Automatic Player Detection and Labeling in Broadcast Soccer Video Using Genetic Algorithm

Golafrooz Davoodifar[1*], Masoume Gholizade[2], Mohammad Rahmanimanesh[2], Rouhollah Haghshenas[3], Hadi Soltanizadeh[2]

***Abstract* -- Due to the increasing amount of video data, a lot of research has been done in the field of retrieving and categorizing this type of data. On the other hand, with the growing popularity of football and the increasing number of its audiences, the importance of automatic and real-time extraction of statistics and information about soccer matches has increased. One of the critical and challenging tasks in soccer video analysis is the detection of players' blobs and regions, along with identifying the teams related to the players. This task encounters many challenges, including grass loss in the playfield, the presence of playfield lines and players' shadows, the overlapping of players with objects and other players, and different shapes of players in different situations. This paper proposes a framework for detecting players and their related teams. For this purpose, an object-sieve-based method is used to detect players' blobs, and a genetic algorithm is used to identify their related teams. Each chromosome of the genetic algorithm is a window that lies on one blob whose fitness function shows how much its color and shape characteristics fit with the uniforms of each of the two teams. The proposed method was evaluated by 50 different frames of broadcast soccer videos, including 563 players, and 40 different sub-images, including 84 players. The results show 98% and 91.6% precision for player detection and labeling, respectively.**

***Index Terms*-- Label detection, Soccer video, Blob Sieve, Genetic Algorithm, Image processing, Video processing.**

I. INTRODUCTION

Soccer is the most popular sport in today's society, with millions of people following it. As a result, in recent years, various video analysis applications have been developed to provide information to fans, officials, coaches, and players. Some of these apps are designed to perform specific functions, such as recognizing players, segmenting the playing field, or registering television images [1],[2]. This task faces many challenges, such as grass loss and decay in the playfield, the presence of playfield lines and shadows, and the overlapping of players with objects and other players. Automatic recognition of images or wearable sensors is a fundamental step toward the development of advanced sports analysis and broadcasting software [3].

Also, the shape of players can be variable in different positions. Players don't have unique shapes while standing, tackling, running, or sitting [4].

After detecting players' regions and blobs, their related teams should also be identified. This is crucial in many applications, such as calculating the percentage of pass accuracy, the rate of block passes, and the rate of possession of the ball for each team. The color similarity of different parts of teams' uniforms, the crowd of players in the area, and the presence of more than one player in one detected blob are the most challenging aspects of this task. For example, imagine two teams whose uniforms have the same colors but different patterns:

- Team 1: red shirts, white shorts, red socks
- Team 2: white shirts, red shorts, white socks

In this case, it is not easy to identify the teams of players, especially if more than one player is identified in the same patch. The use of artificial intelligence algorithms can be effective as a suitable solution to this challenge. Artificial intelligence algorithms include a wide range of applications that can reduce risk in addition to higher operating speeds [5],[6] [7].

In this paper, we first extract the playfield using a color-based method and then detect the blobs related to players using an object-sieve-based method. At last, a genetic algorithm is used to identify the related teams of players and label the detected players.

A genetic algorithm is a type of evolutionary algorithm that uses inheritance, mutation, and selection

1.Computer Engineering, Islamic Azad Mobarake Branch, Mobarakeh, Isfahan, Iran.
2.Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran.
3.Faculty of Human Sciences, Semnan University, Semnan, Iran.
Corresponding author: afroozdf@gmail.com

principles to find the optimal formula to predict or match a pattern. The problems that can be solved by the genetic algorithm must have inputs that are transformed into solutions through a process modeled on genetic evolution, and then the solutions are evaluated as candidates by the fitness function if the exit condition of the problem is met. The algorithm has been terminated.

According to the nature of the problem, the input is a spot identified by the players and an image related to that spot. In the results, it was found that the genetic algorithm in this problem was able to recognize the players who have the same color pattern or overlap in the image.

The remainder of the essay is structured as follows: The research on player detection and team identification is covered in Section II. We offer a technique for removing the players' blobs from the playfield in Section III. A technique for locating the teams of the identified players is provided in Section IV. The results of the suggested approaches are displayed in Section V, and the study is then completed in Section VI.

II. RELATED WORKS

Related works on player detection are divided into two categories: player region detection and player team detection.

A. *Detecting players' regions*

In this step, the blobs that belong to players are detected. The proposed methods are mainly learning-based or contour-sieve-based.

Rao M. and C. Pati in [8] identified non-player areas to detect the player regions. In this study, the image is converted to a binary image after playfield pixels extraction. So that the playfield pixel were shown in black and the foreground pixels were shown in white. Then, lines were found and removed using the Sobel edge detector and Linear Hough Transform. At last, the area of the remaining objects is calculated, and objects having less or more area than a specific range are eliminated.

Heydari and E. Moghadam in [9] initially removed the non-player regions. They identified and removed the playfield pixels and used the largest horizontal line to remove the gallery area and billboards. Then they divided the remaining objects into two-player and non-player classes using a multi-layer perceptron neural network. This classifier had three layers: the input layer, the hidden layer, and the output layer. The input layer has 27 nodes of color features and one area feature; the hidden layer has 30 nodes; and the output layer has one node.

In the paper [10], the authors offered an innovative and effective technique for extracting soccer match events based on numerous cameras. The free-kick, corner kick, penalty kick, and goal are included in their definition since they are sample shots for the audience to witness. The events are crucial for selecting highlights and analyzing sports data.

In [11], Liu et al. used a boosted cascade detector and Haar features to detect players. This detector is trained using 34000 Haar features and the AdaBoost algorithm and randomly selects 0.01 features for each round of training.

In [12] the authors suggested their method for detecting suitable regions by integrating gradient magnitude and direction coherence of text pixels in a novel way. The number of temporal frame clusters formed by K-means clustering on frame differences is determined using candidate regions. Using other algorithms can provide better results in this research. After player detection using HOG features and an SVM classifier, Gerke et al. [13] extracted color features from detected positive samples in HSV color space. These features were divided into two clusters using the k-mean algorithm, and an SVM was made for each cluster. These SVMs were combined with the HOG SVM and used for player detection in the remaining frames. So, both color and gradient features were applied for player detection.

Manafifard et al. in [14], authors first removed the pixels that were out of the grass color range and then filtered the remaining blobs using area, major and minor axis length, and width ratio. Then the eliminated blobs are reprocessed by the proposed multi-scale edge-based detector using a dilation operator, and the blobs without vertical edges are removed.

Yang et al. [15] transmitted the image to grayscale using a saliency map and applied a linear transmission on gray-scale image pixels to enhance the contrast; then used the Otsu algorithm to split the image into foreground and background pixels; and applied the top-hat algorithm to remove playfield lines and extract player blobs.

Lu et al. [16] proposed a cascaded convolutional neural network (CNN) to detect players. This algorithm had two main steps: first, the regions that may contain players were identified, and then the detected regions were evaluated to detect players. This neural network is trained in three different branches to identify player with different aspect ratios in images, and the test phase uses different scales of images to improve the accuracy of results. This research uses a big dataset for the training step.

Kim et al. [17] demonstrated the difference between frames and moving edges for player and foreground detection. In their method, two consecutive frames were subtracted, and a Gaussian filter was applied to the resulting image. Then the moving edges were detected using a Canny edge detector. The edges of the current frame were also used to detect the static players. At last, the combination of the union of the two resulting images and the background subtraction denotes the regions of players and foreground objects.

Zhang et al. [10] demonstrated the YOLO structure of CNN. In their method, the input image is divided into different parts that overlap with each other. The parts that contain at least one object were used in the training phase. In the test phase, only the parts which have common parts with the playfield enter the network to detect players. At last, the parts that were detected as players by the network and whose overlap area was greater than a threshold were merged and considered one player.

B. Detecting players' teams

In [11], Liu et al. manually collected samples of player shirts from the first frames. The Expectation-Maximization (EM) technique and Gaussian Mixture Model (GMM) were then used to cluster their color histograms. The referee cluster is included in the k cluster of the EM method, which is employed to increase accuracy without removing the cluster. Then, four clusters were created by hierarchically merging these clusters. The players are then labeled after using the Bhattacharya distance to compare the histograms of each cluster of players.

Tran et al in [18], applied the RGB color space feature and k-mean clustering to identify the related teams of the players. The histogram of each player was composed of 512 bins from the RGB color space.

In [14], Manafifard et al. used the particles swarm optimization (PSO) algorithm to detect the player teams. Particles in this method were composed of four elements that denote the characteristics of an ellipse, and they spread on the detected player blobs in the image. Each of the particles only performs the search operation on its own blob. The fitness function in this algorithm depends on color histogram features, gradian orientation features, and the rate of grass pixels in the related ellipse.

Lu et al. [16] applied a convolutional neural network to auto-detect the teams related to players in soccer videos.

Montañés Laborda et al. [19] applied the HSV color space and Gaussian Mixture Model to detect the teams of players. In their paper for each team, two different Gaussian Mixture Models were made from jersey and shorts color, and these histograms were used to identify the players' teams.

Lu et al. [20] demonstrated an RGB histogram of a team's jersey colors to train a logistic regression classifier that maps team labels to players.

Chen et al. [21] applied HSV color space to detect teams related to players. For this purpose, two different histograms were created according to a set of training samples containing the color of the shirts of the teams. The peak values were specified in the created histograms and searched and counted in the blob of players. The team of players is identified based on the counted values.

Zhang et al. [10] demonstrated a U-encoder to detect a team of players. This encoder was composed of a convolution layer, max-pooling layer, and decomposition layer, which were connected using the rectified linear unit and extracted a feature vector with 256 dimensions. Then hierarchal clustering is applied to the extracted features, and three clusters are created that show two teams and referees.

III. PROPOSED METHOD

In the analysis of football videos, it is not easy to identify the players' teams, especially in cases where there is more than one player in a detected spot. This situation becomes even more difficult if the players' whole bodies are not placed on the play area because the color of the advertising boards may be the same as the color of the players' uniforms. In this situation, we used innovative and evolutionary algorithms to identify the team related to the players.

A genetic algorithm has been used to identify the team of players. For this purpose, the components and functions of the genetic algorithm must be adjusted according to the problem. The input of this algorithm is a spot detected by the players and an image related to that spot.

A genetic algorithm has been used to identify the team of players. For this purpose, the components and functions of the genetic algorithm must be adjusted according to the problem. The input of this algorithm is a spot detected by the players and an image related to that spot.

We first propose a method to extract the playfield from video frames. Afterward, a blob-sieve-based method for player region detection as well as a solution-space-searching-based method for team detection are proposed.

This research uses a genetic evolutionary method to detect and categorize participants in the match, in contrast to many comparable efforts that employ deep learning networks and enormous data sets for model training. There is no longer a requirement for massive data sets thanks to the use of genetic algorithms. If a deep learning network is used, it must be trained for each color of clothing, requiring a large amount of data for each color pattern, which results in a considerable waste of time and money.

When the genetic algorithm is employed, however, the color pattern is set based on each user, and the values of the fit function required for the fitness genetic algorithm are decided based on the same color patterns supplied by the user, making the process considerably faster.

A. playfield region extraction

To detect the playfield area, we first pass the frame through the following filter:

$$GFil(x,y) = \begin{cases} 1, & (g(x,y) > \\ r(x,y) > b(x,y))\ and \\ 0, & else \end{cases} \tag{1}$$

where $r(x,y)$, $g(x,y)$, and $b(x,y)$ are r, g, and b elements of RGB color space related to the pixel at point (x,y) of the image and $edge(x,y)$ is a binary function that determines whether the pixel is a member of the edge or not. The edge of the frame is obtained using Sobel or Canny filters.

Afterward, we calculate the mean and standard deviation values of r, g, and b of the filtered pixels. Let m_r and σ_r be the mean and standard deviation of r, m_g and σ_g be the mean and standard deviation of g, and m_b and σ_b be the mean and standard deviation of b, respectively. A pixel may be a member of playfield pixels if it satisfies the following condition.

$$\mathrm{ground(x,y)} = \begin{cases} 1 & m_r - \frac{5}{3} * \sigma_r < \mathrm{r}_{(\mathrm{x,y})} < m_r + \frac{5}{3} * \sigma_r \\ & m_g - \frac{5}{3} * \sigma_g < \mathrm{g}_{(\mathrm{x,y})} < m_g + \frac{5}{3} * \sigma_g, \\ & m_b - \frac{5}{3} * \sigma_b < \mathrm{b}_{(\mathrm{x,y})} < m_b + \frac{5}{3} * \sigma_b \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Players and field lines are situated in non-playfield pixels thanks to this filter. By applying the linear Hough transform to non-playfield pixels, field lines are recovered. After that, playfield pixels are filled using morphological operations after the dilation morphology operator is applied to extracted lines together with a significant structural element. We acquire the linked pixels by intersecting the complement of dilated lines and playfield pixels. We then identify the largest component and remove the low-area components. At last, the dilation is applied, and the convex hull of the remaining components is determined using the morphology operator. Fig.1 shows the block diagram of the playfield detection task in the video frames. These resulting images are shown in Fig. 2.

B. Detection of the players' region

In most of the football video frames, the players have appeared like blobs with certain color patterns. This characteristic can be used to identify the players by the following steps:

To eliminate the regions out of the playfield, the intersection of the playfield binary mask (G) and the original image (I) ($B = I \cap G$) is obtained.

Eliminated regions in B are covered with grass color and then blurred using a median filter as follows:

$$B = \mathrm{Median}(B + (\Gamma \cap \bar{G}))$$

where Γ is an image covered by grass color (GC), and $\bar{G}$ is the complement of the playfield binary mask. The values r, g, and b of grass color are obtained using the values of playfield detection steps as:

$$r(gc) = m_r, \qquad g(gc) = whose$$

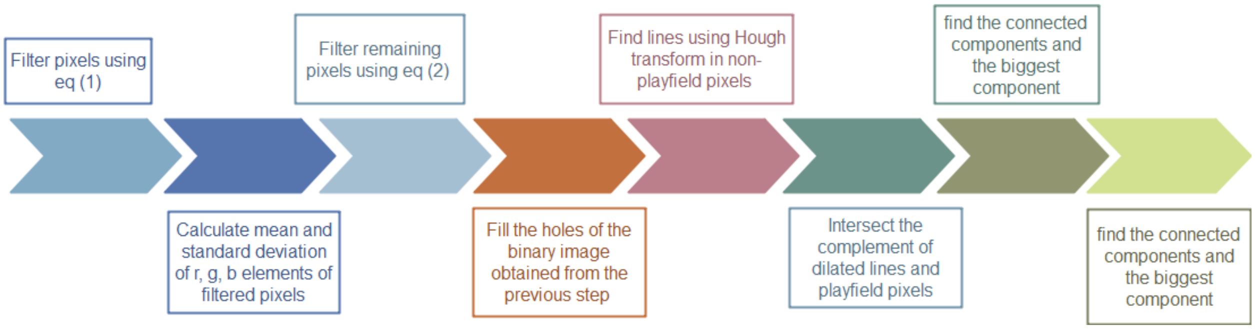

Filter pixels using eq (1)
Calculate mean and standard deviation of r, g, b elements of filtered pixels
Filter remaining pixels using eq (2)
Fill the holes of the binary image obtained from the previous step
Find lines using Hough transform in non-playfield pixels
Intersect the complement of dilated lines and playfield pixels
find the connected components and the biggest component
find the connected components and the biggest component

Fig. 1. steps of playfield detection

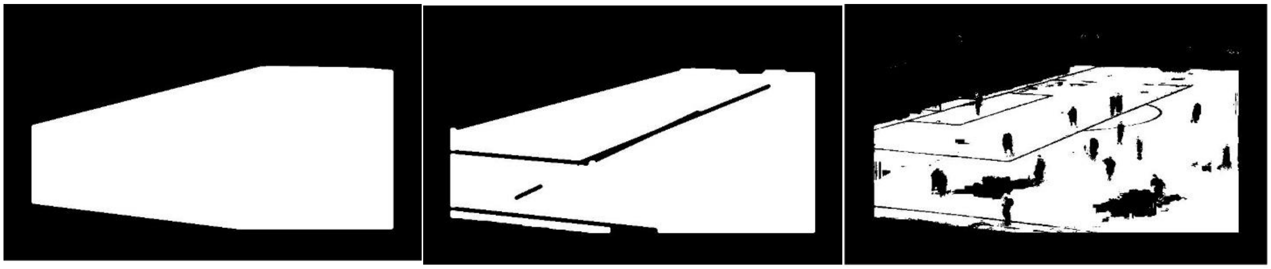


Fig. 2. playfield detection, right: image obtained from eq. (2), middle: filling image and intersecting with the complement of lines, left: removing low area object and finding the convex hull

The pixels whose colors fall within the grass color range $rgb(gc)$, are shown in black, and the others are shown in white. In the obtained image, lots of the playfield lines, shadows, and playfield regions that were mistakenly classified as non-playfield regions and parts of players are removed. The grass color range is determined in the two last equations.

The remaining lines are identified and removed using the linear Hough transformation (LHT). The obtained image is dilated using a large structuring element, and as a result, the dilated blobs cover the whole body of the players.

To remove the intercalary parts from dilated blobs,we use $E \leftarrow E \cap ed$, where *ed* is the edge of the original image (I) and E is the image obtained from step 5.

E is filled using morphological operators, and then the opening operator is applied after using erosion and thinning objects. So, the extra lines that mistakenly fell into the player's regions due to the use of the dilation operator in step 5 are removed.

The connected components (CC) of the image are obtained and put in set Ξ. The non-player objects in Ξ are detected and removed by the following equation:

$$\mathrm{O_i} = \begin{cases} \text{non} - \text{player}, & \mathrm{O_i} \cap \mathrm{C} = \emptyset \text{ , } \mathrm{INS(O_i)} > \mathrm{INT}, \\ & |\mathrm{O_i}| < \mathrm{SI} \\ \quad \text{player}, & \text{otherwise} \end{cases}$$

where O_i is an object in Ξ with size $|O_i|$, $INS(O_i)$ is the mean intensity of O_i, C is the image obtained from step 3, and INT and SI are constant values. This equation expresses that if an object has no common elements with C or has the properties of a ball (small and high brightness), it is a non-player object. The non-player objects are removed from Ξ and the remaining objects are delated. At the end of this step, the player regions are detected.

Algorithm 1 shows the pseudo-code for the proposed method for player region detection; these steps are shown in Fig. 3.

Based on the algorithm introduced in the text of the article, the time complexity of the algorithm is defined based on the region of the players in which the color pattern is searched.

Algorithm1: summary of player's regions detection steps

```
Input: $I, G, grass, rgb(gc), S, SE_1, SE_2, INT, SI$
1:  $B \leftarrow I \cap G$
2:  $grass \leftarrow grass \cap \overline{G}$
3:  $B \leftarrow B + grass$
4:  $B \leftarrow Mdn(B)$
5:  $C = \{ (x,y) | rgb(x,y) \notin rgb\ (gc),\ (x,y) \in B \}$
6:  $lines \leftarrow LHT(C)$
7:  $E = \{(x,y) | (x,y) \notin lines, (x,y) \in C\}$
8:  $E \leftarrow (E \oplus SE_1)$
9:  $E \leftarrow E \cap ed$
10: $Fi \leftarrow fill(E)$
11: $OP \leftarrow (Fi \ominus SE_2) \circ SE_2$
12: $Obj \leftarrow CC(Op)$
13: $for\ i \leftarrow 1\ to\ length(Obj)\ do$
14:   $O_i = Obj[i]$;
15:   $if\ O_i \cap C = \emptyset\ And\ INS(O_i) > INT\ And\ SZ(O_i) < SI\ then$
16:     $remove\ O_i\ from\ Obj$
17:   $end\ if$
18: $end\ for$
19: $PLAYER = \{ (x,y) | (x,y) \in list(2), (x,y) \in Obj\}$
20: $PLAYER = PLAYER \oplus SE_2$
```

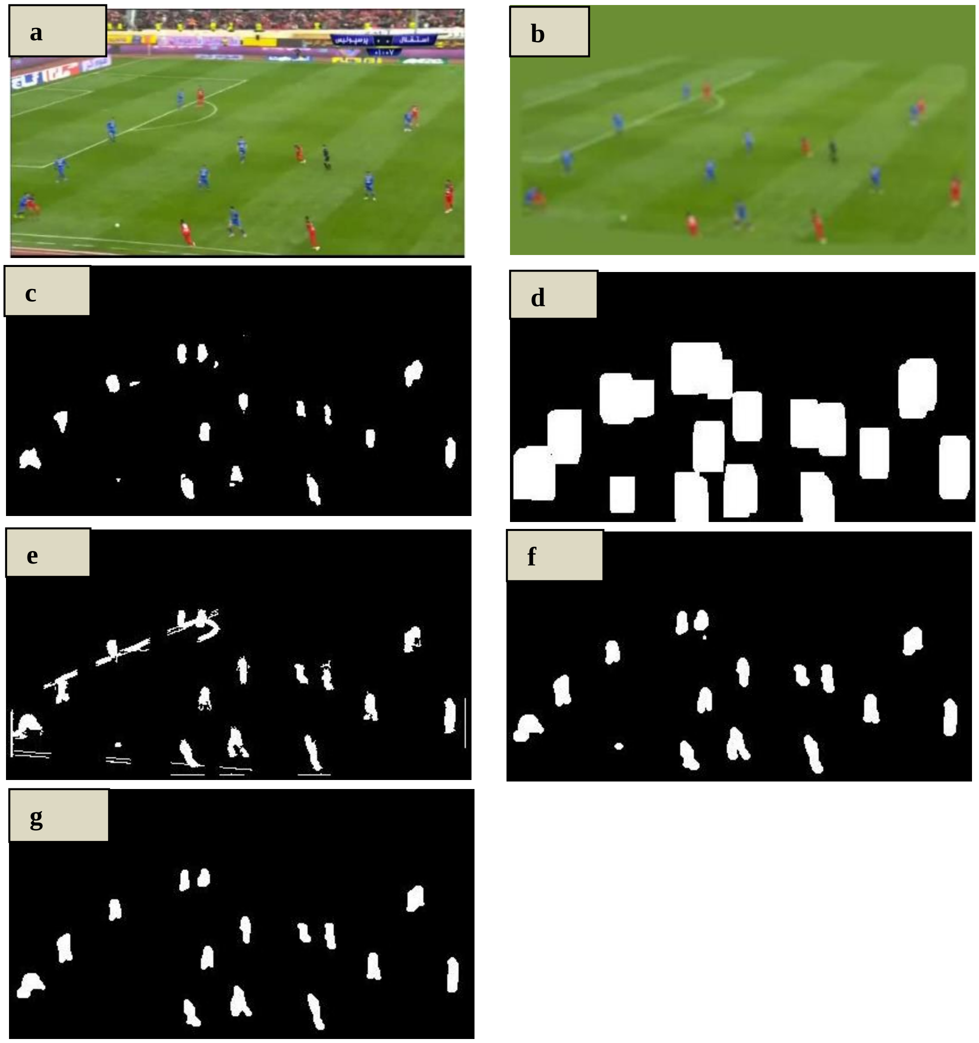


Fig. 3. player region detection steps. a) the original image. b) bluring image and output of step 2. c) output of step 3. d) dilation and output of step 5 e) output of step 6 f) erosion and opening g) removing the objects which have no intersect with image c and final result

C. *Identifying the teams of the blobs*

Identifying the teams of the players in soccer videos is often a challenging task. The difficulties emanate from the existence of similar colors in the uniforms of two teams, the crowd of players in the area, and the presence of more than one player in one detected blob. To resolve these challenges in this paper, we use the genetic algorithm to identify the players' teams.

As seen in Fig.4, the genetic algorithm's inputs include a blob's sub-image and its binary mask, as well as the colors of two teams' jerseys (as predetermined models), and the output has labeled the input blob's mask. We should define task-appropriate evolutionary algorithm settings in order to solve the player labeling challenge.

The fitness function of each chromosome in the evolutionary algorithm displays how well its color, pixel density, and shape match the colors of the two teams' uniforms. Each chromosome is a window that rests on a single blob. The suggested function combines characteristics like the white pixel density in the binary mask of the window, the aspect ratio of the window, and the color similarity between the picture in the window and certain predetermined models.

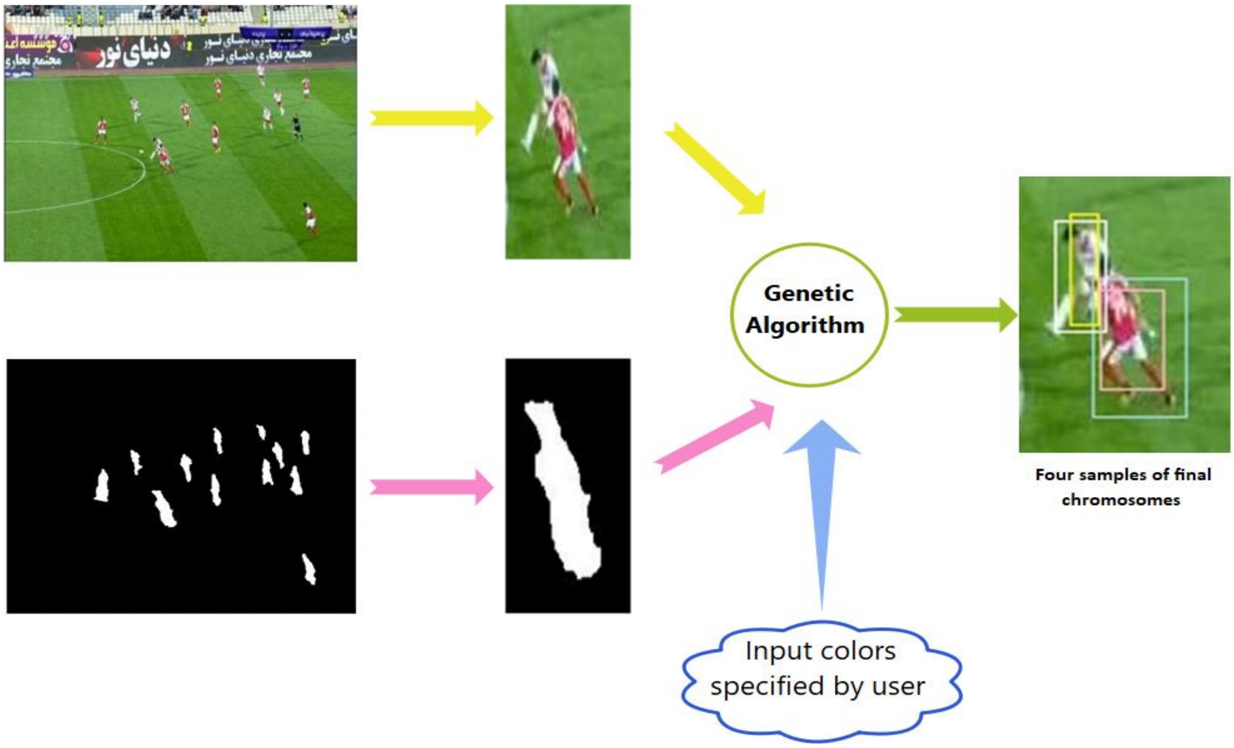

Genetic Algorithm
Four samples of final chromosomes
Input colors specified by user

Fig. 4. inputs and outputs of Genetic Algorithm

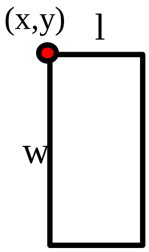


Fig. 5. window of a chromosome

Chromosomes (representation)

Each chromosome (solution) is composed of four genes: x, y, w, and l, which express the features of a rectangular window, (x, y) is the position of the window in the input image, and w and l are width and length of the window, respectively. To exclude irrational sizes, the genes of each chromosome must satisfy the following conditions:

$$\begin{aligned} & x + w <= W \ , \\ & y + l <= L \end{aligned} \tag{3}$$

where W and L are the width and length of the input image, respectively. The scheme of each chromosome is shown in Fig. 5:

The population is initialized with random individuals.

Building predefined models from input colors

The predefined models are made using colors specified by the user. Usually, the first half of the model is considered a shirt color, the next 1/4 of the model is considered a short color, and the next 1/4 is considered a sock color. For example, if the user specifies the colors of the shirt, shorts, and socks of team A as red, white, and red and specifies the colors of the shirt, shorts, and socks of team B as red, white, and red, respectively, the formed models in three different angles in RGB color space are shown as follows:

Genetic Algorithm operators

We use one-point crossover in the proposed method. The parents are selected randomly. The offspring inherit the position (first and second gene) from one parent and inherit the size (third and fourth gene) from the other. Also, the mutation is carried out by random resetting. The new population is checked to satisfy the conditions. After that, they are evaluated and scored based on their fitness, and the individuals with the highest scores are chosen for the next generation. After an appropriate number of iterations, 10 of the best chromosomes determine the team of the different regions in the blob.

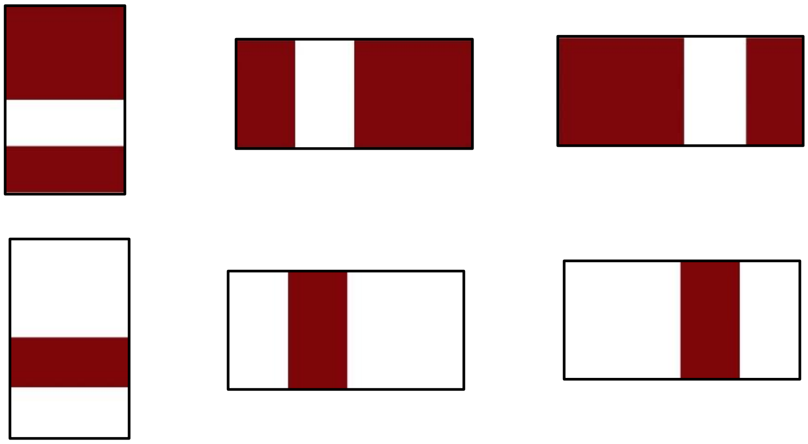


Fig. 6. Models made for two different uniforms at three different angles

Sometimes, two or more windows assign different teams to one region. To overcome this problem, two strategies can be used:

1. Priority Strategy: The window that has the most fitness determines the related team in the region.
2. Majority Strategy: For each team, the scores of the fitness of the related chromosomes are summed, and the team with the highest score is identified as the related team of that region.

Fitness function

Table I demonstrates the parameters that are considered in the calculation of the fitness function along with their descriptions.

TABLE I.

the Parameters Used in Calculating Fitness Function and Their Descriptions

parameter	Description
σ	the maximum color similarity between the image in the window and the predefined models
ω	the white pixels density in the binary mask of the window which shows the pixels related to the player in the window of chromosome
ώ	the white pixels density of the top of the body
β	The bigness of the window
ρ	the aspect ratio of the window

The parameter σ is defined as the maximum color similarity between the image in the window and the predefined models. At first, the models are made to the size of the rectangular window of the chromosome. Then the intersection of the image in the window and the models with the binary mask of the window is obtained. Let BI be the intersection of the image in the window and the binary mask of the window, and BM be the intersection of the model a and the binary mask of the window. Also, suppose R_1, R_2, R_3 are constant values that determine the importance of each channel in the HSV color space, h_{ij} ،s_{ij}, v_{ij} determine the values of H, S, and V channels of pixel (i, j) of the BI in HSV color space respectively, h_{ij}^{p} ،s_{ij}^{p} and v_{ij}^{p} be the values of H, of S and V channels of the pixel in (i, j) position of the BM, and $(mh،\ ms, mv)$ and $(mh^p ، ms^p,\ mv^p)$ be the mean values of H, S, and V channels of BM and BP, respectively. Then, we define

$$dhsv^p = \frac{R_1 \times (1 - \frac{(\sum_{i=1}^{N} \sum_{j=1}^{M} |h_{ij} - h_{ij}^{p}|)}{W})^p + R_2 + R_3 \times (1 - \frac{(\sum_{i=1}^{N} \sum_{j=1}^{M} |v_{ij} - v_{ij}^{p}|)}{W})^p}{R_1 + R_2 + R_3} \tag{4}$$

And

$$dm^p = \frac{R_1 \times (1 - |mh - mh^p|)^p + R_2 \times (1 - |ms - ms^p|)^p + (1 - |mv - mv^p|)^p}{R_1 + R_2 + R_3} \tag{5}$$

Based on these definitions, the σ parameter is calculated using the following equations:

$$\sigma = \min_{1 \leq p \leq 6} \frac{3 \times dhsv^p + dm^p}{4} \qquad (6)$$

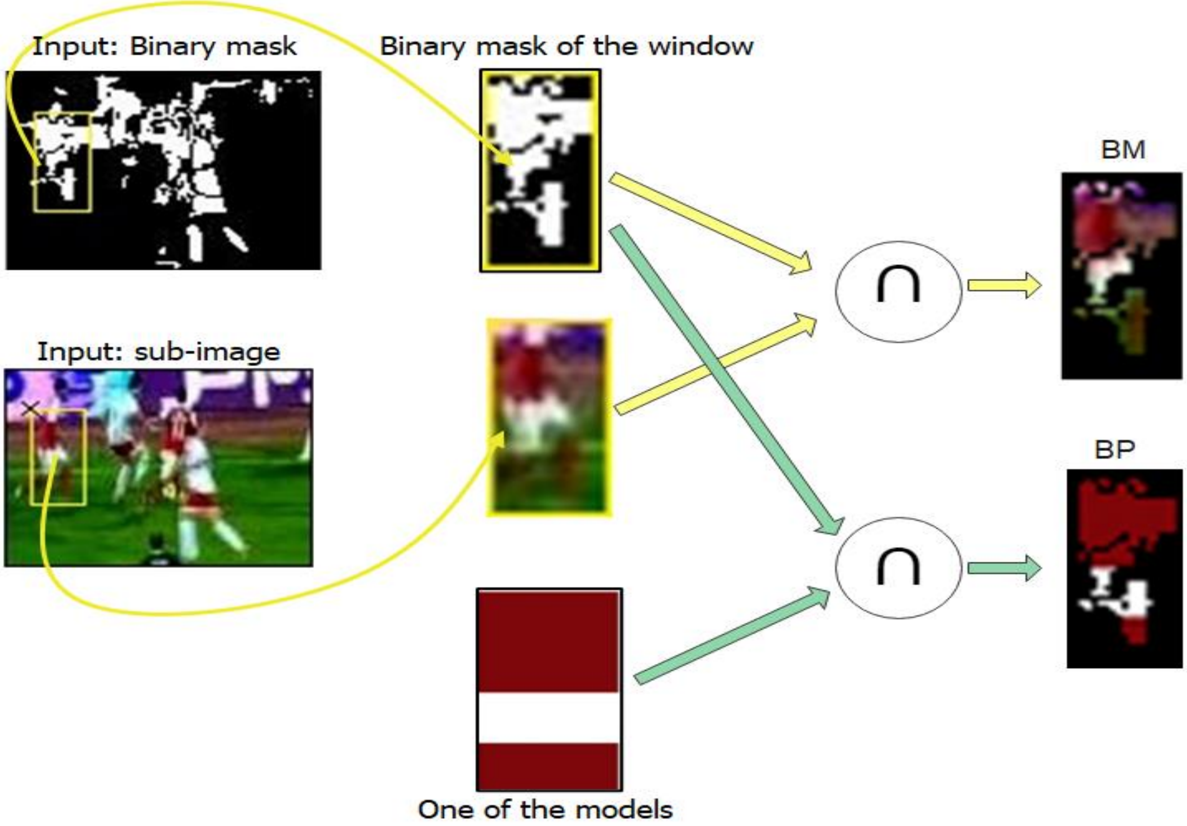


Fig. 7. The method of comparing the chromosome image with the models

The value of the model has the least difference from the image in the window. The comparison method is shown in Fig. 7.

The ω parameter determines the density of white pixels in the binary mask of the window, which shows the pixels related to the player in the window of the chromosome. Suppose $blobim(x, y)$ is the value of pixel (x, y) in the binary mask of the window. Also, suppose $S = \{(x, y) \mid blobim(x, y) = 1\}$, then the parameter ω is calculated as:

$$\omega = \frac{number\ of\ pixels\ in\ S}{w \times l} \qquad (7)$$

The density value in the top of the player's body is more important than the value in the foot part. So, we obtain the density of the top of the body as well and denote it by ώ.

The density value in the fitness function makes the final results small. To overcome this problem, we should apply the window size value in the fitness function. This value is defined as follows:

$$\beta = \frac{\frac{w}{W} + \frac{l}{L}}{2} \qquad (8)$$

The ρ parameter denotes the aspect ratio of the window. The closer the shape of the window to the square, the smaller the ρ value. This parameter is calculated by the following equation:

$$\rho = \begin{cases} 1 - \frac{w}{l}, & w < l \\ 1 - \frac{l}{w}, & w > l \end{cases} \qquad (9)$$

Since the players are more likely to stand in the football frames, we give more scores to the windows that their widths are longer than their lengths. This value is a constant that is defined as:

$$VE = \begin{cases} 0, & w < l \\ C_6, & w > l \end{cases} \quad (10)$$

Finally, the fitness function is defined based on the linear combination of six parameters σ, ω, ώ, β, ρ, and VE, as:

$$f(x) = \begin{cases} C_7 \times fittness, (w < l \, , edge_h > edge_v) \;\; \boldsymbol{O}) \\ fittness, \quad otherwise \end{cases} \quad (11)$$

where

$$fittness = C_1 \times dfinal + C_2 \times white + C_3 \times white1 + C_4 \times big + C_5 \times ratio + VE \quad (12)$$

where $C_i, \; i = 1, \ldots, 7$ are the constant values, and $edge_h$ and $edge_v$, are the horizontal and vertical edges of the input binary mask, respectively.

IV. EXPERIMENTS AND RESULTS

A. player region detection

The proposed methods were implemented using MATLAB 2015 and tested on 50 different video frames of the match between Esteghlal and Persepolis played at Azadi Stadium in Iran in 2017. In total, there are 563 players in these frames. The results are shown in Fig. 8.

In Fig. 8, three images on the left are original images, and the images on the right are the extracted region players obtained from the proposed method for player region detection. Table... shows the confusion matrix earned from applying the proposed method. The matrix demonstrates that the suggested approach can accurately identify 25 player absence zones and 519 player regions. Additionally, the criteria for accuracy, precision, and recall reveal scores of 92.5%, 98.2%, and 93.6%, respectively.

This algorithm correctly removes the shadows of players and playfield lines. Balls that are overlapped with players and parts of playfield lines are among the objects that are mistakenly identified as players. Most of the errors in this algorithm are due to the non-detection of players located on the middle line of the playfield and the non-detection of players on the edges of the playfield, which depend on the position of the camera relative to the playing field. Assume Fig. 8 to understand this. Suppose the cylinders represent the players. In both images, the positions of the players are equal. But the position and angle of the camera in image b make half of the bodies of players b, c, and e out of the field and they may not be identified. On the other hand, only player B may not be identified in image A. Also, the size of the structuring element used in step 5 of the proposed method in Section 3-1 affects the final results. As shown in the red ellipse shown in image (d) in Fig. 4, the size of the structuring element places a portion of the playfield lines in the player's region, and a smaller size of the structuring element fixes the problem. On the other hand, the player's foot specified in the image (s) is lost due to the small size of the structuring element. In these results, the referee's error is not considered an error since it can be identified and removed using a color histogram in the next steps. Despite the problems mentioned, this algorithm shows good results and correctly recognizes 82% of the players.

B. player team labeling

The proposed method for player team identification is implemented in MATLAB. Some results are shown below:

On 40 separate video frames from the 2017 matches between Esteghlal and Persepolis and Padideh-Persepolis played in Azadi, Iran, the suggested method is evaluated. There may be more than one player overlapping in each sub-image. 84 players are included in the frames. The following optimal values for the channels h, s, and v were found experimentally after repeatedly executing the suggested algorithm:

TABLE II
The Optimal Values

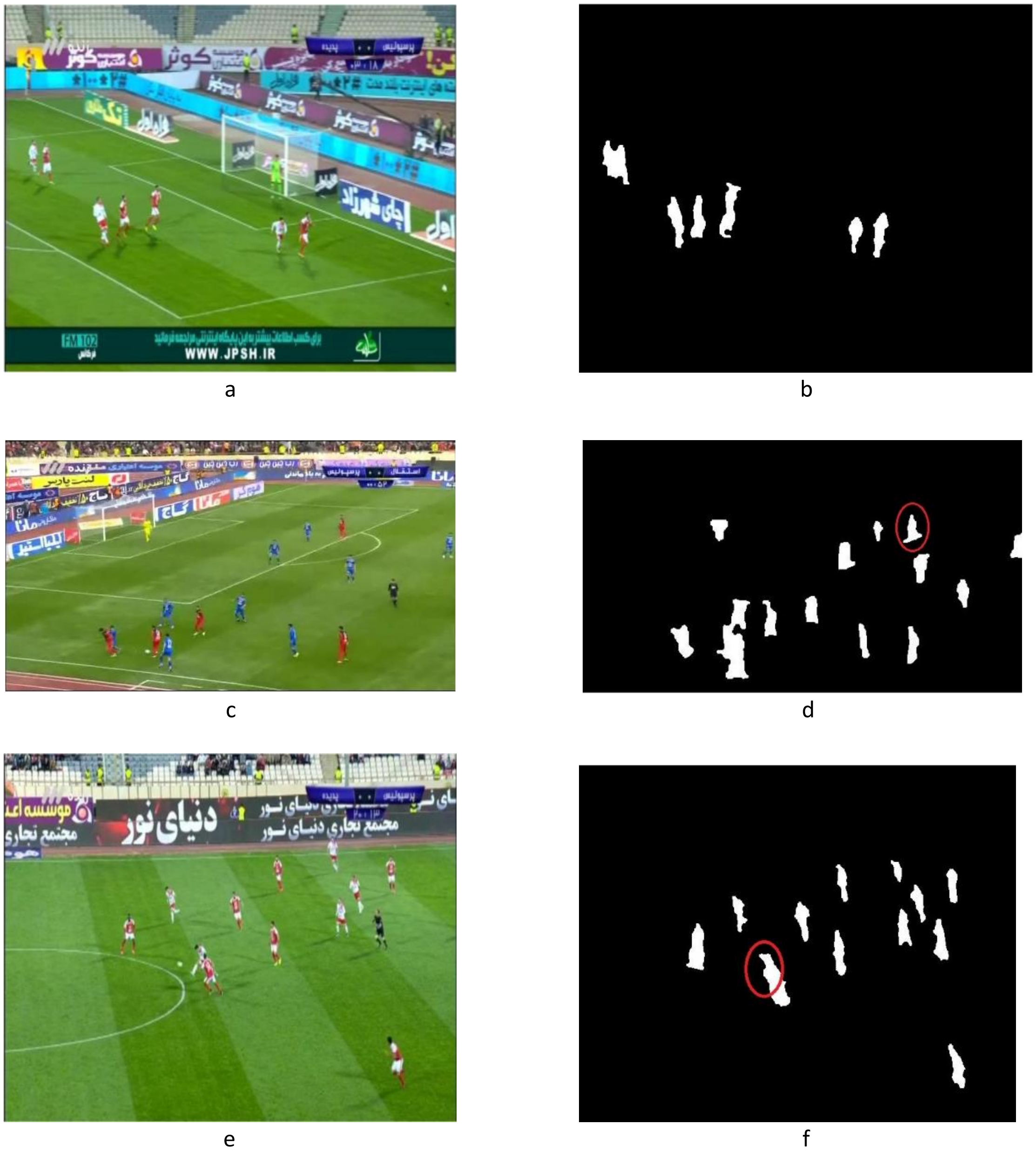
R_1	R_2	R_3
15	20	20

Also, the following values for the fitness function show better results, Table III.

TABLE III.
Optimal Values Obtained For The Coefficients Of The Fitness Function

C_1	C_2	C_3	C_4
2.31	1.1	0.9	0.41
C_5	C_6	C_7	
0.4	0.1	1.6	

- Initial population = 30
- crossover: one-point crossover
- Mutation: random resetting
- End condition: reach 5 iterations



a

b

c

d

e

f

Fig. 8. results obtained from the proposed method for player region detection

Using the priority strategy in determining the team related to the player increases the accuracy of the results. The implementation results are shown in Table IV.

This method shows high accuracy, but the main problem is that it is time-consuming and depends on various factors such as the number of iterations and initial population. To improve this approach, since it is easier to identify the team of players with completely different colors, fewer iterations and fewer populations can be used. Also, the results depend on the input colors entered by the user. However, the algorithm shows good results and correctly detects

91.6% of the teams. This algorithm largely solves the problem of team detection for overlapping players whose uniform colors are similar. Additionally, the results may be strengthened and improved by including various angles for the models and various color schemes for each team, in contrast to the approach employed in the research of 39 teams of players with various positions and angles that have identical colors.

TABLE IV
the Results Obtained for the Proposed Method of Player Team Identification

	Total number of images	Percentage correctly recognized in 10 repetitions
Non-overlapping players with completely different colors	10	100
Overlapping players with completely different colors	23	96.6
Non-overlapping players with similar colors	13	91.3
Players with overlapping and similar colors	38	87.4
Total players in the image	84	91.6

V. CONCLUSION

In this paper, we propose a method for player detection and labeling in broadcast soccer videos. For this purpose, we first detect player regions using an object-sieve method and then identify the teams related to player regions using a genetic algorithm. The proposed algorithm is implemented in MATLAB 2015, and the obtained results show a 98.2% value of accuracy for the player region detection algorithm and a 91.6% value of accuracy for the team detection algorithm. The proposed method for player region detection can identify players in different positions, including running, standing, and tackling, but players who are standing at the border of the playfield in images may not be detected. The final results of this algorithm depend on the size of the structuring element in morphological operations and the position of the camera relative to the playfield. With excellent accuracy, the genetic algorithm recognizes teams that have overlapping players whose colors overlap. The output of this method is determined by the user-specified input colors and preset models.

Those interested in this field are advised to use other algorithms, such as Ant Colony, that are compatible with the nature of the problem to solve similar problems instead of using the genetic algorithm. Also, in the next research, the accuracy of each player's pass can be obtained individually, or by extracting the pass patterns and players' movements, the team's tactics and performance can be evaluated, and more information can be obtained automatically.

VI. REFERENCES

[1] C. Cuevas, D. Quilón, and N. García, “Techniques and applications for soccer video analysis: A survey,” *Multimed. Tools Appl.*, vol. 79, no. 39–40, pp. 29685–29721, 2020, doi: 10.1007/s11042-020-09409-0.

[2] N. Alijanpour, H. Nobari, L. Bolboli, R. Afroundeh, and A. Garcia-Ramos, “Using Global Positioning System to Compare Training Monotony and Training Strain of Starters and Non-Starters across of Full-Season in Professional Soccer Players,” *Sustain.*, vol. 14, no. 6, 2022, doi: 10.3390/su14063560.

[3] L. Morra, F. Manigrasso, and F. Lamberti, “SoccER: Computer graphics meets sports analytics for soccer event recognition,” *SoftwareX*, vol. 12, 2020, doi: 10.1016/j.softx.2020.100612.

[4] R. Gholizadeh, H. Nobari, L. Bolboli, M. Siahkouhian, and J. P. Brito, “Comparison of Measurements of External Load between Professional Soccer Players,” *Healthcare*, vol. 10, no. 6, p. 1116, 2022, doi: 10.3390/healthcare10061116.

[5] M. Kaushal, B. S. Khehra, and A. Sharma, “Soft Computing based object detection and tracking approaches: State-of-the-Art survey,” *Appl. Soft Comput. J.*, vol. 70, pp. 423–464, 2018, doi: 10.1016/j.asoc.2018.05.023.

[6] R. Singh, A. Kumar Singh Kushwaha, Chandni, and R. Srivastava, “Recent Trends in Human Activity Recognition- A Comparative Study,” *Cogn. Syst. Res.*, Oct. 2022, doi: 10.1016/j.cogsys.2022.10.003.

[7] M. Gholizade, H. Soltanizadeh, and M. Rahmanimanesh, “A Survey of Transfer Learning and Categories,” *Model. Simul. Electr. Electron. Eng.*, vol. 1, no. 3, pp. 17–25, 2021, doi: 10.22075/mseee.2021.23310.1062.

[8] U. M. Rao and U. C. Pati, “A novel algorithm for detection of soccer ball and player,” *2015 Int. Conf. Commun. Signal Process. ICCSP 2015*, pp. 344–348, 2015, doi: 10.1109/ICCSP.2015.7322903.

[9] M. Heydari and A. M. E. Moghadam, “An MLP-based player detection and tracking in broadcast soccer video,” *2012 Int. Conf. Robot. Artif. Intell. ICRAI 2012*, pp. 195–199, 2012, doi: 10.1109/ICRAI.2012.6413398.

[10] K. Zhang, J. Wu, X. Tong, and Y. Wang, “An automatic multi-camera-based event extraction system for real soccer videos,” *Pattern Anal. Appl.*, vol. 23, no. 2, pp. 953–965, 2020, doi: 10.1007/s10044-019-00830-2.

[11] J. Liu *et al.*, “Automatic player detection, labeling and tracking in broadcast soccer video,” *BMVC 2007 - Proc. Br. Mach. Vis. Conf. 2007*, 2007, doi: 10.5244/C.21.3.

[12] S. Nag, P. Shivakumara, U. Pal, T. Lu, and M. Blumenstein, “A new unified method for detecting text from marathon runners and sports players in video (PR-D-19-01078R2),” *Pattern Recognit.*, vol. 107, 2020, doi: 10.1016/j.patcog.2020.107476.

[13] S. Gerke, S. Singh, A. Linnemann, and P. Ndjiki-Nya, “Unsupervised color classifier training for soccer player detection,” *IEEE VCIP 2013 - 2013 IEEE Int. Conf. Vis. Commun. Image Process.*, 2013, doi: 10.1109/VCIP.2013.6706424.

[14] M. Manafifard, H. Ebadi, and H. A. Moghaddam, “Multi-player detection in soccer broadcast videos using a blob-guided particle swarm optimization method,” *Multimed. Tools Appl.*, vol. 76, no. 10, pp. 12251–12280, 2017, doi: 10.1007/s11042-016-3625-6.

[15] Y. Yang and D. Li, “Robust player detection and tracking in broadcast soccer video based on enhanced particle filter,” *J. Vis. Commun. Image Represent.*, vol. 46, pp. 81–94, 2017, doi: 10.1016/j.jvcir.2017.03.008.

[16] K. Lu, J. Chen, J. J. Little, and H. He, “Lightweight convolutional neural networks for player detection and classification,” *Comput. Vis. Image Underst.*, vol. 172, pp. 77–87, 2018, doi: 10.1016/j.cviu.2018.02.008.

[17] W. Kim, S. W. Moon, J. Lee, D. W. Nam, and C. Jung, “Multiple player tracking in soccer videos: an adaptive multiscale sampling approach,” *Multimed. Syst.*, vol. 24, no. 6, pp. 611–623, 2018, doi: 10.1007/s00530-018-0586-9.

[18] Q. Tran, A. Tran, T. B. Dinh, and D. Duong, “Long-view player detection framework algorithm in broadcast soccer videos,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6839 LNAI, pp. 557–564, 2011, doi: 10.1007/978-3-642-25944-9_72.

[19] M. A. Montañés Laborda, E. F. Torres Moreno, J. Martínez del Rincón, and J. E. Herrero Jaraba, “Real-time GPU color-based segmentation of football players,” *J. Real-Time Image Process.*, vol. 7, no. 4, pp. 267–279, 2012, doi: 10.1007/s11554-011-0194-9.

[20] W. L. Lu, J. A. Ting, J. J. Little, and K. P. Murphy, “Learning to track and identify players from broadcast sports videos,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1704–1716, 2013, doi: 10.1109/TPAMI.2012.242.

[21] L. H. Chen, C. W. Su, and H. A. Hsiao, “Player trajectory reconstruction for tactical analysis,” *Multimed. Tools Appl.*, vol. 77, no. 23, pp. 30475–30486, 2018, doi: 10.1007/s11042-018-6164-5.