

A COGNITIVELY INSPIRED FRAMEWORK FOR AUTOMATED FOOTBALL COMMENTARY

Gloria Virginia, Jeffrey Susilo and Aditya Wikan Mahastama

Informatics Department, Universitas Kristen Duta Wacana,
Yogyakarta, Indonesia

ABSTRACT

This study presents an automated football commentary system that integrates computer vision, rule-based reasoning, and natural language generation within a cognitively inspired framework. The modular design employs a YOLOv8 model for real-time object detection, K-Means clustering for team classification, and a rule-based reasoning module for event recognition based on spatial and temporal conditions such as ball possession and dead-ball states. Detected events are transformed into expressive, sportcaster-style commentary using a generative language model. Experiments on real football match videos and evaluations by 30 football enthusiasts using a five-point Likert scale demonstrated high detection precision, reliable team identification and accurate recognition of key events. Beyond technical performance, the framework promotes inclusivity through potential extensions such as multilingual commentary, closed captioning, and audio description, supporting equitable access for diverse audience.

KEYWORDS

Cognitive AI, Computer Vision, Rule-Based Reasoning, Natural Language Generation, Automated Football Commentary

1. INTRODUCTION

In sporting events, human commentators play a significant role in enhancing the audience experiences as well as the emotional engagement. However, commentators encounter challenges, including bias arising from subjective interpretation and inconsistencies caused by fatigue due to long matches. From a cognitive science perspective, another issue arises in the restricted capacity of working memory that may prevent the recollection of prior events or tactical situations, while the fast-paced nature of the game often requires simultaneous attention on various activities on the field. In order to overcome these challenges, integrating advanced AI systems that utilize computer vision and natural language processing should substantially improve and strengthen viewer engagement.

Studies on automated sports commentary [1] [2] [3] [4] demonstrate potential results due to its human-centric nature, yet it is highly challenging due to the inherent complexity of the task itself. Rather than reporting facts, commentary also requires interpretation and the ability to convey emotional nuances and excitement [1].

For an automated commentator, object detection represents a fundamental starting point. Casting object detection as a language model problem offers a novel framework for integrating visual and contextual data in commentary generation. For example, the Pix2Seq framework conceptualizes object detection as a sequence generation task, enabling a unified and flexible approach to object

David C. Wyld et al. (Eds): IBCOM, GridCom, SPPR, NLAI, ICCSEA, NECO – 2025

pp. 53-73, 2025. CS & IT - CSCP 2025

DOI: 10.5121/csit.2025.152305

identification within images [5]. This method aligns with emerging trends in Multimodal Large Language Models (MLLMs) for vision-language tasks, such as Zang's [6] work on predicting object names and locations to enhance human-AI interaction.

Gao et al. [7] emphasized that object detection tasks should not rely solely on visual information. Instead, knowledge-driven approaches should assist in acquiring high-level semantic information between objects. This argument is supported by Qi et al. [8] who introduced GOAL, a benchmark for Knowledge-grounded Video Captioning (KGVC), and demonstrated promising directions that the integration of knowledge and visual data may significantly enhance the effectiveness of automated commentary in real-time sports narratives. Similarly, Andrews et al. (2024) utilized spatial-temporal data, comprising the location and movement of players and the ball over time in a football video, to enrich the automated system with deeper contextual insights.

Wang et al. [9] proposed TN2L2K, a large-scale dataset for object tracking using natural language specification. Their work demonstrates that deep learning techniques can effectively identify and monitor multiple objects in complex environments. More recent studies incorporating natural language descriptions into object tracking have shown promise in improving flexibility and robustness [10] [11]. For instance, models using textual prompts to guide object tracking exhibit enhanced performance in identifying and following targets across diverse scenarios [12].

The convergence of computer vision, natural language processing (NLP), and knowledge-based system (KBS) has led to the development of increasingly sophisticated models capable of not only detecting and tracking objects but also generating natural language descriptions that communicate meaningful insights about observed events [3] [13] [14]. This advancement holds substantial implications for automated commentary systems. However, key challenges persist, particularly in ensuring that generated description are coherent, contextually appropriate, and informative.

This study aims to address these challenges by proposing a framework that integrates object detection and tracking with knowledge-based reasoning and natural language generation. The objective is to develop a system capable of interpreting complex scenes and producing descriptive narratives that accurately reflect real-world events. By combining visual data with structured knowledge and linguistic modeling, the proposed approach seeks to enhance both the interpretability and communicative capabilities of AI-based commentary systems.

The proposed automated football commentary system also holds significant potential for promoting inclusivity. It is designed to be accessible to diverse audiences, including individuals with disabilities, non-native speakers, and people from varied cultural backgrounds. Further enhancements, such as closed captioning and audio descriptions, could contribute to a more equitable and accessible digital environment [15] [16].

2. METHODOLOGY

The research methodology comprises a series of systematic components, as illustrated in Figure 1. It is designed to develop an integrated system capable of performing object detection in football matches, classifying teams based on jersey colors, recognizing on-field situations, and generating natural language commentary automatically.

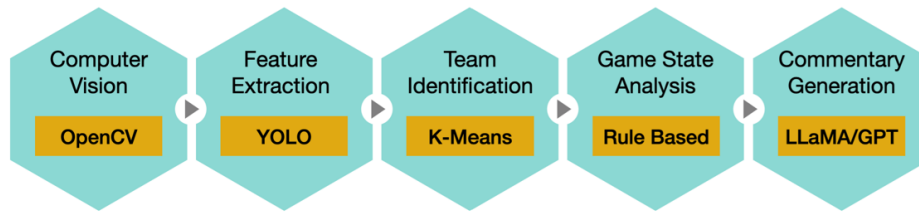


Figure 1. Research methodology.

The first component, Computer Vision, functions as the visual perception module of the system. Using OpenCV, the input video is processed and segmented into individual frames, which are subsequently analyzed by YOLO for object detection. This stage identifies various object classes - such as players, goalkeepers, referees, and the ball -- within each frame and extracts relevant features, including jersey colors and bounding-box coordinates.

Next, Team Identification is conducted through a color-based approach using the K-Means clustering algorithm. The dominant color extracted from each player's jersey is used to group players into two clusters representing the respective teams (Left and Right). The output from this stage, along with positional data from YOLOv8, serve as inputs to a Rule-Based Game State Analysis module, which infers match situations such as ball possession, goal attempts, and player interactions.

Finally, the identified situations are translated into structured prompts that feed into the Commentary Generation component. Leverageing a large language model (LLaMA or GPT) accessed via API, this module transforms symbolic representations into coherent, contextually appropriate sports commentary. Collectively, these interconnected components -- from video processing and object detection to reasoning and commentary generation -- form a unified system capable of real-time video processing, contextual understanding of match dynamics, and human-like automated commentary.

2.1. Requirement

YOLO (You Only Look Once) was introduced by Redmon et al. [17] as a unified model of object detection that treats it as a regression problem. The transformation of the traditional object detection approach makes it extremely fast, implicitly encodes contextual information, and learns generalizable representations of objects. The YOLO framework has significantly evolved, with various iterations improving its accuracy and efficiency in real-time applications [18]. YOLOv8 has introduced further enhancements, such as improved architectural design and advanced training techniques, making it one of the most effective models for real-time object detection in sports contexts [18] [19]. The core components of the YOLOv8 architecture includes the Backbone, Neck, and Head.

The backbone is responsible from difference scales to improve detection accuracy. It extracts features from the input image using several convolutional layers and a structure known as **C2f (Cross Stage Partial Networks)**. The neck is responsible for refining the features extracted by the backbone and merging information from multiple scales. It combines features from different scales to improve detection accuracy. The head predicts the final bounding boxes and object classes, producing outputs such as bounding box coordinates and confidence scores.

The dataset used in this research is **SoccerNet** [20], which consists of football match videos segmented into individual frames with annotations specifying the positions of players and the ball.

This dataset was selected for its strong relevance to the sports domain and its comprehensive coverage of diverse object categories that need to be detected. The annotated data are utilized to train the object detection model, enabling the system to automatically recognize players, the ball, and other on-field personnel. Furthermore, the dataset supports the evaluation of the model's robustness in handling detection challenges under varying environmental conditions and camera perspectives.

The **K-Means** clustering algorithm was implemented to categorize detected objects according to spatial proximity and visual similarity. Following the object detection stage, K-Means groups entities such as players wearing similar jerseys, thereby enabling automated team identification and facilitating situational analysis. This clustering process supports the examination of spatial relationships between players and the ball within dynamic match contexts.

The **Groq API** was implemented to accelerate deep learning inference by leveraging Groq's high-throughput hardware architecture. Designed for low-latency and parallelized computation, Groq provides substantial performance gains for AI and machine learning workloads. In this study, the **LLaMA API** from Groq was employed to execute inference on the trained YOLOv8 object detection model. This integration ensures efficient real-time processing, addressing the computational demands inherent in automated sports commentary systems.

OpenCV, an open-source computer vision framework, was employed in this study for essential image and video processing operations. Its functions include image pre-processing, video frame extraction for YOLO-based object detection, and visualization of detection outputs. Furthermore, OpenCV was applied to track dynamic objects across video frames, supporting the analysis of player and ball trajectories in football match recordings.

2.2. Model Training

Object detection training was performed using the YOLOv8 architecture provided by Ultralytics, chosen for its demonstrated accuracy and efficiency in real-time detection tasks. A customized subset of the SoccerNet dataset was reformatted to align with YOLO's annotation schema. Each frame was annotated for key objects such as players, the ball, referees, and goalposts. Training employed the YOLOv8 configuration, initialized with pre-trained weights and fine-tuned using an image size of 720 pixels, a batch size of 16, and an initial learning rate of 0.03. Conducted in the Google Colab environment with GPU acceleration, the process included automatic checkpointing of both the best and latest weights. This setup is designed to ensure model robustness under diverse conditions, including varying illumination, camera perspectives, and object density.

2.3. Team Identification

Following object detection using YOLOv8, each detected player is classified into their respective team to enable higher-level reasoning such as ball possession estimation and tactical interaction analysis. Team identification is based on extracting the dominant jersey color while minimizing interference from the grass background. To achieve this, each frame is converted to the HSV color space, and a green color mask is applied to isolate the grass region. The mean hue Hg of all green pixels is then computed (Equation 1) to dynamically adapt to varying illumination conditions.

$$Hg = (1 \div N) \sum H_i, \text{ for } i = 1 \text{ to } N \text{ green pixels} \quad (1)$$

The average hue Hg is used to generate a grass mask $mask_{grass}$ (Equation 2) within the range.

$$mask_{grass} = inRange(HSV_{frame}, [Hg - 10, 40, 40], [Hg + 10, 255, 255]) \quad (2)$$

For each detected player, only the upper half of the bounding box $Region_{upper}$ is used (Equation 3), as it typically contains the jersey.

$$Region_{upper} = P_i[0 : h \div 2, 0 : w] \quad (3)$$

A binary mask $mask_{jersey}$ is then generated (Equation 4) to remove green regions, allowing only non-grass pixels to contribute to colour estimation.

$$mask_{jersey} = NOT(mask_{grass}) \quad (4)$$

The dominant jersey colour C_i is computed as the mean colour value within the masked region, where \odot denotes pixel-wise AND operation.

$$C_i = mean(P_i \odot mask_{jersey}) \quad (5)$$

All jersey colour vectors $C = \{C_1, C_2, \dots, C_n\}$ are then clustered using the K-Means algorithm defined in Equation 6 into two groups, corresponding to the two teams.

$$KMeans(C, k = 2) \rightarrow label_i \in \{0, 1\} \quad (6)$$

To identify which label corresponds to the left-side team, the average x-coordinate of all players in each cluster is computed using Equation 7. The cluster with a lower average x-position is designated as the left team as defined in Equation 8.

$$\begin{aligned} \bar{x}^0 &= (1 \div N^0) \sum x_i \text{ for players in team 0} \\ \bar{x}^1 &= (1 \div N^1) \sum x_i \text{ for players in team 1} \end{aligned} \quad (7)$$

$$\begin{aligned} left_{team} &= 0 \text{ if } \bar{x}^0 < \bar{x}^1 \\ left_{team} &= 1 \text{ if } \bar{x}^0 \geq \bar{x}^1 \end{aligned} \quad (8)$$

2.4. Rule-Based for Situation Detection

This module uses domain knowledge to infer match situations through a series of logical and geometric rules involving player position, ball coordinates, and motion vectors. Below are the key rules formulated mathematically:

1. Kick-off detection

Kick-off is detected if players are mostly positioned on their respective sides. Kick-off detection in Equation 9 is used for detecting beginning stage of the game, where L_l is player from team L on the left half and R_r is player from team R on the right half.

$$|L_l| \geq 10 \wedge |R_r| \geq 10 \Rightarrow \text{Kick-off has started!} \quad (9)$$

2. Ball possession

A team is considered in possession if one of its players is closest to the ball. Equation 10 defines the ball possession, where p_i is player position, b is ball position, and θ is dynamic distance threshold.

$$\|p_i - b\| \leq \theta \wedge \|p_i - b\| = \min(\|p_j - b\|) \Rightarrow team(p_i) = closest_{team} \quad (10)$$

3. Attack direction and ball direction

For attack direction, the field is vertically divided into three zones: *Top attack*, *Centre attack* and *Bottom attack*. It depends on the y-position H of the ball b_y as defined in Equation 11.

$$\begin{aligned} closest_{team} \neq \emptyset \wedge b_y < H \div 3 &\Rightarrow Top\ attack \\ closest_{team} \neq \emptyset \wedge H \div 3 \leq b_y < 2H \div 3 &\Rightarrow Centre\ attack \\ closest_{team} \neq \emptyset \wedge b_y \geq 2H \div 3 &\Rightarrow Bottom\ attack \end{aligned} \quad (11)$$

There are 4 ball directions, which are *right*, *left*, *down*, and *up*. The ball direction in Equation 12 is determined using its displacement vector $\vec{d} = (dx, dy)$, where dx and dy are ball movement vector.

$$\begin{aligned} |dx| > |dy| \wedge dx > 0 &\Rightarrow right \\ |dx| > |dy| \wedge dx < 0 &\Rightarrow left \\ |dy| > |dx| \wedge dy > 0 &\Rightarrow down \\ |dy| > |dx| \wedge dy < 0 &\Rightarrow up \end{aligned} \quad (12)$$

4. Shot detection

A shot is assumed if a team has possession and the ball is nearing toward the opponent's goal. Let GK_L and GK_R be goalkeepers of Left and Right, then Equation 13 determines the shooter.

$$\begin{aligned} closest_{team} = L \wedge Direction(b) \rightarrow GK_R &\Rightarrow Shot\ by\ team\ L \\ closest_{team} = R \wedge Direction(b) \rightarrow GK_L &\Rightarrow Shot\ by\ team\ R \end{aligned} \quad (13)$$

5. Dead ball detection

Dead ball detected if it's coordinate does not change after some period, as defined in Equation 14.

$$\vec{d} = (0,0) \text{ for several periods} \Rightarrow Dead\ Ball \quad (14)$$

6. Free kick detection

A free kick is detected if the direct line between ball and goalkeeper is obstructed. Let o be opponent and $line(b, GK)$ be the line to goalkeeper, then free kick detection is defined as Equation 15.

$$Dead\ Ball \wedge \exists o \in line(b, GK) \Rightarrow Free\ kick\ for\ team\ X \quad (15)$$

7. Penalty detection

If the direct line is between ball and goalkeeper is not obstructed, then it is penalty. The situation should satisfy Equation 16.

$$Dead\ Ball \wedge \neg \exists o \in Opponents : o \in line(b, GK) \Rightarrow \text{Penalty for team X} \quad (16)$$

8. Goal kick detection

Goal kick situation is detected if there's no nearby player and *GK* posses the ball, as described in Equation 17.

$$Dead\ Ball \wedge \|b - GK\| < \varepsilon \wedge \neg \exists o \in Opponents : \|o - b\| < \varepsilon \Rightarrow \text{Goal kick for team X} \quad (17)$$

2.5. Commentary Generation

In this phase, natural language commentary is generated using the LLaMA 3.3-70B model via the Groq API. Detected game events — identified through rule-based reasoning, such as kickoffs, ball possessions, or shots — are transformed into concise textual prompts that provide situational context. The model elaborates on these prompts, rephrasing them in the expressive register of professional sports commentary. This process allows the system to deliver dynamic, contextually coherent, and real-time narrations aligned with the communicative style of human sportscasters.

3. RESULTS AND DISCUSSIONS

3.1. YOLO Training Result

The object detection model's performance was evaluated using mean Average Precision (mAP) and Average Recall (AR), which jointly assess detection accuracy across different IoU (intersection over union) thresholds and object sizes. The validation results of the trained YOLO model are summarized in Table 1.

Table 1. Validation Result.

Metric	Value	Metric	Value
AP @ IoU=0.50:0.95	0.649	AR @ 1 detection	0.556
AP @ IoU=0.50 (AP50)	0.874	AR @ 10 detections	0.718
AP @ IoU=0.75 (AP75)	0.742	AR @ 100 detections	0.726
AP for small objects	0.301	AR for small objects	0.446
AP for medium objects	0.687	AR for medium objects	0.771
AP for large objects	0.745	AR for large objects	0.852

As shown in Table 1, the model achieves a mAP of 0,649 across IoU thresholds ranging form 0.50 to 0.95, demonstrating a strong balance between precision and recall. The model performs particularly well on medium and large objects, with AP values of 0.687 and 0.745, respectively. In contrast, performance on small objects is relatively lower (AP = 0.301), indicating a potential area for further optimization.

In addition to evaluation on the official validation set, the trained model was further tested on several randomly selected football match images that were not included in the training or validation

datasets. The images were sampled using Python's *random* library, and the predictions were visualized with OpenCV. The corresponding results are presented in Figure 2.

The qualitative results indicate that the model consistently produced accurate detections of both players and the ball under diverse image conditions, including variations in lighting, camera angles, and field appearance. The predicted bounding boxes aligned well with visual expectations, demonstrating the model's robustness and strong generalization capability beyond the original dataset.

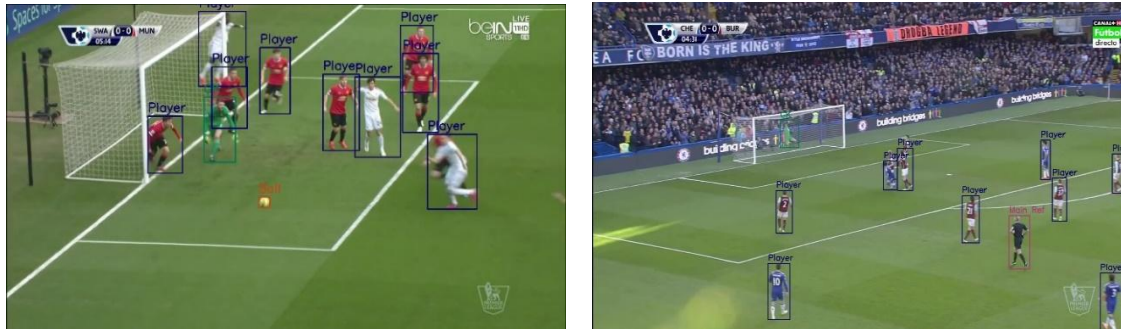


Figure 2. Visualization of model predictions.

Algorithm 1 outlines the process of object detection and feature extraction. This algorithm initializes the system components, including the pre-trained YOLOv8 object detection model, the LLaMA language model, and the OpenCV frame reader. For each video frame, YOLOv8 identifies objects such as players, referees, and the ball while bounding-box data are extracted for further analysis. Player regions are cropped and converted to the HSV color space to suppress the influence of the grass field. A green mask is applied, and the dominant jersey color and spatial coordinates are computed for each player. These features provide essential inputs for subsequent modules, including team identification and match state analysis.

Algorithm 1. Pseudocode for object detection and feature extraction stages.

```

Load YOLO_Model = pre-trained YOLOv8 weights
Load LLaMA_Model = pre-trained language model (via API)
Initialize frame_reader = OpenCV(video_file)
Initialize previous_ball_position = None
Initialize static_ball_frames = 0
Initialize team_colors = empty list
Initialize rules = set of domain rules for match analysis

WHILE frame_reader.hasNextFrame():
    frame = frame_reader.read()

    detections = YOLO_Model.detect(frame)
    # Each detection: {object_class, confidence, bounding_box}

    player_boxes = filter(detections, class="player")
    ball_box = filter(detections, class="ball")
    referee_box = filter(detections, class="referee")

    FOR each player_box IN player_boxes:
        (x1, y1, x2, y2) = player_box.bounding_box
        player_region = crop(frame, x1, y1, x2, y2)

```

```

# Convert to HSV to remove green field influence
hsv_img = RGBtoHSV(player_region)
grass_mask = mask_color_range(hsv_img, green_range)
jersey_color = mean_color(player_region, exclude=grass_mask)

Append jersey_color TO team_colors
Store player_position = center(x1, y1, x2, y2)

```

```

ball_position = center(ball_box.bounding_box)

```

3.2. Team Identification Result

Following successful player detection using the YOLOv8 model, team identification was performed to distinguish players based on their jersey colours. This step is essential for downstream tasks such as ball possession tracking, tactical analysis, and automated commentary generation.

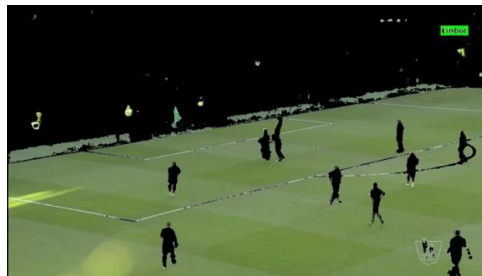


Figure 3. Estimation of grass colour.

The implemented method follows the approach described in Section 2. First, the grass colour was dynamically estimated by converting each frame into the HSV colour space and computing the average hue of pixels within a predefined green range, as formulated in Equation 1. This dynamic estimation compensates for variations in lighting and field appearance. The resulting hue value, denoted as H_g , was then used to generate a binary mask that isolates grass pixels, as shown in Equation 2. The visual outcome this step is presented in Figure 3.

Subsequently, only the upper half of each detected player bounding box was analysed, since it primarily contains jersey regions stated in Equation 3. A binary mask was applied, using Equation 4, to remove green pixels corresponding to grass, producing a segmented region that represents the actual jersey area. From this region, the dominant jersey colour for each player was extracted by averaging the non-grass pixel values (Equation 5). The visual results of this process are shown in Figure 4.

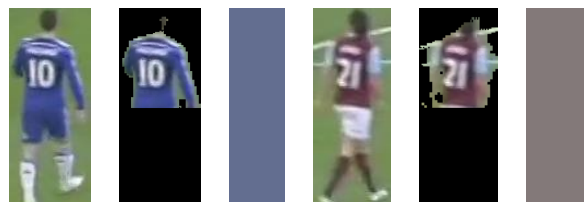


Figure 4. Estimation of jersey colour.

The obtained jersey colour vectors were then clustered using the K-Means algorithm with $k = 2$, effectively separating players into two distinct teams (Equation 6). To maintain consistent labelling

across frames, the average x-coordinate of players in each cluster was computed (Equation 7), and the team with the lower mean x-position was designated as the left-side team (Equation 8). The visual outcomes is illustrated in Figure 5.



Figure 5. K-Means-based team classification.

Empirical testing on several randomly selected match images demonstrated that the proposed K-Means clustering method consistently and accurately grouped players into their respective teams. Figure 5 presents representative results, showing that players wearing similar jersey colours were correctly clustered and colour-labelled. Moreover, team-side identification based on horizontal player distribution proved robust under varying camera angles and jersey designs.

Overall, this method ensures consistent and reliable team classification across frames and match conditions, providing a solid foundation for higher-level reasoning modules within the system.

Algorithm 2 assigns players to teams based on jersey color clustering and determines each team's field side. Using the K-Means algorithm, player jersey colors are grouped into two clusters representing Team L and Team R. Each detected player is then assigned to the nearest color cluster. The algorithm computes the average horizontal (x-axis) position of both teams, and the team with the smaller mean x-value is designated as the left-side team, while the other is assigned to the right.

Algorithm 2. Pseudocode for team identification.

```

IF not teams_defined:
    clusters = KMeans(team_colors, n_clusters=2)
    Assign Team_L and Team_R based on cluster index
    teams_defined = True

FOR each player:
    Assign player.team = closest_cluster(player.jersey_color)

Determine average x-position of each team:
    mean_x_L = mean(positions of Team_L)
    mean_x_R = mean(positions of Team_R)

IF mean_x_L < mean_x_R THEN
    side_left = Team_L
    side_right = Team_R

```

3.3. Object Detection Result

To evaluate the overall system performance, a real-world football match video was used as input to the integrated framework comprising object detection, team classification, and visual annotation modules. This evaluation aimed to demonstrate the end-to-end functionality and stability of the developed system when processing continuous, dynamic video data.



Figure 6. Video object detection.

Figure 6 shows example frame from the annotated output video that successfully detect object and classify the team. Each bounding box is labelled with the appropriate role (e.g. *Player-L*, *GK-R*, *Ball*) and color-coded by class. The left and right teams consistently labelled across frames, reflecting stable team clustering and classification. The system demonstrates accurate localization of players and the ball, even under varying lighting conditions, occlusion, and motion blur.

The integrated process operates within a continuous while-loop until all frames are processed. Upon completion, the output video consistently displays correctly labelled players, goalkeepers, and ball positions. The model maintained robust detection accuracy despite environmental challenges such as lighting variations, occlusions, and camera motion. This results confirm that the end-to-end pipelines (combining YOLOv8-based object detection, dominant-colour team classification, and real-time visualization) performs effectively in complex, real-world football video scenario. This implementation establishes a solid foundation for higher-level reasoning modules, including ball possession estimation and tactical analysis.

3.4. Rule-Based Detection Result

To interpret gameplay events from visual data, a rule-based reasoning module was developed based on the logical conditions defined in Section 2.4. These rules were applied in real time to the outputs of the YOLO-based object detection and team classification modules, enabling the system to infer high-level football events such as kick-off, ball possession, and shooting attempts.

The system was evaluated using multiple video clips of real football matches. As each video was processed frame by frame, the predefined logical conditions were continuously assessed. When the criteria for a specific event were satisfied, the corresponding situation was automatically identified and labelled.

Detected events were visualized by dynamically generating textual annotations on the lower-left corner of each video frame. These in-frame comments provided direct visual confirmation of correct event recognition and demonstrated the system's ability to link low-level visual cues to semantic gameplay situations in real time.

Algorithm 3 infers the current game state based on ball motion and positional data. The system first calculates the ball's displacement between consecutive frames to determine whether it is static or in motion. Based on predefined thresholds and spatial rules, the algorithm classifies situations such as *Kick-off*, *Free Kick*, *Penalty*, *Attack Direction* (Left or Right), *Shot on Goal*, or *Neutral Play*. The analysis integrates temporal ball movement with player positioning to provide real-time recognition of key match events.

Algorithm 3. Pseudocode for game situation recognition.

```
# Compute ball motion
IF previous_ball_position IS NOT None:
    ball_dx = ball_position.x - previous_ball_position.x
    ball_dy = ball_position.y - previous_ball_position.y
    distance = sqrt(ball_dx2 + ball_dy2)
IF distance < STATIC_THRESHOLD:
    static_ball_frames += 1
ELSE:
    static_ball_frames = 0
ENDIF

# Apply Rules
IF all_players_in_own_half() AND ball_centered():
    game_state = "Kick-off"
ELSE IF static_ball_frames > MAX_STATIC_FRAMES:
    game_state = "Free Kick"
ELSE IF is_penalty_condition(ball_position, player_positions):
    game_state = "Penalty"
ELSE IF team_in_possession(ball_position) == side_left:
    direction = detect_attack_direction(ball_position)
    game_state = "Left Attack (" + direction + ")"
ELSE IF team_in_possession(ball_position) == side_right:
    direction = detect_attack_direction(ball_position)
    game_state = "Right Attack (" + direction + ")"
ELSE IF ball_approaching_goal(ball_dx, ball_dy):
    game_state = "Shot on Goal"
ELSE:
    game_state = "Neutral Play"

previous_ball_position = ball_position
```

3.4.1. Kick-Off Detection

During the evaluation using real match video footage, the system successfully detected the kick-off event when both teams were positioned in their respective halves at the start of the game. Once the condition defined in Equation 9 was met, the system generated the corresponding commentary "Kick-off" in yellow font. displayed at the bottom-left corner of the annotated video frame, as

illustrated in Figure 7. This visual confirmation demonstrates that the rule-based logic operated as intended, accurately identifying and annotating gameplay situations in real time.



Figure 7. Kick-off detection.

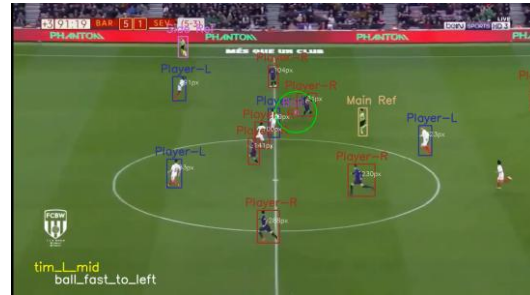


Figure 8. Ball possession detection.

3.4.2. Ball Possession

Ball possession detection was implemented according to the rule defined in Equation 10, which assigns possession to the player nearest to the ball within a dynamic distance threshold. This mechanism is fundamental for enabling higher-level tactical inferences, such as determining the direction of attack and identifying shooting opportunities.

During testing with real match video samples, the system successfully identified the player closest to the ball and accurately attributed team possession. The algorithm continuously recalculated player-to-ball distances for each frame, ensuring real-time updates of possession status.

As illustrated in Figure 8, when a player approached the ball an attacking sequence, the system correctly detected the controlling team and generated contextual commentary, such as "Team L {direction}", displayed in the bottom-left corner of the annotated video frame. This outcome confirms the effectiveness of the rule-based mechanism in providing reliable and interpretable situational awareness.

3.4.3. Attack Direction and Ball Direction

Attack direction was determined based on both the current ball possession status and the spatial position of the ball on the field, as described in Equation 11. Once possession was established as *True*, the pitch was divided into three vertical zones: upper third (top), middle third (mid), and lower third (bottom). The direction of the attack was then inferred from the ball's location within these zones relative to the team in possession.

During implementation, the system continuously evaluated the ball's position and generated contextual commentary such as "Team L bottom", displayed on the annotated video frame (see Figure 8).

In addition to positional heuristics, motion vectors of the ball were computed using differences in consecutive frame coordinates (Δx , Δy). The direction of ball movement (left, right up, or down) was inferred from the relative magnitudes of $|\Delta x|$ and $|\Delta y|$. This information was rendered in white text on the video frame to visualize the ball's motion trajectory and to further validate the orientation of the attacking play.

3.4.4. Shot Detection

In the annotated frame shown in Figure 9, the system accurately detected a shooting event when a player from the attacking team directed the ball toward the opposing goalkeeper. This detection was achieved by evaluating the team's ball possession status and analysing the direction of the ball's movement relative to the goalkeeper's position, as defined in Equation 12. Consequently, the system generated the contextual commentary "Shot_R!" displayed in the lower-left corner of the video frame. This result demonstrates the effectiveness of the rule-based reasoning module in identifying and annotating critical match events in real time.



Figure 9. Shot detection.

3.4.5. Dead Ball

The *dead-ball* condition in football was detected based on ball movement, following the rule defined in Equation 14. When both the current and previous ball positions were available, the system calculated the displacement using the Pythagorean formula. If the ball remained stationary for several consecutive frames, it was considered static; otherwise, the count was reset.

According to the defined rule, when the ball stayed motionless for more than ten consecutive frames (or for a configurable threshold specified by the system) a *dead-ball* state was recognized. Once this condition was satisfied the system determined the closest to the ball by calculating the Euclidean distance between the ball and all detected players, identifying the nearest player and their corresponding team.

3.4.6. Free Kick

In Figure 10, the system successfully identified a free kick situation by detecting that the line between the ball and the goalkeeper was obstructed by players from the opposing team, as specified by the defined rule. This condition, combined with the ball being in a *dead-ball* state, triggered the generation of the contextual commentary "Free kick L!" displayed in the lower-left corner of the annotated video frame. This result demonstrates that the implemented rule-based logic effectively captures and labels free kick scenarios during gameplay.



Figure 10. Free kick detection.



Figure 11. Penalty detection.

3.4.7. Penalty

The system successfully detected penalty situations when the ball was positioned inside the penalty area and no opposing players obstructed the path between the ball and the goalkeeper, as specified in the rule-based logic. When the ball was in a stationary *dead-ball* state, this condition was recognized as a valid penalty scenario. Consequently, the system generated the contextual commentary "Penalty_L", displayed in the lower-left corner of the annotated video frame, as illustrated in Figure 11. This result demonstrates the reliability of the implemented rule-based method in accurately identifying penalty situations during live match sequences.

3.4.8. Goal Kick

The system successfully detected goal-kick situations when the ball was stationary near the goalkeeper and no opposing players were present in the surrounding area. This condition satisfied the rule-based criteria that considered both the *dead-ball* state and player proximity. Once detected, the system generated the contextual commentary "Goal kick L!" displayed in the lower-left corner of the annotated video frame, as illustrated in Figure 12. This result validates the effectiveness of the implemented rule in recognizing real in-game goal-kick scenarios based on visual input and spatial configuration.

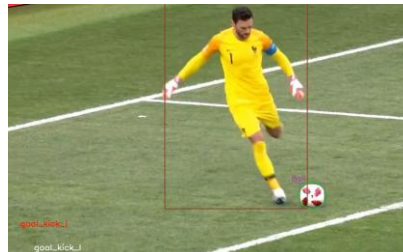


Figure 12. Goal kick detection.

3.5. Commentary Generation

To enhance the expressiveness of the rule-based commentary, a generative component was integrated into the system. This module transforms rule-derived textual outputs into natural sportscaster-style narration, enabling more engaging and dynamic match commentary. A dedicated function was designed to process raw comments produced by the rule-based logic and generate fluent, context-aware commentary in real time.

The function constructs a prompt that instructs the model to emulate a professional sports commentator. The temperature parameter was set to 1.0 to promote diverse and vivid phrasing, effectively simulating the enthusiasm of live broadcast narration.

To evaluate the effectiveness of this generative mechanism, several sample comments were tested. The results showed that the system consistently produced expressive and contextually appropriate commentary, as illustrated in Figures 13 and 14. These examples demonstrate that the system not only conveys essential event information but also presents it in a natural, lively, and audience oriented manner. The generated commentary aligns closely with the visual context and rule triggers, thereby enriching the overall viewing experience.



Figure 13. Improved commentary 1.



Figure 14. Improved commentary 2.

Algorithm 4 produces natural-language commentary based on the detected game state. For each identified event -- such as *Kick-off*, *Free Kick*, *Attack*, or *Shot on Goal* -- a corresponding base comment is generated. The text is then refined by the LLaMA language model, which rephrases it into a dynamic, sportscaster-style narration. The resulting commentary is displayed on the video frame and can optionally be converted to speech for real-time broadcasting.

Algorihm 4. Pseudocode for comment generation.

```

SELECT game_state:
  CASE "Kick-off":
    base_comment = "Kick-off"
  CASE "Free Kick":
    base_comment = "Free-Kick"
  CASE "Penalty":
    base_comment = "Penalty"
  CASE "Left Attack":
    base_comment = "Left Team Attacking"
  CASE "Right Attack":
    base_comment = "Right Team Attacking."
  CASE "Shot on Goal":
    base_comment = "Shot"
  CASE "Neutral Play":
    base_comment = ""

improved_comment = LLaMA_Model.generate(
  prompt = "Rewrite this to sports commentary: " + base_comment )

  DisplayTextOnFrame(frame, improved_comment)
  Optionally: ConvertToSpeech(improved_comment)

END WHILE

```

3.6. System Evaluation

To assess the effectiveness and realism of the developed system from a user perspective, a subjective evaluation was conducted involving 30 respondents identified as football enthusiasts. Each participant was presented with seven short video clips, each depicting a distinct match

scenario, namely penalty, free kick, kick-off, goal kick, ball possession, shooting, and attack direction.

After viewing each clip, the respondents were asked to rate the system's performance in recognizing the situation and generating appropriate commentary using a five-point Likert scale, where 1 indicated *very poor* and 5 indicated *excellent*. The aggregated results of this user evaluation are summarized in Table 2.

As shown in Table 2, the subjective evaluation yielded consistently high ratings across all tested scenarios, with an overall mean score of 4.44 on the five-point scale. The results indicate that respondents generally perceived the system as highly effective in identifying gameplay situations and producing relevant commentary. Among the evaluated scenarios, penalty detection achieved the highest average rating (4.50), followed by ball possession (4.47), reflecting user satisfaction with the system's ability to recognize critical match events.

Table 2. Validation result.

Scenario	Average Rating	Standard Deviation	Respondents	Standard Error (SE)
Kick-off	4,40	0,621	30	0,113
Ball possession	4,47	0,600	30	0,110
Attack direction	4,43	0,502	30	0,092
Shoot	4,43	0,626	30	0,114
Dead-ball – Free Kick	4,43	0,577	30	0,105
Dead-ball – Penalty	4,50	0,509	30	0,093
Goal Kick	4,43	0,571	30	0,104

The standard deviations ranged from 0.50 to 0.63, suggesting low variability and strong agreement among participants. Similarly, the standard errors, all near 0.10, confirm the stability and reliability of the reported mean values. Overall, these findings demonstrate that the developed system performs robustly and delivers realistic, contextually appropriate commentary across various football scenarios, validating its effectiveness from a user experience perspective.

3.7. Discussions

The experimental results and system evaluations presented in the previous section demonstrate the effectiveness of the proposed integrated framework that combines computer vision, clustering, rule-based reasoning, and natural language generation for football match understanding. This section provides a deeper analysis and critical discussion of these findings from both technical and cognitive perspective, highlighting how the system reflects aspects of human perception, reasoning, and linguistic interpretation in sports cognition.

From a perceptual standpoint, the YOLOv8-based object detection module exhibited strong performance on the validation set, achieving high precision and recall. The model effectively detected players, goalkeepers, and the ball in unseen video frames, demonstrating perceptual generalization similar to human visual attention in tracking salient objects under dynamic conditions. Nevertheless, occasional false positive were observed in frames with complex occlusion or partially visible balls, suggesting the need for temporal filtering or motion-based post-processing to approximate the human ability to integrate visual information across time.

The team classification process, which employed dominant jersey colour features and K-Means clustering, also proved robust across diverse video samples. The adaptive adjustment to lighting

conditions via HSV-based grass masking can be viewed as a form of perceptual normalization, analogous to how the human visual system compensates for environmental lighting variations. However, challenges remain when both teams wear similar colour tones or hues, which reduce feature distinctiveness--a limitation comparable to perceptual ambiguity in human colour differentiation under complex visual scenes.

The integration of object detection and team classification in real match video clips demonstrated smooth frame-by-frame annotation and reliable labelling consistency. The resulting visual outputs confirmed that the system can process complete video sequences autonomously, paralleling how human observers continuously track multiple moving agents to form coherent situational awareness. This integration provides a stable foundation for subsequent reasoning modules that emulate higher-order cognitive functions such as event inference and tactical understanding.

The rule-based reasoning components--responsible for identifying match events such as penalties, free kicks, and shots--achieved high detection accuracy. By incorporating *dead-ball* and ball-possession conditions as logical prerequisites, the system effectively minimized premature or incorrect detection. The rule-based structure reflects a form of symbolic reasoning found in human cognition, where event interpretation depends on contextual conditions and temporal dependencies. The ability to use spatial relationships and motion cues for inference aligns with cognitive models of situational reasoning, in which humans infer intentions and outcomes by integrating spatial-temporal information.

The generative commentary component further enhanced the system's interpretability and communicative realism. By transforming symbolic event data into natural sportscaster-style narration, the model operationalizes aspects of human linguistic cognition--particularly language generation grounded in situational context. Although minor variations in tone and length were observed, the generated commentary successfully reproduced features of expressive and context-sensitive human narration, bridging low-level perception with high-level communicative function. Finally, the subjective evaluation involving 30 football enthusiasts confirmed the system's practical and cognitive validity. The average user ratings exceeding 4.4 across all scenarios indicate that participants intuitively recognized and accepted the system's event interpretations and narrative outputs as realistic and meaningful. This alignment between computational processes and human interpretive perception suggests that the framework not only performs effectively in technical terms but also models key aspects of human cognitive behaviour in perceiving, reasoning about, and linguistically describing dynamic sports events.

4. CONCLUSIONS AND FUTURE STUDIES

This study presents an integrated framework for the automated understanding of football matches by combining computer vision, clustering algorithms, rule-based reasoning, and natural language generation. From a cognitive science perspective, the framework can be seen as a computational model that mirrors key aspects of human cognition in sports perception and interpretation--specifically, how humans visually perceive, reason about, and linguistically describe dynamic events.

The proposed system effectively detects players, the ball, and goalkeepers in real time using a custom-trained YOLOv8 model, and classifies players into their respective teams based on dominant jersey colours through a dynamic colour-masking approach. This visual perception process parallels the human visual system's ability to segment and categorize stimuli based on salient features under varying lighting and environmental conditions. The extracted visual data are processed using a set of logically defined rules to infer high-level match events such as kick-offs, ball possession, attack direction, shots, free kicks, penalties, and goal kicks. These rules are context

sensitive, incorporating conditions such as ball possession and *dead-ball* states to improve precision and reliability. This rule-based reasoning component aligns with cognitive theories of symbolic and situational reasoning, in which humans use contextual cues and causal inference to interpret ongoing events.

To enhance interpretability and viewer engagement, an AI-based commentary generator was integrated to produce natural, sportscaster-style descriptions aligned with detected events. This generative mechanism reflects the linguistic dimension of cognition, transforming structured representations of events into expressive, contextually grounded language--similar to how human commentators transform perceptual and conceptual understanding into narrative form.

A subjective evaluation involving 30 football enthusiasts demonstrated that the system's outputs were perceived as accurate, informative, and realistic. The high average ratings across all scenarios confirm the system's ability to provide meaningful and intuitively interpretable match interpretations. This alignment between computational processing and human perception underscores the frameworks' cognitive plausibility--its capacity to model how humans perceive, reason about, and verbally describe complex visual environments.

Looking ahead, several directions are proposed to enhance the system's cognitive and computational performance. The rule-based component can be expanded to include a broader range of match situations, such as drop balls, throw-ins, tackles, and formations. The system could also incorporate higher-order tactical analysis--recognizing attacking patterns, off-ball movements, and real-time ball possession statistics--to enrich commentary with strategic insights similar to human expert reasoning.

Integrating a text-to-speech feature would further advance the communicative dimension of the system by adding auditory expressiveness and real-time responsiveness, thus simulating a live broadcast experience. Additionally, enhancing the casual recognition capability--through the detection of field markings and the identification of players by jersey number, position, or distinctive features--would improve the granularity of perception and contextual precision in generated commentary.

Importantly, future iterations should also emphasize inclusivity and accessibility. By integrating features such as multilingual commentary, closed captioning, and audio description, the system could serve a broader audience -- including individuals with hearing or visual impairments, non-native speakers, and users from diverse cultural backgrounds.

Collectively, these advancement would not only strengthen the system's technical and cognitive dimensions but also promote a more inclusive and equitable digital sports environment. Ultimately, the proposed framework represents a significant step toward cognitively inspired intelligent systems for sport analysis and broadcasting, bridging the gap between artificial perception and human understanding in dynamic real-world contexts.

ACKNOWLEDGEMENTS

The authors would like to express gratitude to Universitas Kristen Duta Wacana for providing research facilities and supporting the publication of this work.

REFERENCES

- [1] Lee, G., & Bulitko, V. (2010). Automated Storytelling in Sports: A Rich Domain to Be Explored. In R. Aylett, M. Y. Lim, S. Louchart, P. Petta, & M. Riedl (Eds.), *Interactive Storytelling. ICIDS 2010. Lecture Notes in Computer Science* (Vol. 6432). Springer. https://doi.org/10.1007/978-3-642-16638-9_35
- [2] Rhodes, M., Coupland, S., & Cruickshank, T. (2010). Enhancing Real-Time Sports Commentary Generation with Dramatic Narrative Devices. In R. Aylett, M. Y. Lim, S. Louchart, P. Petta, & M. Riedl (Eds.), *Interactive Storytelling. ICIDS 2010. Lecture Notes in Computer Science* (Vol. 6432). Springer. https://doi.org/10.1007/978-3-642-16638-9_14
- [3] Siu, W.-C., Chan, H. A., Chan, S., Cheng, W.-H., Yang, B.-C., Chan, C.-Y., Hui, C.-C., & Salahudeen, R. (2023). On the completion of automatic football game commentary system with deep learning. *International Workshop on Advanced Imaging Technology (IWAIT) 2023, I2592*, 125920H. <https://doi.org/10.1117/12.2668398>
- [4] Andrews, P., Nordberg, O. E., Borch, N., Guribye, F., & Fjeld, M. (2024). Designing for Automated Sports Commentary Systems. *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences*, 75–93. <https://doi.org/10.1145/3639701.3656323>
- [5] Chen, T., Saxena, S., Li, L., Fleet, D. J., & Hinton, G. (2022). Pix2seq: A Language Modeling Framework for Object Detection. *International Conference on Learning Representations*. <https://openreview.net/forum?id=e42KbIw6Wb>
- [6] Zang, Y., Li, W., & Han, J. (2025). Contextual Object Detection with Multimodal Large Language Models. *Int J Comput Vis*, 133, 825–843. <https://doi.org/10.1007/s11263-024-02214-4>
- [7] Gao, Y., Zhou, H., Chen, L., Guo, C.-F., & Zhang, X.-Y. (2022). Cross-Modal Object Detection Based on a Knowledge Update. *Sensors*, 22(4), 1338. <https://doi.org/10.3390/s22041338>
- [8] Qi, J., Yu, J., Tu, T., Gao, K., Xu, Y., Guan, X., Wang, X., Dong, Y., Xu, B., Hou, L., Li, J., Tang, J., Guo, W., Liu, H., & Xu, Y. (2023). GOAL: A Challenging Knowledge-grounded Video Captioning Benchmark for Real-time Soccer Commentary Generation. *arXiv. Org*. <https://doi.org/10.48550/arXiv.2303.14655>
- [9] Wang, X., Shu, X., Zhang, Z., Jiang, B., Wang, Y., Tian, Y., & Wu, F. (2021). Towards More Flexible and Accurate Object Tracking with Natural Language: Algorithms and Benchmark. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [10] Zhou, L., Zhou, Z., Mao, K., & He, Z. (2023). Joint Visual Grounding and Tracking with Natural Language Specification. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 23151–23160. <https://doi.org/10.1109/CVPR52729.2023.02217>
- [11] Yu, E., Liu, S., Li, Z., Yang, J., Li, Z., Han, S., & Tao, W. (2023). Generalizing Multiple Object Tracking to Unseen Domains by Introducing Natural Language Representation. *The Thirty-Seventh AAAI Conference on Artificial Intelligence*.
- [12] Nguyen, P., Quach, K. G., Kitani, K., & Luu, K. (2023). Type-to-Track: Retrieve Any Object via Prompt-based Tracking. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in Neural Information Processing Systems* (Vol. 36, pp. 3205–3219). Curran Associates, Inc. <https://doi.org/10.48550/arXiv.2305.13495>
- [13] Mangalika, U. (2024). Object Recognition to Content Based Image Retrieval: A Study of the Developments and Applications of Computer Vision. *Journal of Computing and Natural Science*. <https://doi.org/10.53759/181x/jcns202404005>
- [14] Chan, C.-Y., Hui, C., Siu, W.-C., Chan, K. W., & Chan, H. A. (2022). To Start Automatic Commentary of Soccer Game with Mixed Spatial and Temporal Attention. *IEEE Region 10 Conference*. <https://doi.org/10.1109/TENCON55691.2022.9978078>
- [15] Kumar, A., & Jha, A. K. (2024). Enhancing Accesibility in Digital Broadcasting: A Comprehensive Research Analysis. *JETIR*, 11(2), 775-783.
- [16] Almajali, I. A. (2025). Enhancing Podcast Accessibility: The Role of Audio Descriptions for Visually Impaired Listeners. *Power System Technology*, 49(3), 22-32.
- [17] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- [18] Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716. <https://doi.org/10.3390/make5040083>

- [19] Varghese, R. (2024). *YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness*. <https://doi.org/10.1109/adics58448.2024.10533619>
- [20] Cioppa, A. (2022). SoccerNet-Tracking: Multiple Object Tracking Dataset and Benchmark in Soccer Videos. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 3490–3501. <https://doi.org/10.1109/CVPRW56347.2022.00393>

AUTHORS

G. Virginia received Ph.D. from University of Warsaw and Master of Artificial Intelligence (MAI) from Katholieke Universiteit Leuven (KUL). She did her Bachelor degree in Informatics from Universitas Kristen Duta Wacana (UKW). Her research interests include Applied Cognitive Science, Artificial Intelligence, Machine Learning, and Human-Computer Interaction.



J. Susilo did his Bachelor degree in Informatics from Universitas Kristen Duta Wacana (UKDW). Currently he works as a backend systems programmer. His research interests include Artificial Intelligence, Machine Learning, and Computer Science.



A.W. Mahastama received Master of Computer Science from Gadjah Mada University (UGM). He did his Bachelor Degree in Informatics from Universitas Kristen Duta Wacana (UKDW). His research interests include Image Processing, Computer Vision, and Computer Linguistics.

