

# Prediction of the Ball Location on the 2D Plane in Football Using Optical Tracking Data

<sup>1</sup>Anar Amirli, <sup>\*2</sup>Hande Alemdar

<sup>1</sup> Department of Computer Science, Saarland University, Saarland, Germany

anaramirli@gmail.com, 

<sup>\*2</sup> Department of Computer Engineering, Middle East Technical University, Ankara

Turkey, alemdar@metu.edu.tr, 

## Abstract

Tracking the ball location is essential for automated game analysis in complex ball-centered team sports such as football. However, it has always been a challenge for image processing-based techniques because the players and other factors often occlude the view of the ball. This study proposes an automated machine learning-based method for predicting the ball location from players' behavior on the pitch. The model has been built by processing spatial information of players acquired from optical tracking data. Optical tracking data include samples from 300 matches of the 2017-2018 season of the Turkish Football Federation's Super League. We use neural networks to predict the ball location in 2D axes. The average coefficient of determination of the ball tracking model on the test set both for the x-axis and the y-axis is accordingly 79% and 92%, where the mean absolute error is 7.56 meters for the x-axis and 5.01 meters for the y-axis.

**Keywords:** Deep Neural Networks, Sports Analytics, Ball Tracking, Data Mining

## 1. INTRODUCTION

The rapid advancements in vision-based tracking and statistical tools have transformed many fields. These developments also break their way in sports analytics through many applications which offer new ways of in-detail analysis and observation to assess different aspects of both games and athletes' performance [1], [2]. As a result, sports analytics now has great importance for managers, athletes, sports experts, and even broadcasters since it enriches our knowledge about sports and leads to a more advanced and rich watching experience.

With its wide popularity and high revenue share, football benefits from all these developments the most. An extensive amount of research has already been done in football, from statistical properties of the game to game flow motifs [3]–[8]. Some studies focus on recognizing football events from the spatiotemporal soccer data. Khaustov and Mozgovoy [9] propose a rule-based system for identifying successful and unsuccessful passes and shots. Özdemir and Alemdar [10] develop a random forest classifier to identify corner kicks, free kicks, goals, and penalties. As in most team sports, understanding the strategies in football is a challenging task. It requires all kinds of relevant information, such as the

individual behavior of the players, their collective behavior as a team, and accurate ball location. In addition to sports analytics, new applications such as creating real-time game highlights for the audience experience are another emerging market that relies on accurate ball tracking. However, unlike some sports such as tennis, computer vision-based methods for frame-to-frame ball detection still remain beyond the state-of-the-art solutions for complex ball-centric sports since the ball is occluded most of the time. In football, detecting the ball's location is an even more challenging task mainly because of the nature of the game. Although there are several initiatives to equip the ball with a tracking chip, no such solution has been accepted by the governing organizations yet. One of the main challenges for ball tracking is that the size of the ball is relatively small compared to the vast field that needs to be monitored. Moreover, the players' interaction with the ball occurs in an unexpected way, and most importantly, the view is often occluded as the ball is lost behind the players.

To track the ball location in centimeter-level accuracy, a large number of very expensive cameras are needed. For example, the goal-line technology used to determine whether the ball has passed the goal line requires 14 cameras to detect

\* Corresponding Author

the goals, and that cannot track the ball in the field all the time.

Our study aims to provide a tool that can be used alongside current techniques to simplify and ensure more accurate ball-tracking. Our proposed approach uses players' formation during the game to estimate ball position. This formation-based approach focuses on analyzing within and between segment groups rather than the individual player activities. The main hypothesis we pursue is to deduce the key behaviors of ball movement in the dynamic game flow from spatial attributes such as players' speed and their positional distribution. This method has been motivated by our observations and experiments in deep learning-based approaches and our intuitive reasoning. Our model is designed to predict the ball's location on the 2D plane. Therefore, when the ball is flying, we aim to provide its projection on the 2D plane since this will give more valuable insights to the football professionals.

The rest of this paper is organized as follows. In the next section, we cite several related studies in the literature. In Section 3, we present our method to predict ball location from optical tracking data. In Section 4, we provide our experimental results on our real-world soccer optical tracking data set. Finally, we conclude with Section 5.

## 2. RELATED WORK

Given the importance of the location tracking of the ball in sports, there are several related studies in the literature. Kamble et al. provides a literature survey on the topic and identifies the need for multiple cameras as a challenge in ball tracking in football [11]. According to current regulations, it is impossible to equip the ball with wireless sensor devices; therefore, all of the existing studies that consider official football match data use computer vision-based approaches. There are two main methodological tracks: i) using broadcast videos and ii) having a fixed camera setup in the stadium. More recently, the use of drones [12] has also been suggested, yet it is not as common. Several studies focus on the detection of the ball only, whereas others also propose a trajectory for the ball.

Cardenas and Zuniga propose a two-stage algorithm [13]. In the first stage, they first extract a set of candidate objects from the segmented image. Then they filter objects that do not look like a ball using several features. In the second stage, each ball candidate's features obtained in the previous stage are combined with the dynamics model to form a trajectory. Then all the possible trajectories are ranked. Lhoest [14] proposes a similar two-stage approach starting with a ball detector followed by tracking. A deep convolutional neural network for image segmentation is used for detection, and a Kalman filter-based approach is used for tracking. In [15], an extended Kalman filter is used after the ball detection stage. Naidoo and Tapamo [16] propose another similar two-stage approach that contains soccer ball detection based on coarse analysis and filtering. Ren et al. use an 8-camera system to track the ball's location, and they provide results on a relatively small dataset that consists of a couple of minutes long video footage [17]. When they use a

buffer size of 50 frames, the detection rate is 68.5% only. A deep-learning-based system is also proposed in [18] and [19] for CCTV footage videos. Leo et al. [20] present a multi-step algorithm to detect the ball in image sequences acquired from fixed cameras. Candidate ball regions are selected by probabilistic analysis of locally affine invariant regions around distinctive points.

Durus works on the broadcast videos to track the ball to make tactical analyses [21]. He proposes to detect the ball first and then employs a particle filtering-based approach to track the ball and recover the ball's trajectory. This method requires the ball to be present and visible in the scene in all frames. Komorowski et al. use a deep neural network-based detector for the ball and players detection in high-resolution broadcast recordings [22]. The model produces a ball confidence map together with the position of the detected ball. To improve the discriminability of the ball, the feature pyramid network design pattern is used. In that way, lower-level features with a higher spatial resolution are combined with higher-level features with a bigger receptive field. In [23], a ball detection algorithm is presented. Ball candidates are first extracted using features based on the shape, color, and size. For selecting the best candidate, they use object area, centroid, bounding box, and minor and major axes features with a rule-based algorithm to eliminate the non-ball objects. Niu et al. [24] present an approach for discovering the ball states rather than its actual trajectory to automatically find the attacking patterns by the teams using broadcast videos.

In this study, we propose a machine learning-based approach to relieve the need for the increased number of cameras just for the ball tracking and use the players' and referee's behavior instead to determine the actual location of the ball in football. In our approach, even though the system cannot recognize the ball object, we are able to predict its location since we use the players' and referee's behavior. We train and evaluate our results on a dataset that contains data from a complete season. To the best of our knowledge, this study is unique in its attempt to locate the ball by using the players' and the main referee locations.

## 3. MATERIALS AND METHODS

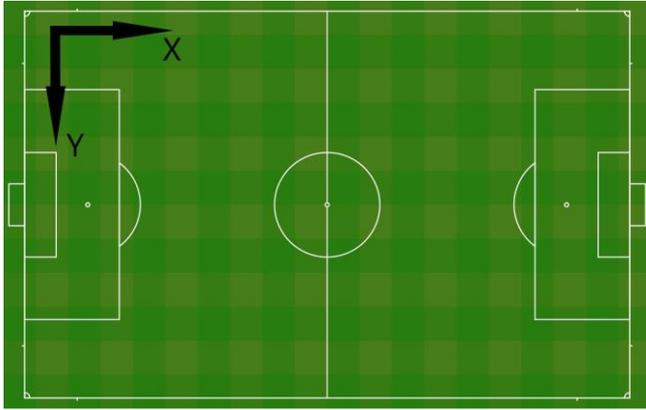
The state-of-art real-time two-camera player tracking system SentioScope, developed by Sentio, collects data from Turkish Super League (TSL) matches [25]. Using this data, we created a dataset for each game to analyze. For each second, position data of players of both teams and the ball in a rectangular coordinate system are saved in this dataset  $\mathcal{D}$ . We identify the home team as  $\mathcal{H}$  and away team as  $\mathcal{A}$ . The ball is labeled as  $\mathcal{B}$ , and the main referee is denoted as  $\mathcal{R}$ . The dataset for a match  $M$  is constructed as follows:

$$\mathcal{D}_M = \{c_i^t = (x_i^t, y_i^t) \mid \forall i \in \mathcal{H} \cup \mathcal{A} \cup \mathcal{B} \cup \mathcal{R}, \quad (1) \\ t = 1, 2, \dots, T_M\}$$

where  $c_i^t$  is the coordinates of the  $i^{\text{th}}$  object (player, referee, or ball) at timestep  $t$ ,  $x_i^t$  is the x-axis coordinate and  $y_i^t$  is the

y-axis coordinate.  $T_M$  is the maximum number of seconds in the match  $M$ .

The dataset contains data for 300 matches of the Turkish Football Federation Super League 2017-2018 season. It encloses the speed and location information of players and the main referee. As the raw data is collected using optical tracking cameras, it suffers from previously mentioned flaws to precisely track the ball [26].



**Figure 1.** Optical tracking software's coordinate system

In the dataset, spatial information of the ball is implicitly available during the frames when some player owns the ball. When the ball is not in play due to the game's pauses, we are also not interested in ball location since it does not have a value. Therefore, we are particularly interested in moments where the ball is in possession of a player. Those are the moments the ball is occluded the most, making computer vision-based ball tracking challenging.

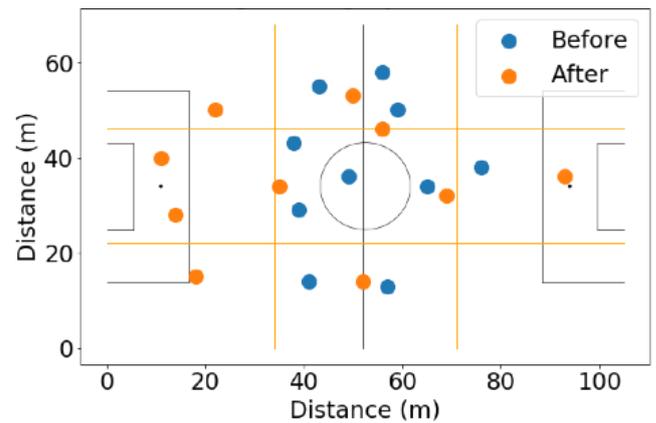
In the following sections, we introduce the formulation of our approach, which can also be applied to other team sports which possess the notion of ball possession, such as handball, basketball, and American football. We first propose a segment-based representation method that handles the ordering problem of the features. After that, we describe our feature extraction methodology. Finally, we present our neural network model to predict the ball location.

### 3.1. Segment-based Representation

There are 11 players for each team in a typical football match, adjusting their positions according to the ball location. Therefore, their collective behavior gives a good indication of the ball's location. For each player on the pitch, the feature set could be represented in a vector. Thus, the collection of individual player attributes forms a matrix that can be used in a machine learning task. However, the number of players can change due to certain events in the game, such as red cards or injuries. Also, due to the substitutions of players, the identities of the players may change. Moreover, in each game, there are different teams and different players playing in different formations. For all these reasons, it is impossible to find a correct ordering for the individual players to be represented in the feature matrix.

In order to address these problems and players' positional interchanges and capture the flow of players' movement, we suggest a data representation method using a role-based approach. The main idea of the proposed method is to divide the pitch into segments and assign players to these segments. This method enables us to set a common data representation regardless of the team and player identities; thus, we can use the same representation for all the matches.

We separate the football pitch into different segments on each axis and assign each player to the corresponding segment on each axis by assessing their movements for the most recent minutes. After grouping players, we use players' coordinates, and speed attributes to extract features, such as average characteristics and attributes of outlier players in the groups with faster speed or slowest speed, for example.



**Figure 2.** Visualization of scaling of average positions

To find out the players' segments, simple averaging of their movements alone is not enough. Football has its own well-established play-book, such as the tendency of teams to keep their formation structure when the opposition team owns the ball. To this end, when segment assignment is carried out, the average positions of a team are calculated over time steps when the rival team has the ball. The average positions are calculated over fixed-width overlapping sliding windows of 15 minutes with a step size of one minute. Due to the averaging, the positions tend to be grouped towards the middle of the field, as depicted in Figure 2 with blue dots. We observe that although the average distribution of the players may show some pattern of formation, it lies around the middle of the field in a squeezed form. In order to represent the average distribution of player formation across the whole pitch, we scale the positions according to the full field size. For each player  $i$ , the scaled coordinate  $c'_i$  is calculated as follows:

$$c'_i = \delta_2 - \frac{[(\delta_2 - \sigma) - (\delta_1 + \sigma)](\alpha - c_i)}{\alpha - \beta} \quad (2)$$

where  $c_i$  is the actual coordinate of player  $i$  for a given axis for a given time step,  $\delta_1$  and  $\delta_2$  are the boundaries of the segment,  $\alpha$  is the coordinate of the player that has the maximum value,  $\beta$  is the coordinate of the player that has the minimum value and  $\sigma$  is the variance of the player coordinates. In this way, for each time step, players' average positions on each axis are scaled to the range  $[\delta_1 + \sigma, \delta_2 -$

$\sigma$ ] based on the dispersion of the players' coordinate distribution. In this way, we obtain the new scaled positions as shown in orange dots in Figure 2.

Updating average positions in overlapping windows and grouping players at each minute based on their scaled averages allow us to capture the trends in players' movements even when a player migrates to different positions during the game or when two players swap their positions. In order to ensure the best representative groups, we partition the field into smaller segments making the area for each segment large enough to host at least one player at each frame. If there are too many segments, most of them will be empty most of the time since players usually stay towards the center and go to some of these segments for a short amount of time, especially the ones at the corners. Their short presence there does not change their overall average position much. For this reason, empirically, we divided the pitch into three different sections on each axis. Segment division on each axis separately helps deduce positional information about players' placement jointly on both axes and mitigate the sparsity of the feature vector introduced by empty segments. The boundaries of these segments are provided in Table 1 and Table 2 for the x and y axes, respectively.

**Table 1.** Segment groups along the x-axis and their boundaries

Segments	Boundaries (m)
Vertical Back (VB)	0, 34
Vertical Middle (VM)	34, 71
Vertical Front (VF)	71, 105

**Table 2.** Segment groups along the y-axis and their boundaries

Segments	Boundaries (m)
Horizontal Top (HT)	0, 22
Horizontal Middle (HM)	22, 46
Horizontal Bottom (HB)	46, 68

We use these segments to assign a role to each player using their scaled average coordinates. Results of the proposed role assignment method reflect the players' positional distribution properly. Having a fixed segment representation also allows us to order the features using the segments. This approach also makes it possible to represent players' data dynamically in a fixed order as they simultaneously change their roles during a match. The order of feature representation can simply be initialized in the form of a set of segments as  $\mathcal{S} = \{HT, HM, HB, VB, VM, VF\}$ . Instead of features calculated individually for each player in that setting, we have features extracted for set the of players in each segment group.

### 3.2. Feature Extraction

In our feature set, we consider many aspects of the football game to obtain the best set of features that can be used to predict the ball location. To begin with, the direction of the game flow depends on the movement of the player who possesses the ball, whose position, in turn, depends on the positional distribution of the other players and their spatial values, such as location, speed, and direction. In order to build a feature set that can help to map from feature space to the game flow at any given moment, we should consider all these spatial features. To capture the relevant connection among all the role groups and team groups (i.e., home team and away team), we calculate features using the groups. Furthermore, we also perform the feature extraction on the combined set of both teams to find the possible interactions among teams. We define our features on different sets:  $\mathcal{H}$  and  $\mathcal{A}$  are the set of the home team and away team's players except for the goalkeepers, respectively.  $\{\mathcal{S}_i \cap \mathcal{H}\}_i^{|\mathcal{S}|}$  and  $\{\mathcal{S}_i \cap \mathcal{A}\}_i^{|\mathcal{S}|}$  represent the set of players for each segment group in each team, and the set  $\mathcal{H} \cup \mathcal{A}$  contains all the players except for the goalkeepers. We represent goalkeepers and the main referee separately. For achieving the unity of expression, we define them as sets that contain a single element. We denote the goalkeepers for home and away teams as  $\mathcal{G}_{\mathcal{H}}$  and  $\mathcal{G}_{\mathcal{A}}$ , respectively. We denote the referee set as  $\mathcal{R}$ .

The speed is also one of the crucial components that provide insight into the ball's location. Teams can develop counterattacks or play with slow tactical passes just before an attack, or when a player dribbles the ball, he runs or sprints to pass his rival. All of these behavior patterns can be used to predict the location of the ball. In order to incorporate this into our prediction model, we categorize players into distinct speed groups. Empirically, we devised two groups. These groups are identified as *Low Intensity* (the speed is less than or equal to 3.5 m/s) and *High Intensity* (the speed is greater than 3.5 m/s). We observed that these speed groups show different characteristics in their relation to the ball's coordinates. For example, the distance of the *Low Intensity* (*LI*) group to the ball is usually more than that of the *High Intensity* (*HI*) group.

The movement direction is another essential component of motion when it comes to finding the ball's location. Thus, for each time step, we calculate the direction of the average movement for a group of players  $dir(G)$  as follows:

$$dir(G) = sign\left(\sum_i c_i^t - c_i^{t-1}\right), \forall i \in G \quad (3)$$

Ball control is an essential element in the football rulebook. Hence, to gain more control of the ball, players approach each other, eventually approaching the ball. The distribution of players gets denser as the game gets close to one of the goal lines. Capturing the form of players' positional distribution is an essential auxiliary element for defining ball location. Thus, the average position of each player group is calculated on each axis separately for each player group.

$$avg(c_p) = \frac{1}{N} \sum_{p=1}^N (c_p), \forall p \in G \quad (4)$$

However, the average position is not helpful when some players in that group are grouped closely, and the remaining players are relatively remote. For this reason, we also use the variance of the player coordinates

$$var(c_p) = \frac{1}{N} \sum_{i=1}^N (c_i - avg(c_p))^2, \forall p \in G \quad (5)$$

for different groups for players such as home team players,  $\mathcal{H}$ , away team players,  $\mathcal{A}$ , all players,  $\mathcal{H} \cup \mathcal{A}$  and player groups found with a density-based clustering (DBSCAN) approach [27]. We apply DBSCAN to have a more robust distribution representation by finding clustered groups of players. The algorithm starts from an arbitrary point in the group  $G$  and finds the cluster of neighborhood points where

at least  $N_{\min}$  of them are directly density-reachable from this arbitrary point with respect to  $\epsilon$  such that

$$|\{p \in G: d(x_{p_i}, x_{p_j}) \leq \epsilon\}| \geq N_{\min} \quad (6)$$

where  $d(\cdot)$  is a distance function. Density-based clustering allows us to find the player clusters where players are closer to each other. The center of this cluster is often close to the location of the ball. In our approach, we used  $\epsilon = 15\text{m}$  and  $N_{\min} = 7$  for the set  $\mathcal{H} \cup \mathcal{A}$ , and  $\epsilon = 15\text{m}$  and  $N_{\min} = 4$  for the  $\mathcal{H}$  or  $\mathcal{A}$  since their group size is smaller than the union set. In addition to DBSCAN features, we also extract features for specific target groups. For example, the referee's coordinates and speed were extracted by considering the fact that referee movements can be determinant since the referee often stands next to positions to resolve any dispute on the pitch. A compact representation of all the features we have extracted is provided in Table 3. In total, we use 251 different features per time step.

**Table 3.** List of features.  $G$  is the player set,  $G^+$  is the set of players with HI speed,  $G^-$  is the set of players with LI speed,  $G^*$  represents the cluster set of players that is found by DBSCAN.  $v_p$  is the speed and  $c_p$  is the coordinate of player  $p$ .

$G = \mathcal{H} \text{ or } \mathcal{A}$	$G = \{\mathcal{S}_i \cap \mathcal{H}\}_i^{ \mathcal{S} } \text{ or } \{\mathcal{S}_i \cap \mathcal{A}\}_i^{ \mathcal{S} }$	$G = \mathcal{H} \cup \mathcal{A}$	$G = \mathcal{G}_{\mathcal{H}} \text{ or } \mathcal{G}_{\mathcal{A}}$	$G = \mathcal{R}$
$avg(c_p) \forall p \in G$	$avg(c_p) \forall p \in G$	$avg(c_p) \forall p \in G$	$c_p, \forall p \in G$	$c_p, \forall p \in G$
$avg(v_p) \forall p \in G$	$avg(v_p) \forall p \in G$	$avg(v_p) \forall p \in G$	$v_p, \forall p \in G$	$v_p, \forall p \in G$
$dir(G)$	$dir(G)$	$var(c_p) \forall p \in G$	$dir(G)$	$dir(G)$
$var(c_p) \forall p \in G$	$var(c_p) \forall p \in G$	$avg(v_p) \forall p \in G^*$	$d(c_p, avg(c_i)),$	$d(c_p, avg(c_i)),$
$avg(c_p) \forall p \in G^+$	$avg(c_p) \forall p \in G^+$	$var(c_p) \forall p \in G^*$	$\forall p \in \mathcal{G}_{\mathcal{H}}, \forall i \in \mathcal{H}$	$\forall p \in \mathcal{R}, \forall i \in \mathcal{H} \cup \mathcal{A}$
$avg(v_p) \forall p \in G^+$	$avg(v_p) \forall p \in G^+$		$d(c_p, avg(c_i)),$	$d(c_p, avg(c_i)),$
$avg(c_p) \forall p \in G^-$	$avg(c_p) \forall p \in G^-$		$\forall p \in \mathcal{G}_{\mathcal{A}}, \forall i \in \mathcal{A}$	$\forall p \in \mathcal{R}, \forall i \in G^*$
$avg(v_p) \forall p \in G^-$	$avg(v_p) \forall p \in G^-$			
$avg(c_p) \forall p \in G^*$				
$avg(v_p) \forall p \in G^*$				
$var(c_p) \forall p \in G^*$				
$min(c_p) \forall p \in G$				
$min(v_p) \forall p \in G$				
$max(c_p) \forall p \in G$				
$max(v_p) \forall p \in G$				
$c_i, i = \underset{p \in G}{\operatorname{argmin}}(v_p)$				
$c_i, i = \underset{p \in G}{\operatorname{argmax}}(v_p)$				
$v_i, i = \underset{p \in G}{\operatorname{argmin}}(c_p)$				
$v_i, i = \underset{p \in G}{\operatorname{argmax}}(c_p)$				

#### 4. PERFORMANCE EVALUATION

We use two separate artificial neural network regression models for predicting ball location along the x-axis and y-axis. In our experimental setup, we use training, validation, and test sets containing data from 243 (81%), 27 (9%), and 30 (10%) matches respectively. We randomly select the matches using their unique identifiers from the whole dataset that contains 300 matches. The matches in the training set were used in the training stage of the neural network models. We used 27 matches as a validation set to perform the

hyperparameter optimization. After the hyperparameter tuning is finished. We tested the performance of the final tuned model on the test set. The match data in the test set were used only at that stage.

The models have been trained with dropout [28]. Moreover, an early-stopping technique has been implemented to avoid overfitting [29]. In our setting, we use L2 loss as the main loss function. Furthermore, mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of

determination  $R^2$  evaluation metrics are used as well in order to carry out fair performance evaluation.

$$MAE = \frac{1}{N} \sum_i |y_i - f_i| \quad (7)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_i |y_i - f_i|^2} \quad (8)$$

$$R^2 = 1 - \frac{\sum_i (f_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} \quad (9)$$

To avoid saturation of activation function, we normalized the target variables using min-max normalization. However, we scaled back the predicted outputs to the normal scale when we did the performance measurement. That method helped our training network to converge to a better local optimum.

The final proposed model is built on a deep neural network through a series of experiments. We optimized the depth of our neural network, the number of hidden nodes, the activation function, the optimization algorithm, and the learning rate of the optimization algorithm. All of the hyperparameter optimizations are performed using the validation set. As a result, we use the rectified linear unit activation [30] as our nonlinear activation function with a gradient-based stochastic optimization algorithm with Adam optimizer [31] with a batch size of 65 and a learning rate of 0.01. The depths of the neural networks are 7 for the x-axis prediction model and 5 for the y-axis prediction model. The number of hidden nodes is 251 for both models.

According to our experimental evaluation, the result of the coefficient of determination on the train set of the x-axis and y-axis are 85% and 94%, respectively. The performance on the test set is 79% and 92% for the x-axis and y-axis, respectively. Our results indicate that the method performs well in its generalization ability. The mean absolute error of each model on the test set was calculated to gain more insight regarding the ball's location on the pitch. On the test, the error is slightly above 7.56 meters on the x-axis while it is 5.01 meters on the y-axis. The mean squared error yields marginally higher results than the mean absolute error. The most significant errors occur in cases when the ball transaction happens unexpectedly and when the ball changes its position from one player to another over a long distance, which often occurs during the long passes and shots on goal.

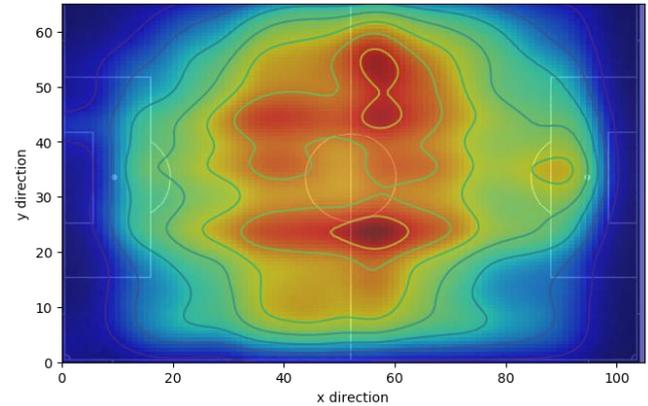
**Table 4.** Regression results on the x-axis

Dataset	MAE	RMSE	$R^2$
Train	6.47	9.90	84.64
Validation	7.39	11.23	80.11
Test	7.56	11.46	79.41

**Table 5.** Regression results on the y-axis

Dataset	MAE	RMSE	$R^2$
Train	4.12	6.23	93.56
Validation	4.83	6.96	92.74
Test	5.01	7.19	92.16

Overall, we can see that our approach performs particularly better on the y-axis. This is because the width of the pitch is almost half of its length and also, the teams are not spread over the y-axis as they try to cover distance mostly on the x-axis in order to reach the opposition goal. The full evaluation results for the x-axis and y-axis are provided in Table 4 and Table 5, respectively.

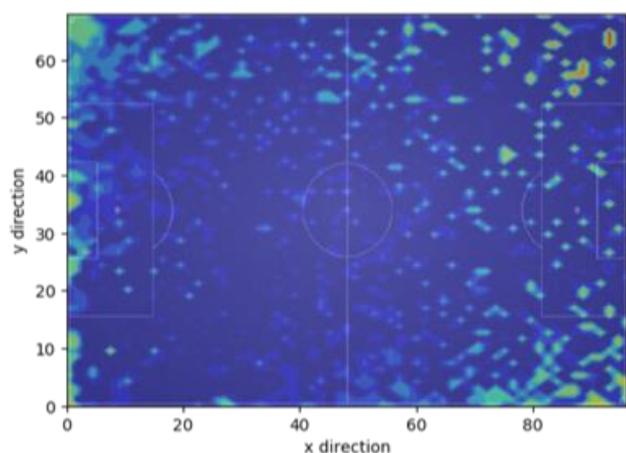


**Figure 3.** Heatmap of the predicted coordinates  $c'_i = (x'_i, y'_i)$  when the Euclidean distance between the actual  $c_i = (x_i, y_i)$  and the predicted coordinate is bigger than the Euclidean distance between the mean absolute error of test set on both axes.

We also visualize our model performance using a spatial representation with heatmaps. In Figure 3, we present the heatmap for the density of the predicted coordinates,  $c'_i = (x'_i, y'_i)$ , when the Euclidean distance between the actual coordinate,  $c_i = (x_i, y_i)$ , and the predicted coordinate is bigger than the Euclidean distance between the mean absolute error of test set on both axes, i.e.,  $\sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2} > \sqrt{x_{MAE}^2 + y_{MAE}^2}$ . This gives us an impression about which parts of the pitch the “faulty predictions” occur the most. Since the ball is at the center region on average, the errors also happen at that region.

In Figure 4, we provide the heatmap of the density distribution of all the errors using the Euclidean distance between the prediction and the actual coordinate. This heatmap visualizes the magnitude of errors the model makes. We observe that regions closer to the goals, corners, wings are the places where the model makes the largest prediction error. The results represented here are consistent with our observations that the model initially struggles to adapt when the goalkeeper starts the game with a long shot, a corner kick is taken, or when there is an unexpected change of attack from one wing to another.

Overall, with these two heatmaps, we provide insights about both the number of errors and the magnitude of errors across the pitch. Additionally, we provide an animated image of our method's performance on real-world match data together with our codebase used for obtaining the results presented in this study at <https://github.com/anaramirli/predict-soccer-ball-location>.



**Figure 4.** Heatmap of the density distribution of all the errors based on the Euclidean distance between the prediction and the actual coordinate.

## 5. CONCLUSIONS

We presented our approach to track the ball location in football, especially when it is occluded. We showed that the ball's coordinates could be estimated from optical tracking data by using machine learning models. In addition to the results obtained by using neural networks, we also present novel ways of extracting features that are the most significant for predicting the ball's location.

The ability of our model's predictions on the y-axis is around 5 meters. The results for the x-axis (~7.5 m) are not as good as that of the y-axis model. The model for x-axis struggles, especially when a goalkeeper starts the game or a player takes a free kick. The model prediction focuses on the player groups on the x-axis rather than a player who is with the ball. During the typical long passing when the ball rapidly changes location halfway through the other side, we observed that the regression models could not identify these changes for the first few frames. However, the achieved prediction rate is good enough to apply it to the existing system as an additional tool and increase their performance.

For our future study, we will explore the ways of improving the performance of the models by employing some auxiliary models such as the detection of the ball from the detection of the game events such as free kick, corner, penalty. The regression models we use for prediction ball location can be combined with the detected event's field lines and thus generalize performance more precisely. Moreover, the accuracy of the model can be improved with more detailed features. It is also important to mention that in further studies, we can utilize Adversarial Generative Networks (GAN) [32] to eliminate the shortcoming of input space representation of individual player attributes that we face in traditional neural networks. Our future work will also focus on pixel-wise detection of the ball location using conditional Pixel2Pixel GAN [33] architecture by incorporating individual features of players together with that of group-based features that we proposed in this study.

**Author contributions:** Concept – H.A., A. A.; Data Collection and Processing - A.A.; Literature Search - H.A., A. A.; Writing - H.A., A. A.

**Conflict of Interest:** No conflict of interest was declared by the authors.

**Financial Disclosure:** The authors declared that this study had received no financial support.

## REFERENCES

- [1] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, "Large-Scale Analysis of Soccer Matches Using Spatiotemporal Tracking Data," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2015-Janua, no. January, pp. 725–730, 2014.
- [2] B. Skinner and S. J. Guy, "A method for using player tracking data in basketball to learn player skills and predict team performance," *PLoS One*, vol. 10, no. 9, pp. 1–15, 2015.
- [3] P. Lucey, D. Oliver, P. Carr, J. Roth, and I. Matthews, "Assessing team strategy using spatiotemporal data," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. Part F1288, pp. 1366–1374, 2013.
- [4] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale, "State of the Art of Sports Data Visualization," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 663–686, 2018.
- [5] A. Rusu, D. Stoica, E. Burns, B. Hample, K. McGarry, and R. Russell, "Dynamic visualizations for soccer statistical analysis," *Proc. Int. Conf. Inf. Vis.*, pp. 207–212, 2010.
- [6] D. Sumpter, *Soccermatics: Mathematical Adventures in the Beautiful Game*. Bloomsbury Publishing Plc, 2016.
- [7] L. Gyarmati, H. Kwak, and P. Rodriguez, "Searching for a Unique Style in Soccer," 2014, pp. 5–8.
- [8] L. Y. Wu, A. J. Danielson, X. J. Hu, and T. B. Swartz, "A contextual analysis of crossing the ball in soccer," *J. Quant. Anal. Sport.*, vol. 17, no. 1, pp. 57–66, 2021.
- [9] V. Khaustov and M. Mozgovoy, "Recognizing events in spatiotemporal soccer data," *Appl. Sci.*, vol. 10, no. 22, pp. 1–12, 2020.
- [10] E. Özdemir and H. Alemdar, "Predicting soccer events from optical tracking data," *26th IEEE Signal Process. Commun. Appl. Conf. SIU 2018*, pp. 1–4, 2018.
- [11] P. R. Kamble, A. G. Keskar, and K. M. Bhurchandi, "Ball tracking in sports: a survey," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 1655–1705, 2019.
- [12] A. E. Abulwafa, A. I. Saleh, H. A. Ali, and M. S. Saraya, "A fog based ball tracking (FB2T) system using intelligent ball bees," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5735–5754, 2020.
- [13] D. G. Cardenas and M. D. Zuniga, "Bullet-Proof Robust Real-Time Ball Tracking," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016, pp. 1–8.
- [14] A. Lhoest, "Deep Learning for Ball Tracking in Football Sequences," University of Liège, 2020.
- [15] H. D. Najeeb and R. F. Ghani, "Tracking Ball in Soccer Game Video using Extended Kalman Filter," *Proc. 2020 Int. Conf. Comput. Sci. Softw. Eng. CSASE 2020*, pp. 78–82, 2020.

- [16] W. C. Naidoo and J. R. Tapamo, "Soccer video analysis by ball, player and referee tracking," *ACM Int. Conf. Proceeding Ser.*, vol. 204, pp. 51–60, 2006.
- [17] J. Ren, J. Orwell, G. A. Jones, and M. Xu, "Tracking the soccer ball using multiple fixed cameras," *Comput. Vis. Image Underst.*, vol. 113, no. 5, pp. 633–642, 2009.
- [18] J. Komorowski, G. Kurzejamski, and G. Sarwas, "BallTrack: Football ball tracking for real-time CCTV systems," *Proc. 16th Int. Conf. Mach. Vis. Appl. MVA 2019*, 2019.
- [19] P. R. Kamble, A. G. Keskar, and K. M. Bhurchandi, "A deep learning ball tracking system in soccer videos," *Opto-electronics Rev.*, vol. 27, no. 1, pp. 58–69, 2019.
- [20] M. Durus, "Ball Tracking and Action Recognition of Soccer Players in TV Broadcast Videos," *Technische Universität München*, 2014.
- [21] J. Komorowski, G. Kurzejamski, and G. Sarwas, "Footandball: Integrated player and ball detector," *VISIGRAPP 2020 - Proc. 15th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 5, pp. 47–56, 2020.
- [22] M. Leo, P. L. Mazzeo, M. Nitti, and P. Spagnolo, "Accurate ball detection in soccer images using probabilistic analysis of salient regions," *Mach. Vis. Appl.*, vol. 24, no. 8, pp. 1561–1574, 2013.
- [23] J. Hossein-Khani, H. Soltanian-Zadeh, M. Kamarei, and O. Staadt, "Ball detection with the aim of corner event detection in soccer video," *Proc. - 9th IEEE Int. Symp. Parallel Distrib. Process. with Appl. Work. ISPAW 2011 - ICASE 2011, SGH 2011, GSDP 2011*, pp. 147–152, 2011.
- [24] Z. Niu, X. Gao, and Q. Tian, "Tactic analysis based on real-world ball trajectory in soccer video," *Pattern Recognit.*, vol. 45, no. 5, pp. 1937–1947, 2012.
- [25] S. Baysal and P. Duygulu, "Sentioscope: A Soccer Player Tracking System Using Model Field Particles," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1350–1362, 2016.
- [26] E. Kùlah and H. Alemdar, "Quantifying the value of sprints in elite football using spatial cohesive networks," *Chaos, Solitons and Fractals*, vol. 139, 2020.
- [27] M. Daszykowski and B. Walczak, "Density-Based Clustering Methods," *Compr. Chemom.*, vol. 2, pp. 635–654, 2009.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [29] L. Prechelt, "Early stopping - But when?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 *LECTU*, pp. 53–67, 2012.
- [30] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, vol. 15, pp. 315–323.
- [31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [32] I. J. Goodfellow et al., "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014, pp. 2672–2680.
- [33] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.