Offside Detection System in Football Matches using Human Pose Estimation

A dissertation submitted in partial fulfillment of the requirements for the award of the degree of

M.Tech.

in

Computer Science

By

Awanish Kumar (Roll No. CS2029)

under the supervision of

Dr. Ujjwal Bhattacharya

CVPR Unit, ISI, Kolkata

INDIAN STATISTICAL INSTITUTE

BT ROAD, KOLKATA – 700108, India

JULY 2022

CERTIFICATE

This is to certify that the dissertation entitled **Offside Detection System in Football Matches using Human Pose Estimation** submitted by **Awanish Kumar (Roll No. CS2029)** to **Indian Statistical Institute, Kolkata** in partial fulfillment for the award of the degree of **Master of Technology** in **Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission

Dr. Ujjwal Bhattacharya

CVPR Unit

Date: 08/07/2022

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to the following people for guiding me through this dissertation and without whom this project and the results achieved from it would not have reached completion.

Dr. Ujjwal Bhattacharya, Associate Professor, Department of Computer Science, for helping me and guiding me in the course of this project. Without his guidance, I would not have been able to successfully complete this project. His patience and genial attitude is and always will be a source of inspiration to me.

Computer Vision and Pattern Recognition Unit, Department of Computer Science, for allowing me to avail the facilities at the department.

I am also thankful to the faculty and staff members of the Department of Computer Science, my parents and friends for their constant support and help.

Awanish Kumar

AWANISH KUMAR

ABSTRACT

Offside decisions are important in context of any football game. In recent times the decision making in sports have been heavily dependent on technology. Football is no exception. Recently the offside decisions and many other similar decisions on the football field have been made by a Video Assistant Referee.These decisions have been broadly inconsistent with the referees.Also these VAR decisions can sometimes take a lot of time causing delays. We can use machine learning techniques to tackle the problem of the offside rule in football. We make use of Keypoint R-CNN so that we can perform human pose estimation of players which information can be used for offside detection and image processing techniques to detect offside in a given frame. This dissertation will tackle all the problems we encounter in the process of offside detection in an image. The dissertation presents an improved offside decision system for football match images. We have also presented various challenges that the current method faces so as to facilitate further research in this area.

Keywords : Offside Detection, Machine Learning, Neural Network, Keypoint RCNN, Human Pose Estimation, Team Classification, Image Processing

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In football, refereeing is a challenging process. We have over the past many years seen many different mistakes which have led to debate over employing alternative methods and technology to aid the referee.
The technology which made a breakthrough is the GOAL line technology. This goal line technology was helpful in making decisions which overruled the possibility of disallowing of genuine goals. The latest technology employed is the VAR which is executed by a set of referees off the field who will review the decision made by on the field referee. These can be a variety of decisions. The decisions can be about Red cards or yellow cards. They can be about checking of Goal or no goal on basis of a foul committed.
VAR was introduced on the thought that it will increase accuracy in refereeing decisions. An increase in accuracy from 82 percent to 96 percent was expected . But what we have seen is that there has been not been sizable increase in accuracy in the given decisions but also we have seen lots of delays in the decision making. This leads to loss of valuable time which may or may not be properly added by the referee in injury time.

The International Federation of Association Football (FIFA) states that a player is offside if his/her playable body part is nearer to his/her opponents' goal line than both the ball and the second last opponent's playable body part. This player in the offside position must also be involved in the gameplay otherwise he does not matter.

Offside decisions of on field referees shows a lot of error. We will try to address the problems of these errors with the use of machine learning.We are using the concept of neural networks/deep learning to get the human pose estimations of the players. We are using a modified form of masked-RCNN trained on keypoints in the COCO dataset to achieve this task. We also use image processing techniques to get a vanishing point for the lines on the football field. Then we use projection of the farthest scorable body part on the player's body for all the players. Then we can use the concept of the farthest defender or the last man where we get the farthest body part of the last defender. Then we check each attacker with respect to the last man. The attackers which are farther advanced in comparison to the farthest man is considered as offside.

We use dataset from [2] which presents a dataset of 500 images which has all kinds of situations which occur in soccer and is a robust one which can be used as an appropriate set to test any future contributions.
The offside evaluation algorithm in our paper has been changed from [2] and we have used the fact that the goalkeeper wears a completely different kind of kit from the rest of the team. So we need a method for determining which team the goalkeeper belongs to which is not manual. Currently this is out of the scope of this dissertation and we just focus on non goalkeeper players and consider an assumption where the goalkeeper is the farthest player.
Next we compare our predictions with the ground truth present with the testing dataset. Then we report the accuracy in form of F1 score.

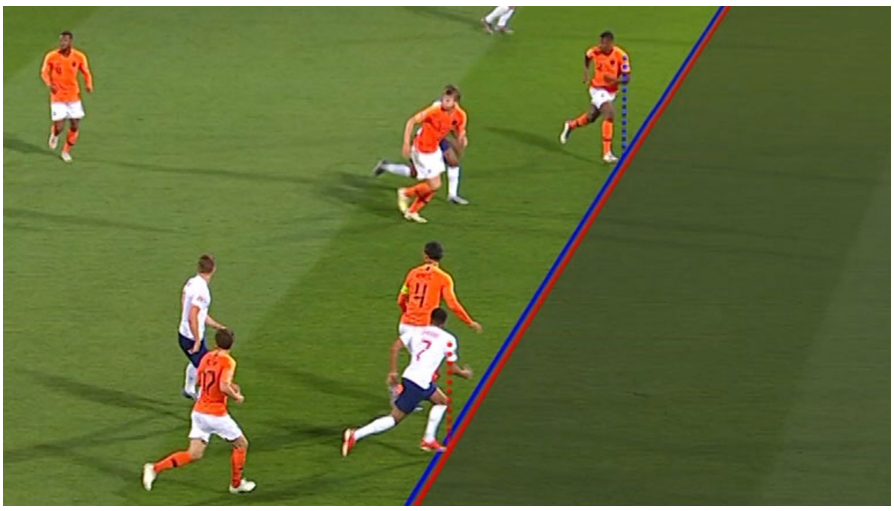

Figure 1.1: Example of an offside[**source:SKY SPORTS**]

CHAPTER 2

RELATED WORK

The study of offside decisions and their errors has been done from a long time. Various Efforts have been made to analyze and understand problems which on field referees face while making decisions[1]. The introduction of VAR has shown that we are able to get some kind of accuracy in handling the decisions made in football matches. But these checks can often lead to time delays which using different available camera angles. Some of these decisions can take as long as 5 minutes which is a significant time in a football game. These decisions made by off field referees are not always right as they can make a mistake in estimating the projection of the players on to the field. The Problems which arise due to occlusion in the referee view can be tackled using video from multiple angles and cameras [3] [4]. However, these works are not sufficently address how the necessary sub-tasks of finding body part positions of players, player tracking and classification into teams can be performed on the image data.

K. Muthuraman. et al. have used Machine learning, Computer vision with image processing to get a dynamically marked offside line [5]. In their work they give a robust method to track the players in the consecutive frames of a football match video where they use the Kanade-Lucas-Tomasi tracker and give a method to get the vanishing point so that we get the relative positions of the player.
But they did not address the important concept of using the playable parts of the body. In football the places such as wrists and elbows and arms cannot be a playable part which means that these cannot be used to score a goal. So positions of these bodyparts need to be ignored while looking at the decision making.
The complex task of offside detection automatically consists of sub tasks of player detection, human pose estimation and the team based classification. For pose estimation, many usual pose estimation techniques given in [6] and [7] can be used. In [8], the authors use a deep convolutional neural network to get the pose of the player. To perform team classification, clustering based approaches and various feature matching have been used. Also implemented is a feature matching algorithm where the authors use the information of prior knowing the colors of the jerseys[9]. This approach needs human intervention and not really automated. These can be automated using some clustering algorithms. P, Sagnolo et al. have used a histograms in the RGB space which are feature extractors, and used unsupervised

clustering to classify players in respective teams[10].
Now let us discuss the previous related work to mask-RCNN.

R-CNN: The Region-based CNN approach [11] is bounding-box detection of object so as to look at manageable number of possible object regions [12] and test convolution network on each [13, 14] RoI. R-CNN was extended [15, 16] so as to enable attending to RoIs on all feature maps using the RoIPool. This has better speed and accuracy. Faster R-CNN [17] learns the attention mechanism using a Region Proposal Network (RPN). This Faster R-CNN is very flexible and pretty robust.

Instance Segmentation: Since the success of R-CNN, the image segmentations have been done using segment proposals. Our method is primarily based on the parallel prediction of the masks and given class labels Li et al. [18] combined the outlined segment proposal system in [19] and the proposed object detection system in [20] for getting a Fully Convolutional Instance Segmentation (FCIS). The common idea here is to accurately predict a bunch of position sensitive output channels which are to be done fully convolutional. These channels will look at object classes, masks and boxes so to be able to make the system fast. But this shows errors in overlapping instances.
Some methods [21, 22] for instance segmentation are based on semantic segmentation. These methods try to divide pixels of same class into different instances.These are a segmentation first based methods. Mask R-CNN[23] is based on the better running instance-first method.

CHAPTER 3

METHODOLOGY

In this thesis we have presented a method for giving image representations of the offside rule decision in football games. We need to perform a series of tasks for achieving our goal. Here we will present the individual tasks that will need to be performed so that we get the final representation. This section shows the individual task.

We have made the pipeline keeping in mind certain things.

The images are captured using a single camera position.
The images are covering one half of the field.
The goalkeepers are assumed to be the last player at one end of the field.
It is assumed that the marking on the fields using white lines are present for facilitating of vanishing points calculation
The attacking direction is predetermined as there is no way to find this unsupervised

The first step is to find vanishing points that can be used to get the positions of the players with respect to each other. These vanishing points are then used to obtain the plane of the football field.
Every player representation has the following -
Pose estimation for the player and positions of 'key' body parts
Team ID for the player showing if the player belongs to the attacking or defending team
The projection of the farthest scorable body part on the plane of the field.
These are the three outputs from three separate tasks
Task 1 is human pose estimation of players using Keypoint RCNN
Task 2 is a clustering algorithm task where we get to cluster the players in 2 groups
Task 3 is the finding the projection of the scorable part on the plane of the soccer field determined earlier by the vanishing points.

Eventually we combine all these information and use the concept of the last defender. This concept says that if we can get a attacking player with its farthest body part projection farther than the farthest body part projection of the farthest defender, then the attacking player is offside and can be marked as such.

Also we can mark the last defender's farthest point and draw a line to the vanishing point. This will serve as the offside line. Then the final representation on the image is the last defender, offside line, offside players and defenders and we are done.

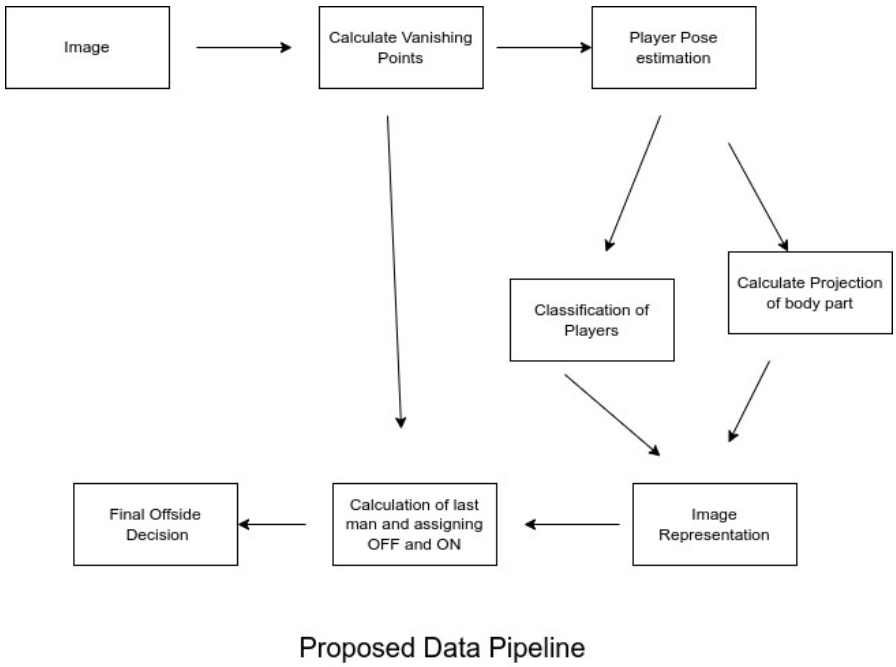

Figure 3.1: The steps involved in the execution of our offside detection system

3.1 Finding the Vanishing Point

The finding of vanishing point is the first and foremost task that needs to be done. This is important to determine the relative position of each player and their respective body parts. We can find the vanishing point using the white marking lines on the football field. These are made on the boundary of the field, in the penalty area and in the centre of the field.

First we use the Canny edge detection algorithm to detect the edges in the image.
To extract the lines the Houghline algorithm is used on the image.
Then we try to get the intersection of these lines which usually falls outside the image.
Then we can find the angles of these lines to the horizontal edge of the image.

Now we need to determine the plane of the field, both the horizontal and vertical vanishing points are calculated and both are perpendicular to one another.
For getting horizontal vanishing point, the lines which are closer to 90 degree to

the vertical are taken.

Figure 3.2: Finding the vanishing points of the lines on football field

In fig 3.2 We can see the detection of two white lines in the penalty area. We extend those lines and they meet outside the picture at a point. That point is the vanishing point

3.2 Human Pose Estimation

We use Keypoint RCNN in the human pose detection part. We know that Keypoint RCNNfollows from Mask RCNN, which arrived after Faster RCNN.
So, to explain Keypoint RCNN, we need to talk about its predecessors.

Let us take some variables at this point.

N is the number of objects given by the Region-Proposal Layer.
C is total amount of classes present in MS-COCO dataset, which is 80.
K is the number of keypoints per person present in COCO dataset, this is 17.

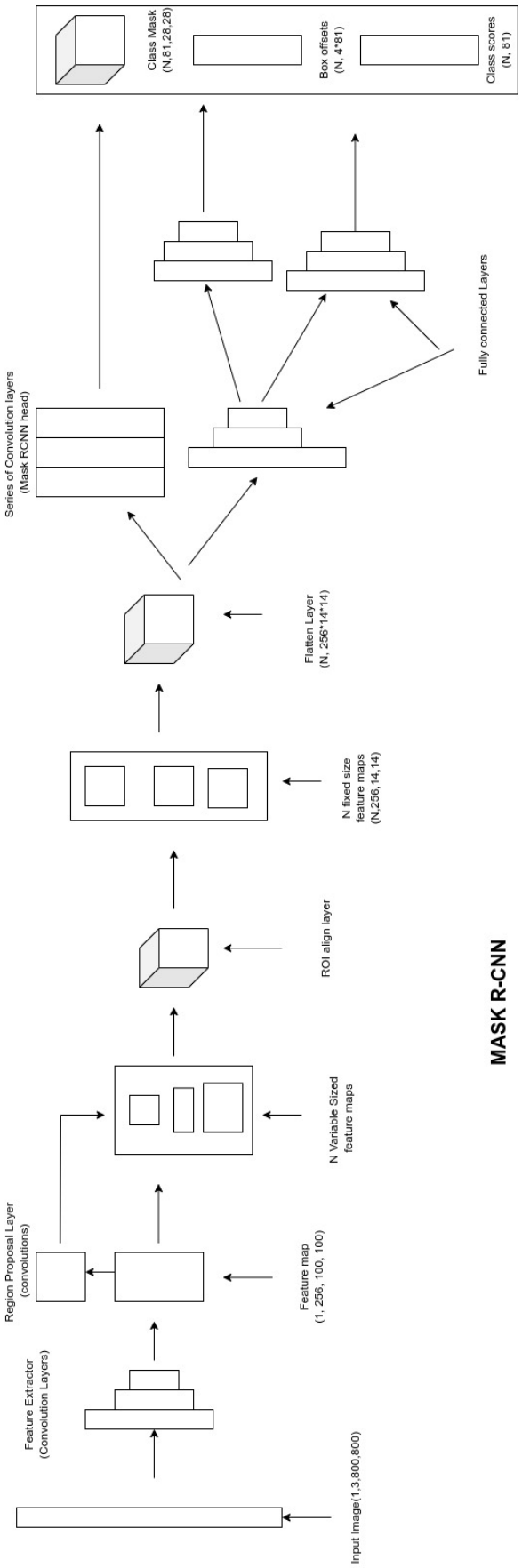

Figure 3.3: A mask RCNN network

Architecture of Mask-RCNN

The architecture of Mask-RCNN has many layers.

The Region-Proposal-Layer here predicts the approximate location of given number of objects(N) which are detected in feature map.
These many variable-sized regions are individually passed to the ROI-Align Layer.
The ROI-Pooling Layer used in Faster RCNN simply resizes the feature-map which was proposed by the RPL to a fixed size. This was done by quantising the variable sized feature map into a fixed size grid. Then we choose the max-values from the map and put them in the grid.

Object Detection needs higher-level information. Mask-RCNN uses the ROI-Align layer. ROI-Align makes use of bilinear-interpolation to put the values in the fixed grid. The generated output the of ROI-Align is given to a branch called Mask-RCNN head .
This is just a series of convolutional layers and final ouput size is [N,C,28,28]
Another branch from the ROI - align goes to a Fully-Connected Layer. These are then further split into fully connected blocks.

One fully connected block then predicts the class score of the proposed object, the output being of size [N, C]
Another fully connected block then adjusts the box coordinates of the object and output size is [N, 4 *C]
Every class has a bounding box with the representation [x-centre, y-centre, width, height]

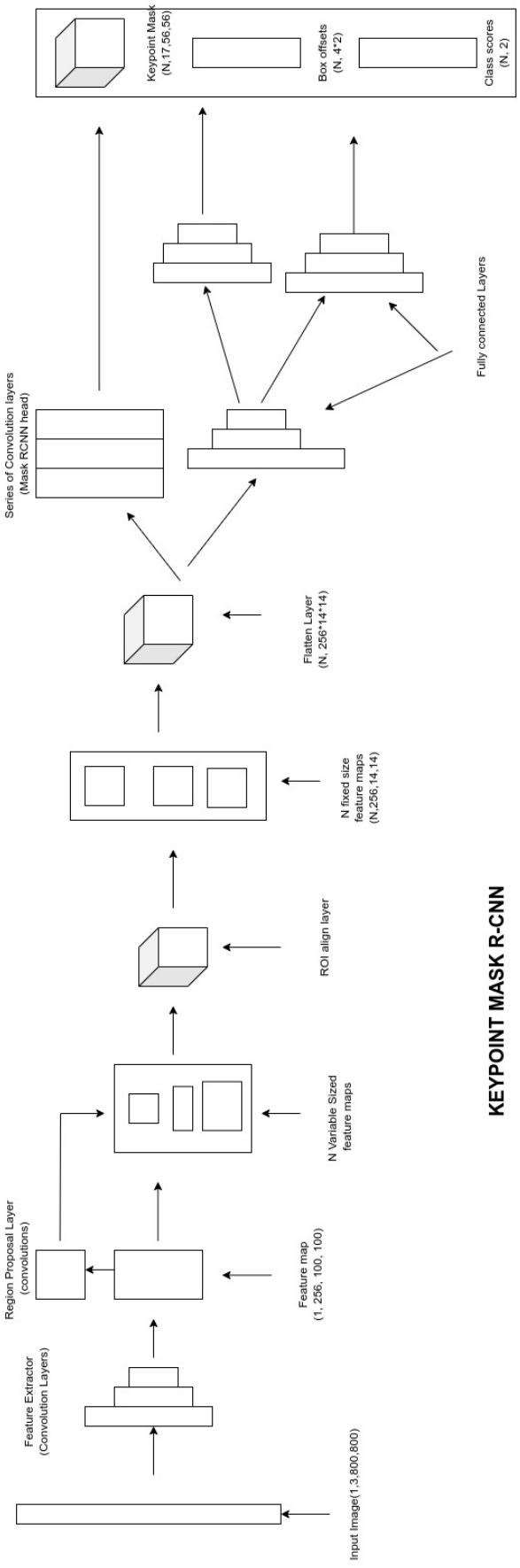

Figure 3.4: A Keypoint mask RCNN network

Architecture of Keypoint-RCNN

The Keypoint RCNN just slightly modifies from Masked-RCNN. They differentiate in output size and manner of keypoint encoding in keypoint mask.

Keypoint RCNN just modifies the Mask RCNN, by using one-hot encoding for the keypoint in the detected object. How these keypoints are then encoded, let us see.
Consider the keypoint detection problem in Mask-RCNN. The following is an image to encode the keypoints in output mask

The output from Keypoint-RCNN is modified version from Mask-RCNN.

Output in Keypoint-RCNN is sized [N, K=17, 56, 56].
K refers to each kind of joint in the human body for example the elbow, the knee, etc.

The class-scores will be the size [N, 2] as we are only concerned with person class
The box-predictions aize will be [N, 2 * 4].

The model uses ResNet-50 FPN as a backbone. Feature Pyramid Network means fusing of feature maps at varying scales so as to be able to preserve information at different levels.
The model is trained on MS-COCO dataset

The model has a mAP score of 62.7 for key point detection on the COCO dataset[23]

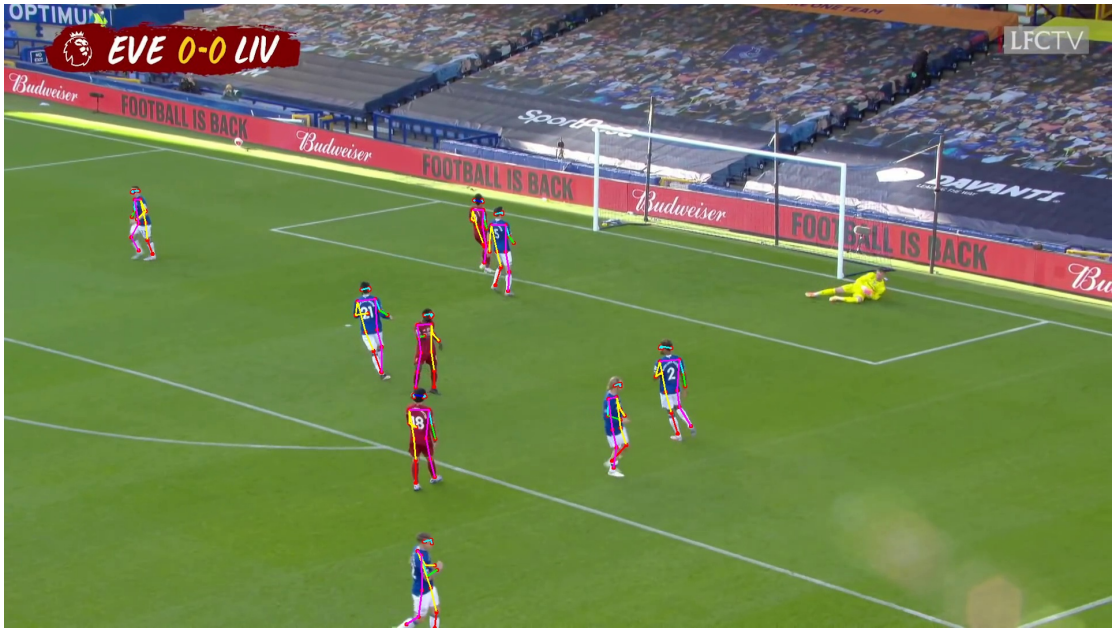

Figure 3.5: The human pose estimation results obtained from our Keypoint RCNN network

The number of keypoints are 17 in COCO. Those keypoints are shown in the figure 3.6. Some of these keypoints are joined in the image to generate the human pose.

Loss function

In Keypoint Detection, every Ground-Truth representing keypoint is one-hot-encoded in the feature map which is of size [K=17, 56, 56], for every single object. This is for all K channels.

For the Ground-Truth, we use channel wise Softmax from the eventual featuremap [17, 56, 56], to minimize the Loss which is Cross Entropy Loss in this case.

$$\text{Loss Function} = \frac{-\sum_{h,w}[Y_{k,h,w} == 1] * (Y_{k,h,w} * \log\left(softmax(\hat{Y}_{k,h,w})\right))}{\sum_{h,w}[Y_{k,h,w} == 1]}$$

3.3 Team Classification

For this task we use the same process as Neeraj et al[1]. Team classification is the process to determine the team each player is belonging to. We use unsupervised clustering algorithm for the same. A specific portion out of the player body is used as defined in [1].

Figure 3.6: Choosing a specific region of interest of the player body

The specific region is the rectangle joining the knees and the midpoints of the shoulder and hip keypoints. This area is uniform across all players of same team and hence is chosen as shown in the figure 3.6

The region of interest for the player helps to extract the features so as to be able to determine the player's team. The pixels in the region are converted into three histograms for each input channel for RGB.

Every histogram gets the features in pixels that is decided by the frequency of occurence.
The histograms are then concatenated and form a representation of the pixels which is a vector.
This vector can be used by clustering algorithms to get the player's team.

Next we are going to identify the noise in the clustering. Here the noise is the Keeper and Referee. We are not going to consider them in our offside evaluation. This is because the jerseys of these entities are usually significantly different from the player jerseys of the two teams.

An ensemble of the two clustering algorithms, KMeans Clustering and DBSCAN is used. The DBSCAN algorithm can identify noisy data and separate them in the process. But we need only 2 clusters and DBSCAN can produce varying number of

clusters. So K-Means is used along with DBSCAN to get the best of both worlds. In this way we can cluster the players into two teams.

An accuracy of 95.75 percent is achieved with this clustering in classifying players[1]

Figure 3.7: Classifying players into teams

Each player is assigned a 0 or a 1 based on his class, as shown in fig 3.7

3.4 Projection of body parts and offside determination

We can now find the projection of body parts. This has been carried out using the method outlined in [2] as this method works farely well for all cases.
The body part is duly projected on the line which connects the vanishing point and ankle lying on the same side on which side is the body part. This is because the line will lie in the ground plane.

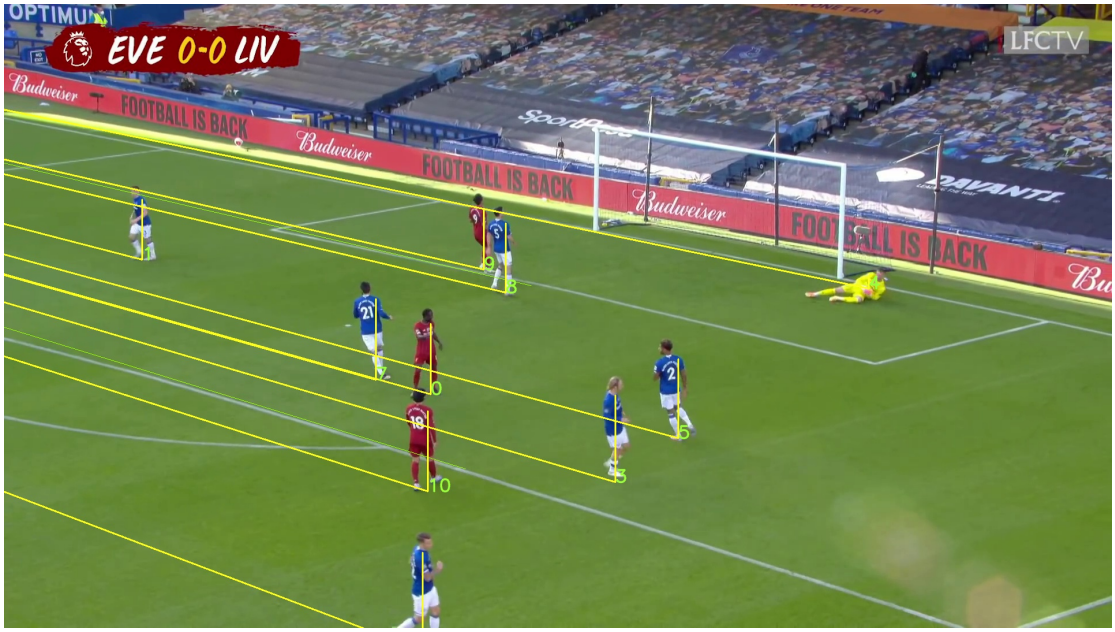

Figure 3.8: Finding the projections of the farthest playable body part for each player

In this figure 3.8 the keypoints are projected on the line joining the vanishing point and ankle. the lines coming from the vanishing point to the projection on football ground will form the lines which will be used for offside determination. these lines are seen in figure

Now we will follow the following steps:-

1. We have the player information, the ID of attacking team and that of the defending team

2. We find the farthest projection point of the last defending player. This is done by first arranging the defending players in the increasing order of angular value. This is achieved by sorting.

3. Now we will choose the player in the beginning of the sorting sequence. This player has least angular value.

4. Now we for each attacking player we will check if the angular value at vanishing point is smaller than angular value of last defending player. Then the attacker is labelled as OFF. Otherwise he is labeled as ON.

Now we have arrived at our final required image representation in result

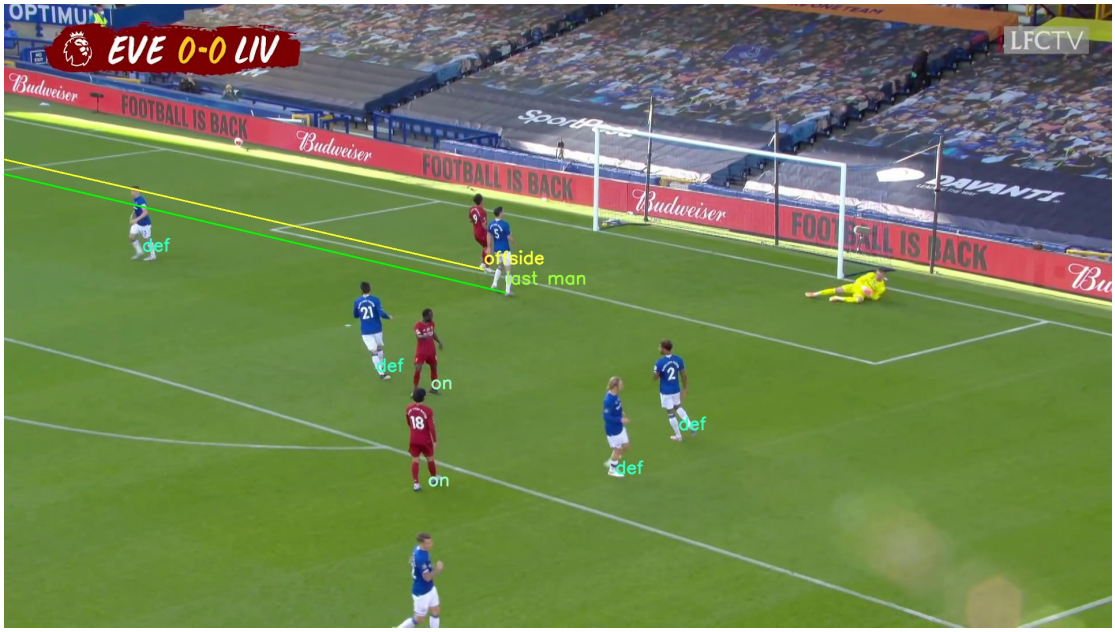

Figure 3.9: The final offside decision

In the figure 3.9 we mark the players in the defending team. And check the projection lines of each attacking player to that of the last defending man. The player whose projection line is farther from the last man is offside and his line is marked as offside. If any player is offside the image result is considered offside otherwise onside

CHAPTER 4

RESULTS AND ANALYSIS

We test our approach on 475 images in the dataset provided by Neeraj et al[2].

We consider the precision, the recall and the F1 score in the system so as to be able to comment on the performance

$$Precision = \frac{\text{Correct Offside Decisions}}{\text{Total Decisions}}$$

$$Recall = \frac{\text{Correct Offside Decisions}}{\text{Total Correct Offside Decisions}}$$

$$F1Score = 2 * \frac{\text{Precision*Recall}}{\text{Precision+Recall}}$$

	Precision	Recall	F1 Score
Neeraj et al [1]	0.72	0.75	0.73
Current Work	0.74	0.78	0.76

Figure 4.1: The various parameters to judge the performance of the offside detection system

We can see in figure 4.1 that our system achives significant improvement over current state of the art approach using pose estimation

There have been certain shortcomings in our system which leads to the accuracy that is achieved.
The errors encountered are mainly due to:
1) Incorrect goalkeeper detection

2) Error in the classification. An improved algorithm might improve this.
3) Position of goalkeeper being ahead of the last non goalkeeper defender
4) A better pose estimation network might improve accuracy

If we can improve on the above points then we can further better the results.

Some more results are presented in the following pages

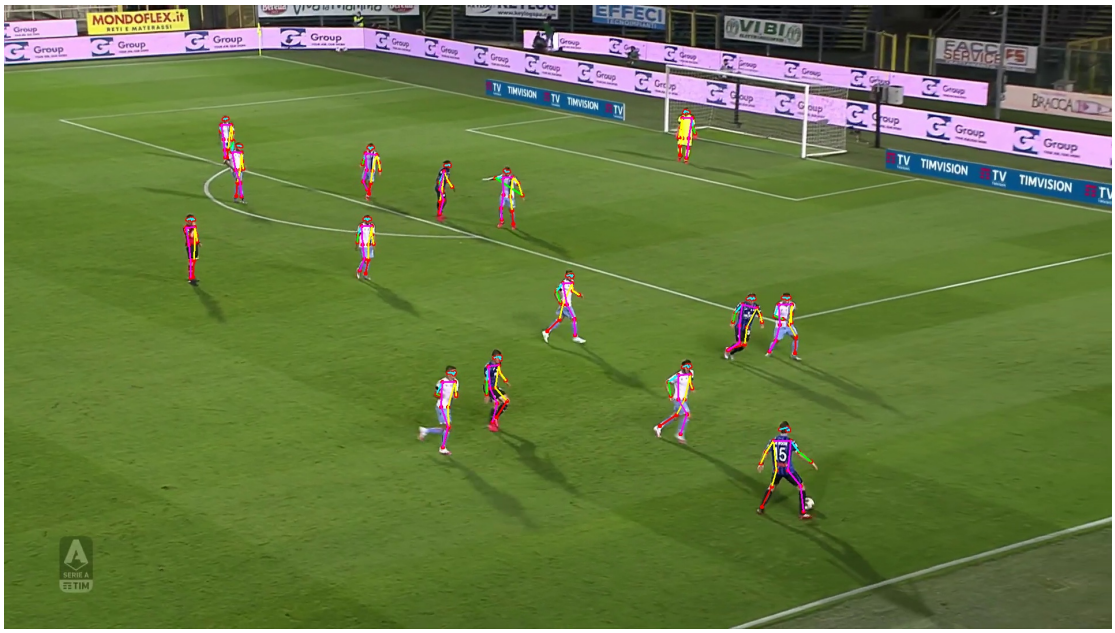

Figure 4.2: Example 1 - pose estimation

Here we can see the detected keypoints of all players. Some of these keypoints are joined to form the pose of all the players

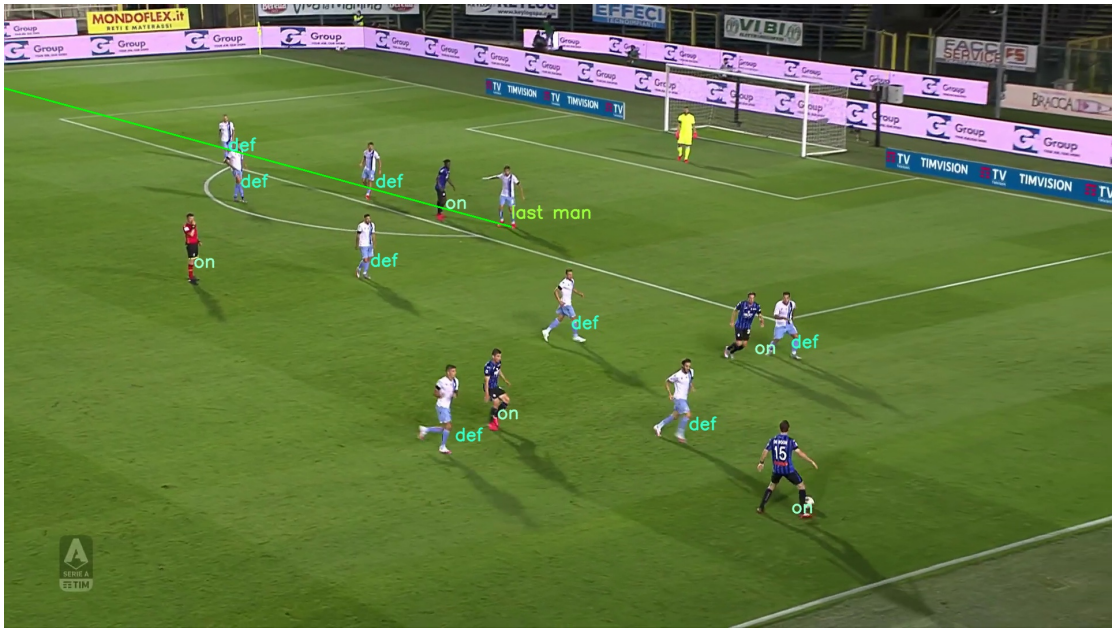

Figure 4.3: Example 1 - final result

Here we can see that all the players in dark blue who are the attacking team are behind the farthest player of defending team which is also known as the last man. So all the players of attacking team are marked on

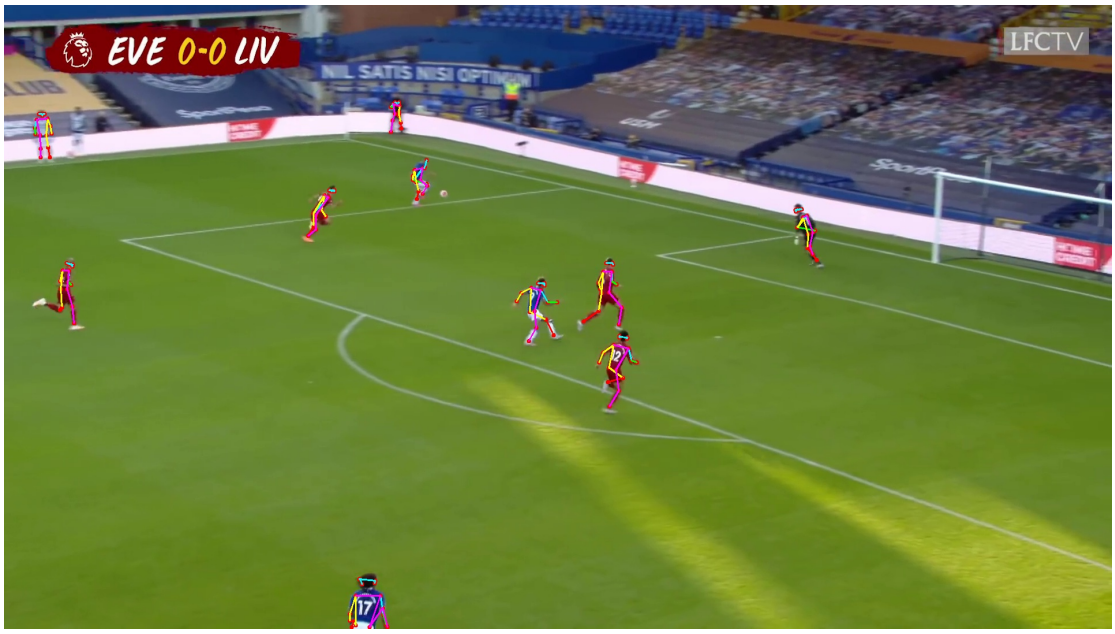

Figure 4.4: Example 2 - pose estimation

We are determining the keypoints of the players and joining some of them to show the pose of these players

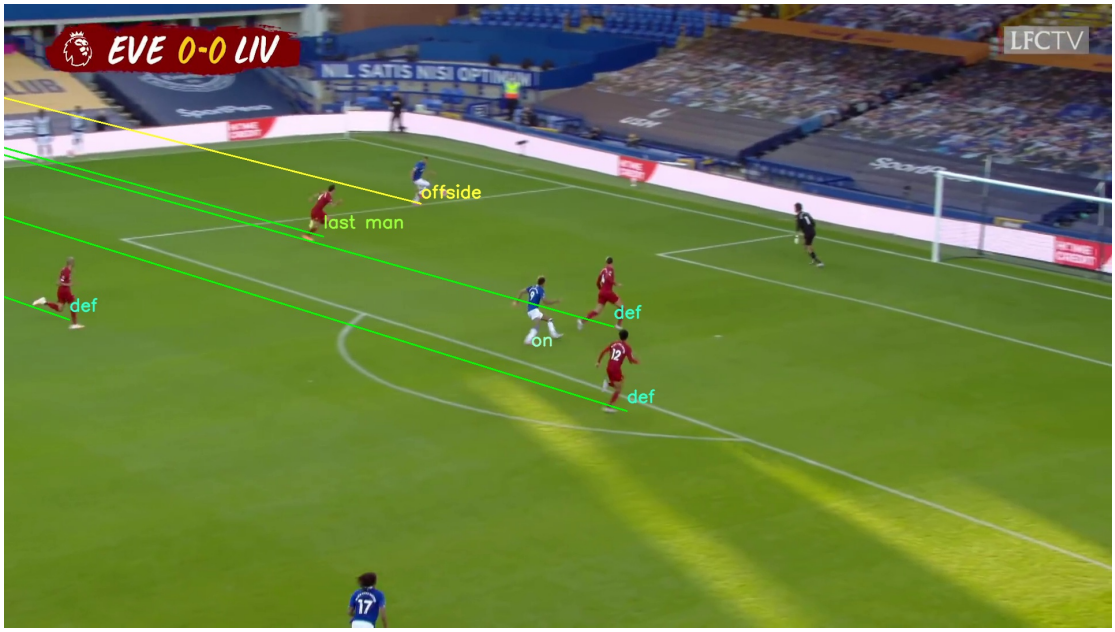

Figure 4.5: Example 2 - final result

In this case we see that the projection line for the topmost blue player is further towards the goal than the last man hence he has been marked in yellow as offside. Other attackers in blue team who are farther from the last man are all marked as on

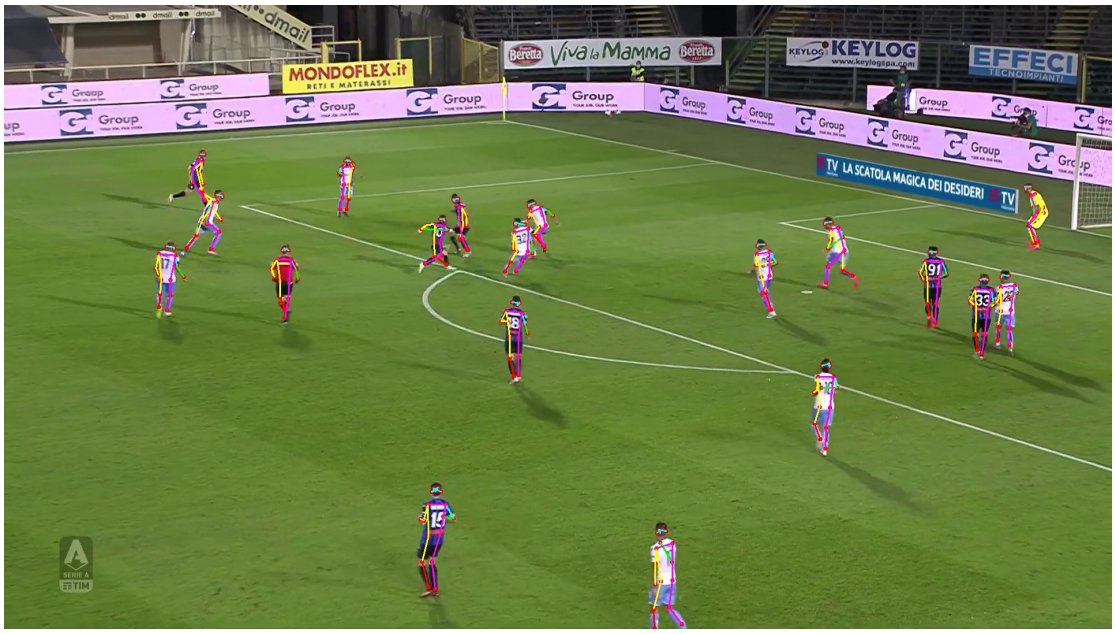

Figure 4.6: Example 3 - pose estimation

We are determining the keypoints of the players and joining some of them to show the pose of these players

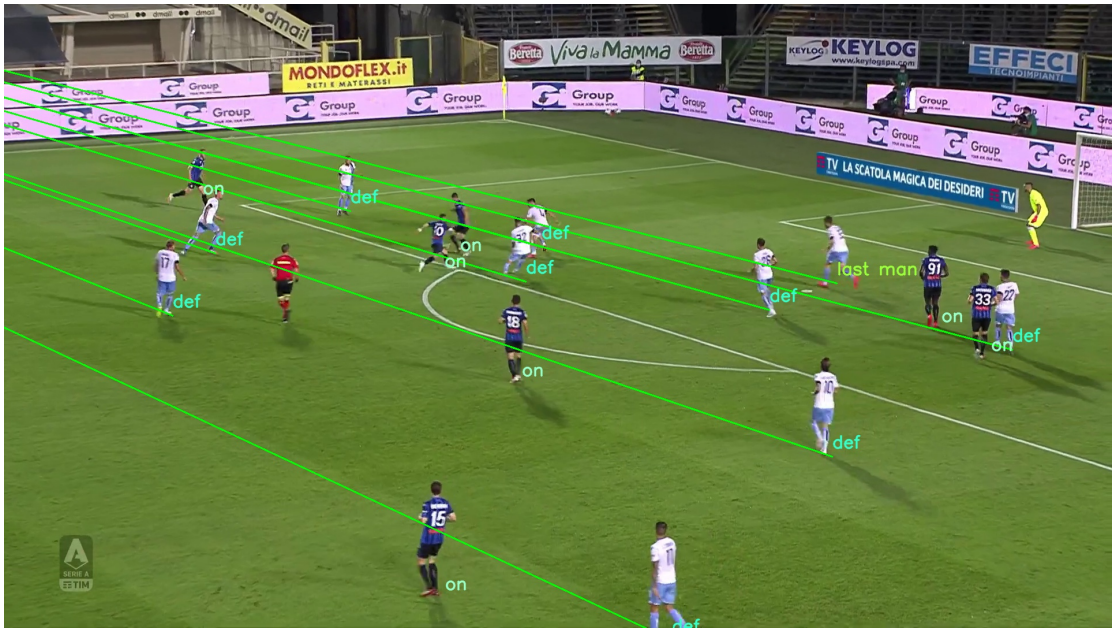

Figure 4.7: Example 3 - final result

Here we can see that all the players in dark blue who are the attacking team are behind the farthest player of defending team which is also known as the last man. So all the players of attacking team are marked on

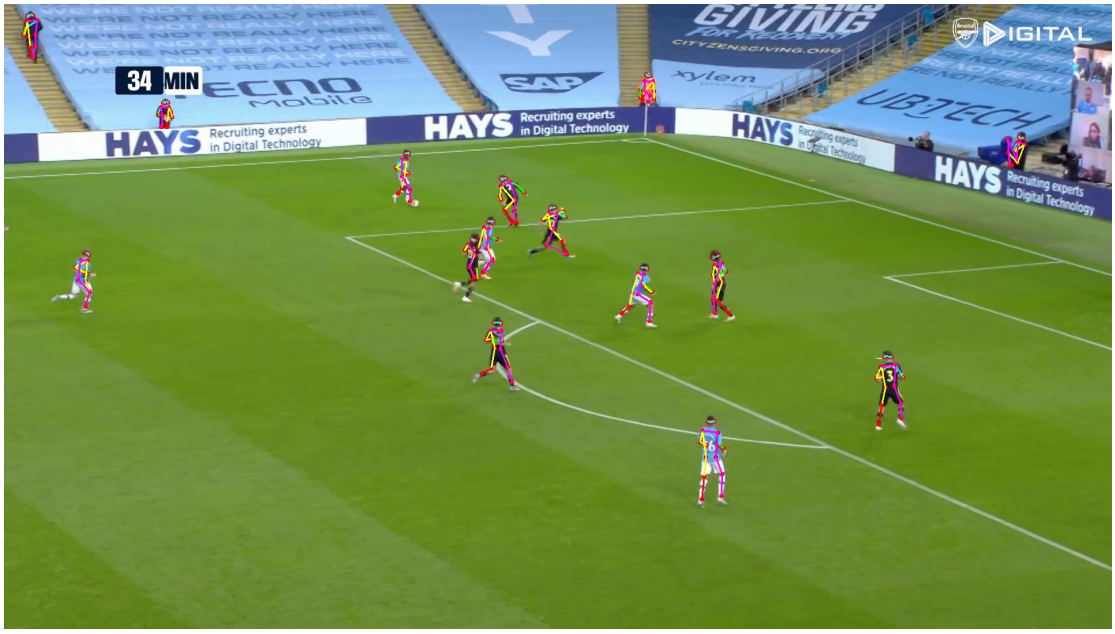

Figure 4.8: Example 4 - pose estimation

Again We are determining the keypoints of the players and joining some of them to show the pose of these players. The total number of keypoints are 17 in COCO dataset

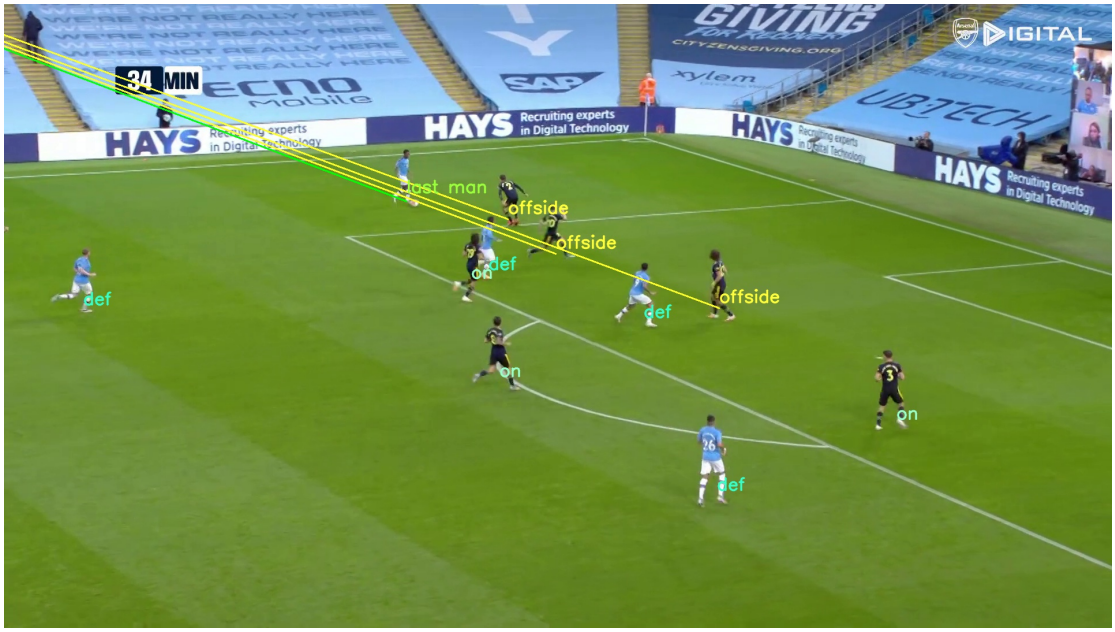

Figure 4.9: Example 4 - final result

Here the blue team is marked the defending team and black is marked the attacking team. Three of the black team members are ahead of the last man is the defending team as their projection lines are closer to the goal than that of last man. So the three players are marked offside in yellow

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Thus we can see that this approach can be seamlessly used either as a standalone system or as an aid for the VAR system already in use in Football. A significant improvement over the current state of the art has been achieved. As can be seen that we are able to get to a F1-score of 0.76 in comparison to the 0.73 achieved previously. This would possibly help in getting quicker and better results when applied to real soccer matches.

Future work has a lot of scope in this field. Research work can be done on improving specific tasks within this system such as improving the human pose estimation network and the clustering algorithms used in classification of the players into two teams.
These tasks can be automated fully in future research and would help in significant improvement.

The pose estimation network can face player occlusion, blurred images and these would need to be improved. A video based Tracking method will be helpful in overcoming this issue.

A final task which needs improvement is goalkeeper detection and this would be a gamechanger in the system and could drive the F1-score as high as 0.95 or more.

To conclude, if considerable research work is undertaken in the mentioned areas then significant progress could be seen in future

CHAPTER 6

REFERENCES

[1] Panse, Neeraj and Mahabaleshwarkar, Ameya, A Dataset AND Methodology for Computer Vision Based Offside Detection in Soccer, M Sports '20: Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports, Oct 2020, pp 19-26

[2] Sirimamayvadee Siratanita, Kosin Chamnongthai, and Mistusji Muneyasu. Saliency-based football offside detection. 17th International Symposium on Communications and Information Technologies (ISCIT), Cairns, QLD, 2017, pp. 1-4., 2017.

[3] Sirimamayvadee Siratanita, Kosin Chamnongthai, and Mistusji Muneyasu. A method of saliency-based football-offside detection using six cameras. Global Wireless Summit (GWS), Cape Town, pp. 127-131., 2017.

[4] Sirimamayvadee Siratanita, Kosin Chamnongthai, and Mistusji Muneyasu. A method of football-offside detection using multiple cameras for an automatic linesman assistance system. wireless personal communications. Wireless Personal Communications 118, 1883–1905, 2021

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

[6] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model. In: ECCV 2016. Lecture Notes in Computer Science(), vol 9910.

[7] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5686-5696

[8] Bridgeman Lewis, Volino Marco, Guillemaut Jean-Yves, and Hilton Adrian. Multiperson 3d pose estimation and tracking in sports. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 2487-2496

[9] Theagarajan Rajkumar, Pala Federico, Zhang Xiu, and Bhanu Bir. Soccer: Who has the ball? generating visual analytics and player statistics. June 2018.

[10] Paolo Spagnolo, Nicola Mosca, Massimiliano Nitti, and Arcangelo Distante. An unsupervised approach for segmentation and clustering of soccer players. Pages 133–142, 2007.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

[12] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. IJCV, 2013

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.

[14] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014.

[16] R. Girshick. Fast R-CNN. In ICCV, 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[18] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In CVPR, 2017.

[19] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In ECCV, 2016.

[20] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In NIPS, 2016

[21] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation.

In CVPR, 2017

[22] A. Arnab and P. H. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In CVPR, 2017.

[23] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask – RCNN. In ICCV, 2017