Ball and Player Detection & Tracking in Soccer Videos Using Improved YOLOV3 Model

Banoth Thulasya Naik
National Institute of Technology Warangal
Mohammad Farukh Hashmi (mdfarukh@nitw.ac.in)
NIT Warangal

Research Article

Keywords: Video Analysis, Ball Detection and Tracking, Occlusion, Deep-Learning, Yolov3, SORT Algorithm, Kalmar Filter

Posted Date: June 3rd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-438886/v1

Ball and Player Detection & Tracking in Soccer Videos using Improved YOLOV3 Model

Banoth Thulasya Naik[1] , Mohammad Farukh Hashmi[2*]

[1-2]Electronics and Communication Engineering Department,
[1-2]National Institute of Technology, Warangal, India
thulasyramsingh@student.nitw.ac.in, mdfarukh@nitw.ac.in

*Corresponding Author: **Mohammad Farukh Hashmi** (mdfarukh@nitw.ac.in)

Abstract

Over the past few years, there has been a tremendous increase in the interest and enthusiasm for sports among people. This has led to an increase in the importance given to video recording of various sports that capture even the minutest detail using high-end equipment. Recording and analysis have thereby become extremely crucial in sports like soccer that involve several complex and fast events. Ball detection and tracking along with player analysis have emerged as an area of interest among a lot of analysts and researchers. This is because it helps coaches in performance assessment of the team and in decision making to obtain optimized results. Video analysis can additionally be used by coaches and recruiters to look for new, talented players based on their previously played games. Ball detection also plays a pivotal role in assisting the referees in making decisions at game-changing moments. However, as the ball is almost always moving, its shape-appearance keeps changing over time and it is frequently occluded by players, it makes it difficult to track it throughout the game. We propose a deep learning-based YOLOv3 model for the ball and player detection in broadcast soccer videos. Initially, the videos are processed and unnecessary parts like zoom-ins, replays, etc., are removed to obtain only the relevant frames from each game. Tracking is achieved using the SORT algorithm which employs a Kalman filtering and bounding box overlap.

Keywords: *Video Analysis, Ball Detection and Tracking, Occlusion, Deep-Learning, Yolov3, SORT Algorithm, Kalmar Filter*

1. Introduction

The advancements in the fields of image and video processing, bolstered by the development of various computer vision and deep learning algorithms, have contributed to the enhancement of applications related to video indexing and analysis. Image and motion-analysis can be employed to retrieve relevant information thereby greatly reducing the manual effort of watching these long videos. This kind of analysis is particularly useful for sports videos, wherein the positions and trajectories of the ball and players can be used for video summarization. In team-sports like cricket, soccer, hockey and basketball better comprehension of game content is essential in devising team strategies, in ball tracking, performance analysis of individual players as well as verifying referee decision. However, most of these games have complex rules for deciding a winner or calculating the points scored by a particular team. This makes annotation of the video quite a challenging task.

Out of the various team-sports, soccer is the world's most popular sport whose fans exist in all of Europe, South America, Oceania, and large parts of Asia. The much-awaited world cup final was watched by over 600 million people around the globe in the oddest times of the day. Apart from this global fest that occurs every four years, numerous leagues and tournaments occur all throughout the year. The demand for the study of the sport using computers has grown exponentially. The coaches, players, and general managers invest heavily in analytics to gain an upper hand. Ball and player tracking on soccer videos can help analyze the flow of the game and enhance the characteristics traits of a team or a player. It can also be used to identify a violation of certain game rules, such as off-side.

While it is clear that the tracking of the soccer ball is crucial, it is, in fact, one of the most challenging tasks is most sports. Some of the common obstacles include misclassification or misdetection due to the small size of the ball, occlusions caused by players, shadowing, shape distortions caused by high velocity, the ball kicked out of frame, and environmental conditions. While the ball is, in most cases, of the same shape and size, the players, on the other hand, can vary considerably. The changes in player's clothes, position, pose, and velocity can greatly hinder most human identification algorithms. While playing, soccer players contort their bodies in unique poses that don't occur in pose estimation datasets. This, again, makes it tougher for pre-existing algorithms to detect and track the players. Soccer as a sport is very dynamic and unpredictable were the ball and the players move quickly in erratic patterns in the large field.

The field, play lines, players' clothing, and the ball are specially designed to be visually discernible in color since soccer is a spectator sport. This can be exploited by computer algorithms to detect and classify various objects in the frame. Teams are not allowed to wear green on the field since the grass itself is similar in color. Different teams are also made to wear distinct colors to not only allow viewers to distinguish between teams but also enable the referees to make fair and accurate calls when overseeing the match. Despite this, the tracking techniques so far have not been very accurate when it comes to the sport of soccer. The International Football Association Board (IFAB) has allowed wearable technology making it possible to collect a large number of accurate measurements from players, but it is a physical device attached to the player that may hinder his/her performance on the field. This is one of the reasons why accurate and high-quality optical ball and player tracking system is necessary.

In this paper, we propose to use the YOLOv3 architecture for detection of the ball and players from broadcast videos. For player detection, a pre-trained version of the model is used (restricting the number of classes to 1) while for ball detection, it is trained on several self annotated frames. Tracking is achieved using the SORT algorithm which employs a Kalman filtering and bounding box overlap. The algorithm achieves high performance and handles challenges like occlusion majority of the times.

The rest of the paper is organized as follows. Section 2 provides a review of the existing literature related to soccer ball and player detection, tracking and analysis. Section 3 talks about the various methods and materials used to achieve the desired results. Section 4 presents the proposed work in detail. The experimental results and performance metrics related to the work are presented in section 5. Finally, the future developments that can be made are discussed in section 6.

2. Related Work

The Computer Vision community has been interested in problems pertaining to sports for a long while. There have been numerous applications in various sports, like cricket [14], soccer [15], volleyball [16],

basketball [17], tennis [18], badminton [19] and hockey [20] among other sports. Moreover, sports have emerged as a popular proxy task for evaluating computer vision algorithms due to complex nature of most sports. In this paper, we restrict our focus on soccer as the sport of interest. In order to track the ball and players on the field, calibration of the cameras used for recording the video is important. Multi-camera calibration can be done by using the known positions in a particular scene. In case multiple cameras are available, Ren et al. [1] provided an innovative method for soccer ball trajectory estimation using multiple fixed cameras. Motion models are used to differentiate the ball from other movements on the field. The likelihood measure of ball detection is assigned to the output of a Kalman filter based on parameters like ground velocity, color features and the normalized size.

2.1 Ball and Player Detection:

Ball as well as player detection is essentially one of the major aspects of sports video analysis. One way this can be done is using algorithms that can distinguish between moving and stationary objects and can continuously track them in a steam of video frames like that done by Sophie et al. [2]. They used an algorithm to obtain the location of the ball at all time and based on the ball coordinates estimate the ball location in the next few frames. In order to initialize the background for all the frames of the videos and the threshold for binarizing, they used a background learning process. For thresholding, a simple and time efficient variance based function is used and noise is removed using a morphological function. However, their algorithm works only when the ball is separated from the player and cannot detect the ball when it is completely or partially occluded by the player. A similar approach is adopted by Wayne et al. [3] for ball detection while overcoming the limitations of [2]. In order to extract the field from the video sequence, they used Hough Transform in order to detect the boundary lines of the soccer field. This information is used in the identification of corners and vanishing points. Here, background subtraction is done in order to identify moving objects in successive frames using median filter on grey-level images. A modified connected components algorithm is proposed by them in order to represent individual objects in a given frame. For ball detection, a coarse analysis stage is present (based on size and color) followed by the filtering stage.

Another way of ball and player detection is by blobs extraction like that proposed by Choi et al. [4]. In this, from a given frame, the player blob and the background is subtracted in order to obtain the ball blob which is then used to obtain the ball trajectory in successive frames. They take into account the frequent occlusions that occur and the model performs well on broadcast videos unlike most of the previous works. A more advanced approach that is used to model the interaction between the player and the ball is that by Andrii et al. [5] which uses a Mixed Integer Program [6]. They account for second order motion of the ball, player and ball interaction along with the different states of the ball. This approach is applied to a number of team-sports and is not just confined to soccer.

2.2 Ball and Player Tracking:

Over the past few years an increasing number of researchers have become interested in tracking the ball in soccer videos, based on the work done in the field of object tracking. One of the commonly used approaches is a trajectory based soccer ball detection and tracking algorithm in broadcast videos proposed by Yu et al. [7]. Their algorithm consists of two important phases: identifying potential ball-candidates in each frame and then obtaining the ball trajectories and processing them. These ball trajectories can then be used to identify the exact ball locations because in general it is difficult to accurately detect a ball due

to the presence of a number of ball-like candidates in the frame. Before identification of the ball-candidates, the videos are preprocessed followed by the ball size estimation step using an anti-model approached based on a set of sieves [8]. A more recent approach for player tracking in consecutive frames is by using the DeepSORT algorithm developed by Wojke et al.[9].

2.3 Player Analysis:

In addition to ball and player detection as well as tracking, a variety of visual player analytics can be generated from the available soccer videos which can be used by recruiters and coaches. Rajkumar et al. [10] used a CNN to identify the player currently possessing the ball while a histogram based approach was used to distinguish players from the two teams. The visual analytics include identifying if a pass was successful or failed based on whether the new player possessing the ball belongs to the same team or not. Enhancing the player analytics generated by [10], Luca et al. [11] detected various spatio-temporal events like fouls, shots, etc. to evaluate player as well as team performance. This was done with the help of 'Soccer-logs' which reconstructs the team's passing network [12], which indicates ball movement between the players of a team. These logs also enable monitoring a player's performance over time using PlayerRank [13]

3. Methods and Material

In this section, we discuss the basics of Convolutional Neural Networks (CNN), an integral part of the deep learning based model proposed in this paper; the challenges that are presented during soccer video analysis; concepts of multiple object detection and tracking, which forms the basis of our work and finally the various performance metrics that can be used for player analytics generation.

3.1 Convolutional Neural Networks(CNN)

The architecture used for classification based on CNN consists of an input layer, a set of convolutional layers followed by an optional pooling and non-linearity layer (Figure 1). Following this is single or multiple fully-connected layers and terminated by a regression layer which gives the classification output. The size of the input image is made equal to that of the first or input layer of the architecture while the output layer contains the same number of neurons as the required classes.

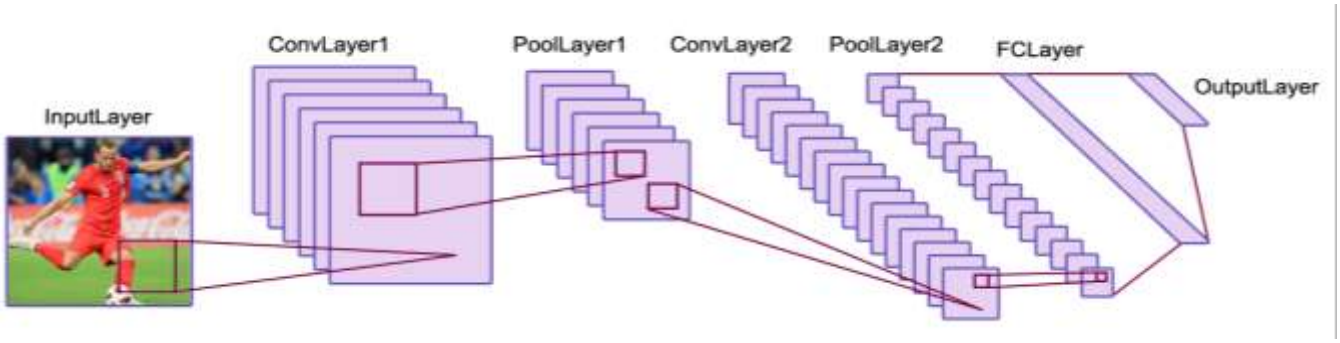

Figure 1. *The basic architecture of a Convolutional Neural Network (CNN)[27]. The figure shows the various layers that are present like conv, pooling and fully connected layers.*

The convolutional layers are initialized with certain weights using which it extracts features from the layer preceding it. The convolution value(x) is obtained by convolving randomly initialized a Gaussian weight matrix of size (FxF) with the image segments of the same size for all the color planes present in the image. During training phase, the weight matrices are updated successively for each layer which is then used to obtain the final feature detection filter for all the layers. As images contain high amounts of data, inputs to the current layer neuron is taken only form a fraction of the entire region (FxF) from the previous layer. Usually, the filter has a greater depth but a small width and height. Activations are produced by performing convolution between the filters and the previous layer. The activation height and width can be computed as follows:

$$V2 = (V1 + 2P_z - F)/S + 1 \tag{1}$$

Where, V1 and V2 are the volume sizes of the input and output respectively, 'F'represents the size of the filter size, 'P_z' the amount of zero padding, and 'S' represents the stride. The output of a particular conv layer is range compressed using a non-linearity function like Rectified Linear Unit (ReLU) [15] which is the most popular. For an input x, the transfer function of the ReLU non-linearity is given as f(x) = max (0, x). Though input normalization is not required in ReLU, Local Response Normalization (LRN) is used for generalization at times [16]. For a kernel i applied to an input volume at (x,y) and having activity $a^i_{x,y}$, the output of ReLU is given below (2).

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta} \tag{2}$$

Here, N represents the total number of kernels in a particular layer while 'n' represents the number of adjacent kernel maps over which the sum is computed. In order to decrease the spatial size of the representation we can insert pooling layers in between. These also help in controlling over fitting of the learning curve as well as reducing the number of parameters. When max pooling is applied, the output dimensions can be calculated as:

$$V2 = (V1 - F)/S + 1 \tag{3}$$

By increasing the number of convolutional layers, we can increase the level of abstraction that the network learns. A fully connected (FC) layer is used at the end of the network to vectorize the output of the last conv layer. As mentioned earlier, the number of FC layers is same as the number of classes that are used for classification (k). A commonly used cross entropy loss function is given below:

$$E(\theta) = -\sum_{i=1}^{n} \sum_{j=1}^{k} t_{ij} \ln y_j(x_i, \theta) \tag{4}$$

Where, Θ indicates the bias parameter vector as well as the weights, target t_{ij} implies that the i^{th} input belongs to j^{th} class and the network associates the input i to the class j, P (tj = 1 | xi) by a probability y_j (x_i, Θ). The transfer function of the output layer is a normalized exponential, i.e. softmax, and is given as:

$$y_r(x) = \frac{e^{a_r(x)}}{\sum_{j=1}^{k} e^{a_j(x)}} \quad (5)$$

During the forward pass data is processed from the input to the output layer and the loss between true and predicted values is computed. During back propagation, Stochastic Gradient Descent with Momentum (SGDM) is employed in order to update the network parameters. Here a conventional SGD is not used as it gets stuck at local minima which can be eliminated using a momentum term as shown:

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) + \gamma(\theta_l - \theta_{l-1}) \quad (6)$$

Here, the learning rate is denoted as α, parameter vector as Θ and the number of iterations as l. γ is a measure of the contribution of previous iteration's gradient to the current one and the loss function gradient is represented as ∇E (Θ). A mini-batch is used for evaluation of gradients by a single pass through the network. An epoch is when all the dataset samples that are to be trained have passed once through the network. The CNN is said to be trained after sufficient number of epochs have passed.

3.2 Challenges in ball detection in soccer videos

Though a substantial work has been done in the field of ball detection from video frames, it continues to remain an extremely challenging task especially in Broadcast Soccer Videos (BSV), where it is difficult to achieve high accuracy. There are various difficulties that are associated with the task of ball detection which include:

• The size of the ball is very small when compared to the size of each frame size. This is a major issue in BSV in which the camera shots are far-view shots.

• The color, size and shape of the ball are not the same in all frames and vary due to ball speed, camera movement as well as the view.

• There is a high chance of getting false-positives due to the presence of several ball-like objects in the given frame.

• Frequent occlusion of the ball by the players takes place which makes detection difficult.

• Just like occlusion, the ball may appear to have merged with the lines on the field or the players in few frames.

3.3 Multiple Object Detection and Tracking

Over the past few years efforts have been made to tune our algorithms to be able to identify overall patterns in photos as well as videos, instead of just looking at them as a set of pixels. Object detection and tracking have been two such technological advancements that have been able to achieve this successfully.

3.3.1 Object Detection

Object Detection can be defined as the process of obtaining the position of an object in a particular image and classifying it (Figure 2). Detection step precedes object tracking as it is important to first identify the object's location in every frame of the video before being able to track it. The output of an object detection algorithm is the coordinates of the object bounding box as well information about the class the object can be categorized into. Some commonly used object detection architectures include R-Convolutional Neural Networks(R-CNN), Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO (You Only Look Once) and SSD (Single Shot Multi-Box Defender). In this paper we use YOLOv3 for player detection (object detection) and not YOLOv2 as more accurate results are obtained from the former when compared to the latter.

In YOLOv3, the frame per second (FPS) is not dropped despite having more number of layers as compared to YOLOv2.The YOLOv3 architecture divides the given frame into a grid of size NxN. In each of these grid cells3 boxes are predicted and each box predicts a totally 4+1+(no. of classes). Here, 4 corresponds to the coordinates of the bounding box x, y, w, h; where 'x' and 'y' indicate the coordinates of the center whereas 'w' and 'h' give the width and height of the box. Next, 1 corresponds to the objectness score which is nothing but the probability that an object is present in a particular box. In order to avoid a compromise in the network's performance, residual layers are added.

3.3.2 Object tracking

Object tracking is the process of fixing onto an object in motion and being able to tell if the object in the current frame is same as that in the previous one (Figure 2). It consists of:

- *Motion detection*: This is usually done at the pixel level in surveillance videos that use a static camera because of constraints in speed.
- *Object tracking*: Tracking is usually done a small number of features that are extracted from the frame like corner points.
- *Motion segmentation*: Based on the different objects in motion that are present in a frame, segmentation is performed.
- *Object localization*: This is aimed to achieve data reduction by focusing only on the area of interest in a frame which is later used as a solution for the correspondence problem.
- *Three-dimensional shape from motion*: This is used to estimate 3D structure from 2D image sequences; thus, it is also called structure from motion.

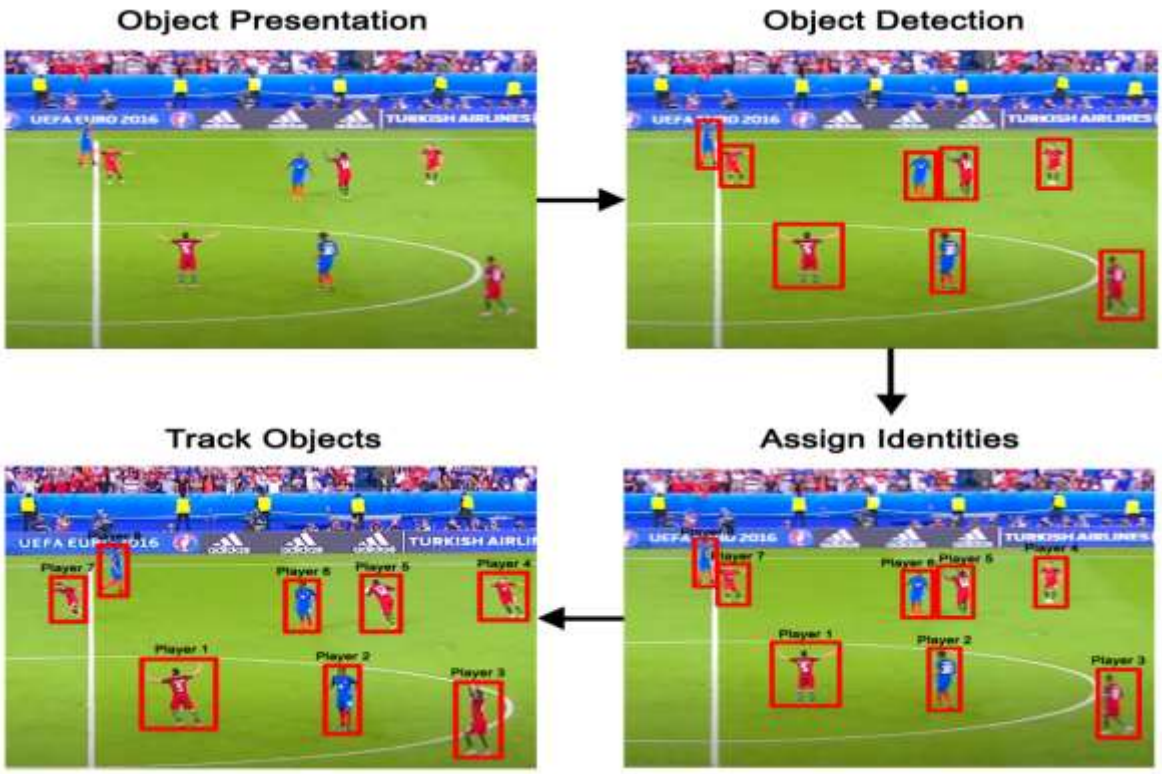

Figure 2. *Stages of multiple object detection and tracking. Here each frame contains two objects namely 'person' and 'car'. The algorithm is trained to detect and track only the people in each frame.*

In soccer, tracking players is not as straightforward as object detection in any other application. This is because it is impossible to apply facial recognition owing to the poor image resolution in broadcast videos. Secondly, it is extremely difficult trying to recognize jersey numbers using off the shelf algorithms. As the orientation and poses of the players keep changing, we resort to identifying players as entities instead of using their faces or player numbers.

3.4 Performance Metrics Evaluation in Soccer

During the course of the game, a team's dominance is often quantified by various metrics and statistical figures. The number of successful passes, and the time duration that the team had possession of the ball shows how well the team played together (Figure 3). While the number of goals a team has scored unmistakably shows a team's merit, it is not a very frequent occurrence in a game, so the number of attempts-at-goal can act as a reasonable substitute. Though it is not impossible for a team with greater attempts-at-goal to be losing, it is far more likely to score a goal given more attempts. When it comes to judging the quality of an individual shot and its likelihood of being converted into a goal, there are many aspects that come into play, such the use of dominant or non-dominant leg, the type of kick, and the locations of the goalie, but the simplest and the most easily identifiable metric is the distance from goal. A player is far more likely to score from within the ten meters of the goal than from fifty, so optical tracking algorithms are also capable of determining such metrics. The number of successful passes and number of

attempts-at-goal should also be possible using video and image based tracking, but so far, this has not been done due to the lack of accurate and robust tracking algorithms.

Figure 3. *Various actions that are used to analyze player performance. Here the first figure(left) indicates 'possession' of the ball while the second figure indicates a 'pass'. A successful pass can be identified based on the jersey color of the player the ball is passed to.*

3.5 Ensemble Learning

In complex deep learning application we usually use more than one model to obtain the final result. After testing all the different models for functionality, we need to 'stack' them in a way such that the output from the first model becomes the input for the next in a pipeline fashion. One way to achieve this stacking is through Ensemble Learning which can also be used to improve the performance of prediction models. The motto behind ensemble learning is to be able to solve a particular problem statement using various models and architectures that cannot solve the overall problem but some part of it. This allows us to use various learning parameters for different models which can give us the desired intermediate results. These results determine the learners for the next model and so on. As the final model is stacked atop the previous set of models, this kind of architecture is referred to as ensemble learning. As a result of this, we succeed in achieving an overall model whose results are way superior to the individual results of our intermediary models, thereby improving the overall performance.

4. Proposed Work

In this section we give a detailed description of the frameworks we have used in this paper along with the various deep learning models that are employed. We divide the task at hand into 2 segments namely player and ball detection and tracking.

Figure 4.*The flow of steps proposed in the paper to obtain detection and tracking of all and players in soccer videos along with information required to perform performance analysis.*

4.1 Detection using YOLO:

The YOLO (You Only Look Once) model is an end-to-end trained neural network that detects objects in an image with the help of its bounding box coordinates along with specifying the class it belongs to. In this, the input frame is divided into a number of smaller cells. If the bounding box center of an object in the input frame falls in one these cells, it predicts its height, width, x & y coordinates of the box as well as the confidence score. In the YOLO model, the final layer uses a linear model while the other layers use a ReLU activation which is given as:

$$\phi(x) = \begin{cases} x, & if\ x > 0 \\ 0.1x, & otherwise \end{cases} \tag{7}$$

The model is optimized on a sum of squared error loss function for the output. The multi path loss function that is used during training the model can be given as:

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [\ (x_i - \hat{x}_i)^2 + \ (y_i - \hat{y}_i)^2]$$

$$+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

$$+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

$$+ \sum_{i=0}^{s^2} \mathbb{I}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (8)$$

Here, $\mathbb{I}_{i}^{obj}$ denotes the appearance of the i^{th} cell of the image while $\mathbb{I}_{ij}^{obj}$ denotes that the prediction in the i^{th} cell is due to the j^{th} bounding box which is present in that cell.

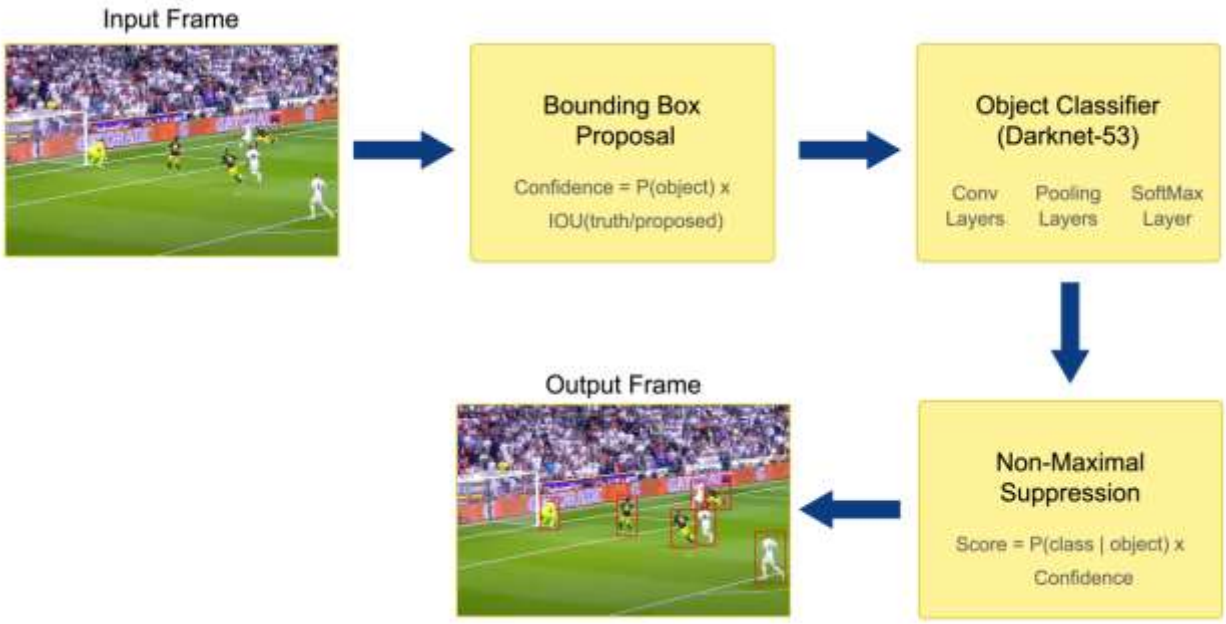

Figure 5.*The Proposed Bock Diagram of object detection and classification by the YOLO model*

The detection of players in this paper is done using YOLOv3 [21] architecture, which is pre-trained on the COCO dataset [22] (consisting of 80 classes). This model was obtained by making further improvements to the YOLOv2 model. Similar to other object detection models like the Faster-CNNs, the YOLOv2 model uses bounding boxes that are tailored with useful sizes as well as shapes during the training phase; these are called anchor-boxes. A k-means algorithm is adopted during training to obtain the bounding boxes for a particular frame. The YOLOv2 model does not directly predict the size and position of the object (in terms of the bounding box) in the frame. Instead, it generates an offset that is used to change the position and resize the pre-defined bounding boxes (anchor-boxes). This reshaping is done with reference to a grid-cell while a logistic regression function is used to obtain accurate results.

In this paper, the pre-trained model is used to detect the players which fall under the 'person' class of the dataset. The architectural details of YOLOv3 are given below.

4.1.1 YOLOv3:

The YOLOv3 network predicts 4 coordinates, t_x, t_y, t_w, t_h for each of the bounding boxes and the predictions are given as:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h} \quad (9)$$

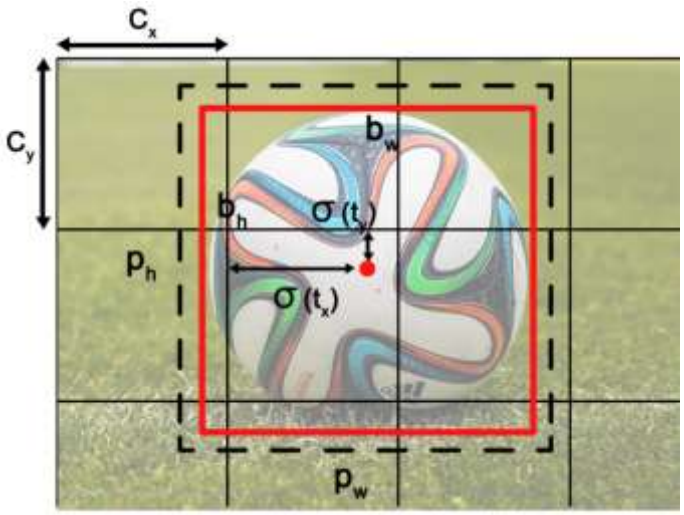

Figure 4. *Bounding box with the priors p_w and p_h along with the predicted width and height [21].*

Where, p_w and p_h are the width and height respectively of the bounding box prior while (c_x, c_y) correspond to the cell offset from the top-left corner of the image. During training the loss is calculated as the sum of squared errors where the gradient is the difference between the ground truth and the predictions of the model(can be calculated from the equations above).

The YOLOv3 architecture (Figure 5) uses logistic regression in order to predict an 'objectness score' for each of the bounding boxes, who's value becomes 1 if the overlap of bounding box prior with the ground truth object is more than that of any other bounding box prior. The prediction is ignored if the overlap is not the best but is greater than a set threshold. A binary cross-entropy loss is employed during the training process in order to predict classes and a multi-label classification is used in case there is more than one class inside the bounding box. The YOLOv3 network uses 3 different scales for box prediction and feature extraction is carried out based on a feature pyramid network [23]. Several convolutional layers are

inserted after the base feature extractor, with the last layer predicting a 3D tensor encoding bounding box, class predictions and objectness.

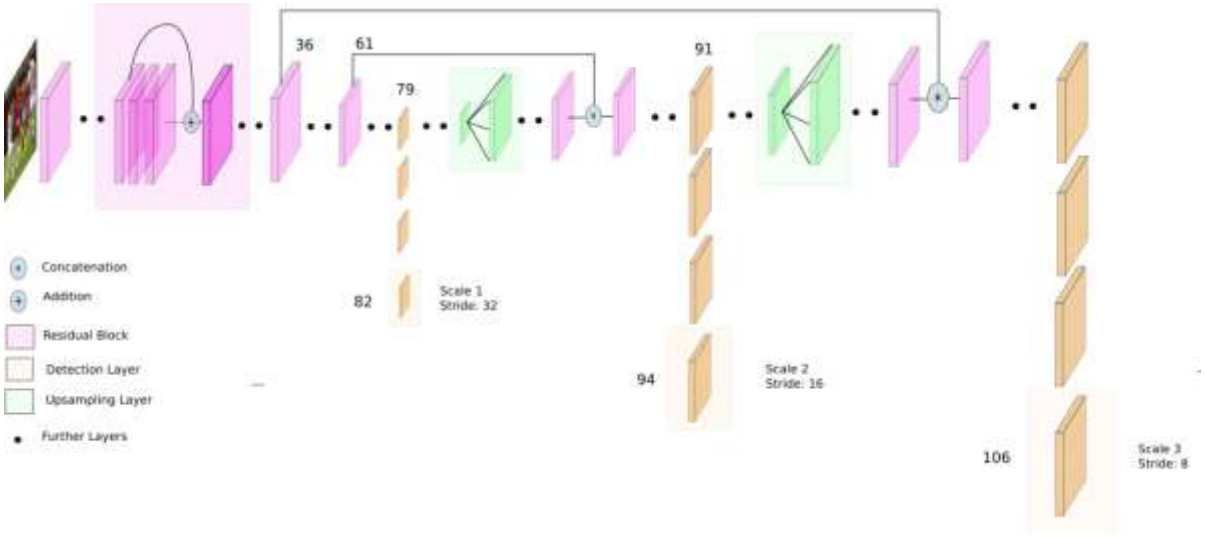

Figure 5. *The architecture of YOLOv3 [28]. The figure indicated various block in the model along with the number of layers in each block and their functionality.*

Then the feature map is extracted from the previous two layers and this is up-sampled by a factor of 2. These features which have been up-sampled are then concatenated with a feature map from within the network, thereby providing us with fine-grained and meaningful semantic information. This process is done again so that the model can predict the final scale bounding boxes. Priors for the bounding boxes are still computed using k-means clustering. However, using the standard k-mean algorithm which uses Euclidean distance, as the size of the bounding box increases, the error associated with it also increases. As the selection of priors which result in better IOU scores are independent of the box size, we use the following distance metric:

$$d(box, centroid) = 1 - IOU(box, centroid) \tag{10}$$

The network contains several 3x3 and 1x1 conv layers along with skip (shortcut) connections between them. As the network has 53 conv layers, it is called the Darknet-53(Table 1).

The performance of the Darknet-53 is comparable with the different state-of-the-art classifiers that are available but with lesser number of floating point operations and higher speed.

4.2 Tracking:

We use the SORT algorithm [24] for tracking the players in our soccer videos. SORT is a simple framework that uses Kalman filtering on each video frame and applies data association on successive frames using the Hungarian method and measures bounding-box overlap which is used as the association metric. For high frame rates, the SORT algorithm is capable of achieving high performance in terms of precision and accuracy.

Table 1.*Architecture of the Darknet-53[21]. The type of each layer(conv, residual, fully connected), number of filters used along with their size and the output of each layer are given in this table.*

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

4.2.1 Hungarian Algorithm:

The Hungarian algorithm utilizes only the bounding box coordinates provided by the YOLO model to track objects and doesn't use any other visual features from the video input. The cost function uses the Euclidean distance between the centroid of the detected bounding box and the predicted bounding box of the object in the same track under consideration. It also calculates the size difference between the current bounding box and that of the previously assigned box for the object in the same track. The cost function for assigning a bounding box to an object in a particular track is given by:

$$d(b,k) = w \cdot d_2(C_b, C_{k+1}') + (1-w)|P_b - P_{k-1}|$$

$$w \in [0,1], k \in N \tag{11}$$

Here, the parameters of the cost function indicate the b^{th} bounding box and the k^{th} track. C_b and P_b are the centroid and area of the b^{th} bounding box C_{k+1}' is the centroid predicted for the k^{th} track. Pk-1 corresponds to the last bounding box's area belonging to the k^{th} track. Here we consider $C_{k+1}' = C_{k-1}$.

4.2.2Simple Online and Real-time Tracking (SORT):

The SORT architecture is based on Faster Region CNN (FrRCNN) detection framework [25]. It consists of 2 stages; features are extracted by the first stage while the second stage classifies the required object. A motion model is used to propagate a particular frame's identity to the next to help in tracking. A linear constant velocity model is employed to approximate the object's inter-frame displacement; this model is independent of other objects in the frame and motion of the camera. Each target's state is represented as a set of vectors give by:

$$x = [\, u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{12}$$

Where, u, v represent the horizontal and vertical locations of the centre of the object while 's' represents the area and 'v' represents the aspect ratio of the object's bounding box. When a detection associated with the target object is obtained, the bounding box is utilized to update the object's state and a Kalman filter framework is used to optimally solve the velocity components.

The geometry of each target object's bounding box is then estimated by predicting its new location in the current frame and this is used to assign detections to the existing objects (Figure 6). The cost matrix is then calculated using the IOU (intersection over union) distance between the detections and the predicted bounding boxes. The Hungarian algorithm is used to optimally solve the assignment. A lower threshold is set (IOU_{min}) to reject any assignments that have an overlap value lower than the threshold. The occlusion that is caused due to other players on the field is handled by the bounding boxes' IOU distance as detections with similar scales are favored by it.

In addition to this, players might enter or exit a particular frame in the video which must be accounted for while tracking by creating new identities for them and destroying these identities accordingly. An untracked object in a frame is identified when the overlap for a detection is less than the IOU_{min}. Each of these trackers is initialized with zero velocity and a large value of covariance of this velocity component. A new tracker is observed for a particular probation period to see if it is associated with a detection or not. This information helps the algorithm avoid detection of false positives. If for a particular T_{lost} number of frames if the tracker remains undetected, it is terminated to avoid errors and prevent an exponential increase in the number of undetected trackers. In case an player reappears in a frame, it is tracked with a new identity.

4.3 Dataset:

We use the Comprehensive Dataset of Broadcast Soccer Videos [26] to train and test the proposed algorithm. The dataset comprises of around 220 broadcast soccer match videos, with each video being about 45 minutes long i.e., half of a game. All video are taken from namely 4 matches: UEFA EURO 2016, FIFA World Cup 2014, Premier League 2016/2017 and AFC Asian Cup 2015. The fps used is 25 and the videos have a resolution of 1080p (HD) and 360p. Each video contains event as well as shot analysis along with sequences for player tracking. There are approximately 71936 shot samples with 5 different shot types and 2 shot-transition types that are included. The event annotations are of 2 types and are available at different granularity. The dataset consists of 6850 event samples and 6294 story samples.

Figure 6. *Shot annotation type examples. Close-view shot (top left), far-view shot (top right), medium-view shot (bottom left) and out-of-filed shot (bottom right).*

There are 2 aspects to the shot annotations for each video: shot boundary and shot type. The shot boundary annotation consists of the gradual transitions and the cut transitions while there are 5 shot types(figure 6) - close-view shot, far-view shot, medium-view shot, playback shot and out-of-filed shot. In addition to these, the dataset contains a special label named 'Other', which accounts for shots containing more than one shot types.

Usually in soccer videos, a playback shot is a sandwiched between two different shot types and can be identified with the help of the competition logo that appears on the screen before the current shot is played again. As the videos in the dataset are taken from 4 major tournaments, the corresponding logos are show in *Figure 7.*

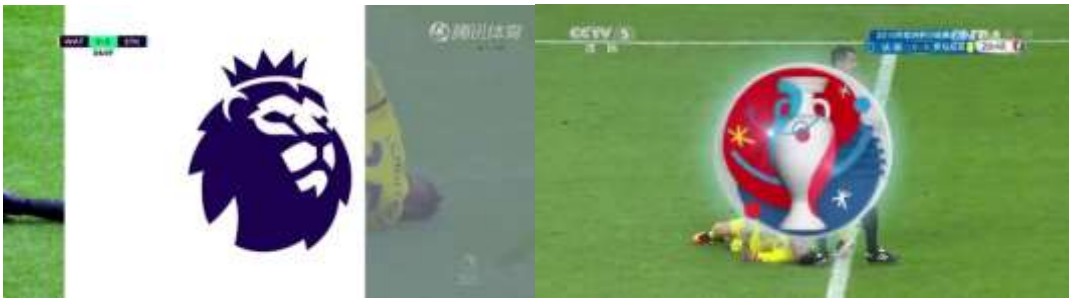

Figure 7. *Tournament logos present in the dataset. These logos can be used to identify and eliminate the playback shots present in the videos.*

4.4 Algorithm for the Proposed Work:

1. Preprocess the raw dataset to separate out the wide-angle shots from the from the soccer videos. This includes eliminating irrelevant frames using the appearance of large logo before and after replays.
2. Load the YOLOv3-SPP model along with the pre-trained weights. Specify the list of classes as 'person' and 'ball' which are part of COCO dataset.
3. Specify the path to the dataset. Open each video and read the frames one by one.
4. For each frame perform object detection using the pre-trained YOLO model.
5. The results that are obtained as the output of the model include the identified class as well the bounding box prediction and its confidence score. Now obtain all players identified with a confidence score> 0.35 and save these output values for each frame. This will become the input for our object tracking model.
6. Now that all players in a frame have been extracted, we need to identify the color of their jerseys. This will help us predict which team a particular player belongs to and perform player-analytics. For this we first need to define the range of colors for each. Then create a color mask and based on the percentage of the specified color to the total number of pixels, the team is identified.
7. For player tracking, load the pre-trained SORT model and give the bounding box coordinates of each player in a frame as input to the model.
8. The output is in the form a video with different color bounding boxes being used to track the players in the given input soccer video. In addition to this, players are also numbered to make facilitate the tracking.
9. Finally combine all the models in the form of a pipeline to obtain detection and tracking of the players and the ball in a soccer video.

5. Results and Discussion

This work has been done on a 64-bit Windows 10 machine using Anaconda environment for Python and an NVIDIA Quadro P600 GPU.


MCI
0-0
MID
06:47
腾讯体育
25 - 27 NOVEMBER
ETIHAD
person 0.63
person 0.74
person 0.70
person 0.60
person 0.72
person 0.67
person 0.80
person 0.57
person 0.71
person 0.44
person 0.55
person 0.80


MCI
0-0
MID
07:11
腾讯体育
person 0.68
person 0.88
person 0.75
person 0.96
person 0.59
person 0.68
person 0.69
person 0.91
person 0.31
person 0.61
person 0.90
sports ball 0.83
person 0.96

Figure 8. *Output frames of the model with bounding boxes for the detections. Each bounding box is associated with the label (person or ball) and a confidence score with which the model made the detection. The SORT algorithm assigns an id is assigned to each player, which helps in tracking. Missed detections occur occasionally when the ball is severely occluded by the players.*

For training the model, we used every 5th frame (size 640x360 pixels) from the video to speed up the training process but for testing all the frames were used(since we require the tracking algorithm to follow the detections).The YOLOv3 model was successfully able to detect almost all the players except those which were severely occluded and quite indistinguishable even to the human eye or those which were very close to the field boundary lines (and a large part of their bodies overlapped with the audience in the frame). The SORT algorithm was then able track all the detected players even if the detections were missing for a few frames in-between.

Table 2.*Experimental results. The trained model is tested on a new video and various performance metrics were obtained to evaluate the model.*

	Player Detection	Ball Detection
Total number of frames tested	400	311
True Positives	4937	203
False Positives	153	6
False Negatives	364	67
Precision (%)	**97**	**97**
Recall (%)	**93**	**75**
F1 Score	**0.95**	**0.85**

Manual inspection was used for the broadcast video sequences to obtain the ground truth. As the ball is not present in all the frames, we use 400 frames for player detection and 311 frames for ball detection which the model was tested. We calculate the values for true positive, false positive and false negative which can be defined as follows: (i) true positive (TP): player/ball detected by the model when player/ball exists in that position; (ii) false positive (FP): player/ball detected by the model when player/ball does not exist in that position; (iii) false negative (FN): player/ball not detected by the model when player/ball exists in that position. Here, the value for each of these parameters indicates the number of players that have been detected/not detected by the model for all the test frames. Using these values, we calculate the percentage precision, recall and F1 score as:

$$Precision = \frac{TP}{TP+FP} * 100\ \% \quad ; \quad Recall = \frac{TP}{TP+FN} * 100\% \tag{13}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{14}$$

From Table 2, we see that the precision (%) for both player and ball detection is high indicating that the output consists of a large proportion of positive identifications that are actually correct. On the other hand, the recall (%) is high for player detection indicating that the output consists of a large proportion of actual positives that are identified correctly. However, this value is low for ball detection as we set the confidence threshold to 0.35 to avoid false positives but it leads to a slight increase in the number of false negatives as the ball was distorted due to its high speed or was very small due to its distance from the camera. We observe that the ball detection yields good results when the ball is in the bottom half of the field (closer to the camera), but the results deteriorated when it was in the top half of the field i.e. farther away from the camera. Tracking errors mainly occur when 2 or more players occlude each other and this leads the model to detecting them as the same player. However, once the players get separated from one another, the tracking algorithm is able to distinguish them as they were before the occlusion.

Performing result comparison with other algorithms in the literature is tough as most of these either do only ball detection or player or concentrate on other video types, while our model uses broadcast soccer videos to perform both ball and player detection and player tracking. We however experimented with the GOTURN model[3] and observed that the model failed greatly in providing ball detections as can be seen in figure 9 even when the ball was not particularly occluded.

Figure 9. *Output frame of the GOTURN model for object tracking. It can be seen that the ball was grossly mis detected by the model which identifies the glove of the goal keeper as the ball because of similarity in color and shape with the ball.*

6. Conclusion and Future Scope

We proposed a deep learning-based model that was successful in detecting and tracking the ball and players in a soccer video. We used a large dataset that eliminated the need for data-augmentation. We evaluated various performance metrics for the model and compared it with the existing work that has been done in this field. We see how the model was able to perform detection despite occlusion at many instances. Future work includes performing various player analytics and determining which player/team has the ball, using a histogram-based technique (as the two teams always have distinct jersey colors). By comparing successive frames, we determine if a 'pass' is made and if so if it is a successful pass or a failed pass. In addition to this, heat maps can be used to identify the footprint of both the teams i.e., identifying which areas were highly occupied by a particular team. Finally, a path for the goalkeeper can also be drawn out.

Funding: No funding
Conflicts of interest/Competing interests: No conflict and Competing Interest
Availability of data and material: Available in paper
Code availability: Available in paper
Ethics approval: Not applicable
Consent to participate: Not applicable
Consent for publication: Not applicable

References

[1] “Tracking the Soccer Ball using Multiple Fixed Cameras”, Jinchang Ren, James Orwell, Graeme A. Jones, Ming Xu.

[2] “Video Based Soccer Ball Tracking”, Sophie Xiaofan Liu, Lijun Jiang, Jacob Garner, Sharayah Vermette.

[3] “Soccer Video Analysis by Ball, Player and Referee Tracking”, Wayne Chelliah Naidoo

[4] “Tracking Soccer Ball in TV Broadcast Video”, Kyuhyoung Choi, Yongduek Seo.

[5] “What Players do with the Ball: A Physically Constrained Interaction Modeling”, Andrii Maksai, Xinchao Wang and Pascal Fua

[6] “Tracking Interacting Objects Optimally Using Integer Programming ”, X. Wang, E. Turetken, F. Fleuret, and P. Fua.

[7] “Trajectory-Based Ball Detection and Trackingin Broadcast Soccer Video”, Xinguo Yu, Hon Wai Leong, Changsheng Xu and Qi Tian

[8] “Antifaces: A novel, fast method for image detection”, D. K. Keren, M. Osadchy, and C. Gotsman

[9] “Simple online and realtime tracking with a deep association metric “, N.Wojke, A. Bewley, and D. Paulus

[10] Manikandan S, Chinnadurai M, Thiruvenkatasuresh M.P, Sivakumar M. (2020). "Prediction of Human Motion Detection in Video Surveillance Environment Using Tensor Flow", International Journal of Advanced Science and Technology, 29(05), 2791 - 2798. Vol. 29 No. 05 (2020): Vol. 29 No. 05 (2020)

[11] “A public data set of spatiotemporal match events in soccer competitions”, Luca Pappalardo, Paolo Cintia, Alessio Rossi, Emanuele Massucco, Paolo Ferragina, Dino Pedreschi & Fosca Giannotti

[12] “The harsh rule of the goals: Data-driven performance indicators for football teams”, Cintia, P., Pappalardo, L., Pedreschi, D., Giannotti, F. & Malvaldi, M.

[13] “PlayeRank: data-driven performance evaluation and player ranking in soccer via a machine learning approach”, Pappalardo, L. et al.

[14] “Fine-grain annotation of cricket videos”, R. A. Sharma, K. P. Sankar, and C. Jawahar

[15] “Camera selection for broadcasting soccer games”, J. Chen, L. Meng, and J. J. Little

[16] “A hierarchical deep temporal model for group activity recognition“, M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori

[17] “Detecting events and key actors in multi-person videos”, V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei

[18] “A technology platform for automatic high-level tennis game analysis“, V. Ren, N. Mosca, M. Nitti, T. DOrazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella

[19] Manikandan, S, Chinnadurai, M, "Effective Energy Adaptive and Consumption in Wireless Sensor Network Using Distributed Source Coding and Sampling Techniques",. *Wireless Personal Communication* (2021), Springer, https://doi.org/10.1007/s11277-021-08081-3

[20] “Classification of puck possession events in ice hockey”, M. R. Tora, J. Chen, and J. J. Little

[21] “YOLOv3: An Incremental Improvement”, Joseph Redmon and Ali Farhadi

[22] Manikandan, S & Chinnadurai, M 2019, ‘Intelligent and Deep Learning Approach OT Measure E-Learning Content in Online Distance Education’, The Online Journal of Distance Education and e-Learning, vol.7, issue 3, July 2019, ISSN: 2147-6454

[23] “Feature pyramid networks for object detection “, T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, andS. Belongie

[24] “SIMPLE ONLINE AND REALTIME TRACKING”, Alex Bewley , Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft

[25] “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, S. Ren, K. He, R. Girshick, and J. Sun

[26] "Comprehensive Dataset of Broadcast Soccer Videos," J. Yu, A. Lei, Z. Song, T. Wang, H. Cai and N. Feng.

[27]“Real Time Object Detection using Deep-Learning and OpenCV,” Suraj Satpute, Harshad Shende, Vikas Shukla, Bharti Patil.

[28] Method of Automated Detection of Traffic Violation with a Convolutional Neural Network - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/YOLO-V3-network-architecture_fig3_337834497 [accessed 1 Jun, 2020]

[29] “A Semi-automatic System for Ground Truth Generation of Soccer Video Sequences,” T. D’Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo.

[30] “Learning to Track at 100 FPS with Deep Regression Networks”, David Held, Sebastian Thrun, Silvio Savarese.

Figures

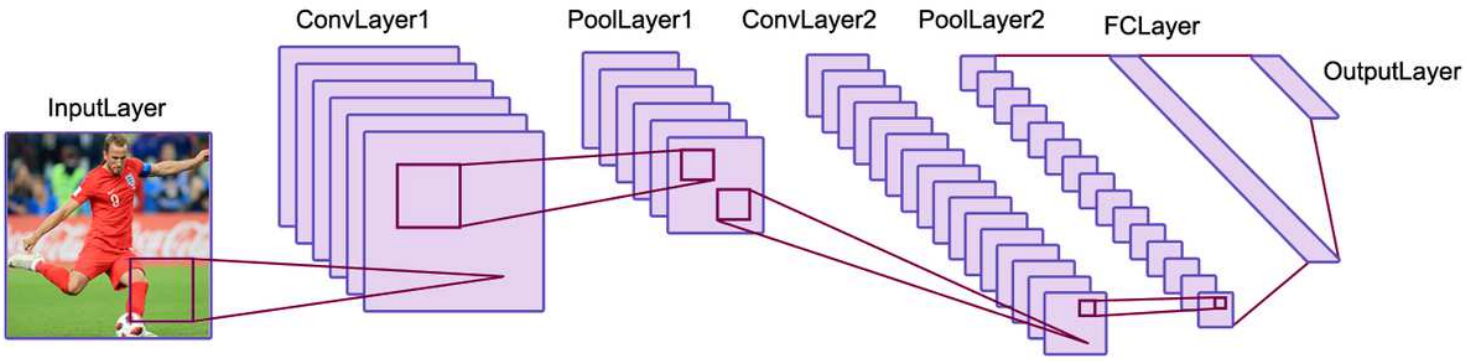

Figure 1

The basic architecture of a Convolutional Neural Network (CNN)[27]. The figure shows the various layers that are present like conv, pooling and fully connected layers.

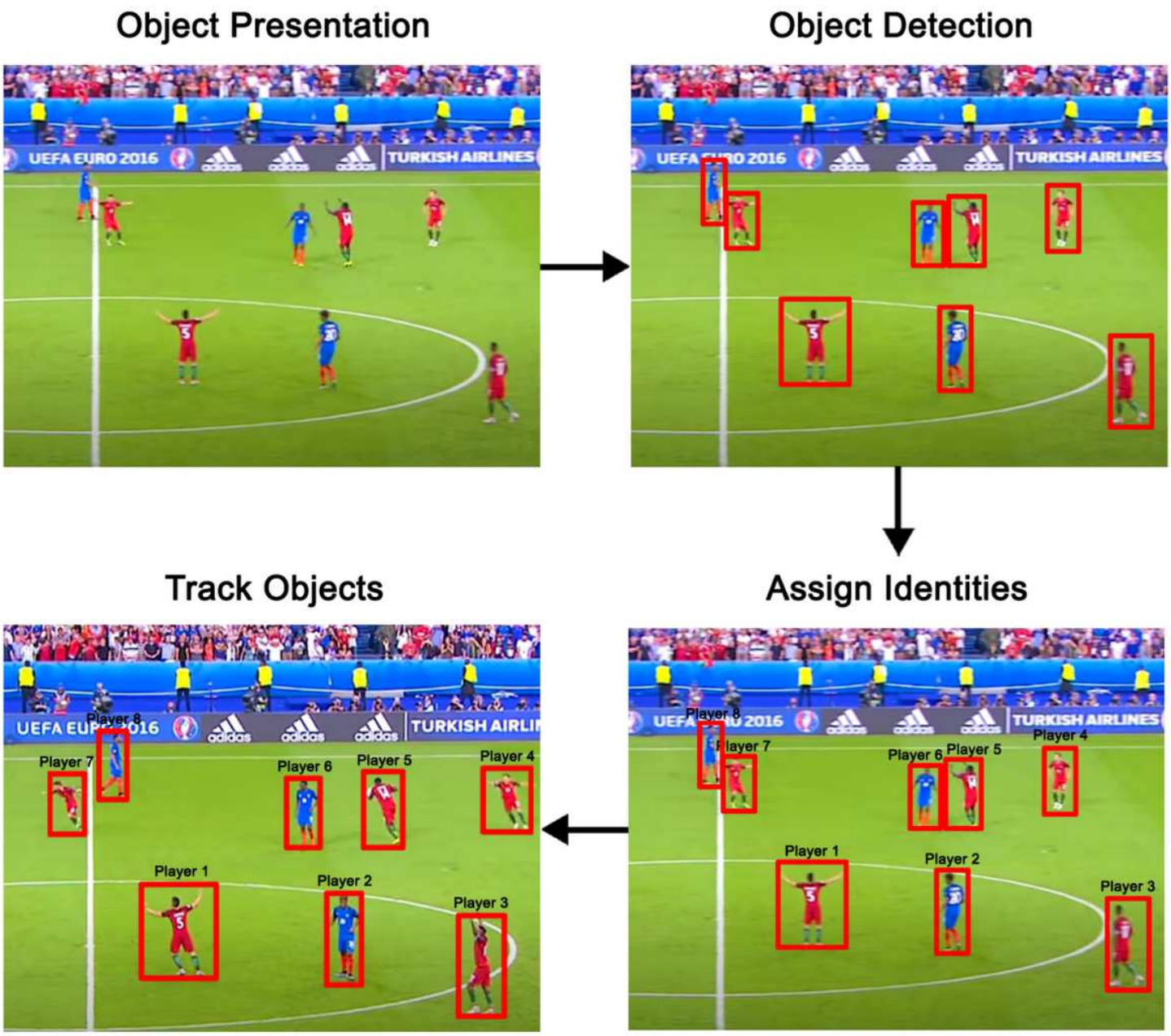

Figure 2

Stages of multiple object detection and tracking. Here each frame contains two objects namely ‘person’ and ‘car’. The algorithm is trained to detect and track only the people in each frame.

Figure 3

Various actions that are used to analyze player performance. Here the first figure(left) indicates ‘possession’ of the ball while the second figure indicates a ‘pass’. A successful pass can be identified based on the jersey color of the player the ball is passed to.

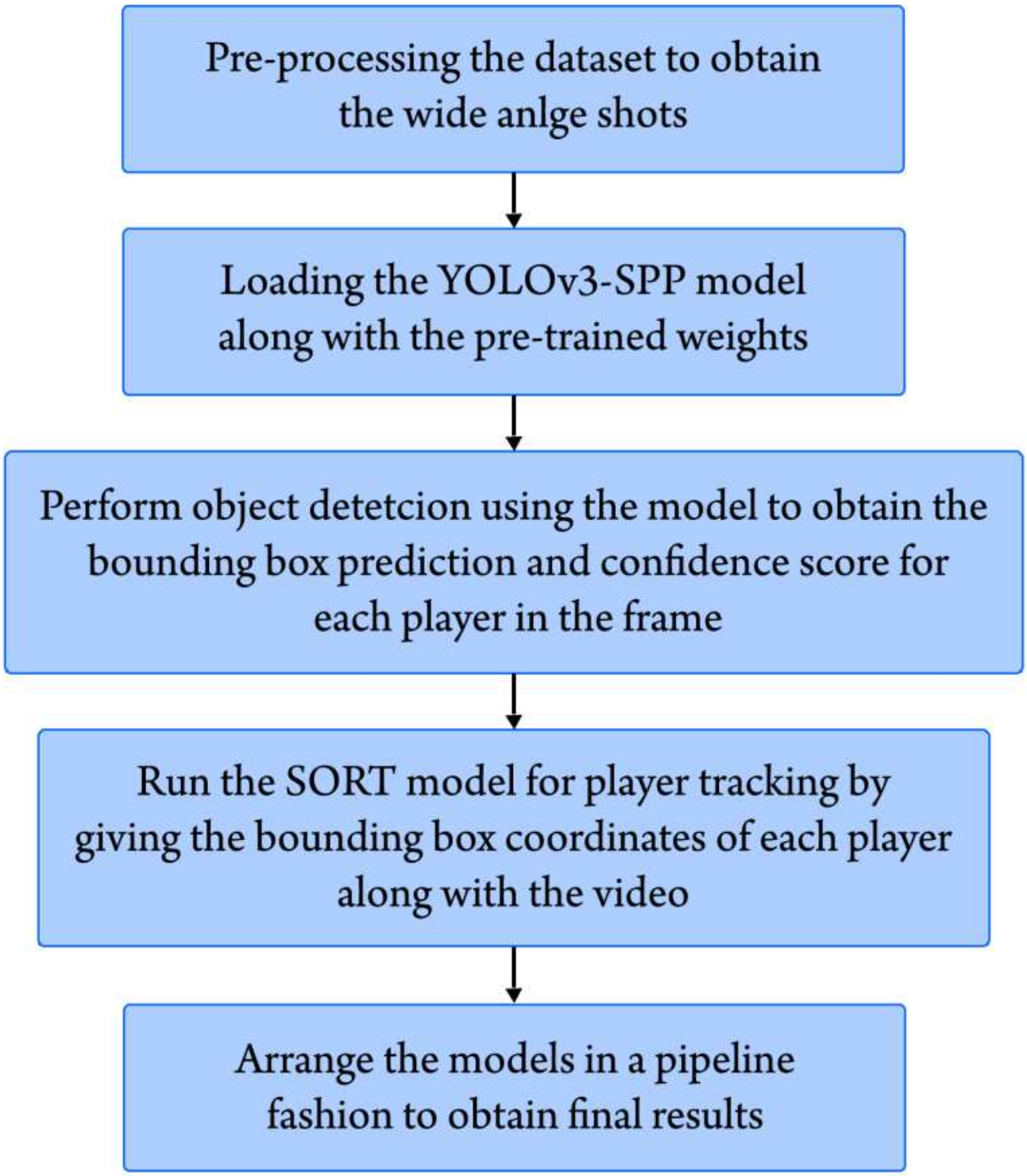

Figure 4

The flow of steps proposed in the paper to obtain detection and tracking of all and players in soccer videos along with information required to perform performance analysis.

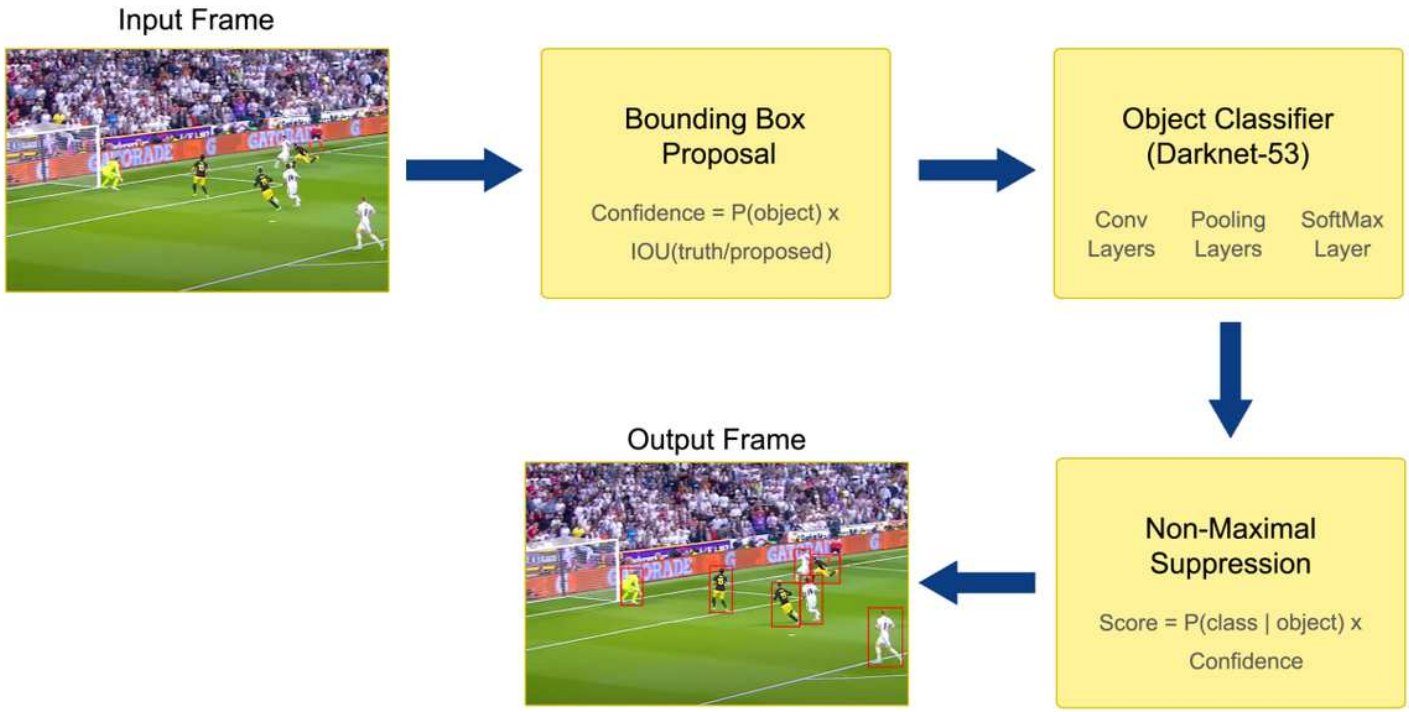

Figure 5

The Proposed Bock Diagram of object detection and classification by the YOLO model

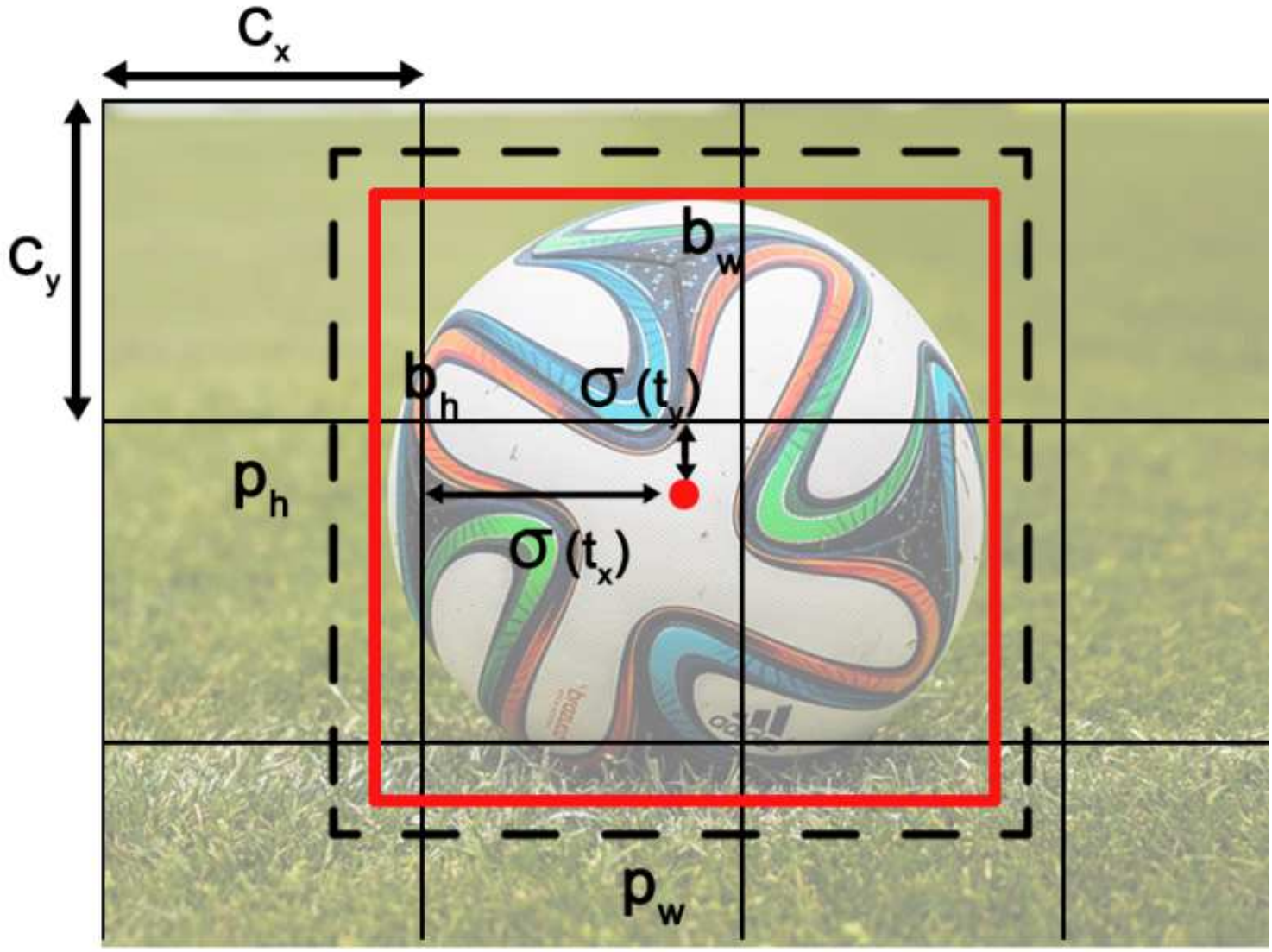

Figure 6

Bounding box with the priors pw and ph along with the predicted width and height [21].

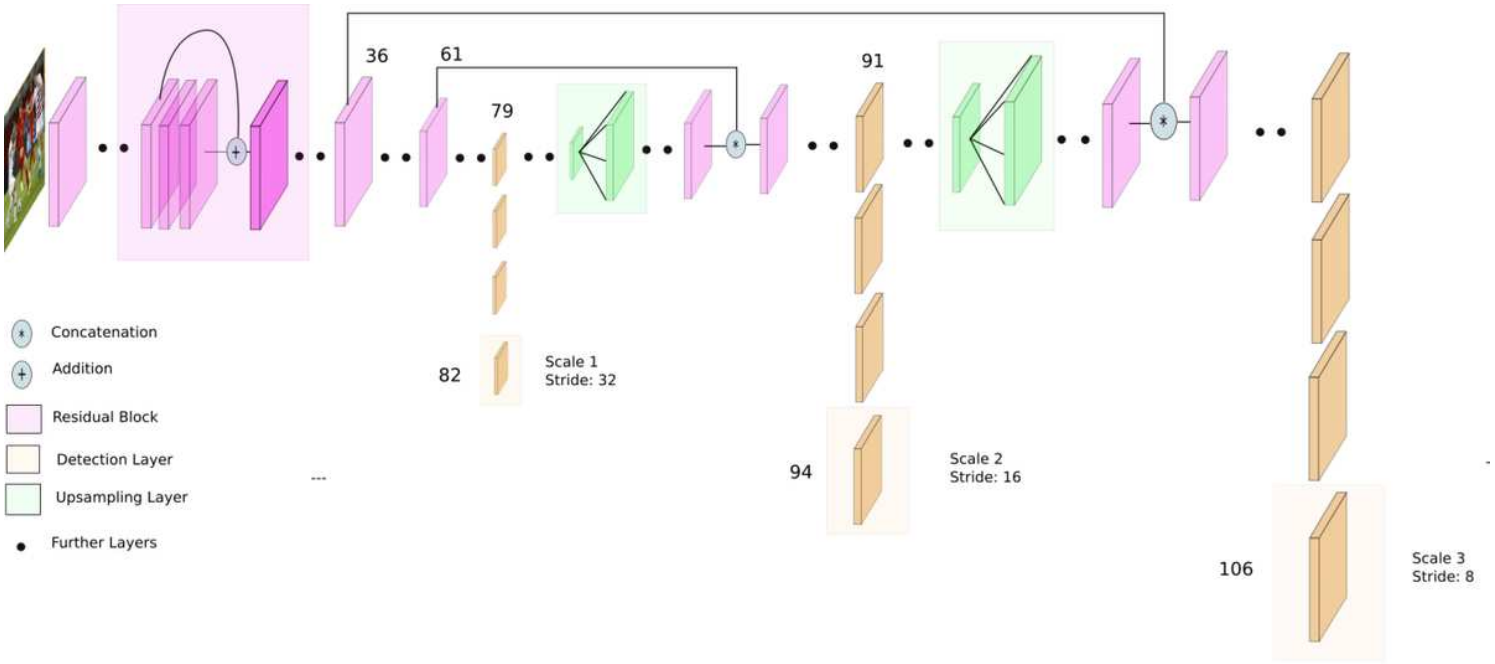

Figure 7

The architecture of YOLOv3 [28]. The figure indicated various block in the model along with the number of layers in each block and their functionality.

Figure 8

Shot annotation type examples. Close-view shot (top left), far-view shot (top right), medium-view shot (bottom left) and out-of-filed shot (bottom right).

Figure 9

Tournament logos present in the dataset. These logos can be used to identify and eliminate the playback shots present in the videos.

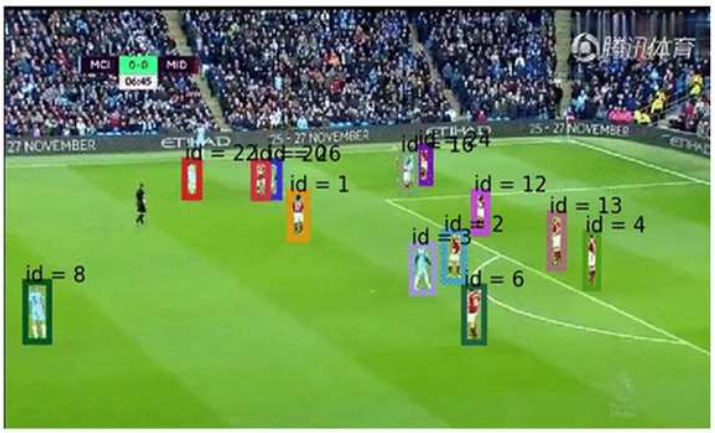

Figure 10

Output frames of the model with bounding boxes for the detections. Each bounding box is associated with the label (person or ball) and a confidence score with which the model made the detection. The SORT algorithm assigns an id is assigned to each player, which helps in tracking. Missed detections occur occasionally when the ball is severely occluded by the players.

Figure 11

Output frame of the GOTURN model for object tracking. It can be seen that the ball was grossly mis detected by the model which identifies the glove of the goal keeper as the ball because of similarity in color and shape with the ball.