Tracking and identification for football video analysis using deep learning

Conference Paper · January 2021

DOI: 10.1117/12.2586798

3 authors, including:

Shreedhar Rangappa
Loughborough University
7 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Ruiling Qian
Loughborough University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Tracking and Identification for Football Video Analysis using Deep Learning

Shreedhar Rangappa, Ruiling Qian, Baihua Li*
Loughborough University, United Kingdom

ABSTRACT

We describe the technique used to train and customise deep learning models to detect, track and identify soccer players, who are recorded during soccer game using custom camera setting. The player detection model is customised to allow detection of person class object from video input. Two newly developed filters, spatial feature filter and bounding box location filter have described that help in classifying players and audience. A new tacking paradigm is illustrated to generate tracks of soccer players with fewer swaps, thereby reducing efforts of human annotators in later stages. A new method of identifying every player by detecting player t-shirt number has been developed and illustrated. This method provides tracks with high confidence and identity to most of the player corresponding to individual t-shirt number. Finally, we provide a unique result assessment technique to judge the performance of the complete model.

Keywords: Detection, Tracking, Feature Matching, Deep Learning Model, Soccer

1. INTRODUCTION

Object detection and tracking have emerged as an important area of Computer Vision (CV) which has found application in many interesting areas such as autonomous driving, robotics, medical surgery and many more. The sports industry is also adopting new developments for recording and analysing the performance of sports personals. There have been recent advancements [1] in the way the sports events are captured using a single [2], multiple [3] and custom [4], [5] cameras setups. Traditionally, the captured sport event data was post-processed by human annotators to pin-point important actions, track players and provide key information, using which the coaches could analyse teams strategy and provide an individual rating to players on their performance during the sports events [6], [7]. But the traditional approach is very time consuming, since the process is very repetitive, and the annotation needs to be performed for almost all the frames. With advancements in CV and Artificial Intelligence (AI), some of the task performed by human annotations can be automated, thus speeding up the process. Performance analysis of sporting events, football, has been carried out for many years and various new techniques are employed to generate quick and accurate results. Some of the main actions in a football match that need to the annotated are player tracking, ball tracking and action recognition. These tasks are challenging due to the nature of the game and camera/s used to capture the match raise the difficulty level.

Extensive research has been carried out to improve the multi-object detection of different shapes and resolutions. New developments in AI has provided many trackers based on deep learning techniques. Most of the tracking algorithms use spatial/temporal features [8]–[10] matching score as key criteria, along with additional criteria's such as Intersection of Union, Kalman Filter prediction and inter-frame distance to generate tracking results. Fewer methods make use of recurrent neural networks (RNNs) [11], [12] to gain an advantage over temporal feature and also generate attention [13], [14]. These methods work well when objects are spatially differentiable. Multi-object becomes challenging when objects tracked are quite similar in spatial domain and location of these objects are overlapping or very close to one another. Due to the nature of football game, the players to be tracked wear the same jersey and the overlapping of players (same team or opposite team) are very frequent that can continue for many frames. We make use of overlap between all detected bounding box and employ the feature matching to only candidates of the individual track to generate fewer swaps and switches, maintaining high confidence in short-term.

Our entire work is divided into 4 sections, player detection, player-audience classification, player tracking and number tracking. Since we are using a custom setup camera system, the object detection models must be optimised to detect person class objects in all the frames. Using static bounding box location and spatial features, the detected person class objects are classified into players and non-players (audience). This helps to reduce the number of objects to be tracked. Using custom tracker, the objects are tracked every frame to produce online tracking results. This result is later used in detecting player t-shirt numbers and to join any gaps in the tracks.

2. PLAYER DETECTION

Player detection is one of the challenging tasks, due to the nature of camera settings used to capture the entire football match. For a successful detection of players using any deep learning-based model, the ground truth annotations play a vital role. In each football match recorded by Statmetrix camera settings, there are usually 22 players from 2

team which are the primary targets to be detected. Apart from players, they are many other person class objects on the pitch such as audience, non-playing players and referee. So, annotation of all objects in the videos require huge man-hours and hence an alternative solution with existing techniques must be used/customised.

Yolo [15] is one of the famous deep learning model used to detect person class objects in images. The frame dimension of Statmetrix videos are in the range of 1K - 2K pixels and 4K - 6K pixels in the height and the width, respectively covering the entire field in one frame. Resizing the full-frame image to Yolo network size (different network size of 32 multiples) and detecting person class objects results in problems such as single bounding box for multiple person objects and irregular bounding boxes as represented in Figure 1. This is due to the uneven resizing of the original image of ratio 1:3(h: w) into square images during inference.

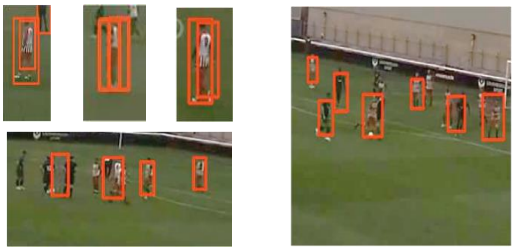

Figure 1 Irregular bounding box (left) and grouping problems (right)

To overcome the problem generated while using full-size images, we split the images into sizes that are in the ratio close to 1:1.5 (h: w). Along with splitting, increasing the network size allows the person class objects that far away from the camera to be detected. This is because the anchors used in Yolo model cannot detect the features when resized to such low resolutions. At a higher network size, the feature dimensions are preserved and detected by the trained anchors, thus increasing the overall Average Precision (AP). Using different spits to provide better results compared to full-size image and provide improved AP over the ground truth annotation (only considering player bounding box, and rest are neglected while calculating AP during inference testing) as shown in Table 1, where AC1 and AC5 are the football match videos used for testing of 5 minutes duration each. The network size and the quantity of spit used in the inference are highlighted in Table 1, that provide high AP in best possible time duration.

Table 1 Average precision of varying image sizes used to detect person class objects using Yolo-V3 model

Network size	Full Image		2 Split Image		3 Split Image	
	AC1	AC5	AC1	AC5	AC1	AC5
416	4.66	20.97	15.92	33.89	26.1	48.74
608	13.95	21.26	26.39	51.98	30.97	50.32
1024	29.51	43.98	31.4	49.98	**33.53**	**56.09**
1536	32.43	45.95	33.51	52.36	33.91	56.33

3. PLAYER-AUDIENCE CLASSIFICATION

The overall aim is to detect 11 players of each team for the successful generation of tracking results. Processing unwanted boxes cost processing time and unwanted errors are introduced during tracking players. This emphasizes the need for an algorithm to filter-out the non-players and audience from the detected person bounding boxes for each frame. We have employed two strategies to remove non-players by using spatial feature and bounding box locations.

3.1. Spatial Feature Filter (SFF)

In the spatial feature filter method, we make use of the spatial differences between players and audience. We collected over 10K images of players and non-players to train a deep learning model based on ResNet-50 architecture to classify the images into 2 categories: player and audience. This filter method accurately identifies the audience who are close to the camera, while produce low confidence to classify images that are far away from the camera. This is mainly due to the resolution of images, as far away images are of resolution 15x40 pixels. Also, the spatial difference between player and audience tend to reduce far away from the camera, and hence spatial feature filter alone cannot be employed for the task as shown in Figure 2.

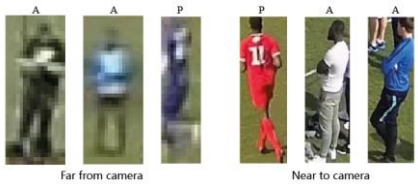

Figure 2 Appearance of Players (P) and Audience (A), far from the camera and near to the camera (images are resized)

3.2. Bounding Box Filter (BBF)

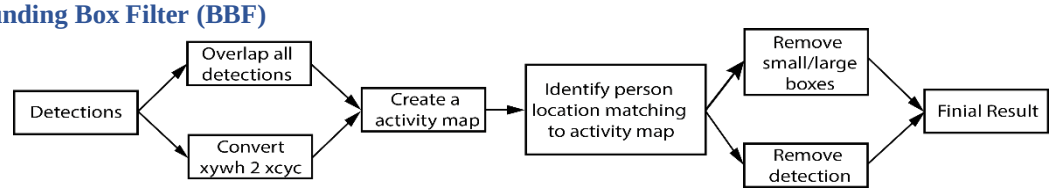

Figure 3 Flow chart of audience removal using bounding box technique

Figure 3 illustrates the outline algorithm use to filter-out non-players and audience per frame in a football match video using bounding box filter method. The main idea behind the algorithm is the nature of the game and how the action (movements) of player and non-player change during a football match. Audiences tend to have constant location compared to a player during a football match and this location displacement is considered as a key to classify player from all detected person class objects.

In the first stage, all the frames recorded during a football game is processed to generate the bounding boxes for person class. The detections are in the format of (x_1, y_1, x_2, y_2) and these detections are later converted to generate the centre location of each bounding box in the format of (x_c, y_c).The bounding box location of the entire match is overlapped to generate an activity map or heat map of the entire game. This map indicates the activity of persons in the football ground in terms of displacement as shown in Figure 4.

Figure 4 Top: Actual recording of a football gameplay Bottom (showing 1st frame only): Corresponding activity map of the entire football and colour bar indicates the number of detections overlapped at location

Depending upon the user requirement the threshold can be set to filter the bright spots. A default value of 0.5 can be used, which indicates a bounding box has the same location during 50% of the entire game and thus the location can be considered as a non-player bounding box.

A mask of non-players and audiences are generated by selecting all pixels that have a pixel value greater than the threshold set by the user. Using (x_c, y_c) information of the bounding box, the non-players are filtered out. All (x_c, y_c) that fall within the mask region are considered as non-players and rest of the detections are considered as players. This drastically reduces the amount of detection per frame to be processed to get necessary tracking information. The resulting mask used to remove the audience from the detection made using Player detection model and the performance of Spatial feature filter, Bounding box filter and the combination is represented in Table 2 and the overall result is shown in Figure 5, were extra bounding box consist of referee's and audiences.

Table 2 Performance of Spatial feature filter, Bounding box filter and combination in audience removal

5 min Video	Ground Truth Bounding Boxes	Detected Bounding Boxes	Filtered Bounding Boxes	Bounding box removed	Extra bounding boxes
AC1	188.9 K	454.1 K	SFM only	51%	17.79
			BBM only	43%	37.02
			BBM and SFM	**55%**	**8.17**
AC5	188.9 K	232.9 K	SFM only	21%	-2.5*
			BBM only	15%	4.79
			BBM and SFM	**18%**	**1.10**

*Removed more bounding boxes compared to required ground truth count

Figure 5 Overall result of Player-Audience classification (green bounding box are considered for next stages)

4. SPLIT FEATURE EXTRACTION AND MATCHING

Tracking players is one of the most crucial tasks in our work, as all statistical data of players such as running, scoring goal, kicking are associate with proper tracking. The video recorded by Statmetrix camera incorporates a unique Field of View (FOV), where the players are recorded at an angle facilitating two different overlapping scenarios. Firstly, the overlapping player completely occludes another player, where the extracted feature represents only one player which is similar to other tracking problems. In contrary, due to camera angle, overlapping players tend to occlude players body rather than an entire player and this occlusion happens very frequently. The feature extracted under this scenario does not provide high confidence to any single player. We employed Split-feature matching system to retain typical full-body feature extraction and gain an advantage over overlapping scenarios.

A split feature matching model is trained to provide player matching results in terms of percentage as shown in Figure 6. The model is trained with ResNet-50 backbone [16] to spatial extract features and Siamese architecture [17] style is used to match the temporal features. The model consists of splitter that splits the test and target image into two parts, Top and Bottom, to extract features that are matched individually to provide a value between 0-1 which is the matching percentage between the test and the target image. Since there are two matching results available after matching, the best matching score is considered as feature matching (FM) score by assigning equal weightage to both feature matching scores. In split feature match technique, variable weightage can be assigned which is not possible for full body matching.

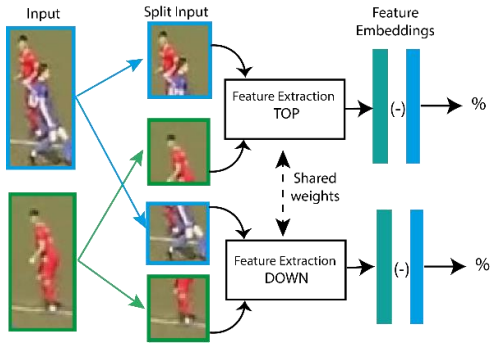

Figure 6 Split Feature matching

5. TRACKING

The primary task of our work is to track the football players accurately to be used by football teams to analyse the performance of individual players. Due to the nature of the football game, the tracking becomes highly difficult in the crowed scenarios and regions that are far away from the camera. Under such situations, human annotators will resolve the tracking issues during post-processing. Hence the main goal of tracking is to produce tracks of high confidence and reduce swaps as far as possible. MOT scores are considered as standard metrics to compare different tracking models and judge the best tracker for a given challenge. But in MOT metrics [18], events such as swap and rename/switch are given the same weightage. This metrics becomes unacceptable in scenarios where the main requirement is to generate tracks with a smaller number of Id swaps and an acceptable number of ID renames. Since to resolve Id rename, human annotator needs to focus in two frames where new Id is created, while in the case of Id swap the human annotator needs to focus on every individual frame to locate the swap frame and then correct the Id. Thus, generating confident short track is better than longer tracks with swaps. some of the keywords that are used in our work are highlighted below.

- Id swap: Id of two players are exchanged.
- Id Rename/Switch: Player Id is changed to a new Id.
- Id Copy: A single Id is assigned to two different players.

5.1. Assignment

Due to the nature of the game, the direction of players changes rapidly, and the action performed by the players changes the bounding box dimensions which some time cause error in Kalman filter prediction (K_p). Therefore, considering only Kalman prediction for assignment is not feasible. In our method, more weightage is given to previous frame detection, i.e. bounding box of available tracks, compared to Kalman filter predictions. Hence, for an assignment, we introduce gating distance (G_d) that provides candidates (c) for every track that have a high probability to be assigned with the tracks, $c \in \mathbb{R}$ as c is a subset of D.

$$ED(t,i) = \sqrt{(xc_t - xc_i)^2 + (yc_t - yc_i)^2} \tag{1}$$

$$G_d(t) = \max(w_t, h_t) \tag{2}$$

Where, ED is the Euclidean distance between the centre, (xc_t, yc_t), of the bounding box of a given track (t) and a detection (a). The bounding boxes are represented by (x, y, w, h). The G_d is the max value between width and height of a tracks bounding box. The detections that are assigned for a track is given by Equation 3.

$$T_{(assignment)}(\tau, i) = D_i \quad \begin{cases} ED\ (t,i) \leq G_d(t) \\ 0 \end{cases}$$
$$T_{(assignment)}(\tau) = \{\, c_0, c_1, \dots \dots c_n \} \in D \tag{3}$$

5.2. Matching

Equation 3 provides a list of candidates with high matching probability to a given track within τ. To select a matching candidate with a track we further generate Feature matching scores (FM) and Track overlap score (δ). We calculate the spatial similarity of two bounding boxes, the track (t) and the candidate detection (c), using a Split Feature matching method (Section 4). Additionally, when the two or more players overlap, the bounding box dimensions differ when there is just one player in the bounding box. To include this feature in the tracking, we calculate the overlap area between the track and candidate detection to measure the similarity of the bounding box area, Area Confidence (A_c). Equation 6 is used as criteria for a candidate detection to be considered for tracking by the user.

$$\text{Track Area} = \tau_{(1\dots t)}(\text{width} * \text{height}) \tag{4}$$

$$\text{Candidate Area} = c(\text{width} * \text{height}) \tag{5}$$

$$A_c(\tau, c) = \begin{cases} 1, & if \dfrac{\min(\text{Track Area}, \text{Candidate Area})}{\max(\text{Track Area}, \text{Candidate Area})} > 0.75 \\ 0, & otherwise \end{cases} \tag{6}$$

The feature matching result provides the spatial similarities for all τ, where $c = (c_0 \dots c_n)$, candidates under consideration.

$$FM(\tau, c) = \mathcal{M}^{mxn} * A_c \tag{7}$$

Track confidence (δ) is introduced as a helper function to FM, to identify the candidates allocated to a track that does not have any bounding box nearby. There are two main reasons for the feature matching model to generate low matching scores. By default, if spatially varying images end up very low matching scores. When the detection of players is not perfectly aligned with the bounding box details stored in existing tracks, the feature matching scores are low even though theoretically detection should have been a good match to a given track. This also helps to neglect some of the candidate bounding boxes that overlap more than the threshold (0.3) with other bounding box and has a high probability of introducing error in the tracking process.

$$\delta\ (\tau) = \mathcal{P}^{mx1} \tag{8}$$

5.3. Tracking Algorithm

Inputs: Tracks $\tau = \{0,1, \dots, t\}$, Detections $D = \{0,1, \dots, d\}$
1. Using G_d assign candidates to tracks, using Equation 2, $T_{(assignment)}(\tau) = \{\, c_0, c_1, \dots \dots c_n \} \in D$
2. Compute Feature matching score, *FM(τ,c)* using Equation 7,
3. Calculate the track overlap confidence, $\delta(\tau)$ using Equation 8,
4. Update tracks τ, generate confirm track list (c_τ) $c_\tau \leftarrow max \| FM(\tau, c) * \delta\ (\tau) \|$
5. Create new tracks, τ' $\tau' \leftarrow D \notin c_\tau$

6. LONG-TERM PLAYER TRACKING BY DIGIT DETECTION

In a football match, the key features that can be used to differentiate individual players are the temporal feature and player locations. Apart from these features, the t-shirt number of each player is one of the important features that can be helpful to identify each player. But, due to nature the video recording the detection of t-shirt number is difficult since the resolution of numbers is low (around 15x15 pixels). Also due to the nature of the game, the t-shirt deformation directly impacts the visibility number. Considering all these challenges, a number detection model is trained with Yolo-V3 as the backbone with a custom dataset.

6.1. Digits to Numbers

The dataset (image data) used for Digit detection model is a subset of football data used for feature extraction. The true bounding box for the players is reused to generate initial database consist of cropped images of players from the football match videos. This initial database consists of players in a different orientation, various locations within the football field. The images used in the digit detection model has been handpicked by human annotators within the initial database, and later annotated for different numbers ranging from 0-9 (10 classes).

The trained digit model predicts the digit in the player images and provides the bounding box detail during inference. Using this information, we can identify the digits on a player t-shirt and locate where exactly the digit is in the image, but there is no information regarding the order of digits. For example, when the model predicts digits like 1, 7 and 3, we are not sure of the order. It can be any of the following 13, 17, 73, 37, and so on. Therefore, it becomes necessary to have an algorithm to sort these digits to numbers. The overview of the algorithm is represented in Figure 7, where digit 1 is in the pairing zone of 7, while 3 is in the non-paring zone. Hence 7 is paired with 1 for further processing and 3 is considered a single digit in this example.

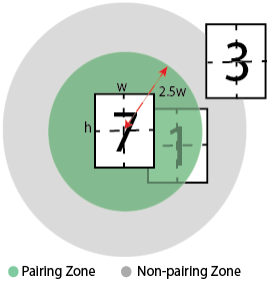

Figure 7 Different zones of digits to number conversion

For a given i^{th} image, let D_n be all the predictions from the digit detection model with bounding box locations at $(x_1, y_1)_n$ and $(x_2, y_2)_n$with $(x_c, y_c)_n$ as their corresponding centres. Let $(w, h)_n$ be the width and height of each prediction in D_n. Let $dist_n$ be the Euclidean distances for all predictions within D_n. Let d_i and d_j be the two detections under consideration with $dist_{i-j}$ being their Euclidean distance. Using the algorithm shown in Table 3, numbers with probability for consideration is at top of $d_{i-j}\ sorted$ list.

Table 3: Digits to Number algorithm

Inputs: D_n ∀ Frames
1. Generate Pairs and Single digits ***For all*** $\boldsymbol{d_i} \in \boldsymbol{D_n}$ ***do***: $d_j = min\|\|dist_n\|\|$ $d_i \xrightarrow{pairs} d_j \ \begin{cases} dist_{i-j} \geq w_i \\ dist_{i-j} \leq 2.5w_i \end{cases}$ $d_i \xrightarrow{pairs} None \ \begin{cases} dist_{i-all} \leq w_i \\ dist_{i-all} \geq 2.5w_i \end{cases}$
2. Sort Left to Right ***For all*** $\boldsymbol{d_{i-j}}$ ***pairs*** $\in \boldsymbol{D_n}$ ***do***: $sort\ Left-Right\ with\ (x_i, x_j)$ as key $d_{i-j}\ sorted \xrightarrow{L-R} d_i * 10 + d_j \quad if\ (x_i < x_j)$ $d_{i-j}\ sorted \xrightarrow{L-R} d_j * 10 + d_i \quad if\ (x_i > x_j)$
3. Sort Centre to Outwards ***For all*** $\boldsymbol{d_{i-j}}$ ***sorted*** $\in \boldsymbol{D_n}$ ***do***: $distance\ of\ d_{i-j}\ and\ i^{th}\ image\ center\ (x, y)\ as\ key$ $d_{i-j}\ sorted \xrightarrow{C-O} d_{i-j}\ sorted$

6.2. Linking and Tracking Numbers

Since the digits are identified on the t-shirt of players who are on constant motion, the digits do not always stay parallel to the camera. Also, the orientation of the players keeps changing hence there are great chances that the model will be able to see just one digit on the t-shirt out of two digits. Hence the high confidence number detected do not always match the true number on the t-shirt. There is a high probability of model predict the wrong number with high confidence (false positive) due to deformation of a t-shirt during gameplay.

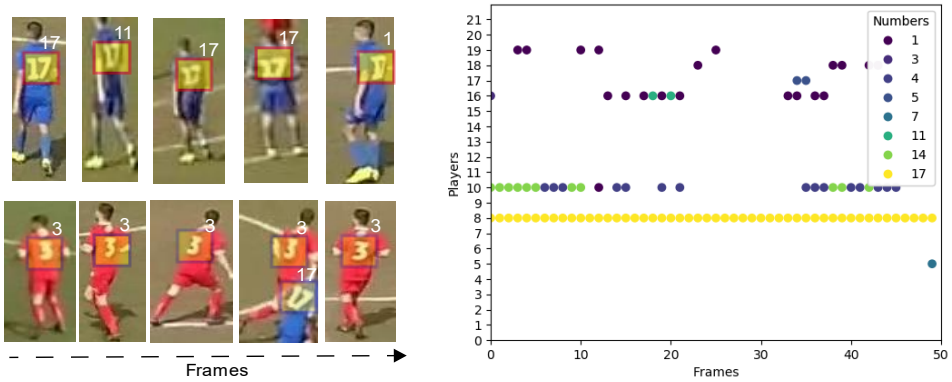

Figure 8 Illustration of Number linking and tracking for 2 players (left), Number tracking for 22 players (right)(only a few frames shown for better visualization, y-axis ticks represent the player index, not the t-shirt number)

Also, additional data can be generated by considering previous and current bounding boxes, to determine the relationship between detections and use the result during assigning a number to the corresponding track as shown in Figure 8 (left) for two players. Since the changes in player orientation are very frequent during football gameplay, probabilities of the wrong classification to the detected number is acceptable which are false positives. This error is hard to identify with a single detection, and hence information of previous detections of the same player is necessary to provide some degree of confidence in considering the classification of detected number. Figure 8 (right) shows different numbers detected for 22 players shown with their index on y-axis and number of frames on the x-axis.

The distance between two corresponding detections, d_{pc} – distance between previous and current detection locations, of the same player, is calculated and used as criteria to assign a t-shirt number to track as well as reduce false positives. For a given short track all digits are converted to numbers using the method explained earlier. Using d_{pc} distance rule, a chain of detected numbers is formed by including the detected number and their decomposed digits. For example, detected number {24} and decomposed digits {2,4}, then numbers considered to generate chain of detected number is {24,2,4}. To assign a number to a specific track lets $T = \{t_0, t_1 \dots \dots \dots t_n\}$ be the list of tracks considered. D_f be the list of frames with a number detected, and D_f^T be the list of D_f for all the T,

$$D_f^T = \begin{cases} D_f^{t_0} = \{I, J, K \dots \dots .\} \\ D_f^{t_1} = \{I, J, K \dots \dots .\} \\ \vdots \\ D_f^{t_n} = \{I, J, K \dots \dots .\} \end{cases} \tag{9}$$

where $D_f^{t_0} = \{I, J, K \dots \dots .\}$ are the list of numbers used to generate a chain of detected numbers, with $D_l^{t_0}$ as the total count of number detected frames. Let the strength of two corresponding detections be given by S_f in Equation. 10,

$$S_f = \frac{1}{e^{-(1-(f_{i+1}-f_i))}} \tag{10}$$

where (f_{i+1}, f_i) are the consecutive frames of a track. Strength of individual track, for a specific number I of track t_0 is given by with $threshold$ being a minimum value set by the user.

$$S_{t_0=I} = \left(\sum_{i=0}^{f} S_f\right) \Big/ D_l^{t_0} \quad \begin{cases} S_f \geq threshold \\ 0 < threshold \end{cases} \tag{11}$$

The final assigned number (AD) for track t_0 is the one with maximum strength among the list of numbers $\{I, J, K \dots \dots .\}$ is given by Equation 12.

$$AD_f^{t_0} = max\left\|\{S_{t_0=I}, S_{t_0=J}, S_{t_0=K} \dots \dots .\}\right\| \tag{12}$$

7. RESULTS

The metrics used in standard tracking, MOT, is difficult to provide the individual components/events that are essential in this work such as swaps, switches, id-copies. Hence, we have come up with simple metrics that illustrate the key components to decide the performance of tracker as well as other additional methods used to provide a complete solution. The ground truth (GT) detections are matched with Test (T) detections using techniques followed in standard MOT benchmark [18], [19]. Additionally, we keep track of swaps, switches, and id-copies along with the age of events to know how long an event occurs, to measure the tracking quality. For example, if an Id swap happens at 45th frame and continues until the 50th frame, the age of swap event is 5 frames. Similarly, all events are recorded along with its age corresponding to all the GT tracks. We have set age/limits of [1,5,10,30] frames to understand the solution generated by models compared to ground truth annotations. For example, 25 swaps at the limit of 5 frames suggest the tracks has 25 swap event which are more than 5 frames, which needs to be resolved by the operator at later stages. The result of tracking is represented in Table 4.

Table 4 Results of tracking

Video	Threshold	Number of Swaps	Number of Switches	Number of Copies
AC1	@ 1 Frame	34	365	0
	@ 5 Frames	28	188	0
	@ 10 Frames	27	133	0
	@ 30 Frames	23	99	0
AC5	@ 1 Frame	198	1035	0
	@ 5 Frames	69	886	0
	@ 10 Frames	36	602	0
	@ 30 Frames	16	366	0

The tracking result is further processed using the number tracking method described in Sec 6. With t-shirt ID as a key, the tracks are joined to further increase the track length. By using this technique, the average track length was increased from 869 frames to 1091 frames by joining 24 tracks out of 365 tracks for AC1 video. Similarly, the average track length of AC5 video was increased from 572 frames to 785 frames by joining 31 tracks out of 338 tracks.

There are few drawbacks of our methods used in this work. Firstly, SFM+BBM method some time fails especially for the goalkeeper, who tends to have constant location during the football match and far audience. Secondly, the tracking method sometimes generates new tracks (switches), as we have introduced area confidence and track overlap score terms. These parameters override the high feature matching score and introduce a new track to provide high confidence short track.

8. CONCLUSION

In our work, we have described different deep learning models that were designed and trained to accomplish the task of detecting players, detecting digits, classification, and tracking players. Due to the camera setting and resolution of players, the Yolo-v3 model was customised to detect most of the person class objects in the video. The technique of image splitting has been adopted which detects most of the bounding boxes, as demonstrated in Table 1. The classification of audience and players using SFM and BBM are accurate over 80%, and the clear advantage of using both the methods together is shown in Table 2. The tracking results on both the videos show that the amount of clicks the operators need to perform is under 350 to resolve the switches. Also, the number of swaps is under 25 in both the videos that requires fewer more clicks. The additional number tracking model helps to join the tracks and increase the average track length. In both the videos the number tracking model was able to join more than 20 tracks and add around 200 frames worth of tracking data.

In our work, few of the deep learning models are customised for the Statmetrix video inputs, to provide results that help to reduce operator effort. But these models can be used for any other sporting video to generate tracking data and provide identity corresponding to the individual's t-shirt number. As future work, we plan to use split-feature matching for other tracking problem to see the advantage over traditional full-body matching.

Acknowledgements

This work was funded by the Innovate UK. Authors acknowledge the collaboration with Statmetrix Ltd. and the ground truth data provided by them.

REFERENCES

[1] M. Manafifard, H. Ebadi, and H. Abrishami Moghaddam, "A survey on player tracking in soccer videos," *Comput. Vis. Image Underst.*, vol. 159, pp. 19–46, Jun. 2017.

[2] N. Babaguchi, Y. Kawai, T. Ogura, and T. Kitahashi, "Personalized abstraction of broadcasted American football video by highlight selection," *IEEE Trans. Multimed.*, vol. 6, no. 4, pp. 575–586, Aug. 2004.

[3] "CaptureMast | Sports Filming Mast Packages | The UK'S leading specialists |." [Online]. Available: https://capturemast.co.uk/. [Accessed: 29-Jan-2020].

[4] "Veo Camera - Veo sports camera everything you need to know." [Online]. Available: https://www.veo.co/camera/. [Accessed: 29-Jan-2020].

[5] "SportsCam Automatic - Provispo." [Online]. Available: https://provispo.com/sportscam-automatic/. [Accessed: 29-Jan-2020].

[6] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9914 LNCS, pp. 17–35.

[7] P. S. Bradley *et al.*, "Match performance and physical capacity of players in the top three competitive standards of English professional soccer," *Hum. Mov. Sci.*, vol. 32, no. 4, pp. 808–821, Aug. 2013.

[8] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings - International Conference on Image Processing, ICIP*, 2016, vol. 2016-Augus, pp. 3464–3468.

[9] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards Real-Time Multi-Object Tracking," in *arXiv preprint arXiv:1909.12605 (2019)*, 2019.

[10] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[11] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," *31st AAAI Conf. Artif. Intell. AAAI 2017*, pp. 4225–4232, 2017.

[12] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 4705–4713.

[13] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. H. Yang, "Online Multi-Object Tracking with Dual Matching Attention Networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, vol. 11209 LNCS, pp. 379–396.

[14] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 1, pp. 666–673.

[15] A. E. Özdener and A. Rivkin, "You Only Look Once: Unified, Real-Time Object Detection Joseph," *Drug Design, Development and Therapy*, vol. 11. pp. 2827–2840, 08-Jun-2017.

[16] T. Akiba, S. Suzuki, and K. Fukuda, "Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes," Nov. 2017.

[17] A. J. Nathan and A. Scobell, "How China sees America," *Foreign Affairs*, vol. 91, no. 5. pp. 956–963, 2012.

[18] K. Smith, D. Gatica-Perez, J. Odobez, and Sileye Ba, "Evaluating Multi-Object Tracking," 2006, pp. 36–36.

[19] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," *Comput. Vis. Pattern Recognit.*, 2016.