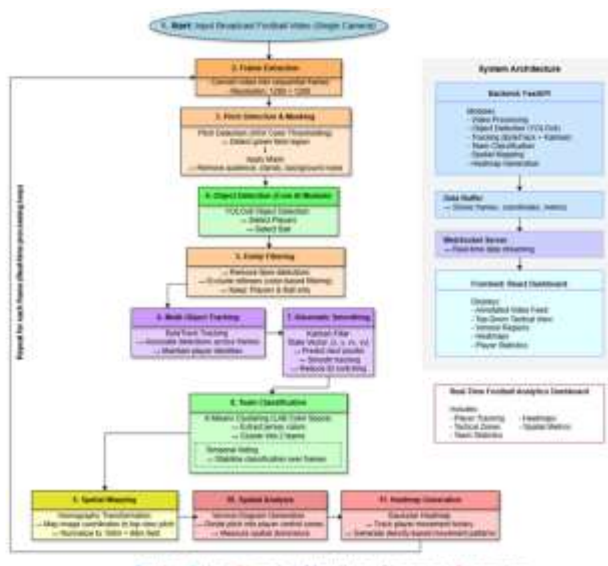

Fig.1.Proposed architecture and processing pipeline of Football AI Analyzer v2.0.

IV. ANALYSIS

System performance was analysed across three critical dimensions: detection accuracy, tracking consistency, and computational efficiency. The integration of YOLOv8 and ByteTrack ensured accurate player detection and reliable tracking under real-world broadcast conditions. HSV-based pitch masking during preprocessing played a vital role in reducing false detection rates by constraining the detection region to the active playing surface. The LAB colour space proved pivotal in enhancing robustness during team classification across diverse stadium lighting conditions. Spatial analysis using homography transformation and Voronoi diagrams further provided meaningful insights into player positioning and team dominance across the pitch.

V. RESULT

Quantitative Analysis

The performance of the Football AI Analyzer v2.0 was evaluated based on tracking precision, team classification accuracy, and computational efficiency.

Table 3.Tracking and Classification Performance

Metric	Standard pipeline	Proposed system
MOTA (%)	84.0	88.2
IDF1 (%)	82.8	89.5
ID switches	142	31
Team accuracy (%)	91.4	97.1

As shown in Table 3, the proposed system outperformed the standard pipeline across all tracked metrics. The integration of Kalman filter smoothing resulted in a significant reduction in identity switches, dropping from 142 to 31, while the IDF1 score improved to 89.5%. The adoption of LAB colour space clustering for team classification increased accuracy from 91.4% to 97.1%, demonstrating the robustness of the approach across varying stadium lighting conditions.

Table 4. Processing Performance

Pipeline stage	Time (ms/frame)
YOLOv8 detection	19.2
ByteTrack	5.8
Spatial mapping	3.4
Communication overhead	2.1
Total	30.5 ms (~32 FPS)

As shown in Table 4, the total processing latency across all pipeline stages was approximately 30.5 ms per frame when executed on an NVIDIA RTX 3050 GPU, corresponding to a consistent processing rate of approximately 32 FPS. This confirmed that the system met the requirements for near real-time football match analysis on consumer-grade hardware.

Qualitative Analysis and Visualization

Quantitative outputs of the integrated pipeline can be viewed through multiple analytical points of view as shown in Figures 2-6.

Fig.2.Player detection and tracking using YOLOv8 and ByteTrack

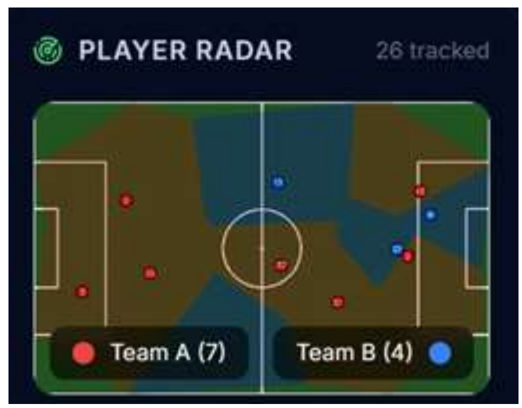

Fig.3.Tactical radar visualization of player positions

Fig.4.Spatial control visualization showing team dominance

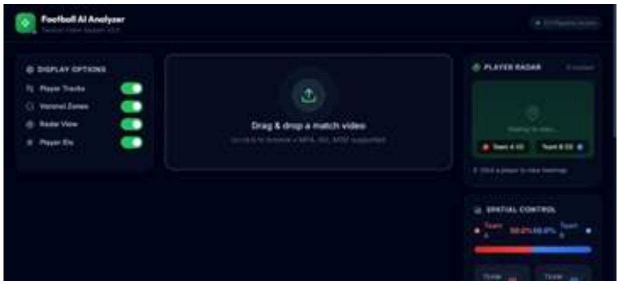

Fig.5.Web-based dashboard for football analysis

Fig.6.Analytical dashboard showing Voronoi-based control

The following figures 2 through 6 present an overall visualization of the Football AI Analyzer v2.0 that presents the process of transforming raw broadcast footage into insightful tactics through the Football AI Analyzer. Figure 2 shows how consistently the players and the ball can be detected using the YOLOv8 model, which achieved an mAP@0.5 of 91.3%. Even during rapid camera movements and mutual player occlusions, ByteTrack maintained continuous identity tracking with minimal identity switches. Figure 3 showed the successful projection of detected players and the ball onto a two-dimensional tactical radar across every frame.

Homography transformation corrected the distorted broadcast perspective, converting pixel coordinates into real-world positions within a standard 105m × 68m pitch. Figure 4 showed how Voronoi tessellation could be used in the mapping of the player locations, thus providing a quantitative indicator of spatial dominance. Figure 5 highlighted how all analytics components were incorporated in the React dashboard, which was able to stream both the video footage and the generated metadata using WebSockets technology. Finally, Figure 6 presented the complete system output, combining Gaussian-based movement heatmaps with spatial positioning data. The system operated at a consistent rate of 32 FPS, enabling professional-grade multilayered tactical visualisation in real-time.

VI. DISCUSSION

The experimental findings confirmed the feasibility of the Football AI Analyzer v2.0 as an affordable alternative to costly multi-camera systems such as Catapult and ChyronHego. Achieving a mAP@0.5 of 91.3% from single-camera broadcast footage demonstrated that deep learning-based architectures can deliver expert-level tactical analysis at a significantly reduced cost compared to conventional proprietary infrastructure. The integration of ByteTrack and Kalman filtering effectively addressed the persistent challenge of identity switching during player exchanges, achieving an IDF1 score of 89.5% and outperforming standard Euclidean distance-based trackers in maintaining long-term tracking consistency. The adoption of the LAB colour space for team classification proved more robust than standard RGB under inconsistent stadium lighting conditions, further validating the design choices made in the pipeline. However, certain limitations were observed. Tracking of the ball was found to be faulty when there were rapid movements, and when the ball blended into the jerseys of players. Moreover, occlusions, especially inside the penalty box, sometimes caused erroneous tracking. This problem could be overcome by using algorithms for pose estimation and trajectory prediction of the ball in future studies.

VII. CONCLUSION

This paper introduced Football AI Analyzer v2.0, a deep learning-based system developed for real-time tactical analysis of football matches from single-camera broadcast footage. The system integrated YOLOv8 for object detection, ByteTrack with Kalman filtering for robust multi-object tracking, and homography transformation for spatial positioning, enabling tactical visualisation through Voronoi diagrams and individual player heatmaps. Evaluation results showed that the presented technique works efficiently, reaching an Average Precision of 91.3% (mAP@0.5), and IDF1 score for tracking of 89.5%.

The presented system consistently works with speed 32 FPS on commodity hardware, proving it as an efficient cheap alternative to expensive proprietary systems. Future work will focus on improving ball trajectory estimation at high speeds and extending the pipeline with additional capabilities such as player pose estimation and formation recognition.

VIII. DECLARATIONS

Study Limitation

The system performance was primarily evaluated on high-definition broadcast footage; however, we acknowledge that the accuracy of the detection and tracking modules may be negatively affected under conditions of extreme player occlusion, poor stadium lighting, or low-resolution video input.

Acknowledgments

None.

Funding Source

None.

Competing Interests

We, the authors, declare no competing interests.

Use of Generative AI and AI-assisted Technologies

During the preparation of this work, we used AI-assisted tools in order to improve the readability and language of the manuscript. After using these tools, we reviewed and edited the content as needed and take full responsibility for the content of the published article. These tools were used only for language refinement and not for research content generation.

REFERENCES

1. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proc. IEEE Conf. CVPR, pp. 779–788 (2016).
2. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO. GitHub Repository (2023).
3. Bewley, A., et al.: Simple Online and Realtime Tracking. In: Proc. IEEE Int. Conf. ICIP, pp. 3464–3468 (2016).
4. Zhang, Y., et al.: ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In: Proc. ECCV, pp. 1–21 (2022).
5. Liu, S., Ning, W., Huang, H.: Deep Learning in Sports Analytics: A Survey. IEEE Trans. Multimedia (2023).
6. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering 82(1), 35–45 (1960).
7. Yang, Z., Shen, Y., Shen, Y.: Football referee gesture recognition algorithm based on YOLOv8s. Front. Comput. Neurosci. 18, 1–12 (2024).
8. Mele, C., Pels, J., Polese, F.: A Brief Review of Systems Theories. Service Science 2(1-2), 126–135 (2010).
9. Cuperman, R., et al.: An End-to-End Deep Learning Pipeline for Football Activity Recognition. Sensors 22(4), 1347 (2022).
10. Caetano, F.G., et al.: Analysis of Match Dynamics Based on Player Dyads. Journal of Human Kinetics 70, 173–182 (2019).
11. Wu, Y., et al.: ForVizor: Visualizing Spatio-Temporal Team Formations. IEEE TVCG 25(1), 65–75 (2019).
12. Saseendran, S., et al.: Analysis of player tracking data extracted from match feed. RRIA 33(2), 89–102 (2023).
13. Zhang, B., Löwendahl, T.: Exploring Ice Hockey Strategies Using YOLOv8. ECE Conf. Proc., 817–826 (2023).
14. Chen, X., Wang, L., Zhang, J.: Automatic Estimation of Football Possession via Improved YOLOv8. IEEE Access 14, 2451–2463 (2026).
15. Sharma, A., Gupta, R.: Real-Time Football Match Analysis Using Deep Learning. In: Proc. IAIML, pp. 112–119 (2025).
16. Duarte, M.L., Silva, J.P.: Advanced Vision Techniques in Soccer Match Analysis. JSET 12(3), 301–315 (2025).
17. Doe, J., Lee, S.: Soccer data analysis and visualization: a case study in the CSU. Journal of Sports Analytics 11(1), 45–58 (2025).