Zonal Distribution of Player Locations based on Football Goal Scene

Kumud Tayal
2019429
kumud19429@iiitd.ac.in

Bhawan Gakhar
2019417
bhawan19417@iiitd.ac.in

Yash Verma
2019456
yash19456@iiitd.ac.in

April 30, 2022

Abstract

Data Analysis is being increasingly used in teams to understand the underlying statistics and tactics. Within football, player positions can be used to understand the concept of 'space,' which is necessary for attack and defense. In the following sections, we automate the process of converting football images to top view 2D images that can directly be used in the analysis. We use deep learning models such as YOLO and some computer vision techniques to arrive at final results. The proposed approach aims to segregate players from their surroundings and differentiate between players of different teams based on the RGB values within them. Next, using the projection matrix and transformation equation, we convert the input images into desired outputs. The desired outcomes consist of player positions in each zone and the number of players within each zone.

1 Introduction

The center circle, along with the entire width of the halfway line, is a necessary prerequisite for the working of the algorithm described. We take general football images and convert them into a 2D top view plane where players would be identified as a point on the field and then counting the number of players in each zone which could be used in further analysis by football experts for better understanding of tactics, develop a better defense for their team and help in working on their mistakes.

Input used are gameplay images from FIFA wherein most of the action occurs near the midway, and the halfway line is clearly evident, along with enire width of halfway line. Output would be the image of a predefined 2D football field divided into various zones and the number of players in each zone. The input image is masked for two conditions: field line detection, which uses masking for white color, and for team detection, wherein green color masking is done.

2 Related Works

Much research has been done on football tracking, converting football to 2D viewpoints, detection of field lines[6], offside detection[4], goal detection[5], player detection, and team detection. Most of the papers have similar steps involving removing noise, detecting lines, and using edges that have been detected. Primarily, documents have used Hough transformation to detect lines that are further used to detect football fields[6]. For player detection different approaches are available.

[3] used HOG descriptors and SVM classification to detect football players. [2] used selective search for the same. [7] used multiple view cameras and matched the data for each player plotted using Tsai's Algorithm. [2, 3] used only one view images, whereas [7] used 8-view images. Our novelty lies in the idea of

Figure 1: **Input**

dividing the field into various zones and calculating the players in each zone. In the future, we can expand our concept to not only the midline and the entire field area.

3 Methodology

The presence of halfway lines helps in player detection manipulation and derivation of the transformation matrix to acquire their mappings in the result.

3.1 Mid Field Line and Center Circle Detection

The circle detection involves taking a masked image wherein the white color pixels within a certain threshold have been shown as white and the rest as black. The next step requires detecting all edges and vertical edges (for the middle line), then the vertical edges are then dilated to add up pixels in the boundaries detection.The largest connected component in the vertical edge binary image is subtracted from the all edges image which would help remove the middle line and help in giving isolated edges detected around of circle. We used Morphological operations on the subtracted image obtained where the structuring element is a disk. After which, we drew a bounding box around the circle using region props.

Line detection also involves taking the masked image, converting it to binary, then dilating it using a rectangle as the structuring element to increase the boundary points around the middle line for easy detection. After obtaining the binary image with a visible straight line, it is detected using Hough Transform. Out of all lines obtained from Hough Transform, the point with minimum y coordinate and maximum y coordinate is being used to detect the middle line's endpoint and starting point.

3.2 Player and Team Detection

To extract player positions on the field and their coordinates, we used the deep learning model YOLO. To supplement player detection for better accuracy, we used the SAHI library [1] that uses the slicing technique over images for better object detection. Out of all detected objects, objects having 'person' labels were segregated, and only those detections with accuracy over comparable limits were differentiated as 'players.'

Using the player locations extracted from YOLO, we created a bounding box over players and separated foreground and background. This differentiation was done using conversion to HSV space and masking green color that occupied most of the background in these bounding boxes. The separated foreground was then used to calculate the mean RGB value that varied among players of different teams but was almost equivalent to members of the same team. After calculating the mean RGB values of all players, we segregate these values into two groups representing two different teams using the Fuzzy C-Means algorithm. The detected teams for each player, along with their coordinates, are sent used for collecting data necessary for the final results.

3.3 Transformation into 2D Topview

To transform from real-time image to the predefined 2D topview, we have calculated a projection matrix which transforms the co-ordinates from the real time image to the coordinates in 2D topview.

The projection matrix looks like this:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

As seen above the projection matrix consists of 8 variables, so minimum of 4 co-ordinates and their corresponding projected co-ordinates must be known to solve the projection matrix.

The transformation equation is :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Where x & y are coordinates from input image and u & v are transformed coordinates.

By using the above mentioned equation, we would have to form a system of equation and solve them for the 8 variables.

But instead we created a matrix equation and solved it.

The equation looks like this:

$$\begin{bmatrix} u1 & u2 & u3 & u4 & u5 & u6 \\ v1 & v2 & v3 & v4 & v5 & v6 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x1 & x2 & x3 & x4 & x5 & x6 \\ y1 & y2 & y3 & y4 & y5 & y6 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Where xi & yi are coordinates from input image and ui & vi are transformed coordinates.

3.4 Zone Formation

The pre-defined field is divided into 18 zones. We hvae divided the field from scratch by plotting lines. The size of the grid depends upon the size of the image. Here, we have taken the size of our predefined football field of 3000 X 6000 pixels. So each grid is of size 1000 x 1000 pixels.

Based on the positions of players detected using YOLO each player is maped to its respective zone and the count of each zone is maintained.

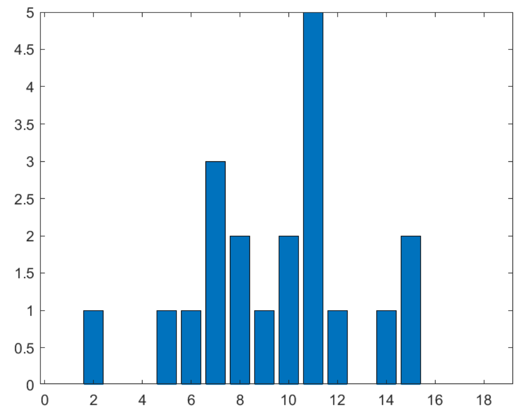

Figure 2: **Number of players in each zone**

4 Experiments

4.1 Field Line and Circle Detection

We tried out various methods for circle detection, including using the imcircle command, which uses hough transform for circle detection, houghcircles function in python, using connected components in the binary images, blob detection, finding out the contours, etc.

The main problem behind using built functions was the shape of the midfield circle and the angle at which it was present in the image.

The main obstacle behind finding the line was the argument that the houglines take the angle at which the line was present. Many noisy lines were found while trying out the function, which we removed either by masking out the field or keeping lines of a certain length.

4.2 Player and Team Detection

Using YOLO without any additional support brought inefficiency in detecting player locations. Some players missed out on being detected, and sometimes two players close to each other would be detected by only one bounding box. To increase

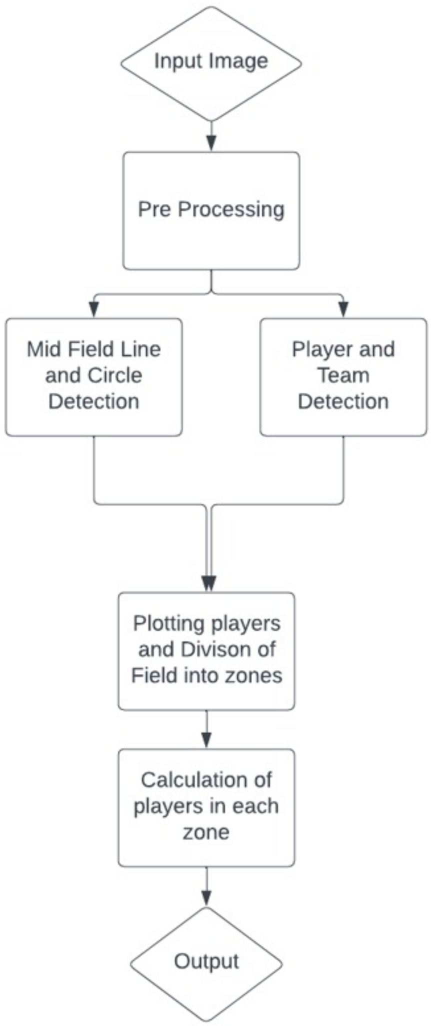

Figure 3: **Methodology**

Figure 4: **Detecting Circle**

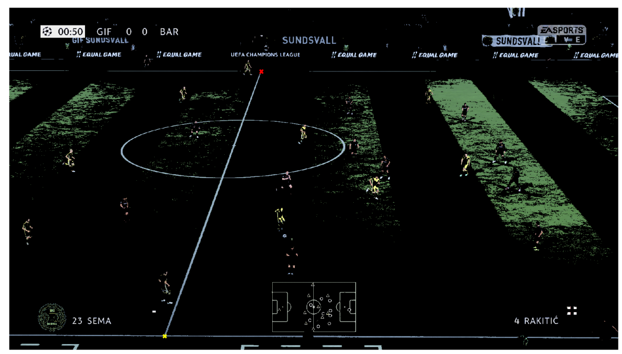

Figure 5: **Detecting Lines**

efficiency, we used the SAHI vision library [1] that uses the slicing technique. The entire image is divided into the number of slices from which object detection is done individually, and results are compiled together as one output.

Image segmentation (using imsegmeans and superpixels) was tried to separate the green background from the rest of the foreground. However, this resulted in inaccurate results as sometimes players' complexion varied from each other, and the orientation and the amount of jerseys apparent in the image varied, which drastically affected the results. Thus, the concept of mean RGB values over players was chosen and implemented to minimize these impacts. To segregate these mean values, clustering methods like K-means and Fuzzy C-Means clustering were tested over the dataset.

4.3 Projection Matrix and Plotting of players

Using the equations mentioned in the section 3.3, first we had to calculate the projection matrix.

Let us assume the projection matrix to be A, the co-ordinates from the real-time image be X and transformed co-ordinates be Y. X & Y are also matrices with each column representing a co-ordinate. So, the equation looks like this:

$$Y = A.X \tag{1}$$

So, using *pinv()* the inbuilt function in matlab, we calculated X^-1 and multiplied to both sides in equation 1.

$$Y.X^{-1} = A.(X.X^{-1}) \tag{2}$$

$$Y.X^{-1} = A.I \tag{3}$$

As $X.X^{-1}$ becomes Identity matrix. Hence from equation 3 we get,

$$A = Y.X^{-1} \tag{4}$$

After we get **A**, which is the projection matrix, we can then map the players in real-time image to their respective co-ordinate in the predefined 2D top view.

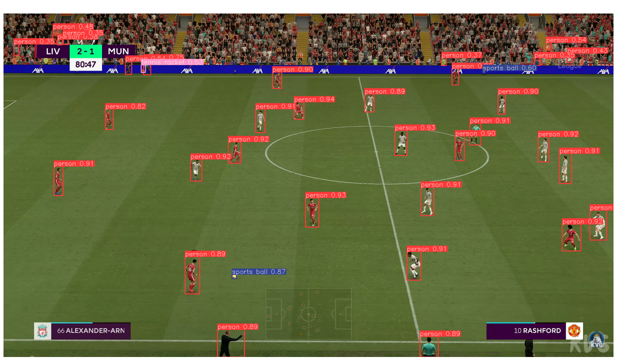

Figure 6: **YOLO Output**

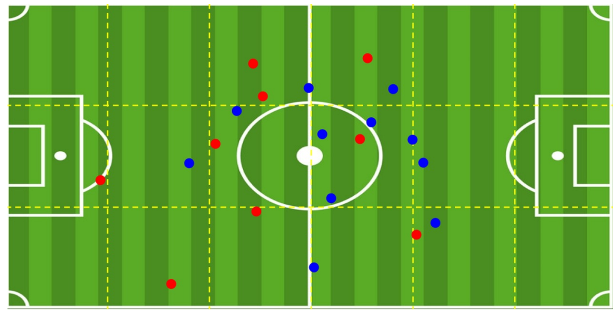

Figure 7: **Output**

We could have calculated the projection matrix using only 4 known co-ordinates which would bound the center circle, but the results were not so good. Players which were apart from the center circle were getting out of frame after projecting the matrix. The players near the center circle were getting projected properly.

Hence to avoid the projections getting out of frame we took the edge points of the halfway line also, due to which all players were getting projected properly, within the frame.

5 Conclusion

Our proposed methodology works well in most cases: detecting players through YOLO with high enough efficiency, locating and segregating players into teams, extracting and creating bounding boxes around the halfway line and center circle, and transforming these images into resulting outputs.

Although the approach is currently naive and intuitive, the idea and methodology provide a unique solution to the problem. There are certain shortcomings due to noise detection (including the person from beyond boundaries, variance in player locations after transformation matrix, the difference in team detected.)

More work and ideas can be used to make this methodology more robust and with greater efficiency. Our approach can be extended to all possible scenarios with further research and further abstraction and trials on the dataset.

References

[1] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. *arXiv preprint arXiv:2202.06934*, 2022. 2, 4

[2] Gabin Kayumbi, Pier Luigi Mazzeo, Paolo Spagnolo, Murtaza Taj, and Andrea Cavallaro. Distributed visual sensing for virtual top-view trajectory generation in football videos. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 535–542, 2008. 1

[3] Sławomir Maćkowiak, Jacek Konieczny, Maciej Kurc, and Przemysław Maćkowiak. Football player detection in video broadcast. In *International Conference on Computer Vision and Graphics*, pages 118–125. Springer, 2010. 1

[4] Karthik Muthuraman, Pranav Joshi, and Suraj Kiran Raman. Vision based dynamic offside line marker for soccer games. *arXiv preprint arXiv:1804.06438*, 2018. 1

[5] Radha Penumala, M Sivagami, and Suresh Srinivasan. Automated goal score detection in football match using key moments. *Procedia Computer Science*, 165:492–501, 2019. 1

[6] Li Sun and Guizhong Liu. Field lines and players detection and recognition in soccer video. pages 1237–1240, 04 2009. 1

[7] Kazuma Tomikawa and Ryosuke Saga. Conversion of player locations from football goal scene videos to a 2d top view. In *International Conference on Human-Computer Interaction*, pages 211–218. Springer, 2018. 1