Semi-Automatic Estimation of Body Orientation in Football

Sobhaan Ul Husan
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
s.j.ul.husan@student.vu.nl

Mauricio Verano Merino
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
m.verano.merino@vu.nl

Elixabete Sarasola Nieto
Vantage
San Sebastián, Spain
eli@vantage.football

Abstract

Body orientation is a crucial component for tactical analysis in professional football. This allows coaching staff to obtain insights into players' attention, decision-making, and strategic positioning. Likewise, body orientation might be an indicator of tactical readiness in football. This paper presents a computer vision pipeline for automatic body orientation estimation during pass events using player pose estimation. To evaluate the aforementioned pipeline, we used a series of video clips from the Women's Super League match between Brighton and Aston Villa in the 24/25 season. To validate our results, we compare them against manual annotations by a professional football analyst. We achieve 75% accuracy on real-world match footage, demonstrating the potential to reduce manual annotation in performance analytics. However, the wrong body orientation estimations were due to issues in the player and ball detection and the pass detection events, and not to the body orientation model. Therefore, we believe that the proposed pipeline can assist coaching staff to semi-automatically determine body orientation in professional football. However, more research is required to improve the ball and players' detection.

CCS Concepts

• **Computing methodologies** → **Object identification**; **Tracking**; **Matching**; • **Applied computing**;

Keywords

Computer Vision, Pose Estimation, Football Analytics, Body Orientation, YOLO, Tracking-by-detection

ACM Reference Format:

Sobhaan Ul Husan, Mauricio Verano Merino, and Elixabete Sarasola Nieto. 2025. Semi-Automatic Estimation of Body Orientation in Football. In *Proceedings of the 8th International ACM Workshop on Multimedia Content Analysis in Sports (MMSports '25), October 27–28, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3728423.3759407

1 Introduction

Statistics in football are widely used to gather insights about players and teams, with the aim of improving the effectiveness of the play and win games. *Vantage* [22] is a company that specializes in 1-on-1 coaching. They evaluate their players by analyzing football matches and gathering statistics about the players in these matches. Currently, many of these statistics are collected manually by reviewing hours worth of footage. This process is cumbersome and can be automated. Gathering these statistics can take up to six hours per game [1]. This research aims to automate this process by proposing a pipeline that can take video footage as input and return the relevant statistics. Specifically, the *body orientation* of a target player at a pass event.

The estimation of player body orientation is a crucial component for tactical analysis in football, providing insights into player attention, decision-making, and strategic positioning [3]. For most scenarios, it is important for the player to be in an *open position*, since it has been discovered that players with open body shapes complete more progressive passes and resist pressure [5]. Furthermore, in the same study, it was found that players who receive on *half position* complete 16% more progressive passes than those who don't. This highlights the importance of the body orientation for coaches because, based on the data, they can train players on body positioning before receiving passes.

In addition to that, there does not seem to be any work that creates a pipeline that can detect, track, and analyze the players to provide body orientation statistics. This thesis aims to bridge the gap in detecting and tracking players, and providing body positioning statistics. This project provides a pipeline that can analyze football videos and gather statistics automatically with minimal human labor.

The contributions can be summarized as follows:

- Pipeline to detect body orientation in elite football players. The pipeline's novelty lies on using exclusively shoulder keypoints; avoiding hip detections in low-resolution frames.
- Validation with a professional video analyst as ground truth, instead of homography steps.
- The proposed pipeline has a 75% accuracy in real-world match footage.

The remainder of this paper is structured as follows: Section 2 presents the motivation and research question of this study. Followed by an overview of the existing literature and background (Section 3). Then, the pipeline for detecting body orientation statistics is presented in Section 4. To test the pipeline, we designed an experiment (Section 5). We analyze the experiment results (Section 6) and discuss the pipeline results (Section 7). Finally, Section 8 presents the conclusions of the paper.

2 Motivation

Currently, several companies are focused on the individual development of elite athletes. For example, *Vantage* [22] in The Netherlands focuses on offering an individual sports development program for elite athletes. In their case, most of the analysis is done manually because players often do not have access to their performance data. Therefore, external football coaches and analysts rely on broadcast video footage of a game in which the player played.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
MMSports '25, Dublin, Ireland
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1835-9/2025/10
https://doi.org/10.1145/3728423.3759407

This already introduces several challenges such as:

Data access Sometimes players do not have video footage, so coaches need to find their footage on platforms such as Wyscout [24] or YouTube.

Licensing rights Due to licensing rights of most football competitions, video footage might not always be publicly available.

Video Quality In many cases, coaches and players have to rely on low-resolution videos from platforms such as Wyscout.

Data Collection Coaches must watch the entire game footage to perform data collection. They must identify the relevant parts where their target player performed.

Manual Annotations Video analysts or coaches must manually annotate the relevant actions.

Video Annotations The original footage needs to be annotated on the basis of the collected data.

Currently, most of these challenges are addressed manually by coaching staff. However, this is a cumbersome activity that takes time and effort, distracting coaches from their expertise in guiding players to become better players. The aim of this paper is to introduce a computer vision pipeline to estimate body orientation metrics in professional football. This is relevant because this would reduce the time coaching staff spend tagging complete video footage. We envision this pipeline to help coaching staff, so that they can focus on the players' movements and positioning.

To address this problem, we have defined the following research question.

RQ: How can a computer vision model help to estimate the body orientation of a football player at a pass event?

3 Background and Related Work

3.1 Body Orientation

In the context of this paper, we define the body orientation of a football player as the orientation of the upper body in relation to the opponent's goal. Body orientation can be classified into three different categories: *open*, *closed*, and *half*. An *open* body position occurs when the player's upper body is facing the opponent's goal completely. A *closed* body position is when the player's upper body is facing his/her own goal completely, and a *half* body position is defined when the player is not facing their own goal or the opponent's goal, but something in between. Figure 1 visually illustrates the three categories of body position including the angles in degrees.

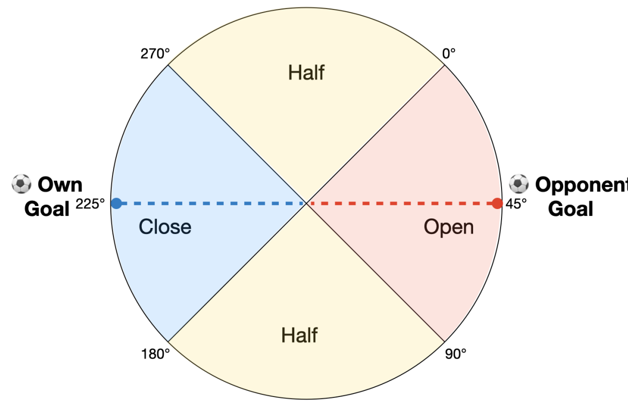


Figure 1: Visualisation of body orientation categories in degrees. The red dotted line shows the direction of the opposing goal, and the blue dotted line shows their own goal.

3.2 Detection and Tracking

This paper presents a computer vision pipeline to estimate body orientation. This task involves two main components: an *object detector* to identify objects and extract information (e.g., bounding boxes) and a component for *tracking* these objects. This type of systems are known as **Tracking-by-Detection** [2, 16].

One of the state-of-the-art detection models is *You Only Look Once* (YOLO) [18]. Recent implementations, exploring enhanced versions specifically adapted for sports scenarios. For example, the YOLOV8 model can accurately predict football players with a precision of 0.981 and a recall of 0.958 [9]. The YOLO architecture works by dividing the frame into a $S \times S$ grid and predicting any objects for each square in the grid. YOLO models process the image in one forward pass in a Convolutional Neural Network (CNN). During inference, it thresholds low confidence boxes and applies non-max suppression to remove duplicates, leading to fast and accurate detections. This differs from traditional pipelines, such as R-CNN variants, which use regional proposals.

To track detected objects, different techniques can be used, such as the *Simple Online and Real-Time Tracking* (SORT) algorithm [7], which combines a Kalman filter for motion prediction with a Hungarian function for data association and Intersection over Union (IoU) as a matching metric. One of SORT's limitations is the high identity switching due to a lack of appearance modelling. *DeepSORT* [23] solved this issue by integrating deep appearance descriptors using CNNs trained on person re-identification systems. However, the field reached new heights with *ByteTrack* [26], which used a hierarchical association strategy to improve tracking. This approach achieved 80.3% MOTA on MOT17 at 30 FPS, demonstrating that low-confidence detections contain valuable information to track continuity.

3.3 Human Pose Estimation

To approximate the body orientation, it is important to find the relevant keypoints of a player's bounding box, which is known as *human pose estimation*. Classical approaches to pose estimation are based on geometric calculations and feature-based approaches to estimate the pose of human subjects or objects [15]. These approaches often fail because of noise, occlusion, and outliers. Lowe [13]'s seminal work on the Scale-Invariant Feature Transform (SIFT) laid the groundwork for reliable key point detection and matching across different frames. SIFT works by detecting similar looking keypoints in different frames (e.g., a distinct looking wall) and using those keypoints to map the other points.

Another approach is the one used by *DeepPose* [19] where they employ CNNs to predict the joint positions of entities directly from an input image. While DeepPose performed well for single-person pose estimation, it did not work well for multi-object pose estimation. To address this limitation, *OpenPose* [8] was created. It popularized a robust, real-time, bottom-up framework that is able

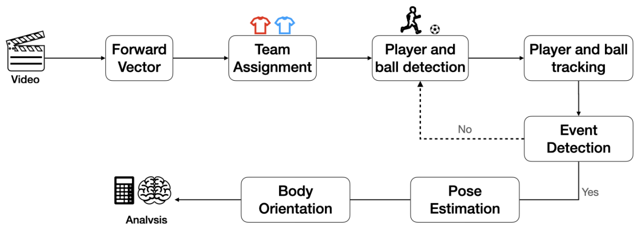


Figure 2: Overview of the body orientation pipeline.

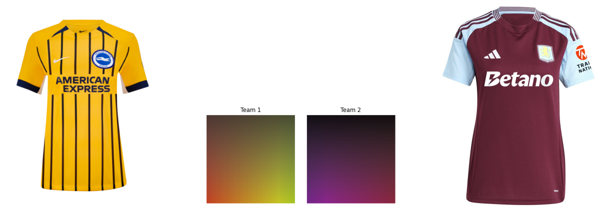


Figure 3: Left: Kit of team 1 with the predicted HSV values. Right: Kit of team 2 with the predicted HSV values.

Left: Kit of team 1 with the predicted HSV values. Right: Kit of team 2 with the predicted HSV values.

to estimate keypoints for multi-person pose estimation. Lastly, the latest pose estimation models are from YOLO. *YOLO pose* estimation models use a similar mechanism to their detection models. In this way, running the YOLO pose estimation models allows for a faster and more efficient running pipeline than using an OpenPose model.

3.4 Pose Estimation in Football

Pinheiro et al. [17] used body pose estimation (based on Open-Pose) with analysis for penalty kick strategies, achieving mean confidence scores of 0.80 ± 0.14. They rely on Part Affinity Fields (PAFs) for 2D pose estimation combined with direct linear transformation (DLT) for field mapping. One of their limitations is that there is not enough data to allow field mapping, and OpenPose estimation models would take a significantly longer time to run inference on compared to YOLO. Moreover, Arbués-Sangüesa et al. [4] adopted computer vision and deep learning techniques to create three vector probabilities that estimate the orientation of a player's upper-torso based on the positioning of the shoulder and hips. This approach provides a 92% accuracy in the left-right orientation directions.

4 Body Orientation Pipeline

An overview of the proposed pipeline is shown in Figure 2. The pipeline starts with the selection of a *forward vector* (Section 4.1) to determine the goal of the opposition team, and *team assignment* (Section 4.2). Then, for each frame, we detect and track the players and the ball. With the ball and players detection, we proceed to search for a *pass event*, which is an event where the possession of the ball changed from one player to another within the same team. If a pass event is detected, a pose estimation model is used to estimate the keypoints of the head and shoulders, which are then used to estimate the player's body orientation. Finally, all statistics are collected and exported into a data frame.

4.1 Forward Vector

Most available football video footage available comes from broadcast sources. Therefore, this type of footage often does not provide a full-view of the pitch and the camera is moving. Without a full-view of the pitch, it can be difficult to determine the direction of the opposing goal. Moreover, since the camera is moving, for every clip or change in camera angle, that direction can change. In this work, the direction of the opposing goal is represented using a *forward vector*. This vector is manually set by the user before running the pipeline. The user is prompted to move an arrow in the direction of the opposing goal. This forward vector is then used in combination with the body orientation categories (open, close, and half) to estimate the player's body orientation.

4.2 Team Assignment via HSV

To automatically find the HSV (Hue Saturation Value) ranges for both team jerseys, the process begins by automatically detecting the players in 60 random frames. Once the players are detected, the detections are cropped to represent only the middle of the jersey. This image of the cropped jersey is then processed by a K-means clustering algorithm to extract the two dominant colors per player crop. This list of colors is applied on a second k-means cluster to identify the color of the two team jerseys. From these two clusters, the system generates an HSV value range. As an example, Figure 3 presents the visualization of the HSV value ranges of two teams using this approach.

4.3 Players and Ball Detection and Tracking

For the detection of players and the ball in this pipeline, a trained YOLOv8 model [11] is used. Moreover, to track players and ball, we used a *Norfair tracker* [20]. This tracker combines elements of both SORT [7] and ByteTrack [26] algorithms. It maintains a collection of tracked objects, each representing a tracked entity (player or ball).

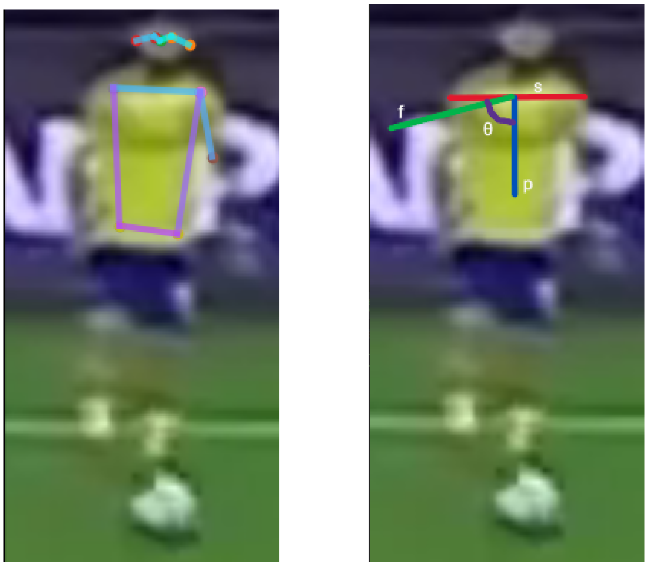
The tracker follows the next steps:

- Remove dead trackers (hit counter system),
- Predict new positions of the entities using the Kalman filter,
- Match detections to existing tracks, and
- Create new tracks for unmatched detections.

The hit counter system is one of the core elements. When an object matches a detection, the hit counter increases. When an object is not matched, the hit counter decreases. Once the hit counter reaches a certain threshold, the track is deleted. Tracks become active only after reaching an initialization threshold. Afterwards, a Kalman Filter is used to predict the new positions of the players and the ball. Then, a matching greedy algorithm is used to match these new positions with the detections. Finally, for the remaining unmatched detections, we create new tracks.

4.4 Pose Estimation

In this pipeline, pose estimation is built on *GluonCV* [10], which takes a custom-trained ResNet-152-v1d model to detect keypoints



(a) keypoints detected by Gluon. **(b) The vectors used in the body orientation calculation.**

Figure 4: Pose estimation and vectors used in calculating body orientation.

using data from the detected bounding boxes [25]. The system outputs 17 keypoints following the COCO format, with each keypoint represented as a triplet (`x`, `y`, `confidence`). For example, Figure 4a shows the relevant keypoints retained from this model.

4.5 Body Orientation Calculation

Using keypoints, it is possible to estimate the direction of the torso. To estimate the orientation with both shoulders [6], a shoulder vector s is computed. This solution considers two scenarios:

(i) When both shoulders are visible (facing the camera completely and facing away completely).
(ii) When only one shoulder is visible to the camera.

4.5.1 Both Shoulders Visible. To estimate the orientation with both shoulders, we define a shoulder vector $\vec{s}$: $\vec{s} = \vec{s}_r - \vec{s}_l$ where $\vec{s}_r$ and $\vec{s}_l$ represent the vector coordinates for the right and left shoulders, respectively. The perpendicular vector $\vec{p}$, represents the forward-facing direction of the player's torso, and is computed as: $\vec{p} = (-s_y, s_x)$ Where s_x and s_y represent the x and y coordinates, respectively, of the $\vec{s}$ vector. This perpendicular vector points outward from the player's chest, perpendicular to the shoulder vector.

Therefore, the body orientation depends on the angular difference between this perpendicular vector $\vec{p}$ and the reference forward vector $\vec{f}$, which represents the camera's viewing direction or a predetermined field direction. This angular difference θ can be represented with a dot product:

$$\theta = \arccos\left(\frac{\vec{p} \cdot \vec{f}}{||\vec{p}|| \cdot ||\vec{f}||}\right)$$

With θ, we can define the orientation of the body. The body orientation thresholds for the three positions (*open*, *close*, and *half*) are the following:

$$\text{Orientation}(\theta) = \begin{cases} \text{Open} & \text{if } \theta \le \frac{\pi}{4} \quad (\approx 45^\circ) \\ \text{Half} & \text{if } \frac{\pi}{8} < \theta < \frac{7\pi}{4} \\ \text{Closed} & \text{if } \theta \ge \frac{3\pi}{4} \quad (\approx 135^\circ) \end{cases}$$

These values correspond to the values used by *Vantage* (Figure 1). Figure 4b shows all three vectors where *Blue* represents the perpendicular vector, *Green* the forward vector, and *Red* the shoulder vector. The angular difference θ in this case was *86 degrees*, and thus the position would have been classified as *half.*

4.5.2 Single Shoulder Visible. Based on the assumption that the camera is on the sideline, we can conclude that if neither shoulder is detected, the player is in *closed* or *open* position. This approach is triggered when one of the keypoints of the shoulders has a low confidence score ($\le$ 0.65). This threshold was defined based on our observations; we found that this is approximately the value when the shoulders are more difficult to see. Based on this, we assume that the shoulder with the lower confidence score is the shoulder that is not visible (or barely visible). Meaning that the player is facing in the opposite direction (e.g., if the left shoulder is not visible, it can be assumed that the player is facing the right side). Depending on the shoulder, the vector $\vec{p}$ is defined as:

$$\vec{p} = \begin{cases} (-1,\ 0) & \text{if left\_conf} > \text{right\_conf} \\ (1,\ 0) & \text{if right\_conf} \ge \text{left\_conf} \end{cases}$$

This means that the perpendicular vector $\vec{p}$ is completely facing the right when the confidence on the right is higher than the confidence on the left of the shoulder, and the other way around. After finalizing the orientation of that single frame, the orientation is put in a dequeue list containing the last x elements (set to 5 for inference). The final orientation is then considered to be the orientation that appears most often in that dequeue list.

5 Experiments

To test the pipeline, a series of clips were used from the Women's Super League match between Brighton and Aston Villa in the 24/25 season. The target player for this test run was set to Player number 10. The total duration of the video is 3.37 minutes.

The parameters used for tracking (Table 1) differ from one tracker to another. Since it is difficult for the detection model to always detect the ball correctly, we decided to adopt a more lenient approach. Therefore, we increase the hit counter and the distance threshold based on the assumption that it is better to have a prediction than not having a prediction at all. The Euclidean distance function also works better than the Intersection over Union (IoU) when detecting the ball under possession, as it is more difficult to find the IoU if a player covers the ball. The Q and R values of the ball aim to trust the motion model more and provide a bit less trust in the detection measurements. The parameters for the player tracker remain mostly standard, except for the hit counter to avoid losing the target player too quickly when not detected for a brief amount of time.

The experiment was run on a local computer with a NVIDIA-1660 SUPER GPU and an Intel-I7 CPU. In addition to this, the detection

Parameter	Player Tracker	Ball Tracker
distance_function	IOU	Euclidean
distance_threshold	0.8	60
initialization_delay	2	0
hit_counter_max	100	200
pointwise_hit_counter_max	4	10
filter_factory	KalmanFilter	KalmanFilter
Q	4	0.05
R	0.1	0.5

Table 1: Tracking parameters for player and ball trackers.

Pass Moment	Actual Pass Time (sec.)	Predicted Pass Time (sec.)	BP Pipeline	BP Vantage
1	9	8.32	Half	Open
2	23	-	-	Closed
3	37	-	-	Open
4	57	57.05	Half	Half
5	72	-	-	Half
6	86	84.69	Half	Half
7	101	-	-	Open
8	118	118.66	Half	Half
9	133	132.62	Closed	Closed
10	149	147.58	Half	Open
11	162	155.02	Half	Half
12	178	177.31	Half	Half
13	193	-	-	Open
14	210	209.59	Half	Half

Table 2: Pipeline and Vantage Data Across Pass Moments. BP: Body Position.

models were converted to TensorRT format for faster detections and less computing time [21].

To validate the pipeline results, we obtained the ground truth from a Vantage analyst. They annotated the clips manually where they identified 14 passes in the video where the target player was involved.

6 Results

To compute the video of the experiment (Section 5) took 69.24 minutes. Of the 14 passes identified by the Vantage analyst, the pipeline detected 8 (≈57%) of them. The body positioning of these 8 passes is compared with the manual analysis (Table 2). The timing of the passes mostly corresponded with the manual analyzed passes. We observed that most passes are within 1 second of the actual pass event. An outlier can be seen at pass event 11, where the pass detected by the pipeline was measured 7 seconds earlier than the actual pass event. For body positioning, six out of the eight (75% of correct pass detections) body orientations were correctly estimated.

7 Discussion and Limitations

7.1 Ball Occlusion

As we present in Section 6, from the total number passes (14), only eight were detected in our pipeline. One of the reasons for the missing six passes is that our detection and tracking model could not detect or track the ball properly at the moment of the pass. In some cases, the body of the ball carrier player partially (or completely) covers the ball, so that not even by increasing the hit counter (Section 4.3), we could track the ball.

Moreover, another situation occurs when two players completely cover the ball because we cannot determine which team has ball possession. This is problematic because the model does not know if the ball came from a deflection, an opponent, or a teammate and can thus not confidently classify it as a pass event. An option to mitigate this would be to have a multi-camera setup, where the ball is visible at all times.

7.2 Pass Detection

The results of detecting player body orientation are promising. Of the eight detected passes, the pipeline correctly identified six of them. Regarding incorrectly classified passes, Figure 5 shows the reason for the wrong classification of the other two passes. The problems occur because the player started in a *half* position (Figure 5a) and then transitioned to a *open* position (Figure 5b). Because the pipeline detected the pass a bit earlier, it was still considered *half*. If the pass would have been detected at the correct time (Figure 5b), the body orientation would have been switched to the correct classification (*open*). In the other missed incorrectly classified pass, we observed the same behavior.

Also, the reason for the wrong classification could be the small smoothing window. A smaller smoothing window would have classified the orientation correctly, but the pipeline would have been less robust and could have seen more mistakes at other pass events.

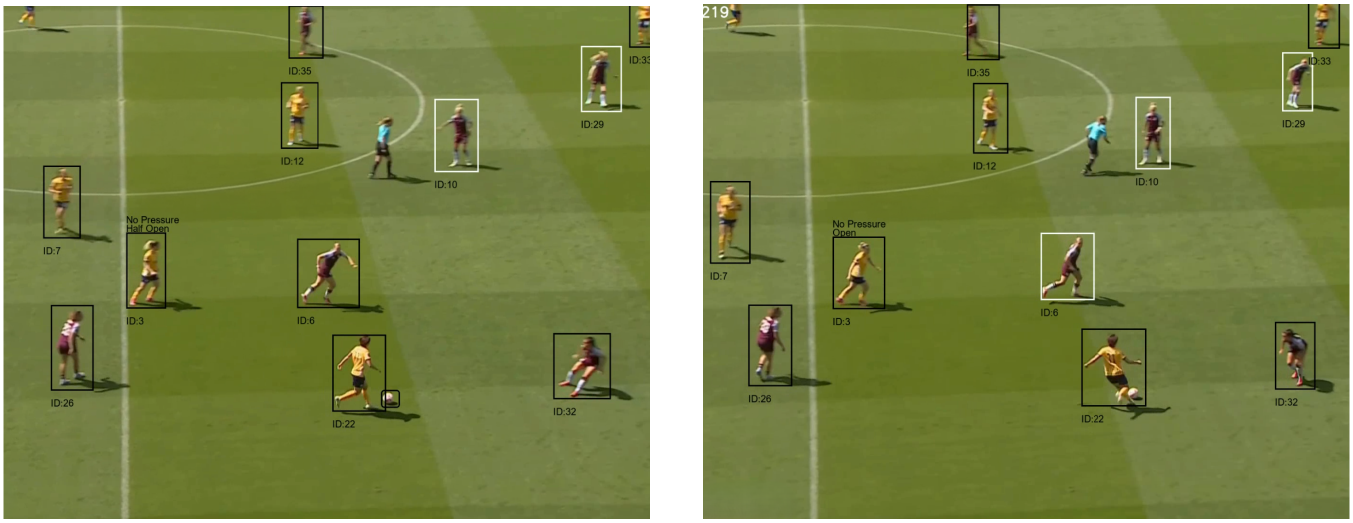
Lastly, the reselection procedure has also played an important role in these results. If the target player is not on the screen, the pipeline will prompt the user to reselect the ID corresponding to the target player. However, this only happens after 1 second's worth of footage. Meaning that, if the player disappeared in the moments leading up to a pass event, the pipeline could not have tracked the player, and thus those scans could have gone missing. One notable instance of this is at pass 10. Where a player is sent deep with a long ball, causing our target player to be not on the screen and prompting the reselection procedure.

Furthermore, there is an outlier in this set of results. The difference in initialization times between the pipeline and the actual pass event is a lot worse. This pass event was a throw-in. The pipeline is configured in such a way that it considers the ball to be in possession if the foot is near the ball. With a throw-in, the ball is relatively far away from the foot, causing the pipeline not to recognize the possession. The pipeline thus concluded that this pass event must have happened the last time a player had the ball in possession, which was eleven seconds earlier.

Lastly, it is important to note that 14 passes from one game are not enough to paint the whole picture. It is not known how the model would behave on a different video where the camera might be slightly tilted, or the weather might have been different, etc. An accurate experiment would have had around 100 data points that came from different videos and situations.

7.3 Forward vector and Automatic Teams

The forward vector (Section 4.1) requires manual intervention at the beginning and every time the target player is lost and reselected. This creates a bias factor and room for human error. An analyst might select a wrong direction, causing the results to be different.



(a) The initiation frame found by the model (half)

(b) The actual initiation frame of the pass (open).

Figure 5: Difference between model detected pass moment vs. actual moment of the pass.

Furthermore, the forward vector can also change with a moving camera. Additionally, if the user is only prompted to re-select at points when the target player is lost, the vector might be wrong at other moments. An option to mitigate this problem might be to increase the number of times a user has to change the vector, but this reduces the automation. A potential solution could include using landmarks and the number of players from one team in a half, to determine which side belongs to what team, and using that information to automatically assign the forward vector. This could reduce manual intervention and increase the robustness and accuracy of the results.

7.4 Team Assignment

The automatic team selection did not always work as expected. For instance, Figure 5 shows how some players are assigned to the wrong team. One problem here is that it is not always possible to determine the center of a player's jersey. In this case, the player with ID 35 (Figure 5), is categorized as team yellow, while they are clearly not. This is because the bounding box is slightly off to the right, so the center is not purple, but green, causing a wrong classification. Another issue can be seen with the player ID 26 (Figure 5). This player is wrongly classified because the center of their jersey is white due to the shirt numbers. One way to fix this is to train a model that can separate players as proposed by Lu et al. [14] who trained a two-stage CNN pipeline to classify the bounding boxes into two teams.

7.5 Performance: Computing Time

As mentioned in Section 6, the computation time for this 3:37 video was 1 hour, 9 minutes and 24 seconds. This means that approximately one minute of video takes around 19 minutes to be analyzed. Analyzing a full 90-minute game would therefore take about ≈ 29 hours. This is a very long time, especially considering that the pipeline requires expert intervention to reselect the target player once the player detection or tracking is lost. An option to mitigate this would be to upgrade the hardware. This experiment was run on a NVIDIA-1660 SUPER, which is not a high-end GPU. Using one of the higher-end GPUs could lead to a significant reduction in time. It is important to mention that this creates a trade-off between performance and costs that needs to be taken into account when making a decision.

Another option could be the adoption of a Re-ID system. This could also reduce the computational time and manual intervention to zero (if the forward vector is also automated). For instance, Deep Re-ID [12] uses *"deep features and auto-encoder-assisted image patching strategy"* to re-introduce the (in this case) players into the frame, which allows them to assign that player the same tracking Id. In this way, the target player will not be lost, and the analyst can leave the pipeline to automatically compute the statistics.

8 Conclusion and Future Work

The proposed pipeline includes a detection and tracking system, automatic team detection, and a system to calculate body orientation at pass events. The pipeline correctly classified 75% of the body orientation cases. However, most of the current problems were due to the detection and tracking aspects of the system. We believe that better resolution video and improved tracking could further enhance accuracy.

The current pipeline requires human input when a player disappears (the object is lost in the tracking), which increases the computational time. Future work could automate the forward vector and implement deep re-identification.

References

[1] Analyisport. 2022. *How Football Data is Used in the Media.* https://analyisport.com/insights/how-football-data-is-used-in-the-media/

[2] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2008. People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on computer vision and pattern recognition.* IEEE, 1–8.

[3] Guillermo Martinez Arastey. 2019. *Automated Tracking Of Body Positioning Using Match Footage*. https://www.sportperformanceanalysis.com/article/automated-tracking-of-player-positioning-using-match-footage

[4] Adrià Arbués-Sangüesa, Gloria Haro, Coloma Ballester, and Adrián Martín. 2019. Head, shoulders, hip and ball... hip and ball! using pose data to leverage football player orientation. *2nd Barca Sports Analytics Summit* (2019).

[5] Adria Arbues-Sanguesa, Adriàn Martín, Javier Fernández, Coloma Ballester, and Gloria Haro. 2020. Using player's body-orientation to model pass feasibility in soccer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 886–887.

[6] Adrià Arbués-Sangüesa, Adrián Martín, Javier Fernández, Carlos Rodríguez, Gloria Haro, and Coloma Ballester. 2020. Always Look on the Bright Side of the Field: Merging Pose and Contextual Data to Estimate Orientation of Soccer Players. arXiv:2003.00943 [cs.CV] https://arxiv.org/abs/2003.00943

[7] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*. Ieee, 3464–3468.

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence* 43, 1 (2019), 172–186.

[9] Bader Dammak. 2023. *Football-Players-Tracking*. https://github.com/Darkmyter/Football-Players-Tracking

[10] Jian Guo, He He, Tong He, Leonard Lausen, Mu Li, Haibin Lin, Xingjian Shi, Chenguang Wang, Junyuan Xie, Sheng Zha, Aston Zhang, Hang Zhang, Zhi Zhang, Zhongyue Zhang, Shuai Zheng, and Yi Zhu. 2020. GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing. *Journal of Machine Learning Research* 21, 23 (2020), 1–7. http://jmlr.org/papers/v21/19-429.html

[11] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. *Ultralytics YOLO*. https://github.com/ultralytics/ultralytics

[12] Samee Ullah Khan, Tanveer Hussain, Amin Ullah, and Sung Wook Baik. 2024. Deep-ReID: Deep features and autoencoder assisted image patching strategy for person re-identification in smart cities surveillance. *Multimedia Tools and Applications* 83, 5 (2024), 15079–15100.

[13] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60 (2004), 91–110.

[14] Wei-Lwun Lu, Jo-Anne Ting, Kevin Murphy, and J.J. Little. 2011. Identifying players in broadcast sports videos using conditional random fields. *IEEE Conference on Computer Vision and Pattern Recognition, 2008*, 3249–3256. doi:10.1109/CVPR.2011.5995562

[15] Trevor Lynn. 2023. *Pose Estimation Algorithms: History and Evolution*. https://blog.roboflow.com/pose-estimation-algorithms-history/

[16] Mustafa Özuysal, Vincent Lepetit, François Fleuret, and Pascal Fua. 2006. Feature Harvesting for Tracking-by-Detection. In *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 592–605.

[17] Guilherme de Sousa Pinheiro, Xing Jin, Varley Teoldo Da Costa, and Martin Lames. 2022. Body pose estimation integrated with notational analysis: A new approach to analyze penalty kicks strategy in elite football. *Frontiers in sports and active living* 4 (2022), 818556.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[19] Alexander Toshev and Christian Szegedy. 2014. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1653–1660.

[20] Tryolabs. 2022. *Norfair*. https://github.com/tryolabs/norfair/

[21] Ultralytics. 2024. *TensorRT Export for YOLO11 Models*. https://docs.ultralytics.com/integrations/tensorrt/

[22] Vantage. 2025. Your Vantage. https://www.instagram.com/yourvantage/ [Online; accessed 16. June. 2025].

[23] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, 3645–3649.

[24] Wyscout. 2025. The world's biggest library of soccer video and data. https://www.hudl.com/products/wyscout [Online; accessed 16. June. 2025].

[25] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple Baselines for Human Pose Estimation and Tracking. arXiv:1804.06208 [cs.CV] https://arxiv.org/abs/1804.06208

[26] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2022. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*. Springer, 1–21.