Soccer: Who Has The Ball? Generating Visual Analytics and Player Statistics

Rajkumar Theagarajan, Federico Pala, Xiu Zhang and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, Riverside, CA - 92521
{rthea001, fedpala, xzhan060}@ucr.edu, bhanu@cris.ucr.edu

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 sports, talent identification and selection are imperative for the development of future elite level performers. Current scenarios involve word-of-mouth, coaches and recruiters scouring through hours of videos and many times manual annotation of these videos. In this paper, we propose an approach that automatically generates visual analytics from videos specifically for soccer to help coaches and recruiters identify the most promising talents. We use (a) Convolutional Neural Networks (CNNs) to localize soccer players in a video and identify players controlling the ball, (b) Deep Convolutional Generative Adversarial Networks (DCGAN) for data augmentation, (c) a histogram based matching to identify teams and (d) frame-by-frame prediction and verification analyses to generate visual analytics. We compare our approach with state-of-the-art approaches and achieve an accuracy of 86.59% on identifying players controlling the ball and an accuracy of 84.73% in generating the game analytics and player statistics.

1. Introduction

Computer vision plays a key role in the world of sports and the best known current application areas are in sports analysis for broadcast. Computer vision is also used behind-the-scenes, in areas such as training and coaching, and providing help for the referee during a game. To date most of the applications for providing sports analysis and player training from video are carried out manually. This requires lots of hours spent watching videos and annotating them.

According to a survey conducted by CNS News [1] and Statista [2], soccer is the number one game played by most students. It is estimated that a total of 838,573 (450,234 boys and 338,339 girls) students all across the USA played soccer for their school for the year 2016/17. From this only 9% of the boys and 11.9% of the girls receive scholarship to go to college which makes it extremely competitive.

Identification of the next generation of sports stars is an important part of a coach's roles and responsibilities. Talent identification has traditionally been based on viewing athletes in a trial game or training session environment, whereby the players aim to impress coaches. This approach to talent selection or recruitment is not informed by scientific evidence, but rather a coach's subjective preconceived notion of the ideal player, which may result in repetitive misjudgments and limited consistency [17] [7] [27]. Therefore, it is of interest to further investigate this area for talent identification and help coaches and recruiters to select potentially talented players more easily and without bias.

The central premise of talent identification and recruitment is to identify and select the most promising young athletes with the potential to excel and become a successful professional senior athlete. In team-based sports, such as soccer, talent identification is a complex process due to the different qualities associated with performance, which includes personal and tactical attributes. Personal attributes refer to how well the player is able to keep the ball possession with him/herself and tactical attributes refer to how successful the player is in passing the ball to the team mates and adapting to different strategies.

In this paper, we propose an approach and design a system to automate the talent identification problem by generating visual analytics and player statistics for soccer from a video using traditional algorithms and deep learning techniques for computer vision. For this purpose, we collected our dataset that consists of 49,952 images which are annotated into two classes namely: players with the ball (12,586 images) and players without the ball (37,366 images). Fig. 1 shows examples of players with the ball and Fig. 2 shows examples of players without the ball, respectively, from our dataset.

2. Related Work and Our Contributions

2.1. Related Work

In this section, we describe various techniques that have been used in commercially-available systems today and how these techniques are being further developed. The main

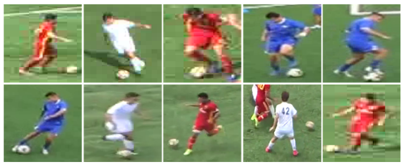


Figure 1. Examples of players with the ball.

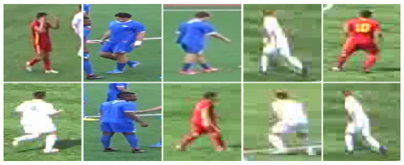


Figure 2. Examples of players without the ball.

applications for sports visual analytics are camera calibration, detecting and tracking players as well as the ball.

Camera calibration is essential for tracking players on the field. Majority of the commercially available systems today use a multi-camera approach for tracking players and the ball. A common approach for multi-camera calibration is to use known positions in the scene. This avoids the need for specially-equipped lenses and mounts. In sports such as soccer where there are prominent line markings on the pitch, a line based calibration is often used. Thomas [26] used the Hough transform to detect the straight lines in the soccer field. The author used the initial pose of the camera and peaks in Hough space to establish correspondence with the lines in the scene and hence calibrate the camera. Homayounfar *et al*. [10] computed the transformation between a broadcast image of a sports field and the 3D geometric model of that field. The authors first detected the vanishing point of the field which helped in reducing the total number of degrees of freedom to be estimated. Next, they performed semantic segmentation to segment the grass field from the field lines and estimated the homography matrix by formulating it as energy minimization in a Markov random field.

After camera calibration, detecting the position of players at a given moment of time and tracking them is the next step for generating useful visual analytics and player statistics, which can be extremely challenging. In most sports, especially in soccer the players appear to be very small from the camera's perspective causing a lot of occlusions and since they wear similar colored jerseys it makes it very difficult to identify players. The most common way to distinguish players is based on the color information of their jerseys [15], [18], but this does not help to discriminate the players individually. Bertini *et al*. [4] used close-up camera shots to identify players individually by the integration of face and jersey numbers. The drawback with this approach is that it requires a high resolution camera setup and resources which may not be available. Moreover, the camera is constantly being panned and zoomed depending on where the action is happening on the field focusing on only a few players which could lead to some players not being detected and, thus, not being able to generate proper statistics.

To address this task there are two commonly used methods: (a) extracting visual features (color [23], texture, motion vectors [19] as cues and then applying deterministic methods such as Support Vector Machines (SVM) [13], (b) considering player identification and tracking as a data association problem, we can detect players in each frame, obtain their tracklets and associate them in contiguous frames. Both of these approaches have problems when players are dense in one small area causing too many occlusions.

Lie *et al*. [15] approached this task by tracking the players using a Markov Chain Monte Carlo (MCMC) data association. Sachiko and Hideo [11] used a joint probability data association filter to associate the players location in the previous frames to the players location in the current frame. Instead of relying just on visual cues, Wei-Lwun *et al*. [16] detected players over multiple frames and used their short-term motion patterns to estimate their homography. Unlike most approaches that relied on matching robust feature points, the authors try to match edge points between players in continuous frames along with their motion patterns.

To date, the only system developed for generating visual analytics for soccer is the system developed by Stensland *et al*. [25]. The authors designed a real-time prototype (Bagadus) for sports analytics application using soccer as a case study which is currently installed at the Alfheim Stadium in Norway. The system integrates a sensor system which uses global positioning and radio based systems for tracking the players, a manual soccer analytics annotation system, and a video processing system with a camera array. Although, the prototype integrates and creates an interactive system for sports visual analytics, the process is not automated and requires hours of videos to be analyzed by scouts to find the most promising talents.

2.2. Contributions of this Paper

In light of the state-of-the-art described above, the contributions of this paper are:

- An approach to automatically generate visual analytics and player statistics for soccer matches from videos.
- Dynamic identification of players controlling the ball using Convolutional Neural Networks.
- Strategy to train Generative Adverserial Networks (GAN) that augment the datasets to improve the performance of the system.
- Extensive experimentation on a dataset collected from different soccer games.
- Trained networks on team dependent and team independent datasets to show the generalizability of the approach during different scenarios of the game.

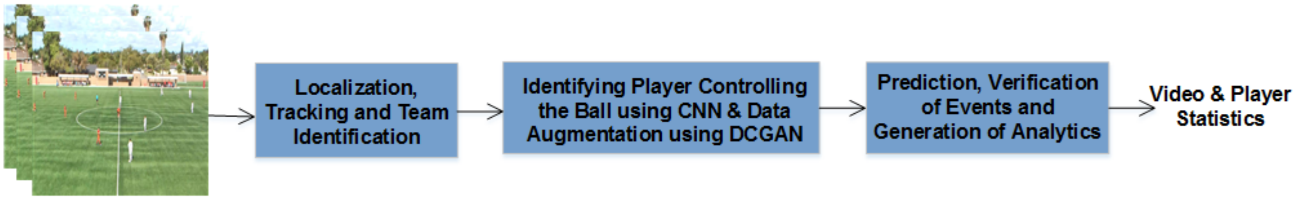

Localization, Tracking and Team Identification
Identifying Player Controlling the Ball using CNN & Data Augmentation using DCGAN
Prediction, Verification of Events and Generation of Analytics
Video & Player Statistics

Figure 3. Overall architecture of our approach.

3. Technical Approach

In this section, we explain the framework of our approach and the individual models we used for training the system. Fig. 3 shows the overall architecture of our approach.

3.1. Localization, Tracking and Team Identification

3.1.1 Localization of Soccer Players

In our approach we detected the soccer players in the incoming video stream using YOLO9000 - real-time object detection proposed by Redmon *et al*. [21]. The framework of YOLO9000 consists of a single CNN that predicts multiple bounding boxes for an image along with the respective class probabilities for each bounding box. YOLO9000 divides the input image into 11x11 grids and for each grid, the CNN predicts a set of bounding boxes along with the conditional probability for each class.

The network was trained on the PASCAL VOC 2007 dataset [6], the COCO 2016 keypoints challenge dataset [14] and Imagenet [22], all of these datasets consist of very diverse images for the class ***People*** which also includes sports players. The images in these datasets have different scale variations, and occlusions which is similar to the scenario on a soccer field. For a given frame, the bounding boxes belonging to the class ***People*** with probability greater than a threshold are considered to be the locations of the soccer players for that frame.

3.1.2 Tracking of Soccer Players

After detecting the soccer players in consecutive frames, we use the DeepSort tracking approach proposed by Wojke *et al*. [28] to track the soccer players over consecutive frames and formulate the association of the soccer players as a re-identification problem. The approach involves training a YOLO9000 based CNN. The CNN detects the players in each frame and extracts a feature set for each player. The authors also concatenate a 8-dimensional state-space feature set (u, v, γ, h, u', v', γ', 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 the bounding box and (u', v', γ', h') are their respective velocities in the image coordinate. The association of the soccer players in the next frame is done by using the visual appearance feature from the CNN and 8-dimension state-space motion feature as input to a Hungarian algorithm.

3.1.3 Histogram Matching for Team Identification

Soccer matches involve two teams wearing different colored jerseys. Each of these jerseys is visually very different from the other, hence in our approach a simple histogram based matching approach was sufficient for identifying the team of a given player.

Before processing the video, we manually crop the Region-of-Interest (ROI) of 10 random players from each team and their corresponding goal keepers and use them as a reference template. Next, after detecting the soccer players for a given frame, we crop the ROI of each soccer player and compute its 64-bin color histogram and compare it with each of the templates. The team with the closest average Euclidean distance is selected as the team of the player. Fig. 4 shows the process of histogram matching for team identification. In the future, we will replace this approach with a more sophisticated approach that does not require any templates for matching.

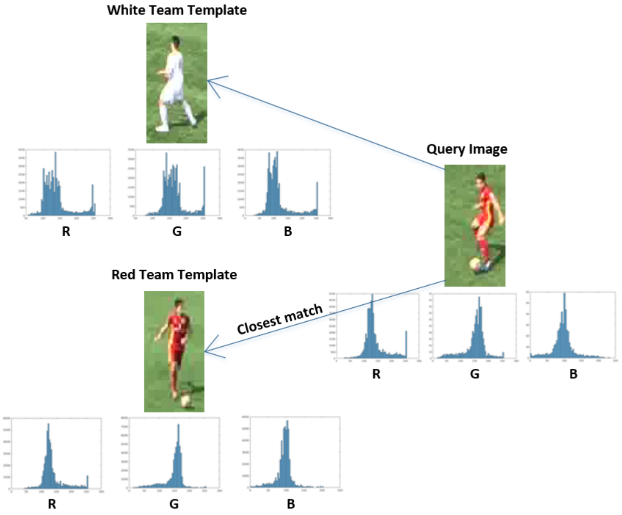


Figure 4. Histogram matching for team identification.

3.2. Identifying Player Controlling the Ball using CNN & Data Augmentation using DCGAN

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 use Convolutional Neural Networks trained to classify a given ROI of the soccer player as either "a player with the ball" or "player without the ball".

3.2.1 CNN for Identifying Player with the Ball

We experimented with our own baseline customized networks and fine-tuned state-of-the-art networks, namely: VGG-16, VGG-19 [24], ResNet18, ResNet34 [9] and AlexNet [12]. We compared each network and observed the features they learned to find the network that gives us the best classification accuracy and generalizability. For all the experiments, the CNN's were evaluated based on the mean accuracy between the two classes.

In order to train the state-of-the-art networks we had to resize all the images to be of size 224x224. We chose a mini batch size of 128 and during every epoch the training data is randomly shuffled and randomly horizontal-flipped. All the state-of-the-art networks were pre-trained on the ImageNet dataset [22]. Since the ImageNet dataset has 1000 classes, we modified the last fully connected layer from 1000 to 2 classes.

We designed customized networks, to determine if preserving the aspect ratio helps in improving the classification performance. The average aspect ratio of the images in our dataset was found to be 0.642. To keep all the images of a uniform size we resized the images to 160x100. Table 1 and Table 2 show the architecture of our networks. In Table 1 and Table 2, conv(x, y, z) represents convolution(kernel size=x, stride=y, padding=z). Furthermore, the weights for SoccerNet 1 and SoccerNet 2 were initialized with uniform Xavier distribution as described in [8]

We performed random parameter search for all the networks to obtain the best learning rate, momentum and weight decay. The networks were optimized using the stochastic gradient descent algorithm with weighted cross entropy loss. Since our dataset is unbalanced, we used the complementary *a-priori* probability of each class as weights in the loss function.

$$C_i = 1 - X_{ci}/X \tag{1}$$

X_{ci} is the total number of images belonging to class C_i and X is the total number of images for all classes.

The random parameter search was done by training and validating a given network with random values within a range for each parameter for 5 epochs, and the combination of parameters that resulted in the highest mean accuracy were chosen as the best parameters for that given network. Table 3 shows the best parameters that we obtained for training all the networks mentioned above.

Input dim.	**Output dim.**	**No. of Feature maps**	**Layer**
160x100	80x50	64	Conv(5,2,2)
80x50	40x25	128	Conv(5,2,2)
40x25	20x12	256	Conv(5,2,2)
20x12	10x6	512	Conv(5,2,2)
10x6	5x3	512	Conv(5,2,2)
7,680x1	2 classes	-	FC layer

Table 1. Architecture of SoccerNet 1.

Input dim.	**Output dim.**	**Number of Feature maps**	**Layer**
160x100	80x50	128	Conv(7,2,3)
80x50	40x25	256	Conv(3,2,1)
40x25	20x12	512	Conv(5,2,2)
20x12	10x6	1024	Conv(3,2,1)
10x6	5x3	1024	Conv(3,2,1)
15,360x1	2 classes	-	FC layer

Table 2. Architecture of SoccerNet 2.

Network	**Learning rate**	**Momentum**	**Weight decay**
SoccerNet 1	$2x10^{-2}$	0.6	$1x10^{-3}$
SoccerNet 2	$7.5x10^{-2}$	0.8	$1x10^{-3}$
VGG-16	$2.5x10^{-3}$	0.6	$1x10^{-4}$
VGG-19	$4x10^{-3}$	0.8	$1x10^{-4}$
ResNet18	$6x10^{-3}$	0.9	$1x10^{-4}$
ResNet34	$6.5x10^{-3}$	0.9	$5x10^{-4}$
AlexNet	$3x10^{-3}$	0.7	$1x10^{-4}$

Table 3. Best parameters for fine tuning the networks.

3.2.2 Dataset Augmentation using DCGAN

In this 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 performance of the network.

To achieve this we trained a Deep Convolutional Generative Adversial Network (DCGAN) [20]. It consists of two deep convolutional neural networks, a generator G and a discriminator D trained against each other. The generator takes a random noise vector, z, and returns an image, $X_{gen} = G(z)$. The discriminator takes a real or a generated image, and outputs a probability distribution $P(S|X) = D(X)$ over the two image sources. The discriminator is trained to maximize the log-likelihood of assigning the correct source while G tries to minimize it. The optimization function V is given by:

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \\ +\mathbb{E}_{x\sim p_z(z)}\left[\log\left(1-D(G(z))\right)\right] \quad (2)$$

The objective is that the two networks converge to the Nash equilibrium so that D is maximally confused and G generates samples that resemble the training data (in our case players with the ball). We followed the suggestion given by [20] for designing a stable architecture for generating images of better quality. The authors suggested replacing pooling layers with convolutional layers for both the generator and discriminator, using batch normalization after convolutional layers, ReLU activation in the generator and Leaky ReLU in the discriminator. Based on these suggestion we designed a Generator and Discriminator network to learn to generate images that resemble players with the ball.

3.3. Prediction, Verification of Events and Generation of Analytics

In this section, we explain the process of generating the visual analytics and player statistics. After tracking a soccer player, determining his/her team and identifying the player controlling the ball, we detect if the player controlling the ball changes over successive frames. If so, we observe if the new player controlling the ball belongs to the same team. If the new player belongs to the same team, we define it as a successful pass otherwise it is a failed pass.

Based on this logic we generate visual analytics describing which player currently controls the ball and when a pass is made. We also keep track of the duration each player controls the ball (ball possession) and the total number of successful passes each player has made, thus, generating player's performance statistics.

When two or more players are very close to each other, it becomes difficult for the network to identify which players controls the ball. To solve this we used a low pass filter to help smooth the transition between player controlling the ball. By doing so some false alarms due to the misclassification of player with the ball were also avoided.

4. Experiments

We trained and evaluated our approach on datasets collected from different soccer matches. The overall framework of our approach is implemented on pytorch [3] with 4 TITAN X GPU's with 7 TFlops of single precision, 336.5 GB/s of memory and 12 GB of RAM memory per board.

4.1. 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. In order to collect high resolution and good quality images with enough pixels on the players body, we allowed the camera operator to pan and zoom depending on where the action is happening on the soccer field.

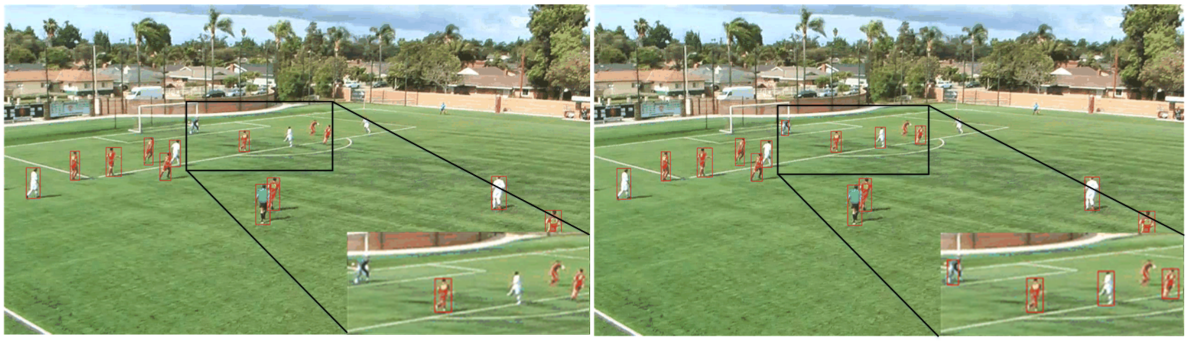
The dataset consists of 49,952 images, and it is annotated into two classes namely: players with the ball (12,586 images) and players without the ball (37,366 images). The dataset was annotated by five experts and the final label for a given image is obtained by taking the majority vote of the five annotators. The dataset is comprised of three teams whose jersey colors are white, red and blue. Out of the 49,952 images, the white team constitutes 27.95% of the dataset (13,959 images), the red team constitutes 34.82% of the dataset (17,392 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 24.81%, 16.46% and 58.72% for players with the ball and 29%, 41% and 30% for players without the ball, respectively. Table 4 shows the data distribution of the three teams for the two classes.

Class	White Team	Red Team	Blue Team
Player With Ball	3,123	2,072	7,390
Player Without Ball	10,386	15,320	11,210

Table 4. Data distribution of the three teams for the two classes.

Clearly from Table 4 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 person 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 25.2% of the data constituting for the class "players with the ball" and 74.8% of the data constituting for the class "players without the ball". The annotated images, the player statistics and the original videos will be released along with the camera ready version of the paper; they will be provided upon request made to the authors.

Furthermore, we used five test videos exclusively for evaluating our tracking, team identification and prediction of game analytics. The videos were categorized based on their complexity as ***easy, moderate*** and ***hard***. In the *easy* complexity case there are only 4 to 5 players spread wide apart usually in the defense zone and do not cause any occlusions. In the *Moderate* complexity case there are 6 to 10 people in the mid-field region causing partial occlusion to the surrounding players and the ball. The *hard* complexity case is when there are more than 10 players gathered within a small area on the field causing a lot of occlusions.



(a) Localization without grid based resizing

(b) Localization with grid based resizing

Figure 5. Comparison of grid based localization. (Note the magnified sub-image).

4.2. Localization, Tracking and Team Identification Results

4.2.1 Localization Results

We experimented with two state-of-the-art CNN's namely: YOLO9000 [21] and OpenPose [5] for the localization of soccer players. We evaluated both of the networks on five exclusive test videos (mentioned in Section 4.1) based on their average Intersection over Union (IoU). The YOLO9000 network achieved an IoU of 84.57% and the OpenPose network achieved an IoU of 69.84%. Both of the networks were able to detect players that were closer to the camera and as the players moved in the opposite direction the camera was facing, the number of pixels on player's body started to reduce making it difficult to detect them.

To solve this we applied a grid based localization approach, where we divided the input frames of size 1920 x 1080 into four equal sized cells. Each cell is of size 960 x 540 preserving the aspect ratio, and we resized each of the cells individually to 1920 x 1080. Next, we did localization individually on these four cells and concatenated the results into a single video frame. By doing this we achieved an IoU of 73.27% and **85.21%** using the OpenPose network and YOLO9000, respectively. Fig. 5(a) and Fig. 5(b) show an example of soccer player detection without and with grid based resizing, respectively. It can be observed that in Fig. 5(b) two of the soccer players that are farther away from the camera and the goal keeper are detected successfully after doing the grid based resizing (see the magnified sub-image). We still encountered some problems after doing the grid based resizing because some players who were close to the opposite horizontal baseline from the camera had too few pixels on their body. Moreover, resizing them increased the pixel distortion and made them unrecognizable.

4.2.2 Tracking Results

We evaluated the tracking algorithm on five test videos. We achieved an average accuracy of **76.54%** ± **6.37%**. The errors in tracking occur in difficult cases when two or more players overlap with each other, which causes the detector (YOLO9000) to detect them as a single player. This mostly occurs only when the striker enters the opposition area to attempt a shot at the goal. Even though multiple players were detected as one player, after these players separated from each other and were detected correctly, the tracking algorithm was still able to distinguish the players as they were before the overlapping occurred.

4.2.3 Team Identification Results

We evaluated our histogram matching approach on five test videos that were used for evaluating the detection and tracking algorithm. We achieved an average accuracy of **92.57%** ± **2.92%**. While calculating the accuracy, we ignored instances when multiple players overlapped each other. There were errors when a player is close to the opposite horizontal baseline away from the camera. The reason for this is that, the players have very few pixels on their body which causes errors while matching their histograms with the templates.

4.3. Results of CNN based Identification of Player with the Ball

In this section, we present the results obtained using different CNN's as described in section 3.2.1. We also show how the color of different team jerseys affects the performance of these networks. In our experiments, we randomly selected 75% of the data in each class as training data, 10% of the data for validation and 15% of the data for testing. We used the validation data to obtain the best parameters for each network as described in section 3.4.

In order to observe how the color of the team jersey affects the networks, we annotated a new set of images that involves soccer players wearing black colored jerseys. These images were not used for training the network and were added exclusively to the testing dataset. Table 5. shows the summary of the data distribution for the training, validation and testing dataset.

Dataset	Player with ball	Player without ball
Training set 75%	9,440	2,802
Validation set 10%	1,258	3,736
Testing set 15% + black jersey	1,888 + 502	5,606 + 3,733

Table 5. Data distribution for training, validation and testing.

We evaluated all the individual networks in three different settings namely: ***15% team dependent, 5% team dependent*** and ***team independent***. In the *15% team dependent setting*, we used 75% of the original dataset (red, white and blue jersey) for training. We used 15% of the original dataset and the black jersey for testing as shown in Table. 5. In the *5% team dependent* setting we used 85% of original dataset for training. We used 5% of the original dataset and the black jersey for testing. Finally, for the *team independent* setting we used 90% of the original dataset for training and the black jersey for testing.

4.3.1 Comparison of Different CNN Architectures

Table 6 and Fig. 6 show the mean accuracy obtained using the individual networks for the team dependent and team independent setting.

Network	15% Team Dependent	5% Team Dependent	Team Independent
SoccerNet 1	62.46%	67.81%	56.23%
SoccerNet 2	61.37%	70.59%	59.98%
VGG-16	80.21%	78.34%	70.71%
VGG-19	**85.37%**	**86.59%**	**76.48%**
ResNet18	75.48%	81.23%	70.62%
ResNet34	76.02%	80.34%	73.59%
AlexNet	69.32%	74.53%	66.82%

Table 6. Mean accuracy of all networks for the 3 settings.

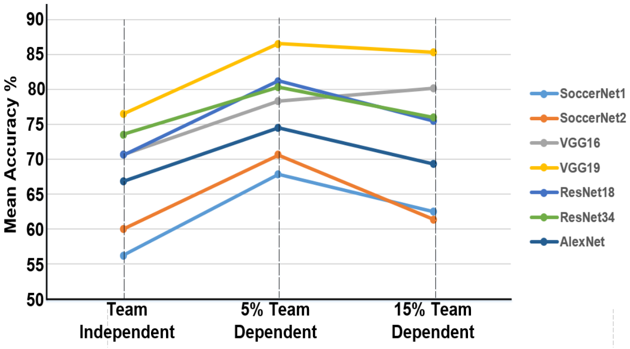


Figure 6. Mean accuracy plot of all networks for each setting.

From Table. 6 and Fig. 6, it is observed that VGG-19 achieved the best performance for all the three settings. The mean accuracy for all the networks in the team independent setting was less compared to their respective team dependent settings. This indicates that, apart from learning the representation of a soccer ball, the convolutional filters are also storing some color information of the player jersey.

4.3.2 Visualization of Features Learned by the CNN

Fig. 7 shows the visualization of the probability score map for VGG-19, when part of the image is hidden by a sliding square window of size 64 x 64 [29]. In Fig. 7, the image on the left is probability score map for the class "player with the ball". The brightness of the pixels in the probability score map indicate how strong of an indicator the corresponding pixel in the image is for detecting if the player has the ball.

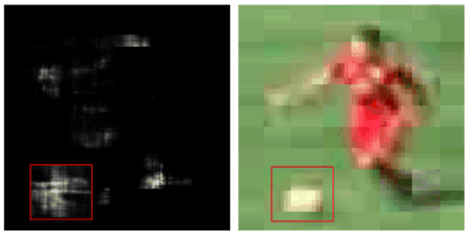


Figure 7. Visualization of features learned by VGG-19.

We further experimented to check if using gray scale images improves the performance for the team independent setting. To achieve this we did random parameter search for VGG-16 and VGG-19 to determine the best parameters for training the network with gray scale images. We used the team independent setting with 90% of the original dataset for training and the black jersey team as the testing set (mentioned in Section 4.3). VGG-16 and VGG-19 achieved mean accuracy of 67.36% and 70.24% respectively.

Fig. 8 shows the visualization of the probability score map for VGG-19 for gray scale images. In order to obtain the gray scale images, the RGB images were converted to HSV and the V plane was used as the gray scale image. The mean accuracy achieved using the gray scale images was less compared to the mean accuracy achieved with the RGB images. The reason for this is that when we convert the image to gray scale, the image loses some of its discriminative properties (white color of the ball) making it difficult for the network to generalize. In Fig. 8, it can be observed that, apart from the bright pixels corresponding to the ball, there are some very bright pixels that correspond to the color of the player's jersey. This indicates that the image is not very discriminative and the network is not generalizing well.

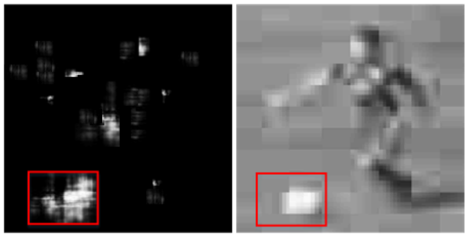


Figure 8. Visualization of gray scale features learned by VGG-19.

4.3.3 Effects of Dataset Augmentation on Selected CNN Architectures

In this section, we observe how augmenting the dataset affects the accuracy of VGG-16 and VGG-19 networks. We used the dataset consisting of 12,586 images of players with the ball to train the DCGAN model. The generator was designed to take as input a 100 x 1 dimensional noise vector randomly sampled from a uniform distribution. The output of the generator is a RGB image of size 128 x 128. The discriminator was designed to take as input a RGB image of size 128 x 128 and predict if the image is real or generated. The learning rate for the generator and discriminator are 10^{-4} and 10^{-5}, respectively, with mini batch size 32 and the weights of the generator are updated after every two mini batches. Both the generator and discriminator were optimized using the Adam algorithm and Binary Cross Entropy loss function [20].

After training the DCGAN, we observed that the model was able to learn the representation of a soccer player but was not able to completely learn the presence of the soccer ball. To overcome this, after partially training the DCGAN (generator is able to generate some reasonable images), we passed the generated images to the already trained VGG-19 network to classify them. Based on the output from the VGG-19, the weights of the generator network are updated again. If the image generated is a player without the ball then the generator is penalized more, thus helping it to learn the correct representation of player with the ball. Fig. 9 shows some of the generated images.

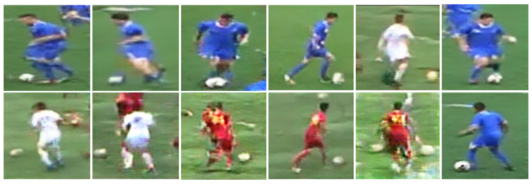


Figure 9. Examples of generated images of players with the ball.

Next, we generated 20,000 images of player with the ball and augmented it to our training dataset. We then trained the VGG-16 and VGG-19 networks and evaluated the network with the team independent setting (mentioned in Section 4.3). VGG-16 and VGG-19 achieved a mean accuracy of **72.13%** and **79.54%**, respectively. Comparing the results of the *team dependent* setting from Table. 6 for VGG-16 and VGG-19, we can observe that augmenting the dataset helped improve the accuracy of the respective networks. Thus, adding more variability helps improve the performance of the network.

4.4. Results on Generating Game Analytics & Player Statistics

We evaluated the accuracy of the generated visual analytics on the five test case videos of varying complexities (easy, moderate and hard as mentioned in Section 4.1). In the *easy* complexity case the proposed system was able to predict the visual analytics (i.e., which player controls the ball and when a pass is made) with accuracy of **84.73%**. We achieved an accuracy of **79.82%** for the moderate complexity and accuracy of **67.28%** for the hard complexity cases.

In the hard complexity case since the players are too close to each other causing occlusions, it is difficult for the network to identify which player is controlling the ball and leads to wrong visual analytics. We can solve this by identifying the player, who controls the ball just before he/she enters the opposition's zone and since he/she is attempting a shot at the goal, he/she is not going to pass the ball. Thus, we can pause the visual analytics processing for that duration and wait till the event is over to predict if the shot at the goal was successful.

To the best of our knowledge, we are not aware of any other automated system that predicts the game analytics and player statistics for soccer. Therefore, we cannot compare our system with any other existing system. The only possible comparison is the CNN based identification of player with the ball and it is shown in Table. 6.

5. Conclusions and Future Work

We proposed an approach and designed a system that is effective for generating automated visual analytics and player statistics for soccer videos. We collected a new dataset that consists of multiple teams. We performed exhaustive evaluation on the dataset with team dependent and team independent settings and observed how these settings affect the performance of the networks. We visualized how training the networks on RGB and gray scale images affects the generalization ability of the network learned and how augmenting more images using Generative Adversarial Networks to the dataset helps further to improve the performance. We also show how different scenarios of the soccer game affects the performance of the system and how we can overcome it. Future works will include collecting more data with players wearing different jerseys, finding more events of interest to improve the visual analytics and generate a more comprehensive statistics for the players.

Acknoledgment

This work was partially supported by a gift from SEVAai, Inc. to the University of California, Riverside.

References

[1] https://www.cnsnews.com/news/article/terence-p-jeffrey/1085272-players-football-remains-no-1-hs-sport-usa.

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

[3] Pytorch: Tensors and dynamic neural networks in python with strong GPU acceleration https://github.com/pytorch/pytorch.

[4] M. Bertini, A. Del Bimbo, and W. Nunziati. Player identification in soccer videos. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 25–32. ACM, 2005.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[7] D. G. Hoare and C. Warr. Talent identification and women's soccer: an australian experience. *Journal of Sports Sciences*, 18(9):751–758, 2000.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] N. Homayounfar, S. Fidler, and R. Urtasun. Sports field localization via deep structured models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5212–5220, 2017.

[11] S. Iwase and H. Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 751–754. IEEE, 2004.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[13] A. Li, F. Tang, Y. Guo, and H. Tao. Discriminative nonorthogonal binary subspace tracking. pages 258–271. Springer, 2010.

[14] 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. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[15] 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):103–113, 2009.

[16] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1704–1716, 2013.

[17] C. Meylan, J. Cronin, J. Oliver, and M. Hughes. Talent identification in soccer: The role of maturity status on physical, physiological and technical characteristics. *International Journal of Sports Science & Coaching*, 5(4):571–592, 2010.

[18] M. Naemura, A. Fukuda, Y. Mizutani, Y. Izumi, Y. Tanaka, and K. Enami. Morphological segmentation of sport scenes using color information. *IEEE Transactions on Broadcasting*, 46(3):181–188, 2000.

[19] J. Perš and S. Kovačič. Tracking people in sport: Making use of partially controlled environment. In *Computer Analysis of Images and Patterns*, pages 374–382. Springer, 2001.

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

[21] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[23] Y. Seo, S. Choi, H. Kim, and K.-S. Hong. Where are the ball and players? soccer game analysis with color-based tracking and image mosaick. In *International Conference on Image Analysis and Processing*, pages 196–203. Springer, 1997.

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

[25] H. K. Stensland, V. R. Gaddam, M. Tennøe, E. Helgedagsrud, M. Næss, H. K. Alstad, A. Mortensen, R. Langseth, S. Ljødal, Ø. Landsverk, et al. Bagadus: An integrated real-time system for soccer analytics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(1s):14, 2014.

[26] G. Thomas. Real-time camera tracking using sports pitch markings. *Journal of Real-Time Image Processing*, 2(2-3):117–132, 2007.

[27] A. M. Williams and T. Reilly. "talent identification and development in soccer," *International Journal of Sports Science*, 18, pp. 657-667, 2000.

[28] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. *arXiv preprint arXiv:1703.07402*, 2017.

[29] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.