



Fusion-Net50-Based Soccer Player Re-Identification and Team Clustering for Speed and Movement Analysis

Syahrul Idhom¹, Moch Arief Soeleman², Catur Supriyanto³

^{1,2,3}Master's Program in Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia

*syahrulidhom05@gmail.com

Abstract

Comprehensive analysis of soccer matches requires substantial human resources as well as specialized equipment with relatively high costs. The development of Computer Vision (CV) technology, particularly the Multi-Object Tracking (MOT) approach, enables contactless and efficient analysis of player performance. However, the main challenges in MOT for soccer, namely player re-identification, tracking ID switches, as well as inter-frame identity consistency, especially under occlusion conditions and low-confidence detections, can cause the analysis data to become invalid. This research aims to improve object identity consistency (Tracking ID) in video-based soccer match analysis by integrating detection, tracking, re-identification, and team identification methods. This research proposes the integration of YOLOv8n- and ByteTrack-based tracking methods with a Re-ID embedding mechanism, low-confidence detection handling, as well as ResNet-50-based Feature-Position Fusion to maintain identity consistency (Track ID) throughout the video sequence. Furthermore, team identification is consistently performed using a Color Embedding approach with a ResNet-50 backbone and the K-Means clustering algorithm. The experimental results show that the integration of all these methods successfully maintains consistent Track ID with a total of 25 Track IDs (Players 21 Track IDs, Goalkeeper 1 Track ID, Referee 2 Track IDs, Ball 1 Track ID). This performance is supported by evaluation metrics indicating strong performance, namely HOTA of 91.9%, MOTA of 89.8%, and MOTP of 81.0%. The effectiveness of the Re-ID module in maintaining object identity consistency is reflected in the IDF1 score of 94.9%. Supported by association metrics (AssA 93.5%, AssRe 94.8%, AssPr 96.3%), these results indicate that trajectories are formed with a relatively low level of association errors, thereby supporting subsequent analyses such as speed estimation and player movement visualization in a more stable manner.

Keywords: ByteTrack; football; K-means Clustering; Re-ID-ColorEmbedding ResNet-50; YOLOv8n

How to Cite: S. Idhom, M. A. Soeleman, and C. Supriyanto, "Fusion-Net50-Based Soccer Player Re-Identification and Team Clustering for Speed and Movement Analysis", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 4, pp. 1065 - 1078, Aug. 2026.
Permalink/DOI: <https://doi.org/10.29207/resti.v10i4.7457>

Received: January 8, 2026

Accepted: June 21, 2026

Available Online: August 13, 2026

*This is an open-access article under the [CC BY 4.0 License](#)
Published by [Ikatan Ahli Informatika Indonesia](#)*

1. Introduction

Soccer remains the number one sport in the world with extraordinary cross-platform appeal. UEFA recorded more than 2.67 million spectators attending stadiums in person during EURO 2024, while more than 5.8 million fans crowded fan zones across Germany. On the other hand, the previous FIFA World Cup reached more than 3.5 billion global viewers through television and digital broadcasts, and digital consumption trends continue to increase sharply until 2025 [1]. Soccer, the most popular sport in the world, has a significant impact on daily life, shaping the economy, social dynamics, and national identity [2]. The soccer community and team performance have a symbiotic relationship. The environment surrounding a team influences support,

identity, financial support, player motivation, and social reflection, which in turn affects team performance on the field [3]. Soccer matches are complex and unpredictable, requiring complex strategies to achieve success. Traditional performance analysis methods emphasize measurable data such as physical activity and basic game events, but often overlook more subtle tactical dimensions that can have a significant impact on match outcomes [4].

Analyzing soccer matches requires substantial human resources and extensive expensive equipment across clubs, including stadiums, training fields, and other key locations. This equipment includes high-quality cameras installed throughout the stadium to obtain an overall view of the match in order to gain insights into

the game both on and off the field [5]. However, such approaches have limitations in terms of cost, flexibility, and accessibility because they rely on specialized hardware such as high-resolution cameras and integrated tracking systems that are not always available at all levels of competition or training environments. The development of Computer Vision (CV) technology can serve as one of the important contactless solutions in the context of soccer performance analysis, with a focus on four main areas: player and ball detection and tracking, motion prediction, tactical analysis, and match event detection, while offering many opportunities to enhance the clarity, accuracy, and intelligence of sports event observation [6].

The demand for flexible and efficient visual tracking has driven the adoption of deep learning-based approaches as alternatives to conventional systems that rely on specialized hardware. Sun et al. [7] and Edozie et al. [8] review the development of object detection methods from traditional approaches to modern deep learning-based models, including CNN-based two-stage detectors (R-CNN, Fast R-CNN, and Faster R-CNN), one-stage detectors (SSD and YOLO along with their variants), as well as transformer-based methods such as DETR, and state that although YOLO still lags behind transformer-based methods in terms of accuracy, this method has a primary advantage in its ability to perform real-time detection, making it more suitable for real-world applications.

The application of YOLO models remains challenging due to complex backgrounds, high object scale variation, and difficulties in detecting small objects; however, many approaches prioritize accuracy while neglecting the balance between precision and efficiency, thereby limiting their applicability in resource-constrained environments [9]. Santos Júnior et al. [10] show that YOLOv8 offers a competitive balance between detection performance and computational efficiency, making it relevant for real-time applications. Wang et al. [11] integrate YOLOv8s as the detector and ByteTrack as the tracker, resulting in an efficient and real-time multi-object tracking system that balances inference speed and detection accuracy. Meanwhile, You et al. [12] combine YOLOv8, ByteTrack, and Kalman filtering to improve the accuracy of overlapping detections and enhance tracking stability under occlusion conditions and temporary detection loss. However, Kalman filter-based approaches still exhibit limitations, particularly in scenarios involving slow-moving or temporarily stationary objects, where motion prediction becomes less informative, causing objects with low detection scores to remain vulnerable to elimination from the tracking process, as reported by Cao et al. [13]. To address the problem of interrupted object detections Stanojević et al. [14], introduce multi-object tracking by exploiting tracklet information, namely short-term tracking trajectories that represent object identity consistency across several consecutive frames, to improve the continuity of low-confidence

object detections. Unlike conventional methods that discard weak detections, these approaches perform confidence boosting and apply adaptive association thresholds, allowing a larger number of objects to be retained during tracking.

Multi-object tracking in sports scenarios has advanced through the integration of deep learning techniques. However, challenges such as player re-identification and ID switching remain major issues, as reported by Sun et al. [15]. In the context of soccer video, Multiple Object Tracking (MOT) frequently suffers from Track ID changes for players, goalkeepers, and referees due to dynamic motion, occlusion, relatively uniform visual appearance, and complex interactions among objects. To address this issue, Naik et al. [16] introduced DeepPlayer-Track for tracking players and referees in soccer videos by leveraging YOLOv4-based detection and jersey color recognition as part of appearance association features with SORT tracking, demonstrating that color information can help maintain identity consistency across frames. However, reliance solely on color features leads to tracking failures during intra-team occlusion and under varying lighting conditions. In response to these limitations, in subsequent work, Naik et al. [17] proposed a new approach based on the integration of Background Subtraction (MOG2), team classification using HSV color masking, and an association mechanism employing EIoU-Distance Loss. As a result, this method is able to improve tracking stability compared to the use of a single metric. However, the study acknowledges a limitation in which the system fails to re-identify players that are temporarily lost due to missed detections, often assigning a new ID when the object reappears. From the perspective of feature representation for re-identification, Zhou et al. [18] developed Omni-Scale Feature Learning (OSFL) for person re-identification by introducing the Omni-Scale Module (OSM), which is capable of capturing local and global details simultaneously, and by integrating Adaptive Instance Normalization (AIN), making the feature representation more adaptive to domain variations, thereby maintaining model consistency despite changes in camera settings, lighting conditions, or environments. Mohammed et al. [19] proposed a hybrid system that combines Convolutional Neural Networks (CNN) and Deep Belief Networks (DBN) for person re-identification (Re-ID), which has been shown to improve feature representation quality and achieve higher accuracy on datasets captured with static cameras.

To produce accurate data, team-based grouping is required so that the data are not distorted by mixing with data from other teams. Tanapatpiboon et al. [20] developed a deep learning-based system to detect players and identify teams in soccer videos using several YOLOv5 variants. Player positions are extracted directly from detection outputs and visualized in the form of movement intensity heatmaps. Although team differentiation can be achieved effectively, this

approach relies on dedicated training to define player classes for each team, making it less flexible when applied to matches with different team combinations or jersey colors. A study by Chen et al. [21] demonstrates that K-Means is capable of forming homogeneous clusters that improve detection robustness against outliers. This finding is further supported by Jardim et al. [22], who reveal that the K-Means algorithm is effective for region extraction in RGB images, even under challenging visual conditions such as low resolution, high noise, and non-uniform lighting variations. The ability of K-Means to separate visual elements provides a strong foundation for its application in automatic inter-team player clustering based on jersey color.

Based on prior studies that form the foundation of this research, the integration of player re-identification and team clustering methods is expected to maintain player identity consistency (Track ID) and team membership throughout a match, thereby supporting the extraction of tactical information and accurate and efficient soccer performance analysis from match videos, without reliance on specialized hardware.

2. Methods

In this study, an integration of YOLOv8n- and ByteTrack-based tracking with Re-ID embedding and low-confidence integration, as well as Feature-Position Fusion for Identity Refinement based on ResNet-50, is proposed. Furthermore, consistent team identification is performed using ResNet-50-based color embedding and K-Means clustering. Therefore, the overall approach proposed in this study is hereafter referred to as the Fusion-Net50-based approach. The overall research methodology flow is illustrated in Figure 1.

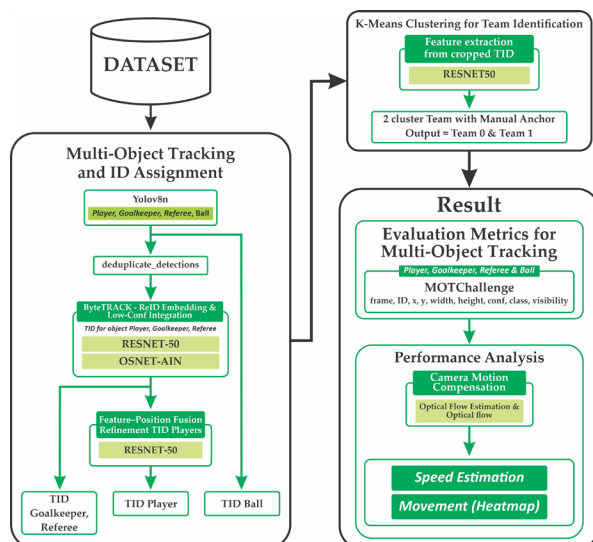


Figure 1. Overview of the proposed research methodology

2.1 Dataset

The dataset is derived from SoccerNet-Tracking: Multiple Object Tracking Dataset and Benchmark in Soccer Videos [23]. This dataset is complex and

consists of several components, including Labels-v3.json and Frames-v3.zip, with splits categorised into train, test, and challenge, and tasks focused on frames. The complete dataset contains 550 matches, including 12 single-camera matches. The total video duration is 764 hours of match content, with a total storage size of 60 GB and label files of 1 GB. The dataset in each match is divided into several folders, each containing 750 image frames, text data in the form of game information, detection data, and label data (ground truth), as well as video information with a frame rate of 25 fps and a resolution of 1920×1080 pixels. In this study, a subset of seven folders was selected for experimentation. Four folders (3,000 frames) from one match were used for detector training. For tracking evaluation, three folders (2,250 frames) were used under a held-out protocol, consisting of one unseen folder from the same match and two additional folders originating from two different matches as shown in Figure 2. This split ensures separation between training and evaluation data while allowing assessment across multiple match contexts.



Figure 2. The three matches used in this study: (left) Match 1, representing unseen segments from the training match; (center) Match 2; and (right) Match 3, originating from different matches

2.3. Preprocessing

The image dataset and its corresponding annotations are converted into the YOLOv8n training format. Four dataset folders comprising 3,000 frames were exclusively used for training the YOLOv8n model. The 3,000 training frames were further randomly partitioned into 70% training, 20% validation, and 10% internal testing subsets to optimize the YOLOv8n detection model.

Each subset contains corresponding image frames and label files. A YAML configuration file was prepared to define the dataset structure and class mappings. The trained YOLOv8n model was then used to generate object detections for Player, Goalkeeper, Referee, and Ball classes. All tracking evaluations were conducted exclusively on three held-out folders (2,250 frames) not used during detector training or validation, spanning three distinct matches: unseen segments from the training match and two entirely different matches.

2.4 Object Detection using YOLOv8n

At this stage, model training is conducted to detect objects including Player, Goalkeeper, Referee, and Ball using the YOLOv8n architecture initialized with pre-trained weights, which is the nano version available in the Ultralytics library.

One of the main advantages of YOLOv8 is its ability to provide real-time predictions with high accuracy, making it highly suitable for various applications such

as video surveillance, sports analysis, autonomous vehicles, and security systems [24].

2.5 Deduplicate Detection

Before being passed to ByteTrack, the YOLOv8n detection results are first filtered using the `deduplicate_detections` function with the parameters `iou_threshold`, `min_center_dist`, and `shrink_ratio`. This function aims to remove duplicate detections, commonly referred to as double bounding boxes. The corresponding equations for the bounding box representation and refinement are presented in Equations 1-6.

$$b = (x_1, y_1, x_2, y_2) \quad (1)$$

$$c_x = \frac{x_1 + x_2}{2}, c_y = \frac{y_1 + y_2}{2} \quad (2)$$

$$w = x_2 - x_1, h = y_2 - y_1 \quad (3)$$

$$w' = \frac{w \cdot \alpha}{2}, h' = \frac{h \cdot \alpha}{2} \quad (4)$$

$$x'_1 = c_x - w', x'_2 = c_x + w' \quad (5)$$

$$y'_1 = c_y - h', y'_2 = c_y + h'$$

$$b' = (x'_1, y'_1, x'_2, y'_2) \quad (6)$$

Bounding box b has a width $w = x_2 - x_1$ and a height $h = y_2 - y_1$. The center point of the bounding box is denoted by (c_x, c_y) , while α represents the shrink ratio used to reduce the size of the bounding box, resulting in a new bounding box b' as defined in Equation 6.

$$IoU(A, B) = \frac{Area_{inter}}{Area_{union}} \quad (7)$$

$$w_{inter} = \max(0, \min(x_2^A, x_2^B) - \max(x_1^A, x_1^B)) \quad (8)$$

$$h_{inter} = \max(0, \min(y_2^A, y_2^B) - \max(y_1^A, y_1^B)) \quad (9)$$

$$Area_{inter} = w_{inter} h_{inter} \quad (10)$$

$$Area_{union} = Area_A + Area_B - Area_{inter} \quad (11)$$

Equations 7–11 define the Intersection over Union (IoU) between two bounding boxes A and B , where $Area_{inter}$ denotes the intersection (overlap) area and $Area_{union}$ denotes the union area. The minimum center distance between two bounding boxes is computed using the Euclidean distance between their center points, as defined in Equation 12.

$$d(i, j) = \sqrt{(c_x^i - c_x^j)^2 + (c_y^i - c_y^j)^2} \quad (12)$$

2.6 Multi-Object Tracking using ByteTrack

In this study, the ByteTrack implementation is adopted from the Supervision library, which provides a practical interface for integration with YOLOv8n detection outputs and supports the configuration of tracking parameters such as track activation threshold, lost track buffer, minimum matching threshold, and frame rate. These parameter settings enable an efficient track management mechanism to maintain object identity (ID) consistency across frames [25]. In addition, an additional filtering mechanism is applied in the form of a minimum consecutive frames parameter, which ensures that an object is considered a valid track only

when it is detected consistently. This mechanism aims to minimize identity switches (ID switches) caused by detection instability across frames.

2.7 Re-ID - Embedding dan Low-conf integrasi

The embedding generation module is designed by leveraging a combination of OSNet-AIN and ResNet-50. OSNet-AIN is used to extract appearance or identity features such as body structure, texture, and clothing patterns. Meanwhile, the ResNet-50-based color embedding with ImageNet pre-trained weights incorporates both Adaptive Average Pooling and Adaptive Max Pooling to capture dominant color information as well as high-contrast color cues from player jerseys. The extracted color features are then projected into a 128-dimensional embedding vector, directly derived from the dataset used. When detection failed, the system executed layered recovery mechanisms to maintain identity consistency: (1) utilizing `lost_tracks` with Kalman prediction, (2) motion-history-based fallback from the last two frames, and (3) fallback from previously stored history frames in `_saved_tracks` when the number of predictions was fewer than 5. Subsequently, a Low-Confidence (Low-Conf) function was employed as a critical additional fallback. YOLOv8n detection bounding boxes with low confidence—which were not absorbed by the deduplication process—were retained to assist when detections were lost. This mechanism aims to maintain the identity consistency of each object even when it has low confidence values due to occlusion or motion blur. ByteTrack was then updated once per frame using these consolidated detections, and tracking history was stored after the stability condition was satisfied.

Once the tracks were stabilized, the Re-ID gallery was updated by matching detection embeddings with active track IDs, but only if the confidence was sufficiently high and the track had passed the `minimum_consecutive_frames` threshold. High-confidence tracks were stored in `frame_tracks` with category-wise ID offsets. Tracks that were temporarily lost but still within the `max_time_lost` tolerance (30 frames) could be recovered through spatial validation against the previously mentioned deduplicated detections or low-confidence predictions, and then updated using Kalman filtering. By prioritizing Low-Confidence detection integration and spatial consistency, the proposed framework avoids using Re-identification (Re-ID) embeddings to directly override tracking identities. Instead, the embeddings serve solely as an auxiliary reference to support identity consistency during recovery stages.

2.8 Feature–Position Fusion for Identity Refinement

At this stage, a Feature–Position Fusion for Identity Refinement mechanism is applied, which combines color feature similarity and spatial position proximity among Track IDs. Color features are extracted using a ResNet-50-based color embedding module that follows the same architectural design as the color embedding

used in the initial Re-identification stage. The model integrates Adaptive Average Pooling and Adaptive Max Pooling to capture dominant and high-contrast color information, producing a 128-dimensional embedding for each object crop. For each Track ID, a set of image crops is collected from several initial and final frames of its appearance. The color embeddings from these crops are then averaged to obtain a more stable feature representation. In addition, object position is represented using the bounding box center point (centroid), which is used as an indicator of spatial proximity among Track IDs.

Based on the extracted feature–position representations, the identity refinement mechanism is performed in a two-stage process, initialized by selecting stable Anchor Track IDs from the initial frames. In the first stage, non-anchor Track IDs are consolidated by matching later tracks with earlier ones based on color embedding cosine similarity and spatial proximity measured using Euclidean distance. In the second stage, the consolidated tracks are compared against the Anchor Track IDs to unify fragmented identities. Finally, a filtration step is applied to remove Track IDs with insufficient crop counts or detections located outside the field boundaries, ensuring stable and reliable tracking results.

2.9 K-Means Clustering for Team Identification

Clustering is performed to identify and group objects into two teams, namely Team_0 and Team_1. The clustering process utilizes the feature extraction results from the ResNet-50 model. Subsequently, several Track IDs are manually selected as anchors (references) to assign the identity of each team. The K-Means clustering formulation is given in Equation 13.

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} |x - \mu_i|^2 \quad (13)$$

$S = \{S_1, S_2, \dots, S_k\}$ expresses the partition of the data into k cluster, where x denotes a data point and μ_i represents the centroid of cluster S_i . The distance between a data point and the centroid is measured using the squared Euclidean distance, defined as $|x - \mu_i|^2$.

Because the numerical labels generated by K-Means are arbitrary and may vary across executions, these anchors are used during the post-labeling stage to provide semantic interpretation of the clustering results, ensuring consistent team assignment. The anchor Track IDs are not involved in the clustering process; they are applied only after clustering is completed and therefore do not affect cluster formation, centroid computation, or membership assignment. The anchors are selected based on tracking stability and visual representativeness, and may vary across different tracking scenarios or datasets.

2.10 Evaluation Metrics for Multi-Object Tracking

The detected object results obtained from the integration of all methods are then exported in the MOTChallenge format and compared against the

existing ground-truth annotations in the SoccerNet dataset. The MOTChallenge files are evaluated using the metrics HOTA, DetA, DetRe, DetPr, AssA, AssRe, AssPR, OWTA, and LocA to comprehensively assess the system performance balance, covering detection quality, inter-frame association accuracy, and bounding box alignment.

In addition, to measure detection error rates (false positives/negatives), positional precision, and object identity consistency (Re-ID) throughout the match video sequences, the metrics MOTA, MOTP, and IDF1 are employed. This set of evaluation metrics is widely used in Multi-Object Tracking evaluation, particularly in SoccerNet benchmarks and challenges, as it effectively represents detection performance, identity association consistency, and overall bounding box quality.

2.11 Speed and movement analysis of players based on the model results

The overall model results are then extracted to generate match analysis, including player speed and team movement analysis. To reduce bias caused by camera motion, the Optical Flow method is used as a tool for camera motion compensation [26], [27]. This method is estimated to play an important role in separating moving objects from the background [28]. Thus, the resulting estimates of team speed and movement become more stable and accurate.

The speed of each player is calculated by converting pixel units to meters based on the field scale in the video. This conversion uses soccer field dimensions as a reference. According to FIFA standards, the field has a length of 110 m and a width of 64 m. The center circle has a radius of 9.15 m or a diameter of 18.3 m.

Player speed computation is based on changes in player positions between frames represented in pixel units. In computer vision–based speed tracking systems, pixel-to-meter conversion is performed through spatial calibration using known real-world distance references. In this study, since the field is not fully visible, the diameter of the center circle (18.3 m) is used as the reference. The obtained meter-per-pixel scale factor is subsequently used to convert player positional displacements between frames into meters. A player's instantaneous speed is computed as the ratio of displacement distance to the inter-frame time interval, which is determined based on the video frame rate, and is expressed in meters per second (m/s). For performance interpretation purposes, the speed values are converted into kilometers per hour (km/h) using the standard International System of Units (SI) conversion factor, namely $1 \text{ m/s} = 3.6 \text{ km/h}$ [29]. In addition to speed analysis, team movement is visualized using heatmaps. The heatmaps are generated by recording the frequency of player position occurrences at each frame based on the extracted coordinates. Player centroid positions at each frame are temporally accumulated to form a positional frequency map. This visualization is

then represented in two-dimensional form to illustrate movement intensity and spatial dominance by a team throughout the match video.

3. Results and Discussions

The experiments were conducted on an ASUS F16 equipped with an Intel Core i7 CPU, 32 GB of RAM, and an NVIDIA RTX 5060 GPU running Windows 11. All experiments were implemented using the Python programming language and executed in the JupyterLab environment.

3.1 Object Detection using YOLOv8n

The best results in this training were achieved using 50 epochs, a batch size of 6, and rect=True to ensure that

images were preserved according to the 1920×1080 pixel aspect ratio. HSV color augmentation was applied to the model with parameters $hsv_h = 0.3$, $hsv_s = 0.5$, and $hsv_v = 0.9$. The model summary consists of 129 layers, 3,011,628 parameters, 3,011,612 gradients, and 8.2 GFLOPs. The YOLOv8n object detection model for Player, Goalkeeper, Referee, and Ball achieved a precision of 86.2%, a recall of 81.7%, mAP@50 of 84.2%, and mAP@50–95 of 54.6%. The complete training results are shown in Figure 3. The trained YOLOv8n model was saved as weights/best.pt and used for object detection, which subsequently produces bounding boxes, confidence scores, and class labels for each frame and then serves as input for the object tracking process.

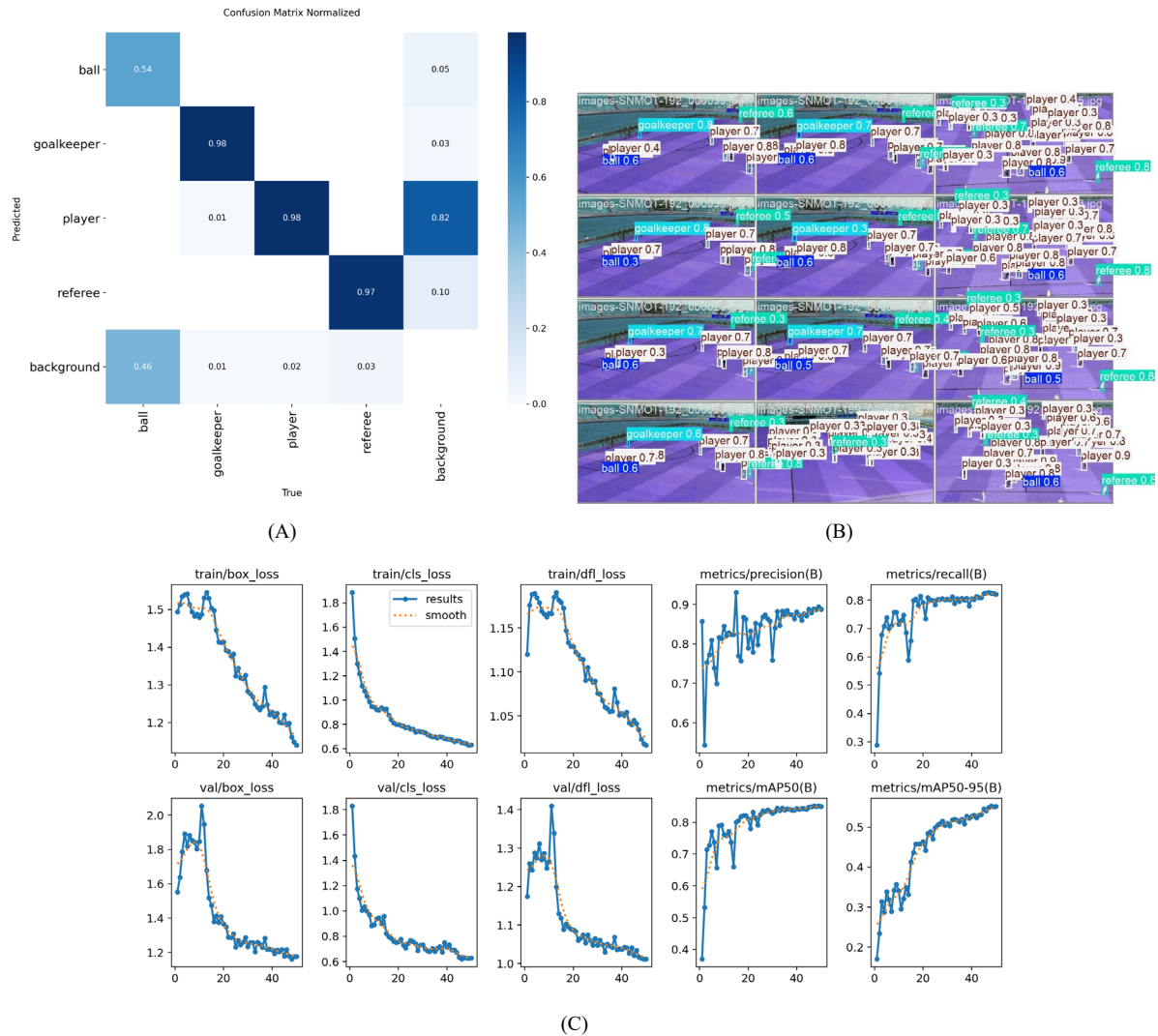


Figure 3. (A) Confusion matrix, (B) Training validation batch, (C) Training result matrix of the YOLOv8n training results

3.2 Deduplicate Detection

The detected bounding boxes were filtered using $conf_threshold = 0.1$, with the objective of retaining all bounding box results for detected objects. At this stage, the deduplicate_detections function re-filters YOLO detection results that still contain duplicate bounding boxes due to low confidence selection and can reduce object trajectory detection errors that will later be

processed by ByteTrack. The results are shown in Figure 4.

The parameters used are $iou_threshold = 0.3$, $min_center_dist = 10$, and $shrink_ratio = 0.6$. The first parameter, $iou_threshold = 0.3$, functions as an overlap filter, where only detections with an overlap of $\geq 30\%$ are further compared using the center distance parameter min_center_dist to determine whether they

are duplicates or not. The shrink_ratio parameter is used to reduce the original bounding box size with the aim of enforcing stricter overlap filtering between bounding boxes. The deduplicate detection results are as follows: the total number of YOLOv8n detections was 38.922 detections, while the total number after deduplication was 14.630 detections.

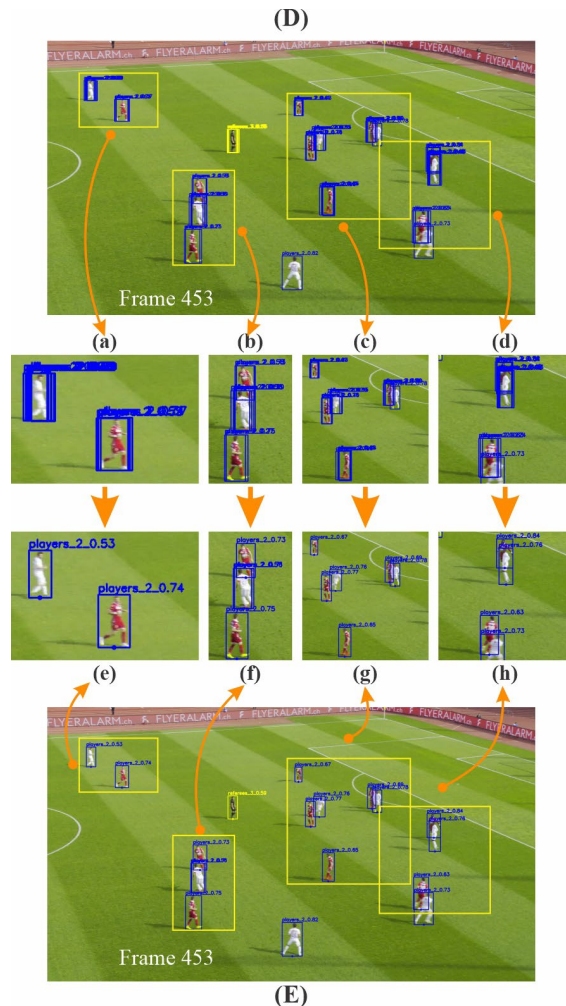


Figure 4. (D); YOLOv8n results, where markers (a), (b), (c), and (d) still contain double bounding boxes. (E); Deduplicate Detection results at markers (e), (f), (g), and (h), where no double bounding boxes are present

3.3 ByteTrack with Embedding-Based Re-ID and Low-Confidence Integration

The default parameters of ByteTrack were maximized and detections were separated according to the Player, Goalkeeper, and Referee classes. The parameters for each class are shown in Table 1.

Table 1. Parameter ByteTrack parameters for each class

Parameter	Player	GK	Referee
track_activation_threshold	0.1	0.1	0.1
lost_track_buffer	300	100	100
minimum_matching_threshold	0.9	1.0	1.0
minimum_consecutive_frames	1	1	1
frame_rate (fps)	25	25	25

Object tracking was performed using two stages, namely the ByteTrack + Re-ID object tracking stage

and the Track ID refinement stage. First, deduplicated bounding boxes were used to extract Re-ID embeddings. The extraction process was conducted using a combination of models: OSNet (512-D) and ResNet-50-based ColorEmbedding (128-D). Each object was preprocessed (conversion to PIL, resizing to a fixed resolution of 256×128 pixels), extracted in batches, and then the embeddings were concatenated and normalized into a 640-D vector as the basis for maintaining identity consistency across frames, performing cross-frame matching, and selectively updating the gallery.

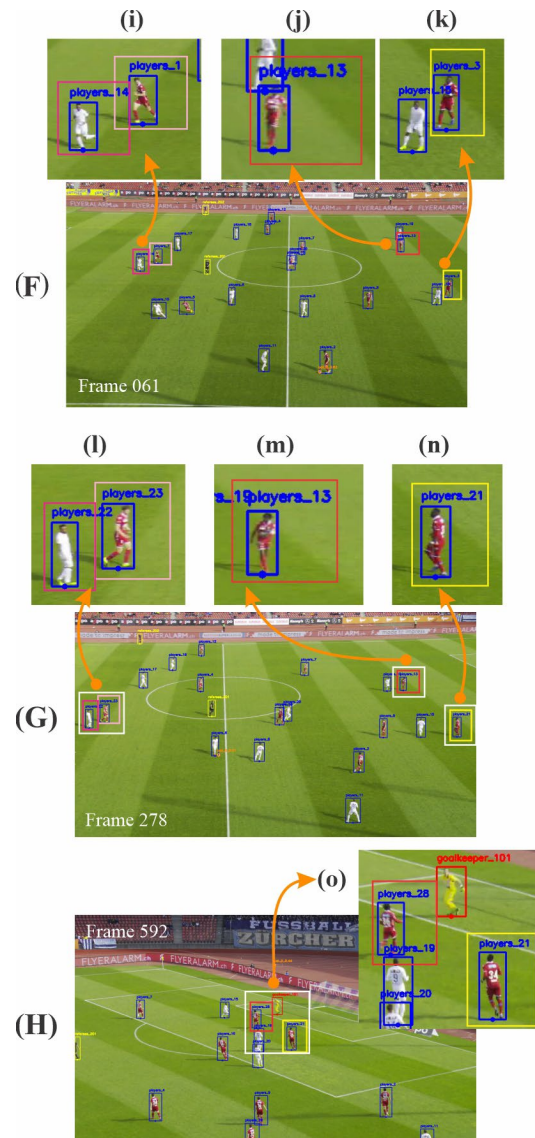


Figure 5. (F); Frame 61, Track IDs were still consistent for all Player objects. (G); Frame 278, three Player Track IDs began to change as indicated in (l) and (n). (H); Frame 592, at marker (o), two additional Player Track IDs were detected to have changed

In this experiment, tracking stability was evaluated at frame 15 by calculating the average number of detections. The system recorded an average of 20.1 detections, exceeding the predefined stability threshold of 15. Once the stability condition was met, the minimum consecutive frames mechanism was activated, requiring at least 2 consecutive frames for

players and 1 frame for goalkeepers and referees. The feature extraction process successfully generated embeddings for each detection, which represent object appearance characteristics, including color, texture, and clothing patterns, and were used as a reference to prevent identity switching between objects with different appearances, without directly overriding the tracking process. Identity consistency was primarily reinforced through the proposed layered recovery strategy, combined with a low-confidence-based fallback mechanism. In addition, Re-ID gallery updates were limited to high-confidence tracks, while the state preservation mechanism (snapshots) was verified to be updated periodically every 3 frames throughout the 750-frame sequence. Ball detection was taken directly from the YOLOv8n output. Ball detections were set with a confidence threshold of ≥ 0.5 to reduce ball detections outside the field, and were then stored in `frame_tracks["ball"]` with bounding box information, center position, confidence, and label. Tracks for Player and Goalkeeper stored bounding box, class label, as well as center position and foot position, whereas Referee tracks only stored bounding box and class label. The results can be seen in Figure 5.

The ByteTrack and Re-ID stage produced 11,694 Player detections (averaging 15.59 per frame), 233 Goalkeeper detections (averaging 0.31 per frame), 1,063 Referee detections (averaging 1.43 per frame), and 738 Ball detections (averaging 0.98 per frame). In total, this stage resulted in 40 unique Track IDs, comprising 35 Player Track IDs, 1 Goalkeeper Track ID, 3 Referee Track IDs, and 1 Ball Track ID.

3.4 Feature-Position Fusion for Identity Refinement

These results indicate that Player Track IDs require refinement because duplicate Track IDs were detected, whereas in football, each team consists of only 10 players and 1 goalkeeper. Therefore, the second stage, Track ID Refinement, was applied. All Track IDs generated by ByteTrack + Re-ID were reanalyzed to reduce remaining duplicate IDs. To proceed with the Track ID comparison and merging process, anchor Track IDs were first determined, defined as track IDs that appeared earliest and were considered valid detections for all Player objects. The Track ID refinement process utilized ResNet-50-based ColorEmbedding (128-dimensional) to evaluate player color and texture similarity.

All non-anchor Track IDs were first compared with one another to identify potential merging using the *first* mode for new non-anchor Track IDs, defined as the position from the initial crop when the Track ID first appeared along with the first 20 bounding box crops. For previous Track IDs, the *last* mode was used, defined as the position from the final crop before the Track ID disappeared along with the last 20 bounding box crops. The results were then compared with the anchor Track IDs as the primary identities using the *first* mode for non-anchor Track IDs and the *last* mode for anchor Track IDs. From these bounding box crops, color

embedding extraction was performed while simultaneously recording player positions (bounding box, center point, and foot position).

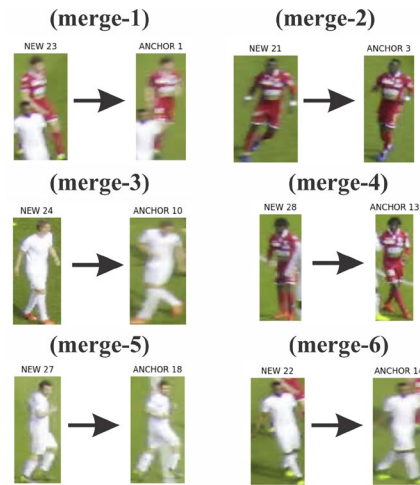


Figure 6. (merge-1) 23 $\rightarrow$ 1 (sim=0.974, dist=22.5), (merge-2) 21 $\rightarrow$ 3 (sim=0.970, dist=22.4), (merge-3) 24 $\rightarrow$ 10 (sim=0.970, dist=33.1), (merge-4) 28 $\rightarrow$ 13 (sim=0.986, dist=4.5), (merge-5) 27 $\rightarrow$ 18 (sim=0.987, dist=6.3), (merge-6) 22 $\rightarrow$ 14 (sim=0.970, dist=22.4)

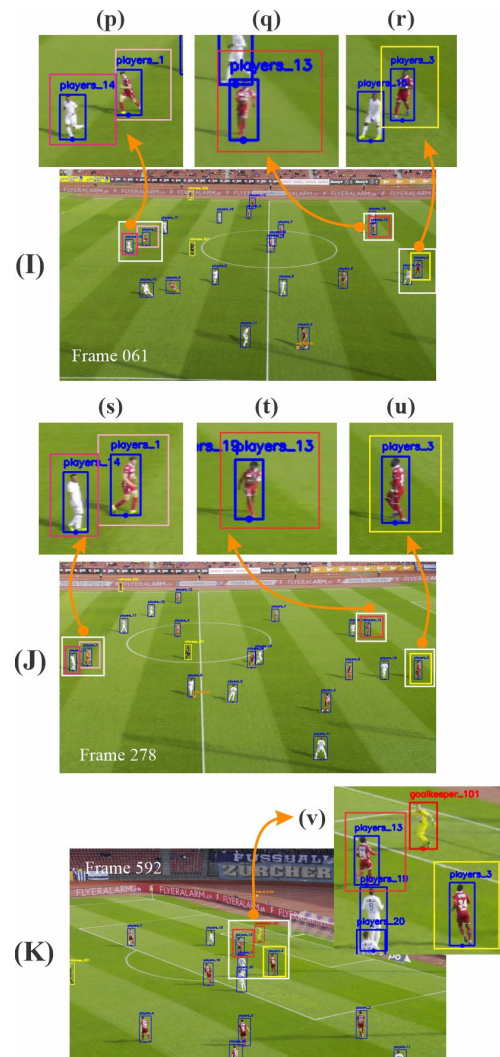


Figure 7. (I); Frame 61 with (p)(q)(r), (J); Frame 278 with (s)(t)(u), (K); Frame 592 with (v), all detected with the same Track ID

The averaged embeddings and positions were then compared using two measures: cosine similarity for color similarity and Euclidean distance for positional proximity. Two Track IDs were considered to belong to the same player if the similarity was $\geq 90\%$ ($\text{sim_threshold} = 0.90$) and the center position distance between them was ≤ 60 pixels ($\text{dist_threshold} = 60$). Several Track IDs that were successfully merged and represent the same object are shown in Figure 6. Subsequently, as a validation step and for outlier removal, tracks with a length of fewer than 20 crops/frames (≈ 0.8 seconds at 25 FPS) as well as track detections located outside the field based on the foot position along the Y-axis ($\text{Y_FIELD_MIN} = 141$ and $\text{Y_FIELD_MAX} = 1015$) were filtered and discarded. This was done to ensure that only consistent and stable player identities were retained. The results of the refinement stage yielded a total of 25 Track IDs, consisting of 21 Player Track IDs, 1 Goalkeeper Track ID, 2 Referee Track IDs, and 1 Ball Track ID. The results of Feature–Position Fusion for Identity Refinement are presented in Figure 7.

3.5 K-Means Clustering for Team Identification

After the Track ID refinement process, the next stage was color clustering to determine each player's team based on jersey color. Feature extraction was performed using ResNet-50, which produced a 128-dimensional color embedding for each player crop. All player crops were processed using the preprocess function (resizing to a height of 300 px while preserving the aspect ratio). For each player (Track ID), a maximum of 20 crops were selected, and embeddings were then extracted in batches of 5 crops ($\text{batch_size} = 5$). The embeddings in each batch were summarized using the median. The final color embedding for each player was obtained by averaging the batch-wise median embeddings, resulting in a single 128-dimensional representation per player.

Features from all players were then clustered using MiniBatchKMeans into two clusters ($n_clusters = 2$) corresponding to the number of teams. The clustering process produced two clusters based on the distribution of player features. After clustering, semantic team labels were assigned based on the predefined anchor selection strategy. In this experiment, Track IDs 4 and 9 were selected as anchors for Team_0, whereas Track IDs 15 and 19 were selected as anchors for Team_1. The details of the selected anchor Track IDs are presented in Figure 8.



Figure 8. (w) Anchor Track IDs manually selected as the labeling reference for Team_0; (x) Anchor Track IDs manually selected as the labeling reference for Team_1

The clustering results were assigned with team labels, followed by the use of the `determine_goalkeeper_team` function to assign the goalkeeper (GK) team based on positional proximity to players at the time of first detection. The final output was `team_map_clustered`, which maps each Track ID to its assigned team along with the detected GK team result. The team clustering results are shown in Figure 9.

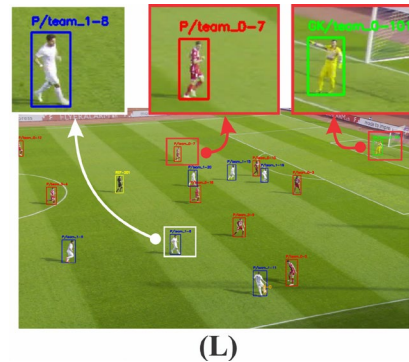


Figure 9. (L) K-Means identified two teams, namely Red Bounding Boxes (Team_0) and Blue Bounding Boxes (Team_1), and the `determine_goalkeeper_team` function successfully identified the goalkeeper with the correct team, namely Team_0

3.6 Evaluation Metrics for Multi-Object Tracking

The overall results from the combination of the YOLOv8n method, ByteTrack with the integration of Re-ID embedding and low-confidence detection, as well as ColorEmbedding ResNet-50 and K-Means clustering, produced a total of 13,652 object detections with an average of 18.21 objects per frame. The complete evaluation metrics show HOTA: 91.9%, DetA: 90.3%, DetRe: 94.2%, DetPr: 95.7%, AssA: 93.5%, AssRe: 94.8%, AssPr: 96.3%, LocA: 81.0%, and OWTA: 93.9%. The evaluation also reports MOTA: 89.8%, MOTP: 81.0%, and IDF1: 94.9%, with 17 identity switches and 196 trajectory fragmentations. These results suggest good tracking performance with relatively consistent player identity preservation in the evaluated match sequence. Object identity consistency is reflected by the high IDF1 value of 94.9%. Supported by high association metrics (AssA 93.5%, AssRe 94.8%, AssPr 96.3%), as well as the relatively low number of identity switches (17) and trajectory fragmentations (196), these results suggest that trajectories are formed with relatively few association errors and generally consistent identity preservation throughout the evaluated sequence. The MOT evaluation metric curves are presented in Figure 10.

The ablation study and cross-match evaluation results are summarized in Table 2. The first four experiment columns represent the ablation configurations evaluated on Match 1, while the last two columns present cross-match evaluations on Match 2 and Match 3.

The ablation study is conducted in a cumulative manner, where each component is incrementally integrated into the baseline model (YOLOv8n + ByteTrack), which serves as the reference without additional enhancements. The integration of detection

deduplication significantly improves tracking performance, increasing HOTA from 0.327 to 0.815 and MOTA from -0.147 to 0.886, while substantially reducing identity switches, false positives, and trajectory fragmentations. The addition of Re-ID further enhances association metrics, particularly AssA (0.869) and IDF1 (0.950), indicating improved identity consistency across frames. The subsequent integration of Feature-Position Fusion, as part of the overall framework, refines identity assignment and achieves the best overall performance, with HOTA reaching 0.919 and IDF1 0.949, while reducing IDSW to 17 and maintaining a relatively low number of fragmentations. Cross-match evaluation on Match 2 and Match 3 shows lower performance (HOTA 0.682 and 0.598, respectively), indicating reduced tracking accuracy under different match conditions.

This performance gap is likely related to the detector training configuration, as YOLOv8n was trained only on data from the first match without exposure to Match 2 and Match 3 sequences. Several failure cases were observed when players temporarily left the frame and later reappeared under different lighting conditions. These variations in visual appearance can reduce feature consistency, leading to identity switches and short trajectory fragmentations, particularly when temporal association is disrupted.

To estimate uncertainty, the 750 frames are divided into five non-overlapping temporal segments and evaluated independently. Confidence intervals are computed using bootstrapping (2,000 iterations) over segment-

level results, and are reported as the mean and 95% confidence intervals for HOTA, MOTA, and IDF1.

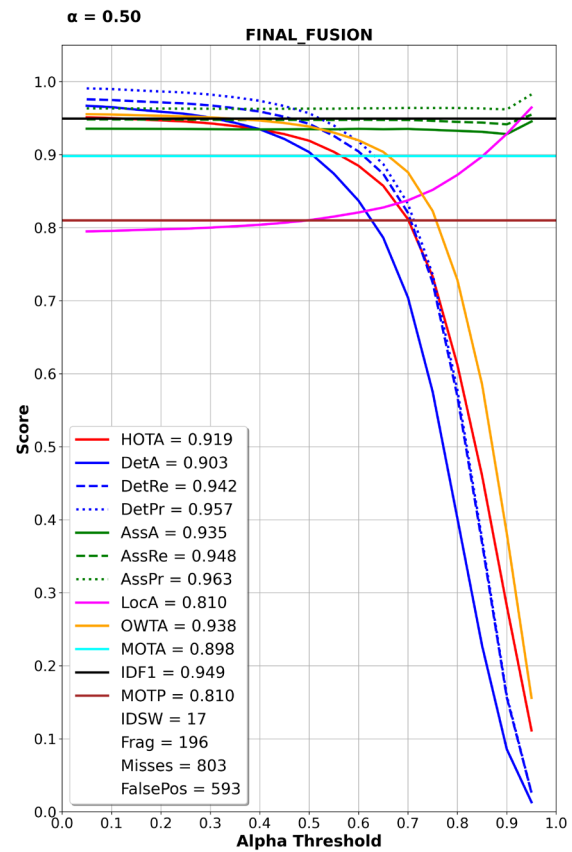


Figure 10. Evaluation Metric Curves

Table 2. The ablation study and cross-match evaluation results

Experiment	HOTA	DetA	DetRe	DetPr	AssA	AssRe	AssPr	LocA	MOTA	IDF1	IDSW	Frag	Misses	FP
Baseline	0.327	0.517	0.952	0.531	0.206	0.209	0.918	0.827	-0.147	0.682	3570	282	672	11654
+Deduplicate	0.815	0.900	0.948	0.947	0.738	0.756	0.946	0.810	0.886	0.947	116	195	727	737
+ Re-ID	0.887	0.905	0.946	0.955	0.869	0.873	0.987	0.810	0.898	0.950	43	194	747	625
+Feature Fusion (Proposed)	0.919	0.903	0.942	0.957	0.935	0.948	0.963	0.810	0.898	0.949	17	196	803	593
Match 2	0.682	0.739	0.810	0.894	0.630	0.705	0.837	0.813	0.690	0.850	320	412	2578	1296
Match 3	0.598	0.701	0.774	0.882	0.511	0.564	0.725	0.831	0.615	0.824	723	469	2996	1371

Table 3. Mean and 95% confidence intervals of HOTA, MOTA, and IDF1 across ablation stages and cross-match evaluation

Experiment	HOTA mean	HOTA ci low	HOTA ci high	MOTA mean	MOTA ci low	MOTA ci high	IDF1 mean	IDF1 ci low	IDF1 ci high
Baseline	0.425	0.371	0.478	-0.109	-0.328	0.111	0.693	0.649	0.737
+Deduplicate	0.788	0.760	0.812	0.884	0.861	0.905	0.946	0.938	0.955
+ Re-ID	0.807	0.786	0.823	0.898	0.885	0.910	0.950	0.944	0.956
+Feature Fusion (Proposed)	0.812	0.794	0.829	0.898	0.887	0.910	0.949	0.944	0.955
Match 2	0.691	0.664	0.708	0.693	0.656	0.731	0.850	0.823	0.876
Match 3	0.650	0.602	0.685	0.613	0.552	0.671	0.823	0.786	0.858

Table 4. Performance comparison with selected references in player tracking and re-identification using standard evaluation metrics

Reference	Dataset	HOTA	MOTA	IDF1	IDSW	FPS
Song et al. (2021)	ISSIA	—	—	—	—	6.5
Naik et al. (2022)	ISSIA	—	—	0.94	—	23
Naik et al. (2024)	ISSIA	—	0.88	0.91	—	6.56
Sun et al. (2024)	SoccerNet	0.831	0.88	0.846	2188	—
Proposed (Ours)	SoccerNet	0.919	0.898	0.949	17	9

As shown in Table 3, the improvement from the baseline to the deduplication stage is clearly reflected in both the mean values and the confidence ranges, while

the gains from Re-ID and feature fusion are more incremental, with partially overlapping intervals. The evaluation on Match 2 and Match 3 shows higher

variability, indicating reduced stability under different match conditions. Table 4 shows that the proposed method achieves competitive results, with high HOTA, MOTA, and IDF1 values and a low number of identity switches (IDSW), reflecting good identity consistency and overall tracking performance

3.7 Speed and movement analysis of players based on the model results

The consistency of player identities produced by the Re-identification (Re-ID) system becomes an important foundation in conducting match analysis. This process also aims to validate the robustness of the Re-ID results when applied to match analysis, such as speed estimation and team movement patterns. Furthermore, camera motion compensation using the Optical Flow method is applied to reduce bias caused by camera movement during match videos. This step aims to ensure that changes in player positions represent the actual movement of players on the field, rather than being caused by shifts in the camera viewpoint. With the presence of camera motion compensation, the results of speed estimation and player movement visualization become more stable and accurate. Player speed calculation is performed by converting the scale from pixels to meters. Since the video frames do not display the entire field, the calculation is based on the size of the center circle of the field. The center circle has a diameter of 18.3 m. Therefore, one frame that displays the center circle line of the field is selected, then 2 (two) straight lines are drawn from the farthest to the nearest sideline forming the perspective of the central part of the field, and then the length of the nearest sideline is taken as 1,385 pixels. The video has a resolution of 1920×1080 pixels with a frame rate of 25 fps. Speed is initially calculated in meters per second (m/s) and then converted to kilometers per hour (km/h) using the SI conversion factor of 3.6. The speed estimation, cumulative distance, and average speed calculations are formulated in Equations 14–18.

Next, the analysis of player movement within each team during the match is conducted. In this process, the centroid positions of players in each frame are accumulated into a 2D array according to the frame size, thereby producing a visual representation of the field areas that are most frequently occupied. This heatmap identifies the dominant areas of each team. In detail, areas with high intensity indicated by red color represent zones where players are most frequently present on the field, green color indicates areas where players are sufficiently often present, and greenish-blue color indicates low intensity where players are rarely present on the field. This enables the identification of the positions and movements of each player during the match. The corresponding equations are presented as Equations 14–18:

Scale Conversion (meters/pixel):

$$Scale = \frac{D_{field}}{D_{frame}} \quad (14)$$

Speed in km/h :

$$v_{km/h} = d_{px} \times S \times F \times 3.6 \quad (15)$$

Distance Estimation (meters) :

$$d_m^{(i)} = d_{px}^{(i)} \times S \quad (16)$$

Cumulative Distance:

$$D_{total} = \sum_{i=1}^N d_m^{(i)} \quad (17)$$

Average Speed :

$$\bar{v} = \frac{1}{N} \sum_{i=1}^N v_{km/h}^{(i)} \quad (18)$$

D_{field} denotes the actual reference distance on the field, D_{frame} represents the corresponding distance in pixels, $d_{px}^{(i)}$ is the pixel displacement at frame i , F is the video frame rate (25 fps), and N is the number of valid frames used for distance and speed estimation.

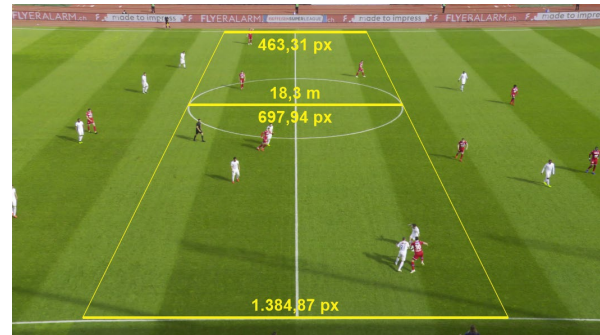


Figure 11. Perspective-based measurement of the center field derived from the total diameter of the center circle

Table 5. Speed and distance analysis data for team_0

Track ID	Role	Top Speed (km/h)	Avg Speed (km/h)	Total Distance (m)	Active Frame
1	players	15.06	5.74	18.29	241
2	players	14.1	4.24	33.29	631
3	players	13.95	4.73	39.52	717
4	players	13.26	6.1	45.0	619
5	players	22.1	5.93	16.38	198
7	players	17.45	5.88	48.4	737
9	players	11.31	4.84	38.48	668
12	players	16.85	6.36	37.43	516
13	players	17.42	5.02	42.32	724
16	players	14.04	4.03	33.52	749
26	players	12.75	7.38	17.25	185

Table 6. Speed and distance analysis data for team_1

Track ID	Role	Top Speed (km/h)	Avg Speed (km/h)	Total Distance (m)	Active Frame
6	players	13.32	5.83	38.18	576
8	players	14.35	5.09	33.42	569
10	players	18.67	6.24	12.59	149
11	players	23.12	6.09	39.93	580
14	players	16.12	6.08	20.78	257
15	players	18.08	5.78	48.13	749
17	players	13.89	6.24	22.32	308
18	players	14.18	6.32	32.38	432
19	players	20.51	5.64	46.91	749
20	players	14.62	4.89	40.69	742

Based on Equation 14, the calculated scale factor is 0.0132 m/pixel, which yields a speed conversion of approximately 1.1891 km/h per pixel using Equation 15. This conversion factor is then applied to estimate player speed, while Equations 16–18 are used to calculate the traveled distance, cumulative distance, and

average speed of each player during the match. The acquisition of the center circle size is shown in Figure 11. The complete results of the speed estimation are presented in Tables 5 and 6. The visualization results of the movement patterns of each team are shown in Figures 12 and 13.

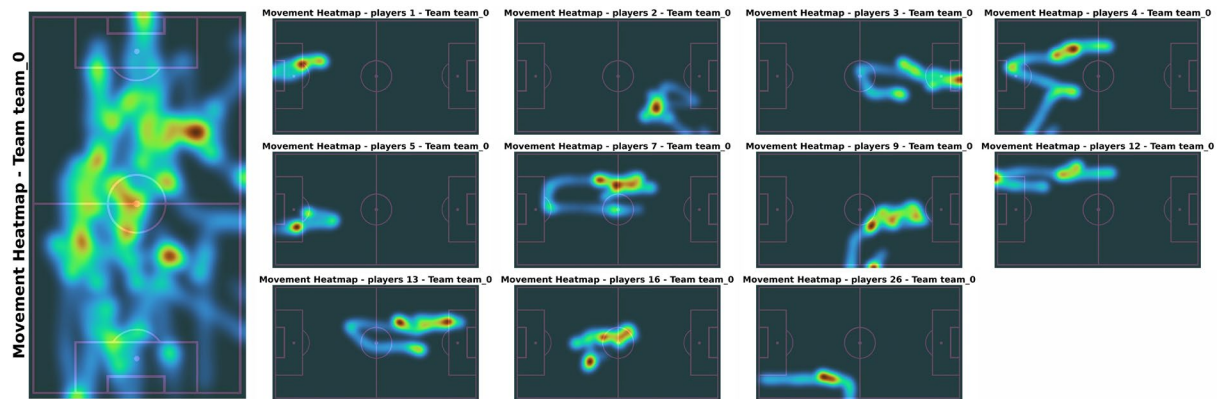


Figure 12. Player movement heatmap of Team 0

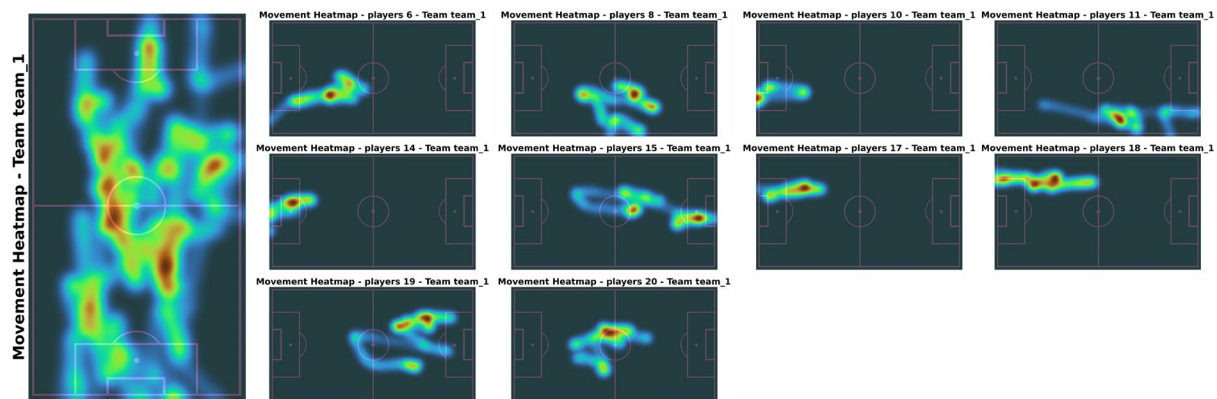


Figure 13. Player movement heatmap of Team 1

4. Conclusions

This study demonstrates that the proposed multi-method approach for tracking and re-identification is able to effectively maintain object Tracking ID consistency in video-based soccer match analysis. The integration of ByteTrack with re-identification embeddings, the utilization of low-confidence detections, and the application of Feature–Position Fusion for Identity Refinement based on ColorEmbedding ResNet-50 are proven to reduce the occurrence of identity switches, as well as to improve and maintain object identity consistency across frames, even under conditions of occlusion, changes in movement direction, and temporary detection loss.

The resulting identity consistency enables the extraction of player speed estimation data and movement patterns to be performed in a more stable and precise manner, thereby addressing the primary objective of this study in providing a reliable foundation for performance analysis and team dynamics. The advantages of this approach are reinforced by the achieved tracking evaluation metrics, including HOTA, MOTA, and IDF1, as well as the reduced number of identity switches (IDSW) and track fragmentations,

which indicate that the system is not only effective in preserving object identities but also in maintaining temporal continuity for advanced match analysis applications.

These findings are further supported by the ablation study results, where the overall pipeline—consisting of detection deduplication, re-identification, and Feature–Position Fusion—operates in an integrated manner to progressively improve performance. The integration of detection deduplication provides a significant initial

contribution by reducing false positives and identity switches through the utilization of low-confidence detections, which is then reinforced by re-identification to enhance association consistency across frames. Subsequently, Feature–Position Fusion refines the identity assignment process, resulting in the best overall performance. However, cross-match evaluation reveals a performance decline under different match conditions, indicating that generalization remains a challenge, particularly when the detector is trained on limited match variations or only a single match. In addition to the ablation analysis, the comparative results in Table 4 further demonstrate the effectiveness of the proposed method. Compared to prior approaches such as Naik et

al. (2022, 2024) and Sun et al. (2024), the proposed method achieves higher IDF1 and significantly fewer identity switches, indicating improved identity consistency. This improvement is mainly attributed to the integration of layered recovery and low-confidence detection handling, which enhances the robustness of identity association across frames. However, the method operates at a moderate processing speed (9 FPS), indicating a trade-off between accuracy and real-time performance.

Nevertheless, this study still has limitations, particularly in fully exploiting the complexity of soccer matches, as the analysis remains focused on moving objects such as players, goalkeepers, referees, and the ball, without integrating static field elements such as field lines and goalposts, along with limitations in distance scale estimation and main camera positioning. Furthermore, the speed estimation process in this study is based on the assumption of camera perspective consistency, utilizing the pitch's center circle as the reference for pixel-to-meter conversion. This assumption may introduce errors due to camera perspective distortion, particularly in regions of the pitch located farther from the reference point. In addition, this study has not conducted a sensitivity analysis, error quantification, or validation against ground-truth player motion; therefore, the error bounds of the estimated speed values cannot yet be quantitatively determined. Future work is expected to further develop this system by integrating field spatial context and comprehensive camera calibration methods (e.g., homography transformation), as well as incorporating ground-truth-based validation, in order to support more comprehensive and accurate tactical analysis of soccer matches. Additionally, semantic team labeling relies on a predefined anchor selection during post-labeling. While it does not affect the clustering process, it limits full automation and could be improved in future work.

Acknowledgements

The authors declare that there is no conflict of interest in this study. This research was conducted without any specific funding support from public, commercial, or non-profit funding agencies.

References

- [1] UEFA, "UEFA EURO 2024: Record Attendance and Fan Engagement Figures," Jul. 2024. [Online]. Available: <https://www.sportspro.com/insights/analysis/euro-2024-stats-tv-ratings-viewership-attendance-social-media-sales/>
- [2] S. Plakias, "Review Articles on Soccer Performance Analysis: A Bibliometric Analysis of Current Trends and Emerging Themes," *Sports*, vol. 13, no. 5, p. 131, 2025, doi: [10.3390/sports13050131](https://doi.org/10.3390/sports13050131).
- [3] C. Huang, M. Zhang, and C. Li, "Tactical Analysis and Performance Metrics," *Psychol. Sport Exerc.*, vol. 73, p. 102695, 2024, [Online]. Available: <https://ssrn.com/abstract=4758324>
- [4] S. Plakias, T. Tsatalas, V. Armatas, D. Tsaopoulos, and G. Giakas, "Tactical Situations and Playing Styles as Key Performance Indicators in Soccer," *JFMK*, vol. 9, no. 2, p. 88, May 2024, doi: [10.3390/jfmk9020088](https://doi.org/10.3390/jfmk9020088).
- [5] B. Mahaseni, E. R. M. Faizal, and R. G. Raj, "Spotting football events using two-stream convolutional neural network and dilated recurrent neural network," *IEEE Access*, vol. 9, pp. 61929–61942, 2021. doi: [10.1109/ACCESS.2021.3074831](https://doi.org/10.1109/ACCESS.2021.3074831).
- [6] F. Zheng, D. Z. Al-Hamid, P. H. J. Chong, C. Yang, and X. J. Li, "A Review of Computer Vision Technology for Football Videos," *Information*, vol. 16, no. 5, p. 355, Apr. 2025, doi: [10.3390/info16050355](https://doi.org/10.3390/info16050355).
- [7] Y. Sun, Z. Sun, and W. Chen, "The evolution of object detection methods," Jul. 01, 2024, *Elsevier Ltd.* doi: [10.1016/j.engappai.2024.108458](https://doi.org/10.1016/j.engappai.2024.108458).
- [8] E. Edozie, A. N. Shuaibu, U. K. John, and B. O. Sadiq, "Comprehensive review of recent developments in visual object detection based on deep learning," *Artif. Intell. Rev.*, vol. 58, no. 9, Sep. 2025, doi: [10.1007/s10462-025-11284-w](https://doi.org/10.1007/s10462-025-11284-w).
- [9] Y.-L. Chen, K.-C. Hung, J.-Y. Zhang, and L.-W. Lin, "Improved YOLOv8n Models for Object Detection in Remote Sensing Images," *IEEE Access*, vol. 13, pp. 97889–97900, 2025, doi: [10.1109/ACCESS.2025.3574856](https://doi.org/10.1109/ACCESS.2025.3574856).
- [10] E. S. d. Santos Júnior, T. Paixão, and A. B. Alvarez, "Comparative Performance of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for Layout Analysis of Historical Documents Images," *Applied Sciences*, vol. 15, no. 6, p. 3164, 2025, doi: [10.3390/app15063164](https://doi.org/10.3390/app15063164).
- [11] Y. Wang and V. Y. Mariano, "A Multi Object Tracking Framework Based on YOLOv8s and Bytetrack Algorithm," *IEEE Access*, vol. 12, pp. 120711–120719, 2024, doi: [10.1109/ACCESS.2024.3450370](https://doi.org/10.1109/ACCESS.2024.3450370).
- [12] L. You, Y. Chen, C. Xiao, C. Sun, and R. Li, "Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack," *Electronics (Switzerland)*, vol. 13, no. 15, Aug. 2024, doi: [10.3390/electronics13153033](https://doi.org/10.3390/electronics13153033).
- [13] J. Cao, J. Pang, X. Weng, R. Khirrodar, and K. Kitani, "Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking," Mar. 2023, [Online]. Available: <http://arxiv.org/abs/2203.14360>
- [14] V. Stanojević and B. Todorović, "BoostTrack++: using tracklet information to detect more objects in multiple object tracking," Aug. 2025, [Online]. Available: <http://arxiv.org/abs/2408.13003>
- [15] J. Sun, H. W. Huang, C. Y. Yang, Z. Jiang, and J. N. Hwang, "GTA: Global Tracklet Association for Multi-Object Tracking in Sports," in *Proceedings of the Asian Conference on Computer Vision*, 2024, pp. 421–434. doi: [10.1007/978-981-96-2644-1_6](https://doi.org/10.1007/978-981-96-2644-1_6).
- [16] B. T. Naik, M. F. Hashmi, Z. W. Geem, and N. D. Bokde, "DeepPlayer-Track: Player and Referee Tracking With Jersey Color Recognition in Soccer," *IEEE Access*, vol. 10, pp. 32494–32509, 2022, doi: [10.1109/ACCESS.2022.3161441](https://doi.org/10.1109/ACCESS.2022.3161441).
- [17] B. Thulasya Naik, M. F. Hashmi, and A. Gupta, "EIoU-distance loss: an automated team-wise player detection and tracking with jersey colour recognition in soccer," 2024, *Taylor and Francis Ltd.* doi: [10.1080/09540091.2023.2291991](https://doi.org/10.1080/09540091.2023.2291991).
- [18] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning Generalisable Omni-Scale Representations for Person Re-Identification," Apr. 2021, [Online]. Available: <http://arxiv.org/abs/1910.06827>
- [19] H. J. Mohammed et al., "ReID-DeepNet: A Hybrid Deep Learning System for Person Re-Identification," *Mathematics*, vol. 10, no. 19, Oct. 2022, doi: [10.3390/math10193530](https://doi.org/10.3390/math10193530).
- [20] A. Tanapatpiboon, W. Kusakunniran, and W. Limroongreungrat, "Deep learning-based detection of players and teams in soccer videos with positioning heatmap generation," *ACI*, 2025, doi: [10.1108/ACI-07-2024-0257](https://doi.org/10.1108/ACI-07-2024-0257).
- [21] S. Chen, X. Ouyang, and F. Luo, "Ensemble One-Class Support Vector Machine for Sea Surface Target Detection Based on k-Means Clustering," *Remote Sens. (Basel)*, vol. 16, no. 13, p. 2401, 2024, doi: [10.3390/rs16132401](https://doi.org/10.3390/rs16132401).
- [22] S. Jardim, J. António, and C. Mora, "Graphical Image Region Extraction with K-Means Clustering and Watershed," *J. Imaging*, vol. 8, no. 6, Jun. 2022, doi: [10.3390/jimaging8060163](https://doi.org/10.3390/jimaging8060163).

- [23] A. Cioppa and others, "SoccerNet-Tracking: Multiple Object Tracking Dataset and Benchmark in Soccer Videos," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 3490–3501. doi: [10.1109/cvprw56347.2022.00393](https://doi.org/10.1109/cvprw56347.2022.00393).
- [24] Ultralytics, "Ultralytics YOLOv8 Docs: Models," 2023. [Online]. Available: <https://docs.ultralytics.com/models/yolov8/>
- [25] Roboflow, "Supervision 0.27.0 – Trackers Documentation," 2025. [Online]. Available: <https://supervision.roboflow.com/0.27.0/trackers/>
- [26] A. Alfarano, L. Maiano, L. Papa, and I. Amerini, "Estimating Optical Flow: A Comprehensive Review of the State of the Art," *Computer Vision and Image Understanding*, vol. 249, p. 104160, 2024, doi: [10.1016/j.cviu.2024.104160](https://doi.org/10.1016/j.cviu.2024.104160).
- [27] H. Yin, R. O. Sinnott, and G. T. Jayaputera, "A survey of video-based human action recognition in team sports," *Artif. Intell. Rev.*, vol. 57, no. 11, Nov. 2024, doi: [10.1007/s10462-024-10934-9](https://doi.org/10.1007/s10462-024-10934-9).
- [28] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *Computer Vision - ECCV 2004*, vol. 3024, T. Pajdla and J. Matas, Eds., in *Lecture Notes in Computer Science*, vol. 3024., Berlin, Heidelberg: Springer, 2004. doi: [10.1007/978-3-540-24673-2_3](https://doi.org/10.1007/978-3-540-24673-2_3).
- [29] Bureau International des Poids et Mesures, "The International System of Units (SI), 9th ed.," 2019, *Sèvres, France*. Available: <https://www.bipm.org/en/publications/si-brochure>