An Automated System for Generating Tactical Performance Statistics for Individual Soccer Players from Videos

Rajkumar Theagarajan, *Student Member*, *IEEE*, and Bir Bhanu, *Life Fellow*, *IEEE*

Abstract— **The world of sports intrinsically involves fast and complex events that are difficult for coaches, trainers and players to analyze, and also for audiences to follow. In fast paced team sports such as soccer, keeping track of all the players and analyzing their performance after every match are very challenging. Current scenarios for identifying the best talents in soccer involve word-of-mouth and coaches/recruiters scouring through hours of manually annotated videos. This is a very expensive and laborious process and also biased by the nature of the recruiters. To alleviate these problems, this paper proposes an automated system that can detect, track, classify the teams of multiple players and identify the player controlling the ball in a video. The system generates three very important tactical statistics for a player: 1) duration of ball possession, 2) number of successful passes and 3) number of successful steals. This is done by training Convolutional Neural Networks (CNNs) to (a) localize and track the players on the field, (b) classify the team of a detected player, (c) identify the player controlling the ball and (d) pooling all the information extracted from (a), (b), and (c) to generate the statistics of players. To overcome the problem that the features learned from specific soccer matches do not necessarily generalize across different soccer matches, the paper proposes minimal amount of match-specific annotation and data augmentation, using a variant of Deep Convolutional Generative Adversarial Networks (DCGAN) to improve the accuracy. Experimental results and ablation studies show that the proposed approach outperforms the state-of-the-art approaches in terms of accuracy and processing speed.**

Index Terms—**Convolutional Neural Networks, video analysis, sports analytics, player statistics.**

I. INTRODUCTION

In recent years, automatic interpretation of sports has gained a keen interest. It is a challenging task especially when it involves rapid changes and long-term dynamics. To date most of the applications for providing sports analysis and player training from videos are carried out manually. This requires lots of hours spent watching videos and annotating them. Computer vision and machine learning play a key role in the world of sports in areas such as player detection, player tracking, action recognition and player analysis.

Soccer is one of the most popular sports played by high school students in the USA. According to a survey conducted by Ranker.com [1] and Statista.com [2] 846,844 (456,362 boys and 390,482 girls) high school students played soccer during the year 2017/18. From this only 9% of the boys and 11.9% of the girls receive scholarship to go to college which makes it extremely competitive.

TABLE I: Statistics of the number of male and female soccer players in high school, NCAA* and MLS* [2].

Year	High school		NCAA			MLS
	Male	Female	Div 3	Div 2	Div1	
2011/12	411,757	370,975	10,117	6,076	5,153	38
2012/13	410,982	371,532	11,097	6,165	4,832	38
2013/14	417,419	374,564	10,870	6,261	5,426	77
2014/15	432,569	375,681	11,679	6,489	5,623	84
2015/16	440,322	381,529	11,889	6,805	5,724	75
2016/17	450,234	388,339	11,256	6,754	5,953	81
2017/18	456,362	390,482	12,322	6,545	5,933	81
2018/19	459,077	394,105	NA	NA	NA	NA

*NCAA stands for National Collegiate Athletic Association
*MLS stands for Major League Soccer

Table I shows the statistics of the number of male and female soccer players in high school, the National Collegiate Athletic Association (NCAA) and the Major League Soccer (MLS) for the years 2011 - 2019. After graduating from high school, less than 6% of soccer players get qualified for the NCAA. The NCAA consists of three tiers namely: Division 1, Division 2 and Division 3. Most of the Major League Soccer (MLS) recruiters seek out only the players in Division 1 and Division 2 and less than 100 of them qualify to become pros in the MLS every year.

The most common reason for players not qualifying for the NCAA or other soccer clubs is that coaches do not have enough time to observe the performance of every player. To alleviate this problem, we propose an automated system that can analyze the performance of all the soccer players, identify the player controlling the ball in a video and generate the tactical statistics of each player.

In team-based sports, such as soccer, talent identification is a complex process due to the different qualities associated with performance; they include technique, tactics, fitness and psychological attributes [3].

Technique involves a player's style such as dribbling and how offensive/defensive the player plays.

Tactics involve attributes such as how well a player is able to control the ball and play with the team-mates.

Fitness involves attributes such as the fatigue, stamina level of the player and history to injuries.

Psychological involves the emotional intelligence of a player and how a player deals with interpersonal and intrapersonal conflicts.

In this paper, we focus on generating directly from the video, three very important tactical statistics for a soccer

This work was partially supported by a grant from NSF 1552454, Bourns Endowment Funds and a gift from SEVAai Inc. The contents of the information do not reflect the position or the policy of the US Government.

Rajkumar Theagarajan and Bir Bhanu are with the Center for Research in Intelligent Systems, University of California, Riverside, CA 92521 USA (e-mail: rthea001@ucr.edu and bhanu@vislab.ucr.edu).

player namely: **(i)** duration of ball possession, **(ii)** number of successful passes and **(iii)** number of successful steals. **To the best of our knowledge this is the first paper in the field of *computer vision* and *circuits and systems for video technology* that can generate tactical statistics for individual soccer players directly from a video.**

Importance of tactical statistics: Ball possession is a very important statistic (stat) for a player as it is has an influence on other statistics such as the number of successful shots at the goal and number of tackles won/lost [4]. The number of successful passes made by a player is very important because the number of overall attempted passes and number of successful passes are important factors in achieving better results leading to winning a match [5] - [7]. It has been shown that the accuracy of successful passes increases significantly five minutes before scoring [8]. The number of successful steals within the 6 yard area has been shown to increase the number of shots at the goal [9]. Interestingly, unsuccessful teams tend to play more within their half of the soccer field which increases the chances of the opposition to steal the ball which in turn increases their chances for shots at the goal [10].

In order to develop an automated system to compute the tactical statistics for players we collected a dataset that consists of 49,950 images of high school soccer players which are annotated into two classes namely: "*Players with the ball*" (12,585) images and "*Players without the ball*" (37,365) images. The first step in our system involves detecting the players using the YOLOv2 framework [11] and tracking them using the DeepSort algorithm [44]. Next, the detected players are passed through a Triplet-Convolutional Neural Network (CNN) that extracts fine-grained features which are used for predicting the team of the player and finding out if the player is controlling the ball. While trying to solve this problem, we also address two key issues: *Speed Vs. Performance* and *Generalizability*.

Speed Vs. Performance: In the field of sports analytics, the speed at which algorithms perform without sacrificing accuracy is very important. For example, on the COCO dataset [12] the algorithms with the best performance are rather slow [13], [14], while the real-time algorithms have lower accuracy [11], [15]. In this paper, we experiment with different architectures of CNNs and show that during inference, our approach is computationally more efficient as compared to the state-of-the-art approaches while not sacrificing too much accuracy. ***It should be noted that our system is not intended to run real-time but instead to be used as a tool for post-match analysis***.

Generalizability: A significant problem is the lack of generalizability, whose origin is at least two fold in sports video analysis: intersport variability, and intrasport variability. It is currently too ambitious to hope for a universal system that can perform accurate player analysis on any sports video, which underlines the need for developing sports-specific models. Besides, even within videos from a single sport, some play conditions may change from one match to the next, such as the outfits of the teams and environmental conditions in the case of outdoor sports such as soccer. Fast algorithms may be less robust to such variations, which might make them non-reusable from one match to the next.

Rather than trying to unify all of these conditions within a single network, it is more appropriate to re-train the network for every match in order to adjust to the conditions [16]. To achieve this we experiment with different soccer matches played by different teams and find the least amount of images that need to be annotated in order to achieve a robust performance. Another problem that arises is that how do we annotate images for a match that has not yet been played? To solve this we annotate images of matches that have been previously played by the same teams and then re-train our models and evaluate them on the match that is to be played.

In summary, the contributions of this paper are as follows:

- ***Tactical Statistics:*** Unlike previous research in the field of computer vision, sports analysis and circuits and systems for video technology (see Table II), this is the first paper that can automatically generate three quantifiable tactical statistics (duration of ball possession, number of successful passes, and steals) of individual soccer players from a video. In addition, an ablation study is carried out to show how different combinations of the individual modules affect the generation of tactical statistics at a match level and an individual player level.
- ***Generation of fine-grained synthetic images:*** This paper designs a novel Triplet CNN-DCGAN architecture for generating fine-grained synthetic images of soccer players controlling the ball. The paper also performs an ablation study to show how data augmentation helps to improve the generation of tactical statistics.
- ***Minimum annotation for robust performance:*** This paper shows that features learned from a specific soccer match do not generalize across all soccer matches. To overcome this problem, the proposed approach requires only 100 annotated images per class (*Player with/without the ball*) from any given soccer match to achieve a robust performance.
- ***Performance evaluation and comparison of individual modules:*** This paper performs extensive evaluation, ablation studies, and comparison of the individual modules used in the proposed approach with the state-of-the-art using a dataset consisting of 49,950 images which are collected from different soccer matches.

This rest of this paper is organized as follows. Section II describes the related work and technical approach is explained in Section III. Experimental results and ablation studies are shown and discussed in detail in Section IV. Finally, Section V concludes the paper.

II. Related Work

In this section, we describe various state-of-the-art approaches that have been used in sports for detecting and tracking players, identifying teams and events, and player analysis. Table II shows a summary of the related work. In

TABLE II: Summary of the related work.

Authors	Application*	Comments
Duh *et al.* [17]	**PT**	Used histogram and spatial similarity matrix to track players
Liu *et al.* [18]	**PT**	Used clustering to detect players and Markov Chain Monte Carlo association for tracking
Chiang *et al.* [19]	**PT**	Used mean shift segmentation to discriminate between a moving player and entire frame
Xing *et al.* [20]	**PT**	tracked players under occlusions using a general tracking and an online model
D'Orazio *et al.* [21]	**PT**	Tracked players by matching their bounding box features with players in the next frame
Khatoonabadi and Rahmati [22]	**PT**	Used histogram based template matching for tracking soccer players
Senocak *et al.* [23]	**PD**	Trained a CNN to extract full body and holistic fisher vectors [24] to detect basketball players
Xu *et al.* [25]	**PD**	Detected soccer players from 4K videos using a CNN based region detection
Liu and Bhanu [26]	**PD**	Used pose-guided RCNN [27] to detect and classify the jersey number of sports players
Istasse *et al.* [28]	**TD**	Distinguished between different teams by extracting the pixel-wise embedding using CNN
Lu *et al.* [29]	**TD**	Distinguished between different teams by using patch based histogram matching
Theagarajan *et al.* [30]	**TD**	Used template based histogram matching to distinguish between different teams
Cai *et al.* [31]	**ED**	Used the Part Affinity Fields [32] and optical flow [33] to detect actions of ice hockey players
Piergiovanni and Ryoo [34]	**ED**	Classified the type and speed of the pitch to predict events in baseball videos
Cioppa *et al.* [35]	**ED**	Used contextual and player grouping information to predict the style of the soccer game
Tora *et al.* [36]	**ED**	Extracted contextual and holistic features using CNN-LSTM to predict events in ice hockey
Fakhar *et al.* [37]	**ED**	Used pooled spatial pyramid feature based sparse representation on highlight videos
Li and Bhanu [38]	**PA**	Used the player key points and dribble energy image to classify the dribbling style
Theagarajan *et al.* [30]	**PA**	Used CNN to detect the player controlling the ball in high school soccer videos
This paper	**PA**	Designed an automated system to extract fine-grained features to identify the team, player controlling the ball and predict the tactical statistics of individual soccer players in a video

***Abbreviations**: **PT**: Player Tracking, **PD**: Player Detection, **TD**: Team Detection, **ED**: Event Detection, **PA**: Player Analysis

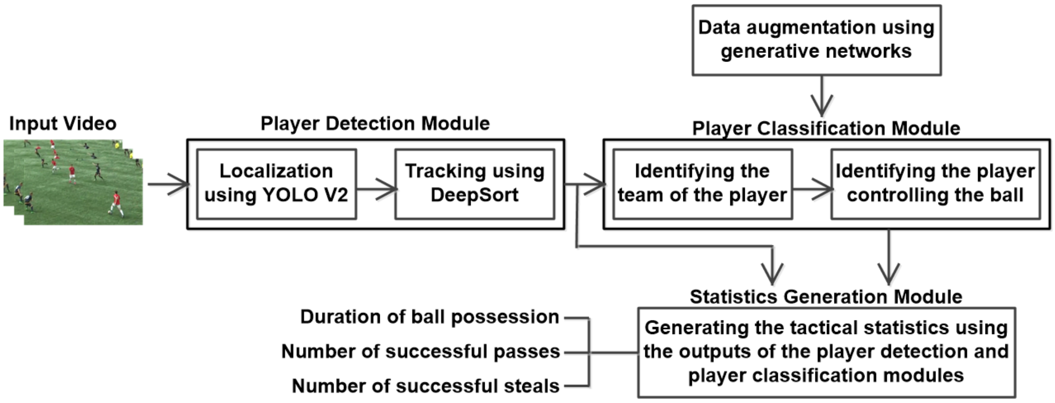
Data augmentation using generative networks
Input Video
Player Detection Module
Localization using YOLO V2
Tracking using DeepSort
Player Classification Module
Identifying the team of the player
Identifying the player controlling the ball
Statistics Generation Module
Generating the tactical statistics using the outputs of the player detection and player classification modules
Duration of ball possession
Number of successful passes
Number of successful steals

Fig. 1: Overall architecture of our approach.

contrast to the state-of-the-art approaches described in Table II our work is significantly different in the following aspects:

- **Generating Tactical Statistics**: To the best of our knowledge, there exists no other work that can generate quantified tactical statistics for soccer at a match, team, and individual player level directly from a video.
- **Team Identification**: Unlike previous approaches that use clustering based techniques such as [18], [22], [28] and ad hoc histogram based matching such as [29], [30], [53] which are susceptible to the player pose and environmental conditions, we evaluate three different approaches using Siamese and Triplet CNNs and show that our approach is more robust and outperforms the state-of-the-art approaches in Table II by 26%.
- **Player Analysis**: Prior work done by Theagarajan *et al.* [30] shows that regular CNNs have trouble in detecting minute details such as the soccer ball in low resolution images which is important for differentiating between a player with and without the ball. To overcome this problem, we extract fine-grained features using a Triplet CNN trained on only 100 images per class (*Player with/without the ball*) and show that our approach outperforms the state-of-the-art by at least 14% and has significantly reduced number of parameters.
- **Generation of Fine-grained Synthetic Images**: This paper shows that regular Generative Adversarial Networks (GANs) often overlook minute details such as the soccer ball when generating synthetic images. To overcome this problem, this paper designs a Triplet CNN-DCGAN for generating fine-grained images of soccer players controlling the ball and performs an ablation study to show the improvement in generating tactical statistics with and without data augmentation.

III. Technical Approach

In this section, we explain the overall framework and its individual modules of our approach shown in Fig. 1. The input video first passes through the player detection module where the soccer players are detected, tracked and cropped. Next, the cropped images are passed through a player classification module which consists of two Triplet CNNs trained to extract fine-grained features and 1) predict the team of the players and 2) identify the player controlling the ball. Next we pool together the outputs of the player detection and classification modules for the entire video to generate the tactical statistics for all the individual players.

A. Localization and Tracking

*1) **Localization of Soccer Players**:* As the number of high school soccer players keeps increasing every year as shown in Table I, the demand for coaches to provide feedback to the players increases significantly. It has been shown that during high school soccer season, 39 hours of videos are uploaded every minute on the Hudl online platform for processing [61]. It is a very laborious task to analyze all the videos manually. Hence, it is important to have a system that can automatically detect soccer players in the video at a reasonably fast speed without sacrificing too much accuracy. In our approach, we evaluated four different object detectors that achieve state-of-the-art performance on the COCO dataset [12] (see Table IV). YOLOv2 [11] achieved a detection accuracy of 84.81% and Mask R-CNN [39] outperformed YOLOv2 by 1.87% in accuracy. However, YOLOv2 operates 10x faster (at 17.2 FPS) than Mask R-CNN. This is very important since it is unreasonable for a coach to wait 30 - 40 minutes to process a 2-minute video at 30 FPS using Mask R-CNN compared to waiting for 3 - 4 minutes using YOLOv2 for a very small trade-off in accuracy.

YOLOv2 initially divides the input frame into a 11×11 grid. Each grid predicts *B* bounding boxes and the confidence score associated for each bounding box. Formally, the confidence is defined as Pr(object) * IOU, where Pr(object) is the probability of an object present and IOU is the Intersection Over Union between the predicted bounding box and the ground-truth bounding box. This probability is conditioned on the grid cell containing one object meaning that if there is no object present on the grid cell, the loss function will not penalize the CNN for a wrong class prediction. The network was trained on the COCO 2016 key points challenge dataset [12]. This dataset consists of diverse images for the class "*Person*" which also includes sports players and the images in this dataset have different scale variations, and occlusions which are similar to the scenario of a soccer field. For a given frame, the bounding boxes belonging to the class "*Person*" with probability greater than a given threshold are considered to be the locations of the soccer players for that frame. In our approach we set the threshold to be 0.5.

*2) **Tracking of Soccer Players**:* In broadcast videos of soccer league matches, the camera operator is located at least 100 feet away from the side lines of the soccer field at a reasonable height. This provides a wide Field-of-View (FoV) for the camera operator and there is very small amount of pan and tilt. In these kinds of videos when a player is moving on the field, the camera operator also pans the camera very gradually such that the Cartesian coordinates of the soccer player in the video has minimal change. In high school soccer videos such as our dataset, due to the availability of limited space, the camera operator is located just 20 - 30 feet from the side lines of the soccer field at a height of 15 feet. This creates a very narrow FoV for the camera operator which leads to a large amount of pan, tilt and zoom even when the soccer players are not running very fast.

Based on these constraints, we experimented with five state-of-the-art tracking algorithms (including algorithms proposed in [45]) and found that DeepSORT proposed by Wojke *et al*. [44] performs the best in our scenario. The reason for this is that, unlike the other algorithms [45] that solely rely on the features extracted from object detectors, DeepSORT also uses 8-dimensional state space vector which is given as input to a Kalman filter. This state space vector contains the information such as velocity and direction in which a player is moving relative to the camera. Assuming that a player usually moves at a constant velocity relative to the camera that is located at a distance and the tracklets of a player always follow a linear model (a player cannot arbitrarily appear at different locations in consecutive frames), the Kalman filter is able to track the individual players more consistently. In the case of soccer videos if a group of players of the same team are close to one another, it would cause association problems if the tracking algorithm solely relied on the features extracted from an object detector. DeepSORT is able to handle these kind of situations with much ease compared to the other methods [45] because it uses the combination of an object detector and a state space Kalman filter that assumes that players move linearly in a video.

In our approach the CNN used for detecting the players is YOLOv2 [11] and the 8-dimensional state-space vector is represented by ($u, v, \gamma, h, u', v', \gamma', h'$) where, ($u, v$) is the image coordinate of the center of the bounding box, γ is the aspect ratio, h is the height of a bounding box and (u', v', γ', h') are their respective velocities in the image coordinate. Finally, the features extracted by YOLOv2 and the state-space motion vector are concatenated and passed as the input to a Hungarian algorithm [46].

It should be noted that our system does not perform any camera calibration and the camera operator is allowed to freely pan, tilt and zoom the camera depending on where the action is happening on the soccer field. Due to this if a player moves out of the field-of-view of the camera and re-appears after a while, the algorithm labels the player as a new person. In this situation our approach will treat such players as new players and continue to generate the statistics. It is very challenging to track players under such conditions using state-of-the-art long-term tracking and re-identification algorithms and it is a problem of its own. In the case of soccer videos, it is more challenging because players belonging to the same team wear the same jersey and they visually look very similar from the camera's perspective.

B. Team Identification

In this sub-section we propose three different approaches (*TI-1, TI-2, and TI-3*) for predicting the team of the players and compare their pros and cons individually.

*1) **TI-1: Cross Dataset Transfer Learning and Feature Matching using Siamese CNN**:* In soccer, since players belonging to the same team wear the same color of jersey, we can formulate the task of player team identification as a person re-identification problem. In this approach we train a Siamese CNN on the Town Center subset of the PETA dataset [48] for the task of pedestrian re-identification with two output classes "*Same person*" and "*Different person*". Fig. 2 shows the architecture of the Siamese CNN. In Fig. 2 Conv(x, y, z) represents the dimension of the filter (x), stride of the filter (y), and padding (z). We used the Siamese loss function for training and it is given by:

$$Loss_{Siamese} = (1-Y)\frac{1}{2}D_W^2 + Y\frac{1}{2}\{max(0, m - D_W)\}^2 \tag{1}$$

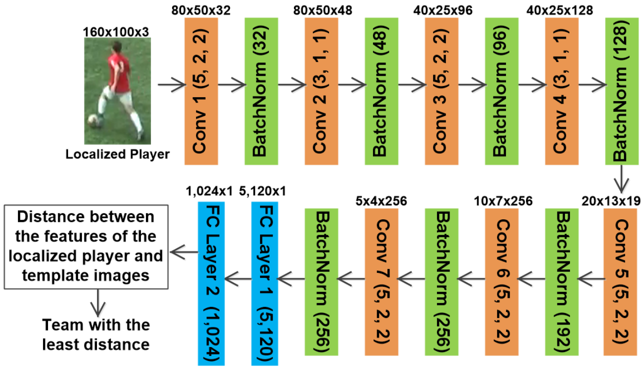

Fig. 2: Architecture of the Siamese CNN.

In Eq. (1), D_W is the Euclidean distance between the outputs of the Siamese networks, *m* is the margin and is chosen as 1. If the inputs are from the same class, then the value of *Y* is 0, otherwise *Y* is 1. After training the Siamese CNN on the PETA dataset [48], we evaluated the CNN on our dataset. Initially we select 10 template images for each team. Next, we pass the detected player through the Siamese CNN and we extract a feature vector M. Similarly, we also extract the feature vectors of the 10 template images of each team N_i where *i* = 1 to 10. Next, we compute the average Euclidean distance between *M* and N_i for both the teams and the team with the least average Euclidean distance is taken as the final prediction. The advantage of this approach is that it requires only training the CNN on a publicly available dataset and is the most generalizable approach.

*2) **TI-2: Fine Tuning and Feature Matching using Siamese CNN**:* This approach is similar to **TI-1** except that after pre-training on the PETA dataset [48] we further fine tune the Siamese CNN with images from our dataset as well. This helps the CNN to learn features specific to the match being played which improves the performance of prediction. During testing similar to **TI-1**, we select 10 template images from each team and compute the average Euclidean distance between the feature vector of the detected player M and the feature vector of the template images N_i for both the teams and the team with the least average Euclidean distance is taken as the final prediction. Experimental results (see Table VII) show that this approach performs better than **TI-1**.

*3) **TI-3: Fine-grained Feature Extraction using Triplet CNN**:* Triplet CNNs are known for extracting fine-grained features while maximizing the interclass variance and minimizing the intraclass variance at the same time [49] - [52]. In this approach we use the same CNN architecture as in Fig. 2 and the only change is that we replaced the final fully connected layer (FC Layer 2) with two output nodes for "*Team A*" and "*Team B*". We use both the Triplet loss Eq. (2) and binary cross entropy loss Eq. (3) for training the Triplet CNN. During testing, this approach does not require any template images for matching.

$$Loss_{Trip} = Max(0, -Y * (G(X_1) - G(X_2)) + m) \tag{2}$$

$$Loss_{BCE} = \frac{-1}{n}\sum_{i=1}^{n} y_i * log(p_i) + (1 - y_i) * log(1 - p_i) \tag{3}$$

$$Loss = \alpha_1 \times Loss_{Trip} + \alpha_2 \times Loss_{BCE} \tag{4}$$

In Eq. (2), X_1 and X_2 are the two anchor images, *m* = 1 is the margin, and *G(X)* is the pairwise distance between the feature extracted by Triplet CNN for the localized player image and the anchor image. If *Y* = 1 it indicates that the anchor image X_1 belongs to the same class as the localized player image, whereas, *Y* = -1 indicates that the anchor image X_2 belongs to the same class as the localized player image.

In Eq. (3), y_i is the ground-truth label, p_i is the output probability score for the respective classes, and *n* is the batch size. In Eq. (4), α_1 and α_2 are constants and are chosen to be 0.5. The advantage of this approach compared to **TI-1** and **TI-2** is that, this approach does not require any template images during inference and gives us the highest accuracy.

C. Identifying the Player Controlling the Ball

To generate player statistics and visual analytics for soccer, we need to identify the player who is in control of the ball at any given point of time. To achieve this, we trained another Triplet CNN with the same architecture used in Section III B.3 (the two CNNs do not share the same weights) to classify a given cropped image of the soccer player as either a "*Player with the ball*" or "*Player without the ball*". The cropped images of the soccer players are resized to size 160×100. We chose this size because the normal aspect ratio of a human body is between 0.6 - 0.7. We chose a mini-batch size of 256 Triplet pairs and during every epoch the training data is randomly shuffled and randomly horizontally flipped. We used a combination of both the Triplet loss and binary cross entropy loss as shown in Eq. (2) - (4) to train the Triplet CNN. Furthermore, we separated a part of our training data as the validation dataset for finding the best training hyper parameters. The validation dataset is used only for finding the best hyper parameters and it is never used for training. More details about the dataset and data partition can be found in the experimental results in Section IV.

We performed random hyper parameter search to obtain the best learning rate, momentum and weight decay. This is done by training and validating the network with random values within a range for each hyper parameter for 5 epochs, and the combination of hyper parameters that resulted in the highest accuracy at the end of 5 epochs were chosen as the best. Based on this we chose the learning rate = 2 $\times 10^{-2}$, momentum = 0.7 and weight decay = 4 $\times 10^{-3}$. Finally, the networks were optimized using the stochastic gradient descent algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.2982580, IEEE Transactions on Circuits and Systems for Video Technology

D. Data Augmentation using Triplet CNN-DCGAN

In this sub-section, we explain on how we performed data augmentation to our dataset. The purpose of data augmentation is to determine if adding more variability to the training dataset helps to improve the generation of tactical statistics. To achieve this we trained the Deep Convolutional Generative Adversarial Network (DCGAN) [41]. It consists of two deep convolutional neural networks, a generator G and a discriminator D trained against each other. The generator takes a randomly sampled Gaussian noise vector, z, and returns an image, $X_{gen} = G(z)$. The discriminator takes a real or a generated image X, and outputs a log probability $P(S|X) = D(X)$ over the two image sources S. The optimization function V is given by:

$$\min_G \max_D V(D,G) = \mathbb{E}_{X\sim p_{data}(X)}[\log D(X)] + \\ +\mathbb{E}_{X\sim p_z(z)}[\log(1 - D(G(z)))] \quad (5)$$

In Eq. (5), $log(D(X))$ is the log probability of the output of the discriminator and D is trained to maximize the probability of assigning the correct label (*i.e.* is the image original or generated) while G is simultaneously trying to minimize it. The significance of Eq. (5) is that by doing a minimax optimization, we are pitting the generator against an adversary that detects if an image is a counterfeit or not. This encourages G to learn the original distribution and generate images that resemble the dataset. The final objective is that the two networks converge to an equilibrium so that D is maximally confused and G generates samples that resemble the training data (in our case "*Players with the ball*").

*1) **Fine-grained Synthetic Image Generation**:* After training the DCGAN, we observed that after the generator and discriminator have reached an equilibrium the generator was able to generate images of the soccer player but, most of these images did not have the soccer ball in it. Generating the soccer ball with the player is the most important feature for distinguishing a "*Player with the ball*" from a "*Player without the ball*". In our novel approach, we solve this problem by introducing a regularizer CNN. The task of the regularizer CNN is to classify the generated images and if the image is classified as a "*Player without the ball*", the generator is penalized more. So now the task of the generator is not only to fool the discriminator but also generate images that resemble a "*Player with the ball*". In our approach we used the Triplet CNN trained to identify the player controlling the ball as the regularizer CNN. Fig. 3 shows the Triplet CNN-DCGAN architecture.

In Fig. 3, it should be noted that the regularizer CNN (i.e., Triplet CNN) is pre-trained to classify between the two classes "*Player with the ball*" and "*Player without the ball*" and its parameters are frozen after pre-training. This means that we do not update the weights of the regularizer CNN while training the generator. In this new architecture, we use the regular DCGAN loss [41] along with the binary cross entropy loss of the Triplet CNN in Eq. (3).

We included the regularizer CNN in the loop only after the generator and discriminator have already reached an equilibrium. The reason for this is that initially while training the DCGAN, the generator does not generate realistic images and if we pass these unrealistic images to the Triplet CNN, it would not be able to recognize the images, resulting in mis-classifications leading to an erroneous back propagation. Hence after the discriminator and generator have reached an equilibrium and the generator is able to generate realistic images of soccer players, only then we include the Triplet CNN in the loop. Fig. 4(a) shows images generated using the regular DCGAN approach [41] and Fig. 4(b) shows images generated using our fine-grained Triplet CNN-DCGAN approach. In Fig. 4(b) the red bounding box indicates the location of the soccer ball in the image.

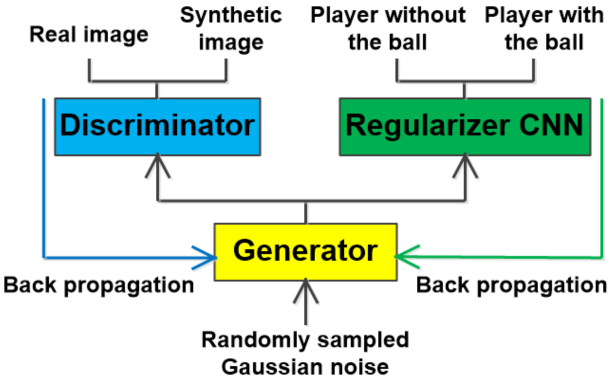

Fig. 3: Architecture of the Triplet CNN-DCGAN.

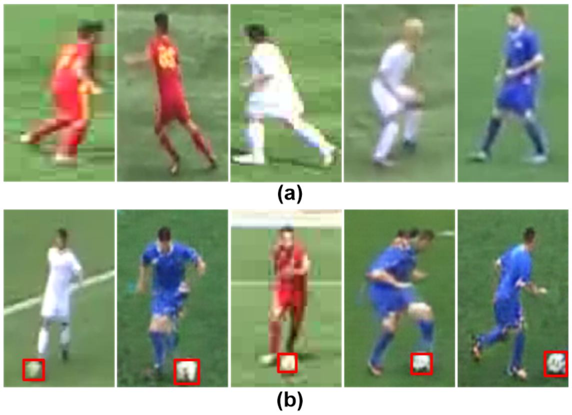

Fig. 4: Generated images of the class "*Player with the ball*" using (a) DCGAN [41] and (b) Triplet CNN-DCGAN.

E. Tactical Statistics Generation

In this sub-section we explain our algorithm for generating the tactical statistics of the soccer players in the video. We pool together the outputs of the player detection module and player classification modules shown in Fig. 1 for the entire video as inputs to our statistics generation module. The outputs of the statistics generation module are the duration of ball possession, number of successful passes and number of successful steals for every player in the video. The pseudo code for generating the tactical statistics is given below. In the pseudo code i is the frame number and j is the number of players detected in frame i. *ID[i][j]* contains the tracking IDs of all j players in frame i. This information is obtained as the output from the player detection module. *x[i]* contains the tracking ID of the player controlling the ball in frame i

and *y[i]* contains the team name of the player controlling the ball in frame *i*. This information is obtained as the output from the player classification module.

Pseudo code for generating the tactical statistics

```
Inputs: 1) List of tracking IDs (ID[i][j]), 2) Player with ball dictionary
(x[i]), 3) Team dictionary (y[i]) where, i is the frame number of the
video and j is the ID of the tracked player in frame i.
Outputs: Ball possession dictionary[], successful passes dictionary[],
and successful steal dictionary[].
Initialize all the key-value pairs for the ball possession, successful
passes and successful steal dictionary to 0
Initialize Players_tracked list to be empty
For i = 1 to N frames in the video
    do
        % This block keeps track of new incoming players
        For j = 1 to length (ID[i])
            if (ID[i][j] does not belong in Players_tracked):
                Append ID[i][j] to Players_tracked
                ball possession dictionary[ID[i][j]] = 0
                successful passes dictionary[ID[i][j]] = 0
                successful steal dictionary[ID[i][j]] = 0
    % This block computes the statistics
    while (i>1 and i<=N):
        % Is the ID of the player controlling the ball in frame i
          and frame i - 1 different?
        if (x[i] is not equal to x[i-1]): % Yes
            % Are the two players in the same team?
            if (y[i] is equal to y[i-1]): % Yes
                % It's a successful pass
                successful passes dictionary[x[i-1]] += 1
                ball possession dictionary[x[i]] += 1
            else: % No, they belong to different teams
                % It's a successful steal
                successful steal dictionary[x[i]] += 1
                ball possession dictionary[x[i]] += 1
        else: % No, it's the same player controlling the ball
          ball possession dictionary[x[i]] += 1
    end
end
```

IV. Experimental Results

We trained and evaluated our approach on a dataset collected from high school soccer matches. The framework of our approach is implemented using 2 TITAN X GPUs.

A. Dataset

We collected a dataset from three different soccer matches. The matches played by the teams were recorded using a single Canon XA10 video camera. The camera was installed at a height of 15 feet and 20 feet away from the horizontal baseline of the soccer field. The resolution of the recorded video is 1280×720. The camera operator was allowed to pan and zoom depending on where the action is happening on the soccer field in order to collect high resolution and good quality images with enough pixels on a player's body. To the best of our knowledge, the only other dataset that has annotations of soccer players in the video, was done by Pettersen *et al.* [47]. The authors used three stationary wide angle cameras installed inside the control room behind the audience in the Alfheim stadium in Norway. Since the control room is very far away from the soccer field (at least 100 feet) the resulting videos have very small number of pixels on the soccer player and even fewer pixels on the soccer ball making it difficult to distinguish the player controlling the ball. Hence we could not use this dataset.

TABLE III: Data distribution for the two classes with respect to the teams

Class	White team	Red team	Blue team
Player with the ball	3,348	2,071	7,400
Player without the ball	10,612	15,319	11,200

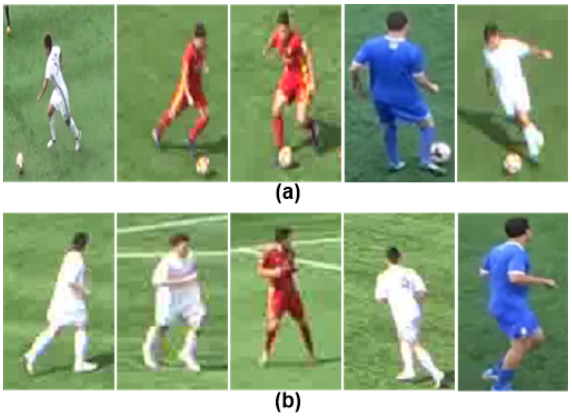

Fig. 5: Examples of players in our dataset for the class: (a) "*Player with the ball*" and (b) "*Player without the ball*".

*1) **Ground-truth for Training Data**:* Our dataset consists of 49,950 images, and it is annotated into two classes namely: "*Players with the ball*" (12,585 images) and "*Players without the ball*" (37,365 images). The dataset was annotated by five experts (initials of the annotators: RT, FP, XZ, YZ, AS) and the final label for a given image is obtained by taking the majority vote of the five annotators. The dataset is comprised of three teams whose jersey colors are white, red and blue. Out of the 49,950 images, the white team constitutes 27.95% of the dataset (13,960 images), the red team constitutes 34.82% (17,390 images) and the blue team constitutes 37.24% of the dataset (18,600 images). Within the two classes, the white, red and blue team constitute 26.12%, 16.16% and 57.73% for the class "*Players with the ball*" and 28.58%, 41.26% and 30.16% for the class "*Players without the ball*", respectively. Table III shows the distribution of the two classes in our dataset, Fig. 5(a) and Fig. 5(b) shows example images of "*Players with the ball*" and "*Players without the ball*" from our dataset, respectively.

It should be noted that in our approach we are not generating tactical statistics for the goal keeper. The reason for this is that a goal keeper is evaluated based on the number of goal shots saved and since we are not generating that statistic we ignore the goal keeper and did not annotate any images of goal keepers in our dataset.

From Table III it can be seen that the dataset is highly unbalanced which makes it challenging. The reason for this is that for every frame of the video only one player can control the ball which leaves 21 other players without the ball. But as the camera is being panned and zoomed not all 22 players are present in a single frame all the time, resulting in 25.66% of the data constituting for the class "*Players with the ball*" and 74.34% of the data constituting for the class "*Players without the ball*".

*2) **Detailed Ground-truth Generation for Testing Videos of Varying Complexities**:* It should be noted that the dataset consisting of the 49,950 images does not have any annotations of player's positions in the video or their tactical statistics. Hence, in order to evaluate the performance of the player detection, tracking and tactical statistics generation module, we annotated seven highlight test videos for *evaluation/testing only* (initials of the annotators: FP, RT). Six of these highlight videos were extracted from matches played between the *Red jersey* Vs. *White jersey* categorized into three different complexities namely: ***Low*** (2 video clips), ***Moderate*** (2 video clips) and ***Severe*** (2 video clip). The duration of these six videos ranges from approximately 90 to 180 seconds. Since our approach uses only a single uncalibrated camera, we selected these six highlight videos such that the camera was not moving (pan, tilt, or zoom) faster than the players on the field. We also ensured that the FoVs of the camera in the highlight videos were large enough such that we do not have players entering/exiting the FoV of the camera for a long duration and then reappearing elsewhere in the video. The 7th highlight video is publicly available on the internet and the match was played between a *white* and *blue* jersey team. The duration of this video is 31 seconds and it contains small segments of the *Low*, *Moderate*, and *Severe* complexity. This video contains segments where the players are entering/exiting the FoV of the camera and the tracking algorithms cannot associate them if the duration of entering/exiting is longer than 5 seconds. In order to evaluate the tracking algorithms in a fair manner, we consider the players who exit/enter after 5 seconds in the video to be new players. It should also be noted that these highlight videos were not used for training the CNNs.

We have carefully chosen these seven videos such that each video shows some level of complexity during different segments of the match. We chose the videos in the *Low* complexity when there are 4 - 5 players of the same team passing the ball among themselves and they are widely spread out. We can see this kind of play in a match when a team is trying to stall the opposing team. It is relatively easy to predict the statistics in this *Low* complexity case as compared to the *Moderate* and *Severe* complexity cases. In the *Moderate* complexity case we chose the videos where the mid-fielders are trying to penetrate the oppositions defense by passing the ball between their team mates while the opposition mid-fielders are trying to steal the ball. This scenario causes a lot of occlusion as there are usually 6 - 10 players in a small area as compared to the *Low* complexity case that makes it even more challenging to generate statistics. In the *Severe* complexity case, the offensive players try to aim for a shot at the goal while trying to avoid the opposition defenders and mid-fielders simultaneously. This scenario involves significant occlusion because usually there are several offensive players from the same team who pass the ball among themselves while there are at least 2 defenders and more than 2 mid-fielders from the opposition trying to steal the ball within a very narrow area of the field. This scenario causes the most occlusion leading to a significant drop in performance of generated statistics (see Table XIII). In summary, these seven videos are representative and the results from our approach demonstrate how the overall system will perform under different segments of a match.

The ball possession for these highlight videos was annotated by identifying the player controlling the ball in all of the frames of all videos. The number of passes and steals made by each player in a video were annotated by observing the video and identifying the tracking ID of the players and the frame numbers of the video when the ball was passed/stolen.

B. Results for the Player Detection Module

In this sub-section we evaluate the performance of the player detection and tracking algorithms using the seven highlight test videos described above.

1) ***Localization Results and their Comparisons****:* We evaluated four state-of-the-art object detection algorithms namely: YOLOv2 [11], Single Shot Detection (SSD) [15], OpenPose [40], and Mask R-CNN [39] for detecting the soccer players in video. We used Intersection Over Union (IOU) between the ground truth and predicted bounding box and the speed of performance during inference in Frames Per Second (FPS) as metrics. Since we do not have any annotated training data with the soccer players position we trained all the above localization approaches on the COCO dataset [12] and then evaluated them on our highlight test videos. We chose the COCO dataset because it has a lot of diverse images for the class "*Person*". Table IV shows the detection results on the seven highlight test videos.

TABLE IV: Performance metrics of different approaches for detecting the soccer players.

Approach	Average IOU (%)	Average Speed (FPS)
YOLOv2 [11]	84.81 ± 2.74	**17.2 ± 0.3**
SSD [15]	74.30 ± 2.03	15.8 ± 0.2
OpenPose [40]	69.45 ± 3.74	6.4 ± 0.2
Mask R-CNN [39]	**86.68 ± 2.58**	1.7 ± 0.4

All the approaches in Table IV were evaluated using two TITAN X GPUs. From Table IV it can be observed that YOLOv2 [11] had the highest average speed of performance with 17.2 FPS and average IOU accuracy of 84.68% while OpenPose had the least average IOU accuracy of 69.53%. SSD had a similar speed of performance compared to YOLOv2 but fell short in its average IOU accuracy. Mask R-CNN [39] had the highest average IOU of 86.47%, but had the least speed of performance of only 1.7 FPS making it impractical to use in our approach compared to YOLOv2.

2) ***Tracking Results and their Comparisons****:* We evaluated three deep learning based [42] - [44] and two handcrafted features based [21], [22] long-term tracking algorithms. Table V shows the average Multi Object Tracking Accuracy (MOTA), Mostly Tracked (MT), and processing speed of each algorithm evaluated on our 7 highlight videos. MOTA is the accuracy of assigning the correct tracking ID to a player during all the frames the player is detected. MT is the accuracy of assigning the correct tracking ID to a player for at least 70% of their tracking duration. In Table V since we are only comparing the performance of tracking algorithms, for fair comparison we replaced the object detection in [21] and [22] with YOLOv2 [11] and used [42] and [43] as proposed in their paper.

TABLE V: Performance comparison of tracking algorithms

Approach	Average MOTA* (%)	Average MT* (%)	Average speed (FPS)
Feichtenhofer *et al.* [42]	72.60 ± 7.37	60.88 ± 3.51	14.1 ± 0.4
Bertinetto *et al.* [43]	63.87 ± 6.55	49.02 ± 4.34	16.3 ± 0.3
D'Orazio *et al.* [21]	29.78 ± 3.22	15.63 ± 4.22	**20.6 ± 0.4**
Khatoonabadi and Rahmati [22]	23.35 ± 2.13	10.89 ± 2.27	20.4 ± 0.4
DeepSORT [44]	**76.59 ± 6.32**	**63.57 ± 3.85**	17.2 ± 0.4

*MOTA and MT refer to Multi Object Tracking Accuracy and Mostly Tracked, respectively.

From Table V it can be observed that DeepSORT had the best MOTA and MT with a speed of 17.2 FPS. The hand-crafted approaches of [21] and [22] had the worst MOTA and MT but had the best processing speed. Most of the errors for the 5 different approaches occurred in *Severe* complexity cases when multiple players overlap with each other, causing the detector to detect them as a single player. This kind of situation arises when a player approaches close to the opposition team aiming for a shot at the goal.

We currently do not account for any camera calibration and the players are tracked based on the Euclidean coordinates of the video frame and not the actual coordinates of the soccer field. As a result, when the camera changes its focus from one part of the field to another, some players do not appear in the video for a while and when they re-appear, they are detected and tracked as new players. It is very challenging to track re-appearing players if the duration between disappearing and re-appearing is very large. Possible solutions to eliminate this problem are:

Player re-identification: 1) We can a use a player re-identification framework to associate the players when they re-appear, but the challenge with this is that, the players are wearing the jersey of the same color which makes it difficult to re-identify them under all situations. 2) We can associate the soccer players by recognizing their jersey numbers as proposed by Liu *et al.* [26], but this is possible only when the soccer player is displaying the jersey number to the camera.

Additional hardware: 1) We can use multiple static cameras on opposite sides of the soccer field such that the collective FoV of the cameras spans the entire soccer field [62]. 2) We can use unique GPS trackers attached to the jersey of the soccer players along with camera calibration to get the physical location of the players on the soccer field.

C. Results for the Player Classification Module

*1) **Team Classification Results and their Comparisons**:* In this sub-section we evaluate and compare the performance of our three different team identification algorithms **(*TI-1, TI-2, TI-3*)** as described in Section III B. For the *TI-1* approach, we trained the Triplet CNN on the Town Center subset of the PETA dataset [48] and then evaluated the CNN on all of the 49,950 images from our dataset. For the *TI-2* and *TI-3* approaches, we randomly split our dataset consisting of 49,950 images evenly based on the number of images in our dataset for each team into 65% for training, 10% for validation and 25% for testing. The validation dataset was selected randomly and fixed for all of the experiments for team identification. We used the validation dataset only for finding the best hyper parameters. Table VI shows the distribution of the training, validation and testing datasets and Table VII shows the results and comparison of the four-fold cross validation for team identification.

TABLE VI: Data distribution for training, validation and testing datasets for team identification.

Team	Training	Validation	Testing
White Team	9,075	1,395	3,490
Red Team	11,303	1,739	4,348
Blue Team	12,090	1,860	4,650
Total Images	32,468	4,994	12,488

TABLE VII: Results and comparison for team identification.

Approach	Average Accuracy (%)	Requires template?
Theagarajan *et al.* [30]	71.28	yes
TI-1*	83.56	yes
TI-2*	93.17 ± 1.07	yes
Otsu [53]	64.07 ± 5.37	no
ResNet18 [54]	82.55 ± 1.56	no
ResNet34 [54]	89.40 ± 1.62	no
ResNet50 [54]	93.58 ± 1.40	no
AlexNet [56]	85.73 ± 1.44	no
VGG-16 [55]	82.07 ± 1.31	no
TI-3*	**97.46 ± 1.19**	no

*TI stands for Team Identification

In order to evaluate the approach of Theagarajan *et al.* [30], we randomly selected 10 template images from the dataset for each team and evaluated the performance on all of the remaining images from our dataset. We can observe from Table VII that our approach (*TI-3*) outperforms all of the state-of-the-art methods. Otsu's method [53] had the least accuracy followed by Theagarajan *et al.* [30] because even within the same match if the pose of the detected player is in the profile view and the templates consists of images that are frontal/back view of the player, the histograms will look very different and it is not realistic to collect new templates dynamically during a match.

From Table VII in the *TI-1* setting, we trained the Siamese CNN on the Town Center subset of the PETA dataset [48] for the task of pedestrian re-identification. After training the CNN, we evaluated it on our dataset by extracting features of the detected player and matching it with the features for the template images. This approach achieved an accuracy of 83.56% and did not require any knowledge of the soccer match making it generalizable to other soccer matches.

The *TI-2* setting is similar to the *TI-1* setting except that after pre-training on the subset of the PETA dataset [48], we further fine tune the Siamese CNN with our soccer dataset which helps to improve the accuracy to 93.17%.

In the *TI-3* setting, we train the Triplet CNN using both the Triplet loss and the binary cross entropy loss. The difference between *TI-1*, *TI-2* and *TI-3* is that *TI-3* does not require any template images for matching the features and it provides the highest accuracy of 97.46%. In *TI-3* we need to train the Triplet CNN for every single match, making it less generalizable compared to *TI-1*. Although the *TI-3* setting is less generalizable compared to the *TI-1* setting, in applications such as sports analytics, coaches and fans of the sport prefer to have a system that provides the highest accuracy compared to having a generalizable system that provides a significantly lower accuracy.

2) ***Results for Identifying the Player with the Ball and Comparison of Algorithms****:* In this sub-section we evaluate and compare the performance of our Triplet CNN for identifying the player controlling the ball using the prediction accuracy and speed of performance during inference as the performance metrics. For this purpose, we split the dataset evenly based on the number of images for the two classes from our dataset into 65% for training, 10% for validation and 25% for testing. Table VIII shows the data distribution for the training, validation and testing datasets and Table IX shows the results and comparison of the four-fold cross validation for identifying the "*Player with the ball*". Similar to Table VI, the validation dataset was used only for finding the best hyper parameters and was never used for training.

TABLE VIII: Data distribution for training, validation and testing for identifying the player controlling the ball.

Class	Training	Validation	Testing
Player with the ball	8,333	1,281	3,205
Player without the ball	24,135	3,713	9,283
Total Images	32,468	4,994	12,488

TABLE IX: Results of the four-fold cross validation for identifying the player controlling the ball.

Network	Average accuracy (%)	Average speed (FPS)	Number of parameters
ResNet18 [54]	81.70 ± 3.21	8.2 ± 0.7	111.7m
ResNet34 [54]	81.18 ± 3.58	6.3 ± 0.7	212.8m
ResNet50 [54]	82.51 ± 4.76	5.9 ± 0.6	235.1m
AlexNet [56]	79.51 ± 2.07	13.4 ± 0.8	61.1m
VGG-16 [55]	81.29 ± 3.22	7.9 ± 0.6	134.2m
Theagarajan *et al.* [30]	83.47 ± 4.02	7.5 ± 0.7	139.5m
This paper	**90.66 ± 2.46**	**26.7 ± 1.2**	**3.7m**

From Table IX it can be observed that our approach had the highest accuracy and speed of performance compared to the state-of-the-art. Moreover, our CNN performs at least 2x faster with less than 16x the number of parameters than the state-of-the-art. The reason for this is that most of the state-of-the-art CNNs are built for more generalized tasks such as classifying the ImageNet dataset [57] which has more than 1,000 classes requiring more number of parameters and computation time. On the other hand, our approach for generating statistics of soccer players is for a specific and time-critical task which requires a Triplet CNN to extract fine-grained features for achieving higher accuracy with less number of parameters. Moreover, as shown in Theagarajan *et al.* [30] regular CNNs have trouble in detecting minute details such as the soccer ball in low resolution images which is the only feature that distinguishes between a player with and without the ball. By using a Triplet CNN, the CNN is able to learn fine-grained features that help in distinguishing a "*Player with the ball*" and further improves the accuracy.

D. Generalization Across Different Matches

1) ***Results on Generalizability and Comparison with other Algorithms****:* In this sub-section we evaluate and compare our approach for its generalizability across different matches. Generalizability is a very important metric for determining a classifier's robustness. Many studies have shown that a classifier that is generalizable across multiple domains does not necessarily have the best performance on all of the domains, similarly a classifier that has the best performance in one domain does not necessarily generalize across multiple domains [58], [59].

To evaluate the generalizability we trained the CNNs on all the training images (32,468 images) in our dataset as shown in Table VIII and evaluated them on soccer matches played by different teams that were never included in our dataset. We selected four high school soccer matches where two matches were played by *Pink jersey Vs Black jersey* and two matches were played by *Green jersey Vs Black jersey*. In order to validate the performance, we annotated 100 images per team per match for the two classes "*Player with the ball*" and "*Player without the ball*", resulting in a total of 800 images. Table X shows the results and comparison of different CNNs for their generalizability across different soccer matches.

TABLE X: Results on the generalizability across different matches for identifying the player controlling the ball.

Network	Accuracy (%)	
	Pink Vs Black	***Green Vs Black***
ResNet18 [54]	59.31	54.22
ResNet34 [54]	57.60	61.23
ResNet50 [54]	60.44	60.76
AlexNet [56]	55.81	58.29
VGG-16 [55]	**63.36**	61.29
Theagarajan ***et al.*** [30]	58.35	61.07
This paper	60.26	**62.28**

From Table X it can be seen that all the CNNs fell short in their performance compared to Table IX. This indicates that the features learned from one match do not necessarily transfer over to another match played by two different teams. This is similar to the findings reported by the authors of [30], [35] and [60]. Moreover, it is not feasible to collect data that resembles all the different conditions to train the network, hence, it is more appropriate to re-train the CNNs for every match with as minimal annotation as possible.

E. Match Specific Annotation for Robust Performance

In this sub-section we use a minimum number of images annotated for specific matches in varying proportions to fine tune the different CNNs and observe the performance. In order to validate the match specific performance, a problem that arises is that how do we annotate images for a match that has not yet been played? To solve this we annotate images of matches previously played by the same teams for training our models and evaluate it on the match that is to be played.

As mentioned in the previous sub-section, we annotated 100 images per team per class (*Player with/without the ball*) from four different matches, where two matches were played by *Pink jersey Vs Black jersey* and two matches were played by *Green jersey Vs Black jersey*. We took the two matches played by the same teams and used one match for training and the other match for testing. Based on this we perform two-fold cross validation. Table XI and XII shows the match specific performance for the different CNNs.

TABLE XI: Match specific performance of different CNNs for the game played between *Pink jersey Vs. Black jersey*.

Network	**Average accuracy (%)**		
	50 images per class	**75 images per class**	**100 images per class**
ResNet18 [54]	64.32 ± 2.56	64.88 ± 3.13	65.72 ± 2.01
ResNet34 [54]	65.60 ± 3.97	65.89 ± 3.42	67.22 ± 3.19
ResNet50 [54]	68.23 ± 4.15	68.42 ± 2.91	69.01 ± 2.88
AlexNet [56]	61.98 ± 4.23	62.34 ± 3.85	64.56 ± 3.76
VGG-16 [55]	66.39 ± 2.54	68.25 ± 4.31	69.14 ± 3.67
Theagarajan *et al.* [30]	65.78 ± 3.58	67.41 ± 3.19	70.22 ± 4.29
This paper	**84.05 ± 2.56**	**84.73 ± 3.02**	**84.81 ± 2.71**

In columns 2 and 3, we randomly selected 50 and 75 images, respectively, for training.

TABLE XII: Match specific performance of different CNNs for the game played between *Green jersey Vs. Black jersey*.

Network	**Average accuracy (%)**		
	50 images per class	**75 images per class**	**100 images per class**
ResNet18 [54]	63.02 ± 3.01	64.51 ± 2.78	64.43 ± 3.44
ResNet34 [54]	65.13 ± 3.44	66.58 ± 4.28	66.77 ± 2.95
ResNet50 [54]	66.08 ± 3.37	66.70 ± 3.56	68.35 ± 3.51
AlexNet [56]	63.28 ± 2.80	63.97 ± 3.89	64.89 ± 3.25
VGG-16 [55]	65.41 ± 4.05	66.98 ± 2.19	70.36 ± 2.71
Theagarajan *et al.* [30]	67.88 ± 4.69	67.13 ± 4.11	69.24 ± 3.53
This paper	**82.49 ± 3.76**	**84.17 ± 3.04**	**84.68 ± 3.32**

In columns 2 and 3, we randomly selected 50 and 75 images, respectively, for training.

From Table XI and XII it can be seen that, as we fine tune the CNNs on images for a specific match, we can observe an increase in performance. Comparing Table X with Table XI and XII we can observe an increase in performance across all CNNs, but our approach significantly outperforms the state-of-the-art CNNs. One possible reason for this is that although the training dataset consists of only 100 images per class (*Player with/without the ball*), we can create more than 75,000 Triplet pairs and train the Triplet CNN to learn fine-grained features by increasing the inter-class variance and decreasing the intra-class variance. Furthermore, this finding is consistent with the works reported by the authors of [49] - [52], wherein Triplet networks are able to outperform regular CNNs in the presence of very limited data.

F. Ablation Study for Generating the Tactical Statistics

1) ***Generating Match Level Tactical Statistics****:* In this sub-section we perform an ablation study to observe how using different combinations of player detectors and classifiers for identifying the team and player controlling the ball affects the generation of match level statistics. We do not need any tracking algorithm for generating match level statistics, hence we use only the outputs of the player detector and classifier for identifying the team and player controlling the ball. Table XIII shows the performance and comparison of our approach with the state-of-the-art approaches for generating match level statistics with and without data augmentation. Additionally, there is no other work that can directly provide the tactical statistics for the number of passes and steals from a video, hence, we cannot compare the state-of-the-art approaches in Table XIII with our work for these statistics. The accuracy of the ball possession is calculated by identifying the correct player controlling the ball in all of the frames in the videos. Based on this we can observe from Table XIII that using our approach for classification outperforms all other approaches for computing the match level ball possession accuracy. Additionally, it is observed that using Mask R-CNN [39] as the detector slightly improves the accuracy for ball possession compared to using YOLOv2 [11].

- ***Effect of Data Augmentation***: From, the data distribution shown in Table III, we can observe that the class "*Player without the ball*" has 3x more training data than the class "*Player with the ball*". In order to observe the effect of data augmentation for generating the tactical statistics, we generated and augmented 20,000 synthetic images for the class "*Player with the ball*" to our dataset using our Triplet CNN-DCGAN approach explained in Section III D.1. Next, we trained all the classifier approaches in Table XIII using the augmented dataset and compared their results without any data augmentation. Based on this we can observe that performing data augmentation helped improve the performance of all approaches in Table XIII. We were able to improve the performance for our approach by 2.59% and 4.43% using YOLOv2 [11] and Mask R-CNN [39], respectively.

Our approach was able to successfully detect 7/8 passes and 2/3 steals in the *Moderate* complexity and 3/5 passes and 1/3 steals in the *Severe* complexity. There is a drop in performance in the *Severe* complexity because the players are too close to each other and since we are using only one camera, it causes a lot of occlusions. Hence, it is difficult for the network to identify which player is controlling the ball leading to a drop in performance.

2) ***Generating Individual Level Tactical Statistics****:* In this sub-section we perform an ablation study to observe the performance in generating individual player level tactical statistics using different combinations of player detector and tracking algorithms and fixing our approach for predicting the team and player controlling the ball. For this purpose, we selected a 45 second clip from a video (recorded at 30 FPS) of moderate complexity where *Players ID # 6* and *10* belonging to the *white* team were passing the ball between them while *Player ID # 13* belonging to the *red* team was trying to steal the ball. Towards the end of the video *Player ID # 13* successfully stole the ball from *Player ID # 10*. Table XIV shows the performance and comparison of our approach with the state-of-the-art approaches for generating individual player statistics. In Table XIV, the duration of ball possession is shown in frames, this can be converted into time by dividing it by the frame rate of the video.

From Table XIV, we can observe that using DeepSORT achieves better performance in generating the statistics with

TABLE XIII: Ablation study for comparing the performance of our system for generating the tactical statistics at a match level on the moderate and severe complexity highlight videos. Acc. is Accuracy.

		Moderate complexity			Severe complexity		
Detector	**Classifier**	**Avg. ball possession acc. (w/ DA)/w/o DA(%)***	**No. of passes**	**No. of steals**	**Avg. ball possession acc. (w/ DA)/w/o DA(%)***	**No. of passes**	**No. of steals**
YOLOv2	**ResNet50** [54]	(73.56 ± 3.19)/72.17 ± 2.44	-	-	(68.24 ± 5.11)/65.57 ± 4.74	-	-
	AlexNet [56]	(71.92 ± 3.87)/70.46 ± 4.47	-	-	(63.18 ± 3.79)/61.53 ± 4.21		
	Theagarajan *et al.* [30]	(74.82 ± 5.02)/71.17 ± 4.65	-	-	(64.28 ± 4.81)/63.17 ± 4.42	-	-
	Our Approach	**(84.19 ± 2.38)/81.83 ± 2.56**	**7/8**	**2/3**	**(76.65 ± 3.41)/74.06 ± 2.72**	**3/5**	**1/3**
Mask R-CNN	**ResNet50** [54]	(73.78 ± 4.24)/72.55 ± 3.13	-	-	(69.83 ± 4.76)/66.02 ± 4.31	-	-
	AlexNet [56]	(72.49 ± 3.17)/70.36 ± 3.20	-	-	(64.02 ± 4.16)/63.20 ± 3.77	-	-
	Theagarajan *et al.* [30]	(73.38 ± 3.86)/72.64 ± 3.41	-	-	(67.35 ± 4.53)/65.18 ± 4.39	-	-
	Our Approach	**(83.27 ± 2.56)/82.34 ± 2.15**	**7/8**	**2/3**	**(76.78 ± 3.82)/72.35 ± 2.56**	**3/5**	**1/3**

*w/ DA and w/o DA refer to with Data Augmentation and without Data Augmentation, respectively.

TABLE XIV: Ablation study for comparing the performance of our system for generating the tactical statistics on an individual level for a 45 second clip from a video of moderate complexity. The Triplet CNNs proposed in this paper are used for identifying the team and the "*Player with the ball*".

			Player ID #6; Team: White		Player ID #10; Team: White		Player ID #13; Team: Red	
Detector	**Tracker**	**Operating speed (FPS)**	**Duration of possession***	**No. of pass/ (No. of steal)***	**Duration of possession***	**No. of pass/ (No. of steal)***	**Duration of possession***	**No. of pass/ (No. of steal)***
YOLOv2	**DeepSORT** [44]	16.9	**498/533**	**2/2/(0/0)**	**314/371**	**1/1/(0/0)**	291/337	**0/0/(1/1)**
	D'Orazio *et al.* [21]	20.6	162/533	0/0/(0/0)	107/371	0/0/(0/0)	82/337	0/0/(0/0)
Mask R-CNN	**DeepSORT** [44]	1.6	**498/533**	**2/2/(0/0)**	**314/371**	**1/1/(0/0)**	**294/337**	**0/0/(1/1)**
	D'Orazio *et al.* [21]	1.6	162/533	0/0/(0/0)	106/371	0/0/(0/0)	85/337	0/0/(0/0)
	Feichtenhofer *et al.* [42]	14.4	483/533	1/2/(1/0)	**314/371**	**1/1(0/0)**	274/337	**0/0/(1/1)**

*The cells in Duration of possession are formatted as *x*/*y* where *x* is the total number of frames the player was predicted to control the ball and *y* is the ground truth total number of frames the player was controlling the ball. The cells in the No. of pass/(No. of steal) are formatted as *p*/*q*/(*r*/*s*) where *p* and *q* are the number of predicted and ground-truth passes, and *r* and *s* are the number of predicted and ground-truth steals, respectively.

the highest processing speed using two TITAN X GPUs. Although, Mask R-CNN [39] outperformed YOLOv2 [11] by 3 frames in predicting the duration of ball possession for *Player ID #13*, there is no other significant change. On the contrary using YOLOv2 had the highest processing speed of 16.9 FPS which is a 10x improvement compared to Mask R-CNN. This is significant because, although our approach is offline it is unreasonable for a user to wait 10x longer to analyze a video using Mask R-CNN compared to using YOLOv2 for a very small trade off in accuracy.

It can also be observed that the algorithm proposed by Feichtenhofer *et al.* [42] did not detect a pass of *Player ID # 6* and also had a false negative in predicting the steals of *Player ID #10*. The reason for this is that the algorithm had an identity flip for that player during which the pass was made leading to incorrect stats.

G. Discussion of Results and Application to Internet of Things (IOT) Environment

1) ***Discussion of results****:* In this sub-section we analyze the results and provide high level conclusions of the individual modules for generating the tactical statistics.

• ***Player detection and tracking***: In our approach we evaluated various player detection and tracking algorithms and found the best combination for detecting and tracking players are YOLOv2 [11] and DeepSORT [44], respectively. Although the Mask R-CNN [39] approach was able to slightly outperform YOLOv2 in terms of IOU, YOLOv2 has a 10x improvement in processing speed. This is a very important trade-off in terms of processing speed. In terms of tracking we observed that deep learning based approaches proposed by [42]-[44] outperform some of the hand-crafted approaches described in [45]. The reason for this is that the approaches proposed in [45] are not very generalizable across different matches and do not handle player occlusions well.

• ***Team identification***: We proposed three different team identification algorithms (*TI-1*, *TI-2*, and *TI-3*) and found that *TI-3* outperformed all state-of-the-art approaches as shown in Table VII. A drawback of *TI-3* is that we require an annotated dataset for training the CNN making it less generalizable. A solution for this problem is that since team jerseys do not often change, we can choose a match that was played in the past by the same teams and annotate those images for training the CNN. In cases where datasets are not available we can still use *TI-1* which is the most generalizable approach for a slight trade off in performance.

• ***Identifying the player controlling the ball***: We proposed to use a Triplet CNN for identifying the player controlling the ball throughout all the frames in a video. Prior work done by [30] showed that regular CNNs often overlook minute details such as soccer balls which is the most important feature for identifying the player controlling the ball. We empirically showed that by training Triplet CNNs to extract fine-grained features our approach outperforms the state-of-the-art classifiers for this task. A general drawback of all the approaches shown in Table IX is that, they do not generalize to matches beyond the dataset. Our approach solves this problem by requiring only 100 annotated images per class (*Player with/without the ball*) per match in order to achieve a reasonable performance and it outperforms the other approaches shown in Tables XI and XII.

*2) **Application to IOT**:* Internet of Things (IOT) is an environment where individual devices sense and collect data which is shared through the internet where the data can be processed and interpreted in real time. This technology has been widely used in areas such remote monitoring, healthcare and recently in sports [61]. In our case the proposed approach can be integrated into a multi-camera system in order to generate more robust statistics and usually this would create a bottleneck problem in terms of processing speed. This kind of problem can be solved by moving heavy computations onto a cloud based IOT-environment as shown in [61]. Additionally, in order to make the tracking more robust, we can attach cheap GPS tracking sensors on the jerseys of the players which transmit the data to a cloud server where all of the data are being collectively processed in real-time.

V. Conclusions

We proposed and designed a system for analyzing the performance of soccer players and generating three tactical statistics of each player (except goal keeper) from a video. We collected a dataset consisting of 49,950 images from high school soccer matches and performed exhaustive evaluation and comparison of algorithms on the dataset and our approach achieved the best performance in terms of accuracy and computation time. Moreover, we observed that although our approach achieves the best performance on matches played between teams in our training dataset, the features learned do no generalize well across matches played by teams that are not in our dataset. To solve this we employed a minimum amount of match specific annotations using a novel Triplet CNN-DCGAN architecture and showed that by fine tuning the network with only 100 annotated images per class (*Player with/without the ball*) we can obtain robust performance. Finally, we performed an ablation study that showed how individual modules of our proposed approach and data augmentation affect the generation of tactical statistics at a match level and individual player level. The Future work will include using multiple wide lens stationary cameras and GPS trackers in an IOT based cloud environment which will provide real-time performance. Additionally we will integrate more actions in our system such as shots on the goal, dribbling detection and player style classification [38] which will be used for generating a more comprehensive performance characterization of an individual soccer player.

Acknowledgment

We would like to thank Ashwini Shandilya and Eric Ebert from SEVAai Inc. and Don Ebert for providing us with the dataset.

References

[1] https://www.ranker.com/crowdranked-list/most-popular-american-sports

[2] https://www.statista.com/statistics/267963/participation-in-us-high-school-soccer/

[3] https://www.footballscience.net/special-topics/performance-analysis/

[4] C. Lago-Peñas, and A. Dellal, "Ball possession strategies in elite soccer according to the evolution of the match-score: The influence of situational variables," *Journal of Human Kinetics*, 25, pp. 93-100, 2010.

[5] K. Saito, and M. Yoshimura, "Pass appearance time and pass attempts by teams qualifying for the second stage of FIFA world cup 2014 in Brazil," *Journal of Sports Science*, 4, pp. 156-162, 2016.

[6] K. Saito, M. Yoshimura, and T. Ogiwara, "Pass appearance time and pass attempts by teams qualifying for the second stage of FIFA World Cup 2010 in South Africa," *Football Science*, 10, pp. 65-69, 2013.

[7] A. Janković, B. Leontijević, M. Pašić, and V. Jelušić, "Influence of certain tactical attacking patterns on the result achieved by the teams participants of the 2010 FIFA World Cup in South Africa," *Physical Culture*, 65(1), pp. 34-45, 2011.

[8] A. Redwood-Brown, "Passing patterns before and after goal scoring in FA Premier League Soccer," *International Journal of Performance Analysis in Sport*, 8(3), pp. 172-182, 2008.

[9] M. A. Gómez, M. Gómez-Lopez, C. Lago, and J. Sampaio, "Effects of game location and final outcome on game-related statistics in each zone of the pitch in professional football," *European Journal of Sport Science*, 12(5), pp. 393-398, 2012.

[10] A. Scoulding, N. James, and J. Taylor, "Passing in the Soccer World Cup 2002," *International Journal of Performance Analysis in Sport*, 4(2), pp. 36-41, 2004.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[12] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *European Conference on Computer Vision*, pp. 740-755, 2014.

[13] Z. Cai, and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154-6162, 2018.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision*, pp. 2961-2969, 2017.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *European Conference on Computer Vision*, pp. 21-37, 2016.

[16] P. Parisot, and C. De Vleeschouwer, "Scene-specific classifier for effective and efficient team sport players detection from a single calibrated camera," *Computer Vision and Image Understanding*, pp. 74-88, 2017.

[17] D. J. Duh, S. Y. Chang, S. Y. Chen, and C. C. Kan, "Automatic broadcast soccer video analysis, player detection, and tracking based on color histogram," *Intelligent Technologies and Engineering Systems*, pp. 123-130, 2013.

[18] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, "Automatic player detection, labeling and tracking in broadcast soccer video,"*Pattern Recognition Letters*, 30(2), pp. 103-113, 2009.

[19] T. K. Chiang, J. J. Leou, and C. S. Lin, "An improved mean shift algorithm based tracking system for soccer game analysis," *Asia-Pacific Signal and Information Processing Association*, pp. 380-385, 2009.

[20] J. Xing, H. Ai, L. Liu, and S. Lao, "Multiple player tracking in sports video: A dual-mode two-way Bayesian inference approach with progressive observation modeling," *IEEE Transactions on Image Processing*, 20(6), pp. 1652-1667, 2010.

[21] T. D'Orazio, M. Leo, P. Spagnolo, P. L. Mazzeo, N. Mosca, M. Nitti, and A. Distanter, "An investigation into the feasibility of real-time soccer offside detection from a multiple camera system," *IEEE Transactions on Circuits and Systems for Video Technology*, 19(12), pp. 1804-1818, 2009.

[22] S. H. Khatoonabadi, and M. Rahmati, "Automatic soccer players tracking in goal scenes by camera motion elimination", *Image and Vision Computing*, 27(4), pp. 469-479, 2009.

[23] A. Senocak, T. H. Oh, J. Kim, and I. So Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling,' *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1732-1739, 2018.

[24] T. Jaakkola, and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in Neural Information Processing Systems*, pp. 487-493, 1999.

[25] J. Xu, L. Kanokphan, and K. Tasaka, "Fast and accurate object detection using image cropping/resizing in multi-view 4K sports videos," *ACM International Workshop on Multimedia Content Analysis in Sports*, pp. 97-103, 2018.

[26] H. Liu, and B. Bhanu, "Pose-guided R-CNN for jersey number recognition in sports," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[27] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015.

[28] M. Istasse, J. Moreau, and C. De Vleeschouwer, “Associative embedding for team discrimination,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[29] W. L. Lu, J. A. Ting, K. P. Murphy, and J. J. Little,“Identifying players in broadcast sports videos using conditional random fields,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3248 - 3256, 2011.

[30] R. Theagarajan, F. Pala, X. Zhang, and B. Bhanu, “Soccer: Who has the ball? Generating visual analytics and player statistics,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1749-1757, 2018.

[31] Z. Cai, H. Neher, K. Vats, D. Clausi, and J. Zelek, “Temporal hockey action recognition via pose and optical flows,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[32] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291-7299, 2017.

[33] T. W. Hui, X. Tang, and C. Change Loy, “Liteflownet: A lightweight convolutional neural network for optical flow estimation,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[34] A. J. Piergiovanni, and M. S. Ryoo, 2018. “Fine-grained activity recognition in baseball videos,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1740-1748, 2018.

[35] A. Cioppa, A. Deliège, and M. Van Droogenbroeck, “A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1765-1774, 2018.

[36] M. R. Tora, J. Chen, and J. J. Little, “Classification of puck possession events in ice hockey,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 147-154, 2017.

[37] B. Fakhar, H. R. Kanan, and A. Behrad, “Event detection in soccer videos using unsupervised learning of Spatio-temporal features based on pooled spatial pyramid model,” *Multimedia Tools and Applications*, 78(12), pp. 16995 - 17025, 2019.

[38] R. Li, and B. Bhanu, “Fine-grained visual dribbling style analysis for soccer videos with augmented dribble energy image,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE International Conference on Computer Vision*, pp. 2961-2969, 2017.

[40] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields,” *arXiv preprint arXiv:1812.08008*, 2018.

[41] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.

[42] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Detect to track and track to detect,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3038-3046, 2017.

[43] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional Siamese networks for object tracking,” *European Conference on Computer Vision*, pp. 850-865, 2016.

[44] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” *IEEE International Conference on Image Processing*, pp. 3645-3649, 2017.

[45] M. Manafifard, H. Ebadi, and H. A. Moghaddam, “Survey on player tracking in soccer videos”, *Computer Vision and Image Understanding*, 159, pp. 19-46, 2017.

[46] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval research logistics quarterly*, 2(1-2), pp. 83-97, 1955.

[47] S. A. Pettersen, D. Johansen, H. Johansen, V. Berg-Johansen, V. R. Gaddam, A. Mortensen, R. Langseth, C. Griwodz, H. K. Stensland, and P. Halvorsen, “Soccer video and player position dataset,” *International Conference on Multimedia Systems*, pp. 18-23, 2014.

[48] Y. Deng, P. Luo, C. C. Loy, and X. Tang, “Pedestrian attribute recognition at far distance,” *ACM International Conference on Multimedia*, pp. 789-792, 2014.

[49] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li, “Breast cancer multi-classification from histopathological images with structured deep learning model,” *Scientific Reports*, 7(1), 2017.

[50] R. Theagarajan, and B. Bhanu, “DeephESC 2.0: Deep generative multi adversarial networks for improving the classification of hESC,” *PloS one*, 14(3), 2019.

[51] X. Wu, R. He, Z. Sun, and T. Tan, “A light CNN for deep face representation with noisy labels,” *IEEE Transactions on Information Forensics and Security*, 13(11), pp. 2884-2896, 2018.

[52] R. Theagarajan, B. X. Guan, and B. Bhanu, “DeephESC: An automated system for generating and classification of human embryonic stem cells,” *International Conference on Pattern Recognition*, pp. 3826-3831, 2018.

[53] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 1979.

[54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[55] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv:1409.1556*, 2014.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[57] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255, 2009.

[58] R. Wiehagen, and C. H. Smith, “Generalization versus classification,” *Journal of Experimental & Theoretical Artificial Intelligence*, 7(2), pp. 163-174, 2007.

[59] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, “Consistent binary classification with generalized performance metrics,” *Advances in Neural Information Processing Systems*, pp. 2744-2752, 2014.

[60] A. Cioppa, A. Deliége, M. Istasse, C. De Vleeschouwer, and M. Van Droogenbroeck, “ARTHuS: Adaptive real-time human segmentation in sports through online distillation,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[61] `https://aws.amazon.com/solutions/case-studies/hudl/`

[62] B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos (Eds.), “Distributed video sensor networks”, *Springer Science Business Media*, 2011.

Rajkumar Theagarajan (S’14) received the B.E degree in electronics and communication engineering from the Anna University, Chennai, India, in 2014 and M.S. degree in electrical and computer engineering from the University of California, Riverside, CA, USA, in 2016. He is currently working towards the Ph.D degree in electrical and computer engineering at the Center for Research in Intelligent Systems, University of California, Riverside, CA, USA. His research interests include computer vision, image processing, pattern recognition and machine learning.

Bir Bhanu received B.S. (with Hons.) from IIT-BHU; M.E (with Distinction) from BITS (Pilani); S.M. and E.E. in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA; Ph.D. in electrical engineering from the University of Southern California, Los Angeles, CA and M.B.A. from the University of California, Irvine, CA. He is the Bourns endowed University of California Presidential Chair in Engineering, the Distinguished Professor of electrical and computer engineering and the Founding Director of the interdisciplinary Center for Research in Intelligent Systems and the Visualization and Intelligent Systems Laboratory (Vislab) at the University of California, Riverside (UCR), CA. He is the Founding Professor of electrical engineering with UCR and served as its first Chair (1991-94). He has been the cooperative Professor of computer science and engineering (since 1991), bioengineering (since 2006) and mechanical engineering (since 2008). Recently he served as the Interim Chair of the Department of Bioengineering from 2014-16. He also served as the Director of the National Science Foundation graduate research and training program in video bioinformatics with UCR. Prior to joining UCR in 1991, he was a Senior Honeywell Fellow with Honeywell Inc. He has published extensively and has 18 patents. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human-computer interactions, and biological, medical, military and intelligence applications. Dr. Bhanu is a Fellow of IEEE, AAAS, IAPR, SPIE and AIMBE.