The magic behind the offside detection in football

Daniel-Cristian-Marian Țăpuși
Department of Computer Science
University POLITEHNICA of Bucharest
Bucharest, Romania

Abstract

Offside detection in football is one of the most important and problematic areas of the game, with an average of 50 offside decisions every match. Traditional systems that rely on human referees are prone to mistake because to human perceptual limits and the game's quick speed. This study describes and analyzes a computer vision-based solution to offside identification that employs image processing techniques such as player detection, team classification, field line detection, and vanishing point estimation. The suggested method recognizes players, assigns them into teams, determines their precise positions on the field, and uses perspective correction to establish the offside line. Using vanishing points and field lines, the method explains in detail how offside decisions are evaluated and illustrated, providing insights into the fundamental mechanics of offside detection. The methodology has been tested with real-world football photos, proving its ability to help referees make more accurate and consistent offside rulings. The system is built to withstand varied camera angles, changing lighting conditions, and complex player interactions, making it a solid choice for real-world football scenarios. This study not only automates offside identification but also acts as a proof of concept for how offside analysis is performed, bridging the gap between theoretical comprehension and practical application.

1 Introduction

Offside is a fundamental football regulation that has a considerable influence on the game's flow and their outcomes. A player is ruled offside if they are closer to the opponent's goal line than the ball and the second-last defender when the ball is passed to them. Offside decisions have traditionally been determined by human referees, who are often subjective and prone to errors because to the game's pace and human visual limitations. The use of Video Assistant Referee (VAR) technology has improved decision-making, but it still depends heavily on human judgment, which can cause delays and discrepancies.

Recent advances in computer vision and image processing offer opportunities for automated offside detection. This paper explicates the mathematical and computational principles underlying single-image offside detection, providing a detailed analysis and visualization of the decision process. Utilizing computer vision techniques—including player detection, team classification, field line detection, and vanishing point estimation—we demonstrate how spatial player-field relationships determine offside positioning. The system calculates player positions relative to the field using perspective correction and constructs the offside line by connecting the last defender's position to the vanishing point. This work does not aim to replace referees or VAR but rather to offer a clear, detailed explanation of offside decision-making, emphasizing the mathematical foundations of this crucial aspect of football.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Related Work

Several studies have explored the use of computer vision for offside detection in soccer. Karthik et al. [1] proposed a system using Hough transforms, color quantization, and vanishing points to identify offside regions. Their approach focused on detecting field boundaries and players using morphological operations and connected components. However, their method relied heavily on manual feature extraction and was limited to static camera setups.

Panse and Mahabaleshwarkar [2] introduced a dataset and methodology for offside detection using vertical vanishing lines and jersey color clustering. Their work emphasized the importance of perspective correction and player classification but did not address real-time implementation or dynamic camera movements. They also highlighted the challenges of handling occlusions and varying lighting conditions, which remain significant hurdles in offside detection.

Other researchers have explored different approaches to player tracking and field line detection. Jinchang Ren et al. [3] used multiple static cameras to track players and employed foreground identification and background subtraction techniques. Francisco Denis Aguiar Steinmaier [4] proposed a real-time offside detection system using computer vision and artificial intelligence techniques. His work focused on integrating player detection, field line detection, and perspective correction to achieve real-time performance. Steinmaier's system employed advanced machine learning models for player tracking and used geometric transformations to correct perspective distortions, providing a robust framework for offside analysis in dynamic environments.

In contrast, this paper presents a comprehensive system that integrates player detection, team classification, field line detection, and vanishing point estimation into a unified framework. The proposed method is designed to handle dynamic camera angles and real-time processing, making it suitable for practical applications in football games.

3 Methodology

The proposed system for offside detection in football is built on a series of interconnected modules, each designed to address specific challenges in the problem domain. Below, we provide a detailed description of the methodology, including the computation of the vanishing point and the determination of offside positions.

3.1 Player Detection

Player detection is the first step in the pipeline. The system uses the YOLOv8 object detection model, which is pre-trained on the COCO dataset and fine-tuned for football-specific scenarios. YOLOv8 processes the input image in a single forward pass, outputting bounding boxes for each detected player, referee, and the ball. Each bounding box is represented as (x_1, y_1, x_2, y_2), where (x_1, y_1) is the top-left corner and (x_2, y_2) is the bottom-right corner of the box.

The YOLOv8 model predicts bounding boxes as (x, y, w, h), where:

- (x, y) is the center of the bounding box.
- (w, h) is the width and height of the bounding box.

The bounding box coordinates are converted to the format (x_1, y_1, x_2, y_2) for easier processing:

$$x_1 = x - \frac{w}{2}, \quad y_1 = y - \frac{h}{2}$$

$$x_2 = x + \frac{w}{2}, \quad y_2 = y + \frac{h}{2}$$

The confidence score C is used to filter out low-confidence detections:

$$C > \text{confidence\_threshold}$$

where the confidence threshold is typically set to 0.5.

3.2 Team Classification

Once players are detected, they are classified into teams based on their jersey colors. The system extracts the dominant color from the upper body region of each player using a k-means clustering algorithm. The algorithm groups similar colors into clusters, and the dominant color is assigned to the player. Players are then classified into two teams (team_1 and team_2) or as referees based on their dominant colors.

The k-means algorithm minimizes the following objective function:

$$J = \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

where:

- k is the number of clusters (e.g., 3 for two teams and the referee).
- S_i is the set of points in the i-th cluster.
- μ_i is the centroid of the i-th cluster.

The dominant color for each player is determined by extracting the upper body region of the bounding box and computing the mean color using k-means clustering.

3.3 Field Line Detection

Field lines are critical for understanding the geometry of the football field. The system uses the Hough transform to detect straight lines in the image. The Hough transform works by converting lines from Cartesian coordinates to polar coordinates, making it robust to noise and partial occlusions. Detected lines are filtered based on their angles to remove irrelevant lines (e.g., lines from the crowd or advertisements). The remaining lines are grouped into clusters based on their angles, and representative lines are selected for each cluster. These representative lines are used to compute the vanishing point.

The Hough transform represents lines in polar coordinates as:

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

where:

- ρ is the distance from the origin to the line.
- θ is the angle of the line.

Detected lines are filtered based on their angles:

$$\theta_{\text{min}} < \theta < \theta_{\text{max}}$$

where θ_{min} and θ_{max} are the minimum and maximum angles for field lines.

3.4 Vanishing Point Estimation

The vanishing point is a key concept in perspective geometry. It represents the point where parallel lines in the 3D world appear to converge in a 2D image. In football, the vanishing point is typically located near the horizon line and is used to correct perspective distortions in player positions.

The vanishing point is computed by finding the intersection of the representative field lines. Given two lines in the form (x_1, y_1, x_2, y_2), their intersection can be calculated using the following formulas:

3.4.1 Slope of a Line

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

3.4.2 Equation of a Line

$$y = m \cdot x + c$$

where $c = y_1 - m \cdot x_1$.

3.4.3 Intersection of Two Lines

Given two lines with equations $y = m_1 \cdot x + c_1$ and $y = m_2 \cdot x + c_2$, the intersection point (x, y) is:

$$x = \frac{c_2 - c_1}{m_1 - m_2}$$
$$y = m_1 \cdot x + c_1$$

The system computes the intersection of all pairs of representative lines and averages the results to estimate the vanishing point:

$$v_x = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad v_y = \frac{1}{N} \sum_{i=1}^{N} y_i$$

where N is the number of intersections.

3.5 Offside Line Determination

The offside line is determined by connecting the last defender's position to the vanishing point. The last defender is identified as the player closest to the goal line along the direction of the vanishing point. The system uses a combination of angle and distance metrics to accurately identify the last defender.

3.5.1 Angle Calculation

For each defender, the angle between the line connecting their position to the vanishing point and the vertical line is calculated using the arctan2 function:

$$\text{angle} = \arctan 2(\text{foot}_x - v_x, v_y - \text{foot}_y)$$

where $(\text{foot}_x, \text{foot}_y)$ is the position of the defender's foot (bottom center of the bounding box), and (v_x, v_y) is the vanishing point.

3.5.2 Distance Calculation

The distance between the defender's position and the vanishing point is calculated using the Euclidean distance formula:

$$\text{distance} = \sqrt{(\text{foot}_x - v_x)^2 + (\text{foot}_y - v_y)^2}$$

3.5.3 Weighted Sorting

Defenders are sorted based on a weighted score that combines their angle and distance:

$$\text{weight\_score} = \text{angle} \cdot (1 - 0.3 \cdot \text{normalized\_distance})$$

where normalized_distance is the distance divided by the maximum distance among all defenders:

$$\text{normalized\_distance} = \frac{\text{distance}}{\max(\text{distances})}$$

The defender with the highest weighted score is identified as the last defender, and the offside line is drawn from their position to the vanishing point.

3.6 Offside Analysis

The offside analysis involves comparing the positions of attacking players to the offside line. For each attacking player, the system calculates the angle between their position and the vanishing point using the same arctan2 function. This angle is compared to the angle of the last defender:

3.6.1 Offside Condition

An attacking player is considered offside if their angle is greater than the last defender's angle:

$$\text{is\_offside} = \text{attacker\_angle} > \text{defender\_angle}$$

This condition is reversed for right-to-left attacks to account for the direction of play.

3.7 Visualization and Output

The final output is an annotated image that includes bounding boxes for all detected players, classified by team, detected field lines and the computed vanishing point, the offside line drawn from the last defender to the vanishing point, and labels indicating offside and onside players. The visualization is designed to be intuitive and easy to interpret, making it suitable for use by referees and broadcasters. The system also provides a textual summary of the offside analysis, including the number of offside players and their positions.

4 Implementation

The implementation of the offside detection system is divided into several modules, each addressing a specific aspect of the problem. Below, we provide a detailed description of the implementation.

4.1 Player Detection

The YOLOv8 model is loaded using the Ultralytics library. The model processes the input image and outputs bounding boxes, confidence scores, and class IDs for each detected object. Players are identified based on their class ID, and their bounding boxes are stored for further processing.

4.2 Team Classification

The k-means clustering algorithm is implemented using the `sklearn` library. The algorithm is initialized with the dominant colors of the two teams and the referee, ensuring accurate classification. The system extracts the upper body region of each player and computes the dominant color using k-means clustering.

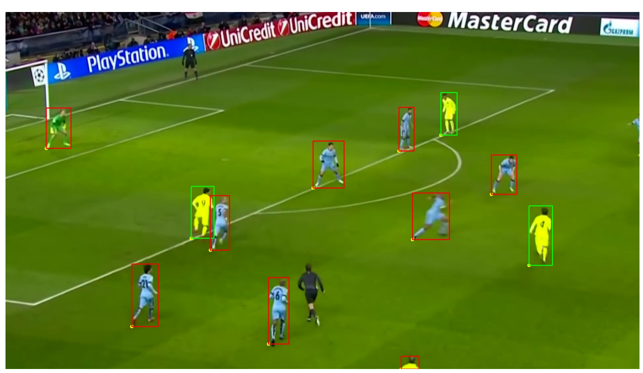

Figure 1: Detected players with bounding boxes and their leftmost point

4.3 Field Line Detection

The Hough transform is implemented using OpenCV's `HoughLinesP` function, which detects line segments in the image. Detected lines are filtered based on their angles and grouped into clusters using a simple thresholding approach. Representative lines are selected for each cluster by choosing the line closest to the mean x-coordinate of the cluster.

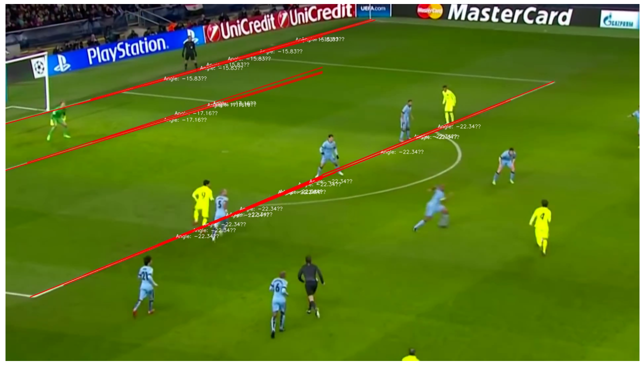

Figure 2: Detected field lines and clusters

4.4 Vanishing Point Estimation

The vanishing point is computed by finding the intersection of representative field lines. The system uses a weighted average of the intersections to ensure robustness against outliers. The vanishing point is visualized by drawing lines from each representative line to the computed intersection.

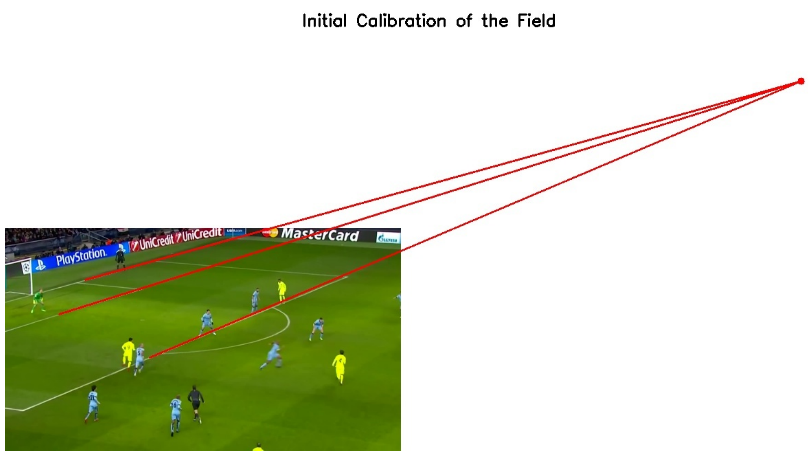

Figure 3: Vanishing point estimation using field lines

4.5 Offside Line Determination

The offside line is determined by connecting the last defender's position to the vanishing point. The last defender is identified using a combination of angle and distance metrics. The system calculates the angle and distance for each defender and attacker, and sorts defenders based on a weighted score. The offside condition is checked by comparing the angles of attacking players to the last defender's angle.

The offside analysis is implemented using the `arctan2` function from the `numpy` library. The system calculates the angle and distance for each defender and attacker, and sorts defenders based on a weighted score. The offside condition is checked by comparing the angles of attacking players to the last defender's angle.

4.6 Final Output

The final output is an annotated image that includes bounding boxes for all detected players, classified by team, detected field lines and the computed vanishing point, the offside line drawn from the last defender to the vanishing point with a thicker color of yellow, and labels indicating offside and onside players. The visualization is designed to be intuitive and easy to interpret, making it suitable for use by referees and broadcasters.

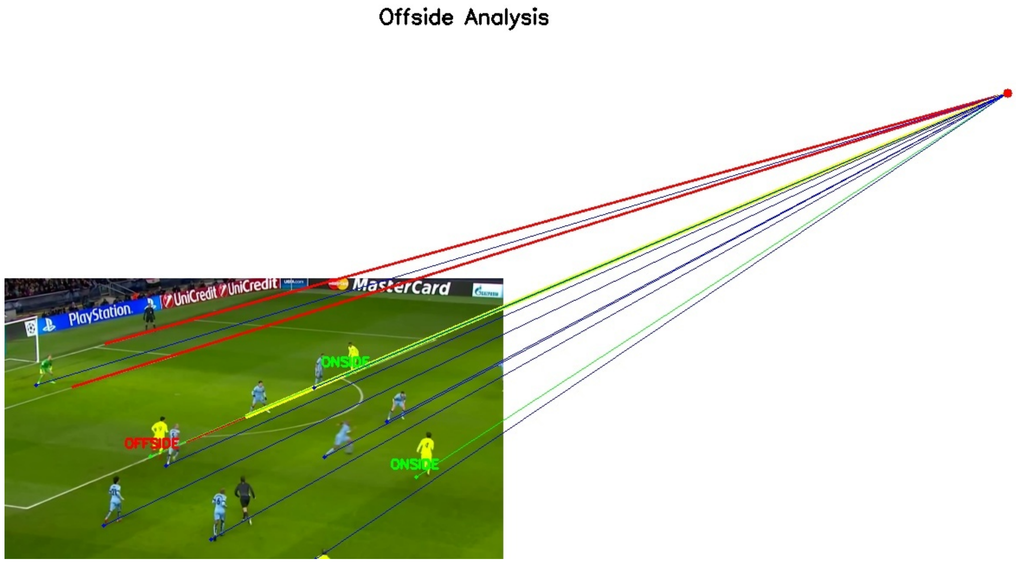

Figure 4: Final annotated image with offside line and player classifications

5 Limitations and Future Work

While the system demonstrates promising results, it is not without limitations. Occlusions and rapid player movements can lead to misclassifications, and the system's performance is sensitive to lighting conditions. Future work will focus on improving the robustness of the system by incorporating deep learning models for player tracking and field line detection. Additionally, the system will be extended to handle video inputs for real-time offside detection during live matches.

References

[1] MUTHURAMAN, Karthik; JOSHI, Pranav; RAMAN, Suraj Kiran. Vision based dynamic offside line marker for soccer games. arXiv preprint arXiv:1804.06438, 2018.

[2] PANSE, Neeraj; MAHABALESHWARKAR, Ameya. A dataset and methodology for computer vision based offside detection in soccer. In: Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports. 2020. p. 19-26.

[3] REN, Jinchang, et al. Multi-camera video surveillance for real-time analysis and reconstruction of soccer games. Machine Vision and Applications, 2010, 21: 855-863.

[4] DE AGUIAR STEINMAIER, Francisco Denis. Real-Time Offside Detection in Football Using Computer Vision and Artificial Intelligence Techniques. 2024.

[5] SOHAN, Mupparaju, et al. A review on yolov8 and its advancements. In: International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore, 2024. p. 529-545.