IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 1, 2019, accepted April 19, 2019, date of publication April 30, 2019, date of current version May 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913953

Analyzing Basketball Movements and Pass Relationships Using Realtime Object Tracking Techniques Based on Deep Learning

YOUNG YOON[1], (Member, IEEE), HEESU HWANG[2,3], (Member, IEEE), YONGJUN CHOI[1], MINBEOM JOO[4], HYEYOON OH[1], INSUN PARK[1], KEON-HEE LEE[2], AND JIN-HA HWANG[2]

[1]Department of Computer Engineering, Hongik University, Seoul 04066, South Korea
[2]Department of Material Science and Engineering, Hongik University, Seoul 04066, South Korea
[3]SmartIndividuals Co., Ltd., Seoul 06627, South Korea
[4]Cube AI Inc., Seoul 04066, South Korea

Corresponding author: Jin-Ha Hwang (jhwang@hongik.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) through the Basic Science Research Program, Ministry of Education, under Grant NRF-2016R1D1A1B03931324, in part by the National Research Foundation of Korea (NRF) Grant supported by the Ministry of Science and ICT for convergent research through the development program for convergence research and development over traditional culture and current technology under Grant 2016M3C1B5906486, and in part by the 2019 Hongik University Research Fund.

ABSTRACT In this paper, we present techniques for automatically classifying players and tracking ball movements in basketball game video clips under poor conditions, where the camera angle dynamically shifts and changes. In the core of our system lies Yolo, a realtime object detection system. Given the ground truth boxes collected by our data specialists, Yolo is trained to detect the presence of objects in every video frame. In addition, Yolo uses Darknet that implements convolution neural networks to classify a detected object to a player and to recognize its jersey numbers of specific movements. By identifying players and ball possessions, we can automatically compute ball distributions that are reflected on complex networks. With original Yolo system, player movement can be interrupted, when the players move out of the frame due to camera shift and when players overlap each other on a two-dimensional frame. We have adapted Yolo to keep track of players even under such poor condition by considering contextual information available from the framework preceding and/or succeeding problematic video frames. In addition to the novel movement inference method, we provide a framework for analyzing the pass networks in various perspectives to help the managing staff to reveal critical determinants of team performance and to design better game strategies. We assess the performance of our system in terms of accuracy by making a comparison with the analytical reports generated by human experts.

INDEX TERMS Sports analytics, object detection, complex networks, deep learning, video processing.

I. INTRODUCTION

Sports analytics have been recognized as a growing multidisciplinary field which incorporates sports, statistics, mechatronics, electrical engineering and computer science with imaginative and intuitive approaches [1]–[3]. Ever-increasing interests in sports have led to the development of various analytical methods for understanding the underlying sports parameters and for finding the clues for enhancing the performance of teams and individual athletes. Sports analytics have gained much attention especially in the world of popular team sports such as soccer, basketball, and volleyball. The statistical analyses are combined with both on-field and off-field measurements [4] in order to improve on-field player-based performances and to assist sports management. In particular, the on-field statistics and analyses have been continuously developing sophisticated tools and performance indicators, with the purpose of enhancing and maximizing players' performance and efficiency with minimal investments. The current R&D trends are moving their approaches from simple descriptive descriptions to in-depth relationship analyses within teams and tactics as multivariate functions.

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

Meanwhile, to draw the relationships or interactions, there should be involved massive human labors with high concentration efforts such as repeating classification and judgment on interactive assignments for recognition of players and balls and event-tagging [5]–[10]. Furthermore, Halvorsen et al. reported an integrated system which is composed of sensors, annotation systems, and a camera image acquisition system to cope with massive amounts of sports data stored in image file formats [11]. Nonetheless, pre-existing attempts on complicated sports analytics suffer from massive, labor-intensive and time-consuming manual operations that are prone to errors such as false negatives and false positives [5]–[11].

In this paper, we aim to devise a novel system for automated sports analytics. For this objective, we propose to use a novel combination of deep learning techniques [12], [13] and network sciences [14].

The recent advancement of deep learning algorithms has paved a way for many successful applications. The computer visions powered with convolutional neural networks [15] and realtime region-proposal algorithms [16]–[19] have exhibited close-to-human object detection and classification capabilities. In this paper, we focus not only on locating players and identifying their jersey numbers but also on tracing ball pass movements and interceptions by adapting the Yolo framework [16]. We made significant improvement over Yolo to reliably track player and ball movements even under the poor condition where the players move out of the frame due to frequent camera shifts and overlay each other on two-dimensional video frames. We devised a novel algorithm to infer missing players and prevent jersey number misclassification by considering contextual information from video frames preceding and succeeding problematic frames.

From a sequence of automatically recognized players and their actions, we generate player interaction networks as directed graphs. With the graph information, we can employ network science to systematically understand and analyses team performance. Network science is a rapidly-growing academic field which study complex networks ranging from social networks, information networks, epidemiology, and biology networks [14], [20]–[23]. Recently, a number of research works using network science surfaced in the field of sports analytics [5]–[10]. Analyzing the networks has been shown to be effective particularly for popular team sports [5]–[10]. Such approaches have offered the opportunities to interpret the performance of individual players from different team perspectives with new parameters and data management. Besides the conventional network parameters, we provide the means to assess the importance of each player with an adapted multivariate Eigen Centrality function [24], which we refer to as Player Centrality. In addition, we offer the intuitive visual cues that reflect both the individual and team behaviors on a single integrated view.

In the rest of this paper, we present our technical contributions in the following structure: (1) in Section II, we devise a novel machine learning methodology for identifying ball possessions based on Yolo, a realtime object detection system and ball movement tracking based on the identified object; (2) in Section III, we extract complex networks that represent ball movement patterns and explain how it can be used to find the key determinants of team performance. Also, we evaluate the accuracy of our novel approaches; (4) in Section IV, we introduce related works; and (5) in Section V, we briefly discuss future works and conclude.

II. SYSTEM DESIGN

In this section, we present the mechanism of object detection from video clip frames and a novel machine learning procedure for recognizing players and their movements.

A. OBJECT DETECTION METHODOLOGY

We adapted Yolo for the realtime detection of objects in recorded video frames [16]–[18]. We selected Yolo, as it has outperformed notable existing object detection systems such as Fast-RCNN [25] and Faster-RCNN [19] in terms of mean average precision (mAP) and object detection speed in FPS (frames per second). The overall procedure for Yolo's object detection and classification mechanism is outlined in Fig. 1.

Yolo first dissects a given video frame into a D x D grid. For each cell, Yolo proposes bounding boxes and computes a confidence value that each box contains an object. We represent the dimension of a bounding box with a coordinate of its center (x, y) and width and height (w, h). Yolo uses K-means clustering algorithm to distinguish distinct groups of dimensions for bounding boxes based on the collected ground truth boxes. The clusters of K = 5 is a good tradeoff between model complexity and high recall [18]. The confidence of each proposed bounding box is computed based on Equation 1. IOU (Intersection Over Union) is the area of overlap between the proposed bounding box and the ground truth box, as shown in Fig. 2.

$$P(object) \times IOU_{prop}^{truth} \tag{1}$$

As shown in Fig. 3, the output that is predicted by Yolo is a $13 \times 13 \times (5 \times 5 + C)$ tensor, where C is the number of classes a proposed bounding box can be associated. We set the grid size to 13×13 considering the high resolution of the basketball game recordings we have. Each cell proposes 5 bounding boxes. We use two Yolo models to identify the exact ball possessors. The first Yolo model distinguishes whether a detected object is a ball possessor or a player without a ball (C = 2). Once a ball possessor is detected, we use the second Yolo model to identify the possessor's jersey number. We first detect the ball possessor without any identification of the jersey number. For the second Yolo model, C = 10, as there are 10 classes, i.e., there are 10 jersey numbers on the floor. Yolo uses Darknet which is a C implementation of Convolutional Neural Network (CNN) [15]. For each cell, Yolo computes the conditional probability that a detected object can be categorized into a class as shown in Equation 2.

$$P(Class|Object) \tag{2}$$

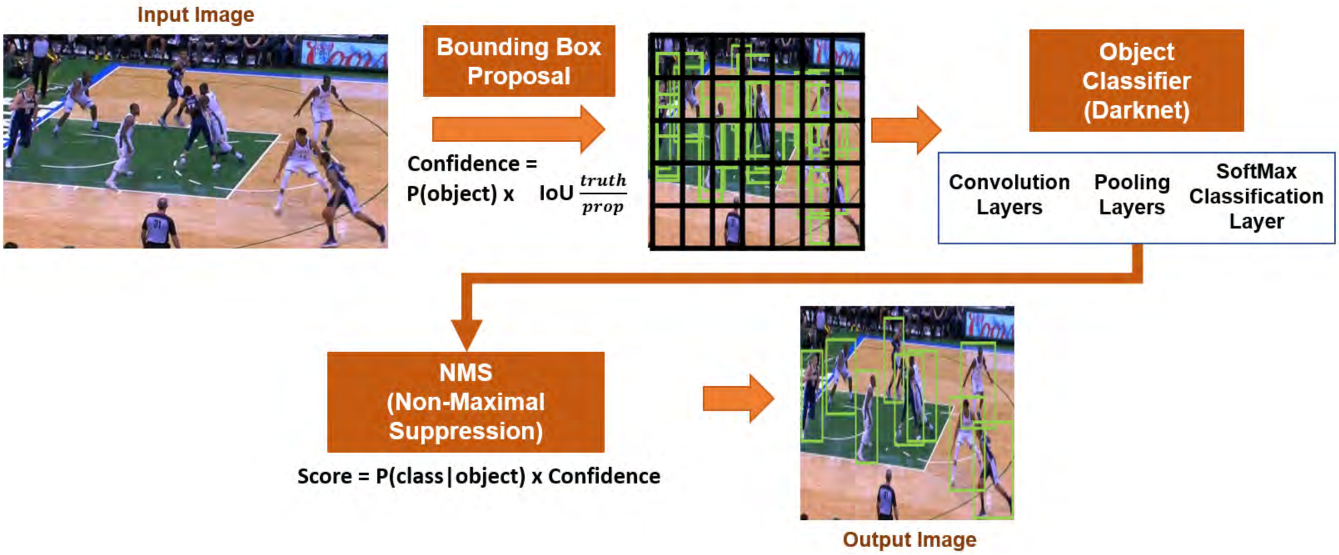


FIGURE 1. Object detection and classification procedure with Yolo.

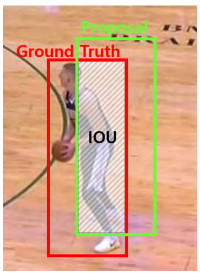


FIGURE 2. An example of IOU evaluation.

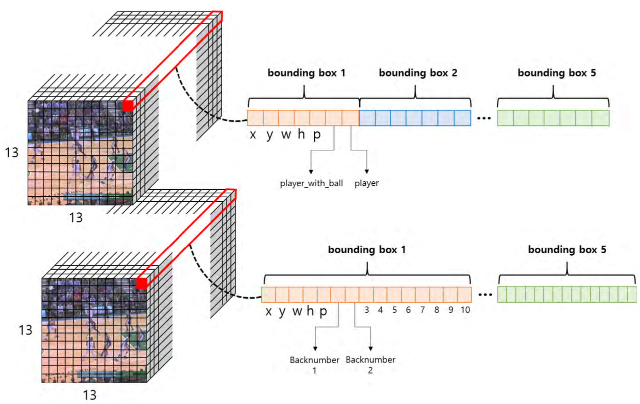


FIGURE 3. Tensors Yolo learns for detecting a ball, players and jersey numbers.

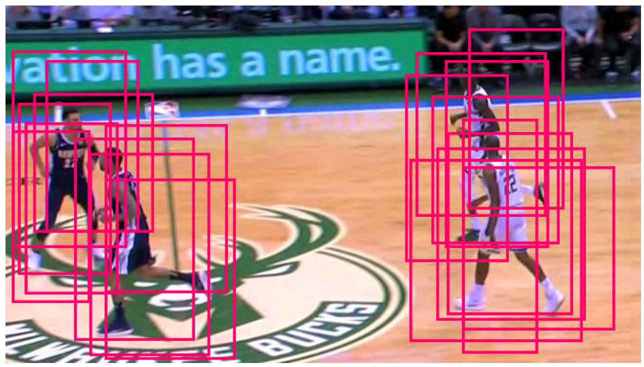


FIGURE 4. An example of redundantly proposed boxes around players.

Algorithm 1 Non-Maximal Suppression

Input: Proposal_Boxes[0…n]
Output: Final_Boxes
Sort Proposal_Boxes of by score in descending order;
while *boxes to check in Proposal_Boxes* $\neq \phi$ **do**
  max_box ← the box with the highest score;
  foreach *box* ∈ *Proposal_Boxes[1…n]* **do**
    if *IOU (max_box, box) > threshold* **then**
      the score of box ← 0
  Remove all boxes with scores=0;
  Insert max_box to Final_Boxes;
  Remove max_box from Proposal_Boxes;

Then, Yolo returns the score of each proposed bounding box that by calculating Equation 3.

$$P(Class|Object) \times P(Object) \times IOU_{prop}^{truth} \tag{3}$$

Grid cells may propose redundant and overlapping bounding boxes especially when they are adjacent to each other, as shown in the examples in Fig. 4. To remove redundant boxes and pick the best bounding box for an object, we run the Non-Maximal Suppression (NMS) method that is specified in Algorithm 1. NMS first sorts the bounding boxes by their scores in descending order. While NMS iterates through the sorted list of bounding boxes, it discards any bounding box whose IOU with the highest-scored box is more than the configured threshold. We regard the highly overlapping bounding boxes to be referring to the same object. NMS continues the iteration until it encounters a box whose IOU with the highest-scored box is less than the configured threshold. NMS finalizes the class association for the highest-scored bounding box and then repeats the aforementioned steps to remove the boxes that intersect with the newly encountered bounding box more than configured threshold.

B. MACHINE LEARNING PROCEDURE

To train the object detector, we collect a set of ground truth boxes for each class we want to recognize. For players possessing a ball, we make sure the ball and the torso of the ball possessor are captured within a ground truth box, as shown in Fig. 5. For players without a ball, we capture the entire body within a ground truth box, as shown in Fig. 6. We also box jersey numbers. Samples of the captured jersey numbers are shown in Fig. 7. Each box is tagged with a team information and the actual jersey number.

FIGURE 5. Boxes showing the possession of the ball.

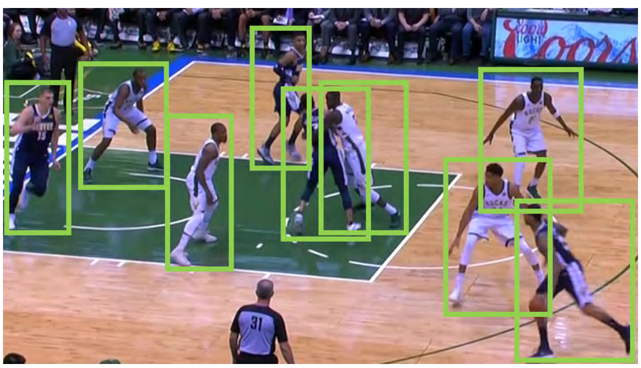


FIGURE 6. Collecting ground truth boxes of jersey numbers for each team.

As mentioned in the previous section, a ball possessor is first identified, and then the detected ball possessor's jersey number is recognized. To implement this two-phase recognition method, we generate two separate models that identify ball possessors and jersey numbers. We had to resort by separating the models, because capturing every player with a visible jersey number was a reckless task especially when the jersey number was not always clearly visible in the video clip. Furthermore, if we collect jersey numbers only from the

FIGURE 7. Collecting ground truth boxes of ball possession.

ball possessors, then we cannot get enough training data to identify jersey numbers.

C. THE GENERATION OF PASS NETWORK INFORMATION

Given the sequence of video frames with ball possessors, we track the ball movements among players. In order to trace the passing relationships, we first need to identify the players and track their movements. Yolo does not attempt to recover the lost track of players and balls when scene shift occurs. Also, Yolo cannot distinguish players when they overlap each other on a two-dimensional video frame without absolute coordinate information. Therefore, we adapted Yolo and devised a new tracking algorithm named Joy2019. The algorithm is specified on Algorithm 2. We first assign a unique tracking ID (tid) for each identified box that encapsulates

Algorithm 2 Joy2019, The Player Tracking Algorithm

```
Input: Frames: video frames with bounding boxes
Output: LabeledFrames: Frames and their bounding
        boxes labeled with track IDs and jersey
        numbers
foreach frame F ∈ Frames do
    if F is the first frame then
        Assign a unique trackingID from 1 to 5 to each
        bounding box;
        Set F as the preceding frame;
    else
        foreach bounding box c ∈ F do
            Find the closest bounding box, b, in the
            preceding frames;
            c.trackID ← b.trackID;
            c.jerseyNumber ← b.jerseyNumber;
            if the jerseyNumber of c recognized then
                From LabeledFrames, retrieve
                P = {p|p.trackID = c.trackID};
                foreach p ∈ P do
                    if there is no jersey number for p
                    then
                        p.jerseyNumber ←
                        c.jerseyNumber;
            Insert the current frame and its bounding
            box information to LabeledFrames;
```

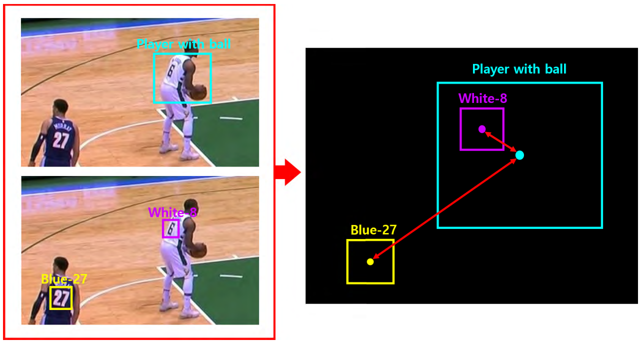


FIGURE 8. Identifying the jersey number of a ball possessor.

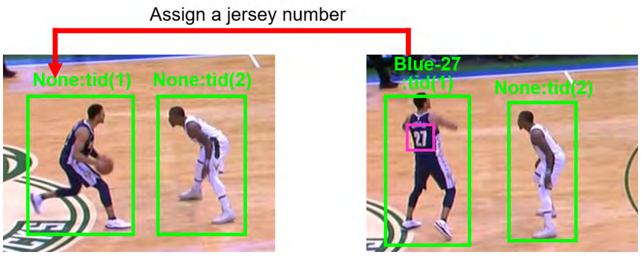


FIGURE 9. An example of assigning a jersey number to a box with the same tid.

a player. A jersey number of each box is not assigned until the back number is detected. As illustrated in Fig. 8, we check if a player box includes a bounding box that contains a jersey number. Once a jersey number box that is recognized to be belong to a specific player box, we iterate back through the previous frames and designate the jersey number to the unassigned boxes with the same tid. For example, Fig. 9 shows that a player box with tid(1) is associated with the jersey number 27. With Joy2019, we can detect the pass relationship as shown in Fig. 10. We can spot the moment when the player with jersey number 7 passes the ball to its teammate contained in a box with tracking ID 3.

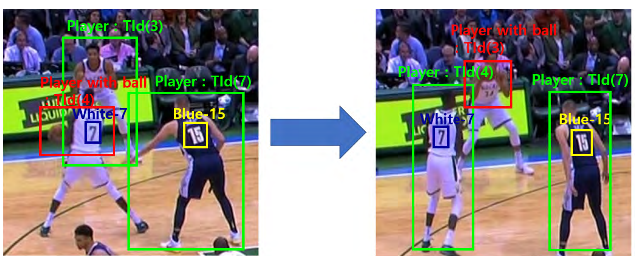


FIGURE 10. Example of recognizing and tracking pass actions.

Upon assigning a tracking ID to a player box in a given frame, we search through 10 preceding video frames to find the box with the same tid that is within the closest proximity in the frame coordinate (Algorithm 2:7). We look up beyond the immediately preceding frame as it may not show a recognized box that matches the current player box being examined. The number of frames to look through is set to 10,

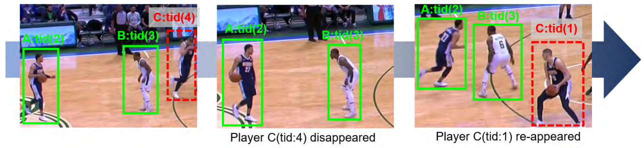


FIGURE 11. The problem of tracking a reappeared player box.

as the range of player's movement is within 150 pixels, meaning that player's relative position in the framework does not change drastically within these 10 frames.

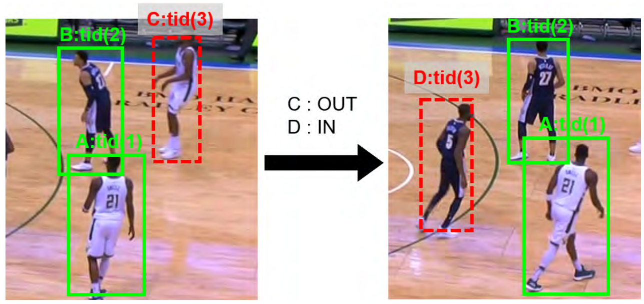


FIGURE 12. The box that disappeared from the scene and the box that reappeared share the same tracking ID.

There are more cases that prevent Joy2019 from tracking movements completely and correctly as follows. Due to dynamically-moving camera angles, a player box may temporarily disappear from the video frame. When it re-appears on the scene, tracking ID information can be lost. For this case, we assign the lowest unused tracking ID to the re-appeared player box and resume the procedure of finding its jersey number. This causes some issues as shown in Fig. 11 and 12. In Fig. 11, Player C with tid:4 disappears from the scene and then reappears in the following frame. If we cannot find the closest box within a configured distance in the set of previous frames, we assign the lowest available tracking ID to the reappeared box. Therefore, a player can be associated with different tracking IDs over time. This is problematic as multiple players can have the same tracking ID, as shown in Fig. 12. Player C:tid(3) disappears, and reappeared Player D:tid(3) takes over Player C's tid. We can correct the tracking IDs when a jersey number is recognized in one of the following frames. This can cause the two different players to have the same jersey number as well since jersey numbers are not visible in every frame. Suppose Player D:tid:(3) in Fig. 12 is assigned a jersey number, our Algorithm searches through the preceding frames to find the boxes with the same tracking ID but without a jersey number. It is susceptible that our Algorithm assigns the jersey number of Player C:tid(3) to Player D:tid(3) as shown in Fig. 12.

Another frequently occurring situation is where players overlap each other in a frame, as shown in Fig. 13. Player A:tid(1) and Player B:tid(2) competing for the ball are overlapped. Yolo confuses the overlapped image as a single player. Player B:tid(2) that eclipsed Player A:tid(1) is

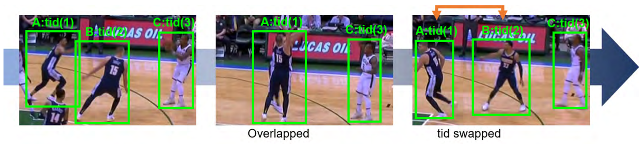


FIGURE 13. Tracking IDs of the two overlapped player boxes being swapped.

accidentally associated with tid(1). In the subsequent frame, the tracking IDs of the two overlapped players are swapped, which can cause wrong player informations to be associated with the recognized boxes.

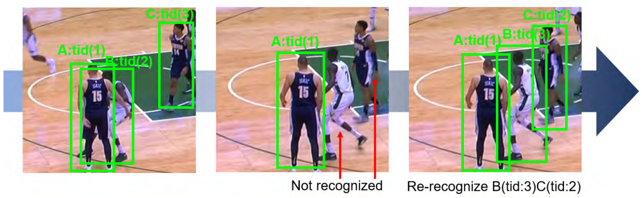


FIGURE 14. Temporarily unrecognized boxes receive different tracking IDs.

Fig. 14 illustrates a case where Yolo temporarily unrecognizes Player B:tid(2) and Player C:tid(3). Joy2019 may assign incorrect tracking IDs when they reappear after a significant number of frames have passed.

It is non-trivial to avoid the aforementioned issues, as Joy2019 relies on a method that identifies a player by its jersey number that is not always visible. We can fix this problem to some degree by obtaining more frames recorded from different camera angles. Another solution is to have a video footage that shows the entire floor instead of zooming in and out. We can also consider training our classifier to identify the player by their unique physical appearance including facial characteristics. However, such an approach may require expensive ultra-high resolution video images and many close-up shots.

III. EVALUATION

Our evaluation was conducted on a deep learning machine equipped with Intel R Core I7-8700 CPU, Nvidia GeForce GTX 1080Ti GPU, and 32GB of memory. This machine is operated with Ubuntu 18.04 LTS. For this evaluation, we used a publicly-available video recording of a full NBA game. The video clip is encoded with MPEG4 at a bit-rate of 1200 Kbps. The resolution and the frame speed are 1280x720 and 30 FPS, respectively. We collected over 16,000 ground truth boxes of balls, players and jersey numbers from 1204 video frames. We took the collection of ground truth boxes as an input to Yolo that is pre-trained with Microsoft's Coco dataset [26]. For testing, we extracted a 3-minute partial recording from the game footage that was not used for training.[1]

[1]The videos with annotated tracking results to be made available on IEEE Access

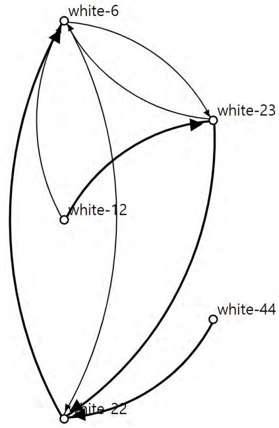


FIGURE 15. The human-generated graph showing passing relationship among players of Team White.

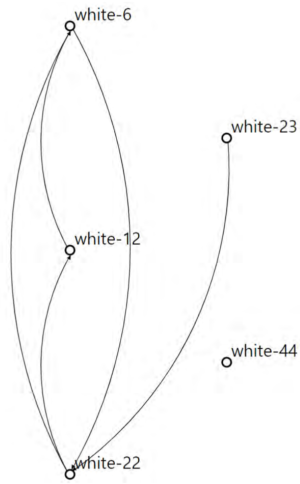


FIGURE 16. The Yolo-generated graph showing passing relationship among players of team white.

TABLE 1. Accuracy of recognizing jersey numbers and players.

	Human	Yolo	Joy2019
Jersey number (Precision)	100%	34.3%	74.3%
Player (Recall)	100%	89.5%	89.8%

From the video clip, we extracted a network showing passing relationships among players who are denoted as vertices (nodes). The directed edges (links) between nodes indicates an inbound or an outbound pass. We visualized the pass graphs generated by human, Yolo and Joy2019 for each team, as shown in Fig. 15, 16, 17, 18, 19 and 20. We produced the graphs with D3.js [27]. Table 1 shows the accuracy in recognizing jersey numbers and players. In terms of player recognition, our algorithm, Joy2019, shows slightly higher recall than Yolo, and it is on par with the human. False classification of a player into another object did not occur. Joy2019 exhibited more than 100% improvement over human

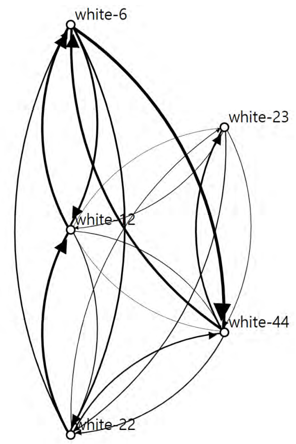


FIGURE 17. The Joy2019-generated graph showing passing relationship among players of team white.

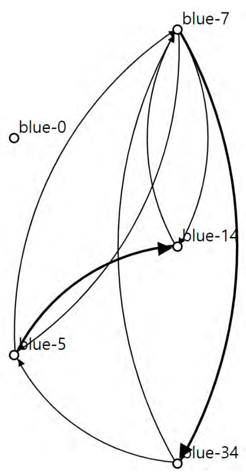


FIGURE 18. The human-generated graph showing passing relationship among players of team blue.

and Yolo with regards to the precision of recognizing jersey numbers. Note that Joy2019 attempts to infer jersey numbers even in the case where jersey numbers are not visible. Joy2019 used the tracking information based on the preceding or succeeding frames for the jersey number inference. Thus, Joy2019 showed much higher recall than Yolo. However, we did not include the recall of jersey number recognition as it was not feasible to count only the visual detection of the jersey numbers by Joy2019, and make fair comparison against Yolo.

Average node degree of each graph is shown in Table 2. Mean absolute percentage error (MAPE) is at most 33.5%. In addition to identifying pass relationships, we assessed and quantified the performance of each player using a multivariate Eigen-centrality algorithm which we refer to as the Player Centrality (PC). We adapted the NodeRank algorithm that we devised for a network security problem [24]. The terms that we used for computing the PC are defined as follows: Ψ is a set of play actions such as passing to a teammate, receiving

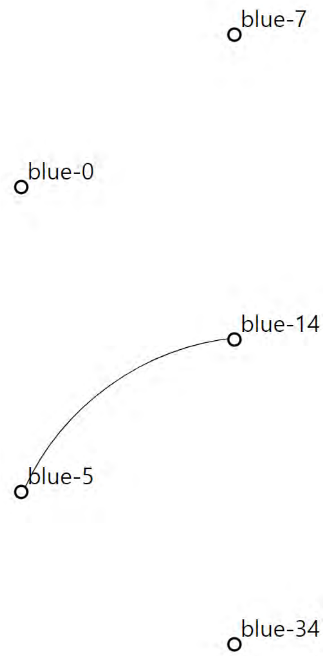


FIGURE 19. The Yolo-generated graph showing passing relationship among players of team blue.

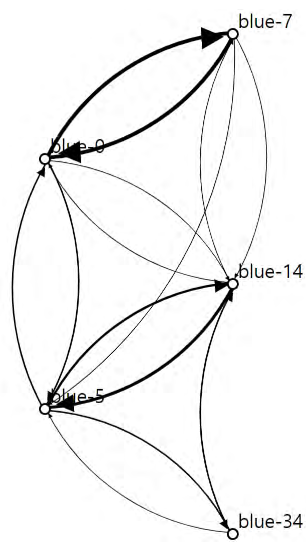


FIGURE 20. The Joy2019-generated graph showing passing relationship among players of team blue.

a pass, intercepting the ball from an opponent, and getting intercepted by an opponent. The PC of a node is a weighted sum as defined in Equation 4.

$$R(v_x) = \sum_{i \in \Psi} \alpha_i R_i(v_x) \tag{4}$$

α_i is a weight for a play action, i. For pass actions, the weight is set to 0.3. For the action of being intercepted by an opponent, we set the weight to -0.7, since it negatively affects the team performance. On the other hand, we set the weight to 0.7 for intercepting the ball from an opponent,

TABLE 2. Average node degree of pass graph for each team. (H and J stands for interpretation by human and Joy2019, respectively. The notation is consistently used in the tables in the rest of this section.).

	Total Node Degree		Outbound Node Degree		Inbound Node Degree		
Team	H	J	H	J	H	J	MAPE
White	5.4	7.2	2.8	3.6	2.6	3.6	33.5%
Blue	4.6	5.6	2	2.8	2.6	2.8	23.1%

as such an action can given positive impact to the team performance. Sports analysts can extend Equation 4 to account for more actions such as turnovers, scoring and fouls.

$R(v_x)$ in Equation 4 is score of v_x, which is defined in Equation 5.

$$R_i(v_x) = \frac{1-d}{n} + d\Big(\sum_{t \in G} a_{i(v,t)} \cdot \frac{R_i(v_t)}{C_i(v_t)}\Big) \quad (i \in \Psi) \tag{5}$$

$a_{i(v,t)}$ is a stochastic adjacency matrix. Every row in $a_{i(v,t)}$ sums to 1. $C(v_t)$ is used for normalizing the PC value, which is defined in Equation 6.

$$C_i(v_t) = \Big(\sum_{t=1}^{n} a_{i(v,t)}\Big)(i \in \Psi) \tag{6}$$

d in Equation 5 is a constant called a damping factor which reflects the phenomenon that not all players engage in a play action directly with every other player on the flow, in a given observed time frame. We set the damping factor to 0.85, which is the value used by PageRank [28] to discount scores of incoming web page links by 15%. While PageRank considers only the incoming links for scoring the rank of a node, we consider both outbound and inbound actions for assessing a node's centrality.

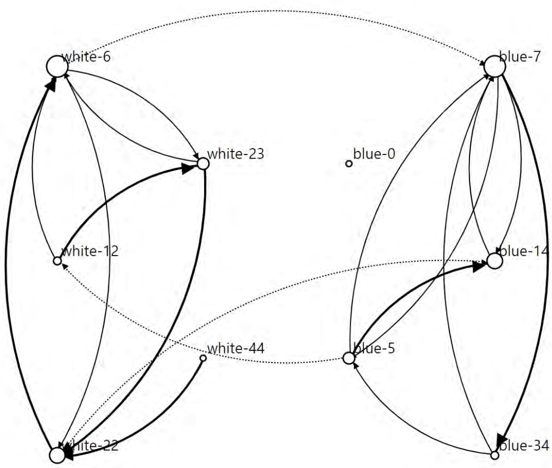


FIGURE 21. The human-generated graph showing passing relationship among players.

We reflected the PC value of each player on the graphs in Fig. 21, 22 and 24. A solid arrow indicates a pass to a teammate. A dotted arrow a ball being stolen (intercepted) to an opponent. We visualized the PC of each player as the circular size of player node. That is, a player with larger node size indicates a higher PC value. With this visual cue, we can easily locate the key players on the flow and their relative impact on the team.

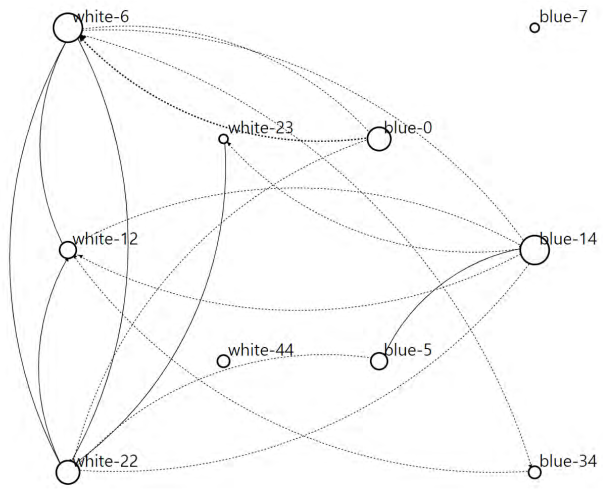


FIGURE 22. The Yolo-generated graph showing passing relationship among players.

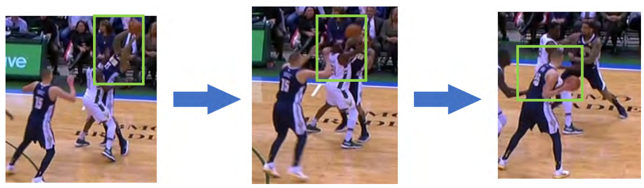


FIGURE 23. An example showing Joy2019 to incorrectly identify the defender as a passer.

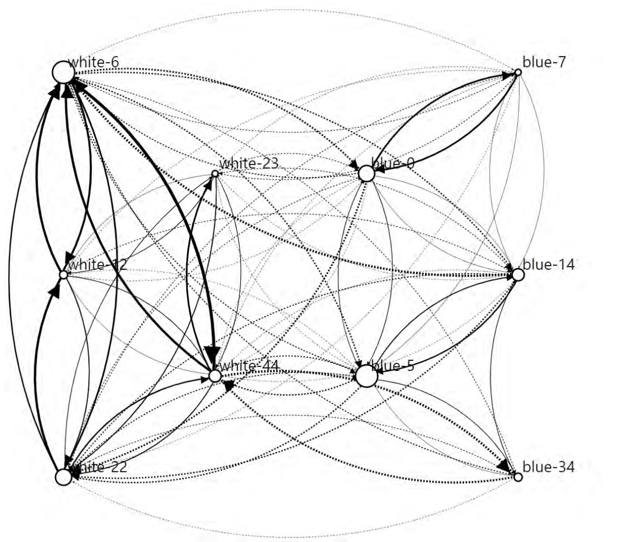


FIGURE 24. The Joy2019-generated graph showing passing relationship among players.

TABLE 3. Pass distribution to teammates in percentage for each player in team white (generated with Yolo).

Passer	White-6 H	White-6 Y	White-12 H	White-12 Y	White-22 H	White-22 Y	White-23 H	White-23 Y	White-44 H	White-44 Y	MAPE
White-6	0.0%	0.0%	0.0%	0.0%	50.0%	100.0%	50.0%	0.0%	0.0%	0.0%	200.0%
White-12	33.3%	100.0%	0.0%	0.0%	0.0%	0.0%	66.7%	0.0%	0.0%	0.0%	300.3%
White-22	100.0%	50.0%	0.0%	50.0%	0.0%	0.0%	0.0%	50.0%	0.0%	0.0%	50.0%
White-23	33.3%	0.0%	0.0%	0.0%	66.7%	100.0%	0.0%	0.0%	0.0%	0.0%	149.925%
White-44	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%

TABLE 4. Pass distribution to teammates in percentage for each player in Team Blue (generated with Yolo).

Passer	Blue-0 H	Blue-0 Y	Blue-5 H	Blue-5 Y	Blue-7 H	Blue-7 Y	Blue-14 H	Blue-14 Y	Blue-34 H	Blue-34 Y	MAPE
Blue-0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	None
Blue-5	0.0%	0.0%	0.0%	0.0%	33.3%	0.0%	66.7%	100.0%	0.0%	0.0%	149.9%
Blue-7	0.0%	0.0%	25.0%	0.0%	0.0%	0.0%	25.0%	0.0%	50.0%	0.0%	300.0%
Blue-14	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%
Blue-34	0.0%	0.0%	50.0%	0.0%	50.0%	0.0%	0.0%	0.0%	0.0%	0.0%	200.0%

TABLE 5. Pass distribution to teammates in percentage for each player in team white (generated with Joy2019).

Passer	White-6 H	White-6 J	White-12 H	White-12 J	White-22 H	White-22 J	White-23 H	White-23 J	White-44 H	White-44 J	MAPE
White-6	0.0%	0.0%	0.0%	30.0%	50.0%	25.0%	50.0%	0.0%	0.0%	45.0%	75.0%
White-12	33.3%	53.9%	0.0%	0.0%	0.0%	23.0%	66.7%	7.6%	0.0%	15.4%	75.9%
White-22	100.0%	23.5%	0.0%	41.1%	0.0%	0.0%	0.0%	11.7%	0.0%	23.5%	76.5%
White-23	33.3%	0.0%	0.0%	28.5%	66.7%	42.8%	0.0%	0.0%	0.0%	28.5%	67.9%
White-44	0.0%	43.7%	0.0%	6.2%	100.0%	18.7%	0.0%	31.2%	0.0%	0.0%	81.3%

As shown in Figure 16 and 19, Yolo failed to derive pass relationships due to poor recognition of jersey numbers. In contrast, as shown in Figure 17 and 20, Joy2019 generated more edges than the human. This is because Joy2019 falsely identify players in between a ball passer and a ball receiver to be involved in the ball passing. For instance, Joy2019 can confuse the defender (the player in a white jersey) to be relaying the ball between the two players in blue jerseys (as shown in Figure 23). Since each video frames two-dimensional images, Joy2019 cannot fully capture the three-dimensional spatial information such as absolute coordinates of each player, which is critical for analyzing player's movement. We expect Joy2019 to perform better if depth information and video frameworks recorded at multiple angles are made available.

We have measured the difference between pass distributions to teammates in terms of absolute percentage error (APE) and mean percentage error (MAPE), as shown in Table 3, 4, 5 and 6. The result exhibits lower MAPE by Joy2019 compared to Yolo. We also compared the two algorithms in terms of normalized rank displacement of players, as shown in Table 7 and 8. The difference of the rank for each player is measured as the amount of displacement. For instance, Player W-12's ranking by Yolo is displaced by one rank whereas Joy2019 does not show any displacement for this player. Overall, as shown in Table 7 and 8, Yolo and Joy2019 showed similar displacement values.

As shown in Table 9 and 10, we ranked the players by their PC values. Overall Joy2019 showed lower MAPE in computing the PC-based rank compared to Yolo. Given the in-depth analysis of the various cases that hamper the recognition and tracking method, we plan to employ additional measures such as using multiple cameras, high-resolution videos and

TABLE 6. Pass distribution to teammates in percentage for each player in team blue.

Passer	Receiver										
	Blue-0		Blue-5		Blue-7		Blue-14		Blue-34		
	H	M	H	M	H	M	H	M	H	M	MAPE
Blue-0	0.0%	0.0%	0.0%	25.0%	0.0%	62.5%	0.0%	12.5%	0.0%	0.0%	None
Blue-5	0.0%	28.6%	0.0%	0.0%	33.3%	0.0%	66.7%	42.8%	0.0%	28.6%	67.9%
Blue-7	0.0%	71.4%	25.0%	14.3%	0.0%	0.0%	25.0%	14.3%	50.0%	0.0%	42.8%
Blue-14	0.0%	16.7%	0.0%	66.7%	100.0%	16.6%	0.0%	0.0%	0.0%	0.0%	83.4%
Blue-34	0.0%	0.0%	50.0%	33.3%	50.0%	0.0%	0.0%	66.7%	0.0%	0.0%	66.7%

TABLE 7. Normalized rank displacement of players. For each team, average normalized rank displacement (ANRD) is computed. Comparison between human (H) and Yolo (Y).

Team White				Team Blue			
Player	H	Y	Displacement	Player	H	Y	Displacement
W-6	1	1	0	B-0	5	2	3
W-12	4	3	1	B-5	3	3	0
W-22	2	2	0	B-7	1	5	4
W-23	3	5	2	B-14	2	1	1
W-44	5	4	1	B-34	4	4	0
ANRD			0.2				0.55

TABLE 8. Normalized rank displacement of players. For each team, average normalized rank displacement (ANRD) is computed. Comparison between human (H) and Joy2019 (J).

Team White				Team Blue			
Player	H	J	Displacement	Player	H	J	Displacement
W-6	1	1	0	B-0	5	2	3
W-12	4	4	0	B-5	3	1	2
W-22	2	2	0	B-7	1	5	4
W-23	3	5	2	B-14	2	3	1
W-44	5	3	2	B-34	4	4	0
ANRD			0.2				0.5

communication devices attached to players in order to make further improvement.

IV. RELATED WORK

In this section, we put our work in the context of related works.

Several general-purpose works used image processing techniques to identify objects from video streams. Gavrila and Philomin presented an efficient shape-based object detection method based on distance transforms and described its use for realtime vision on-board vehicles [29]. Nascimento et al. provided novel methods to evaluate the performance of object detection algorithms in video sequences [30]. W4S is a real time visual surveillance system for detecting and tracking people by integrating realtime stereo computation into an intensity-based detection and tracking algorithms [31]. While these works mainly used traditional image processing methods, approaches based on machine learning are becoming more prevalent recently. For instance, Wang and Yeung [32] trained a stacked de-noising autoencoder offline to learn generic image features from auxiliary natural images to get the trajectory of a moving object in a video with complex background. [19] introduced a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. VoxNet is a deep learning architecture to tackle the object detection problem by integrating a volumetric Occupancy Grid representation with a supervised 3D Convolutional Neural Network (3D CNN) [33]. VoxNet is used in various fields including robotics. In this work, we learned that we cannot utilize the off-the-shelf solutions out-of-the-box. For visual sports analytics, some customization is necessary. We introduce notable related works in the following.

With the help of deep learning, SportVu modeled the ball trajectory without a priori knowledge of physics [34]. SportVu recognizes ball movements from the images recorded and streamed at 16 different locations around the basketball court. SportVu also classifies offensive play calls with unique variants of neural networks [35]. In contrast, we used a publicly-available video clip showing dynamically shifting and changing camera angles. Despite the use of multiple cameras, SportVu is known to suffer from errors as well and requires engineers to be stationed on-site to manually revise the analysis results. Comparing the performance between our solution and such proprietary systems is not trivial. Nevertheless, it would be worthwhile to assess the trade-offs among cost, human efforts, resource requirements, accuracy, and speed. We have provided a solution that can be applied to video clips recorded with commodity cameras, which can appeal to sports teams that have limited hardware and human resources. More importantly, we transparently disclose the detailed reasons for imperfect player tracking especially under much poorer condition in order to prompt the research community to seek further enhancements. HawkEye is another commercial solution similar to SportVu, which is used in a number sports including baseball and tennis [36]. HawkEye specializes more in ball tracking than inferring relationships among players in

TABLE 9. Comparison between human and Yolo in terms of player rank values.

Team White				Team Blue			
Player	H	Y	MAPE	Player	H	Y	MAPE
W-6	0.10	0.25	144.5%	B-0	0.03	0.05	53.3%
W-12	0.05	0.17	188.2%	B-5	0.13	0.04	68.0%
W-22	0.08	0.21	143.6%	B-7	0.19	0.02	87.8%
W-23	0.06	0.02	61.9%	B-14	0.17	0.07	56.9%
W-44	0.02	0.02	4.1%	B-34	0.13	0.03	72.6%

TABLE 10. Comparison between human and Joy2019 in terms of player rank values.

Team White				Team Blue			
Player	H	M	MAPE	Player	H	M	MAPE
W-6	0.10	0.14	42.7%	B-0	0.03	0.11	210.2%
W-12	0.05	0.06	2.8%	B-5	0.13	0.11	12.4%
W-22	0.08	0.12	37.5%	B-7	0.19	0.07	60.7%
W-23	0.06	0.06	9.7%	B-14	0.17	0.10	37.1%
W-44	0.02	0.12	404.7%	B-34	0.13	0.08	35.3%

team sports. In Barlett et al.'s work, the football player's coordinates and movement patterns were analyzed with Prozone3 computer vision solution [37]. Granot et al. tracked the player's movements with hardware sensors instead of using computer visions [38]. It used triangulation with multiple sensors to locate the coordinates of players. Our solution relies on computer visions only and tracks relative movement of each player. Alahi et al. analyzed past player trajectories to predict the player's movements [39]. However, they did not analyze interactions between players. We compute relative importance of each player that is measured as a multivariate Eigen centrality value. In [40], SORT is developed based on Yolo v2.0 to detect players in realtime. However, SORT does not identify the jersey numbers and the player actions such as dribbling, interception, and passing. Jiang et al. analyzed beach volleyball games using multiple hypotheses tracking (MHT) method [41]. Our solution adapted Yolo that exhibits a higher mean average precision than MHT in detecting players. Lamas et al. analyzed the goalkeeper's decision making and how it determined team performance [42]. Sampaio et al. and Skinner et al. analyzed basketball team performance based on statistical player tracking data [43], [44]. However, it does not explain how it obtained player tracking data. Our work is keen on automating the process of tracking play actions seen on the stream of images. Lastly, Gerke et al. used convolutional neural networks to recognize soccer jersey numbers [45]. However, Gerke at al. focused only on classifying a manually cropped jersey numbers [45], whereas our solution attempts to detect the regions that contain jersey numbers and associate them with player objects to track their movements as well as their interactions with others.

V. CONCLUSION

In this work, we presented a novel system that automatically recognizes basketball players and their interaction with athletes on the floor such as passing and interceptions in realtime from a publicly-available NBA game footage that was used for broadcasting. We employed Yolo for object detection and classification. In addition, we implemented a novel player tracking algorithm that performance better than Yolo under the poor condition where camera angle dynamically shifts and changes. Our algorithm also handles the case where players overlap can hamper the correct tracking and identification of players due to the lack of absolute coordinate information on two-dimensional frames. We took advantage of movement history extracted from preceding video frames to recover any missing track information. We applied network science to analyze the passing relationship among players and assessed the importance of each player in terms of a newly defined multivariate Eigen centrality. Our solution have shown some shortcomings in terms of accuracy due to inherent limitation of current deep learning algorithms that are not entirely error-free. However, we gained valuable insights into the technical issues when attempting to construct a fully-automated sports analytics machine, and we envision such findings to motivate the community to enhance the system further. Besides the efforts to improve the accuracy, we plan to identify more player actions such as shoot attempts, scoring, rebound, and missed shots.

ACKNOWLEDGEMENT

We thank Juno Moon, Jiwon Oh, Won-Jun Jang, Myeong-Jin Lee, and Sung-Woo Yang for carrying out the painstaking tasks of collecting and extracting appropriate training data from thousands of video frames.

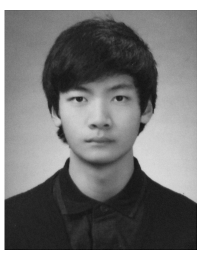
REFERENCES

[1] B. C. Alamar, *Sports Analytics: A Guide for Coaches, Managers, and Other Decision Makers*. New York, NY, USA: Columbia Univ. Press, 2013.
[2] E. Heichler. *Sports Analytics*. Accessed: May 2, 2019. [Online]. Available: https://sloanreview.mit.edu/tag/sports-analytics
[3] (Aug. 2018). *Sports Analytics*. [Online]. Available: https://en.wikipedia.org/wiki/Sports_analytics
[4] H.-A. Jacobsen *et al.*, "Grand challenge: The bluebay soccer monitoring engine," in *Proc. 7th ACM Int. Conf. Distrib. Event-Based Syst.* New York, NY, USA: ACM, 2013, pp. 295–300.
[5] B. Gonçalves, D. Coutinho, S. Santos, C. Lago-Penas, S. Jiménez, and J. Sampaio, "Exploring team passing networks and player movement dynamics in youth association football," *PloS One*, vol. 12, no. 1, 2017, Art. no. e0171156.
[6] J. Duch, J. S. Waitzman, and L. A. N. Amaral, "Quantifying the performance of individual players in a team activity," *PloS One*, vol. 5, no. 6, 2010, Art. no. e10937.
[7] D. Lusher, G. Robins, and P. Kremer, "The application of social network analysis to team sports," *Meas. Phys. Edu. Exerc. Sci.*, vol. 14, no. 4, pp. 211–224, 2010.
[8] T. U. Grund, "Network structure and team performance: The case of English Premier League soccer teams," *Social Netw.*, vol. 34, no. 4, pp. 682–690, 2012.
[9] J. H. Fewell, D. Armbruster, J. Ingraham, A. Petersen, and J. S. Waters, "Basketball teams as strategic networks," *PloS One*, vol. 7, no. 11, 2012, Art. no. e47445.

[10] Y. Yamamoto and K. Yokoyama, "Common and unique network dynamics in football games," *PloS One*, vol. 6, no. 12, 2011, Art. no. e29638.
[11] P. Halvorsen *et al.*, "Bagadus: An integrated system for arena sports analytics: A soccer case study," in *Proc. 4th ACM Multimedia Syst. Conf.* New York, NY, USA: ACM, 2013, pp. 48–59.
[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[13] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[14] A.-L. Barabási *et al.*, *Network Science*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[15] J. Redmon. (2013). *Darknet: Open Source Neural Networks in C*. [Online]. Available: http://pjreddie.com/darknet
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.
[18] J. Redmon and A. Farhadi. (2018). "YOLOv3: An incremental improvement." [Online]. Available: https://arxiv.org/abs/1804.02767
[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[20] D. Lazer, D. Brewer, N. Christakis, J. Fowler, and G. King, "Life in the network: The coming age of computational social," *Science*, vol. 323, no. 5915, pp. 721–723, 2009.
[21] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Nat. Acad. Sci.*, vol. 106, no. 36, pp. 15274–15278, 2009.
[22] L. A. Meyers, "Contact network epidemiology: Bond percolation applied to infectious disease prediction and control," *Bull. Amer. Math. Soc.*, vol. 44, no. 1, pp. 63–86, 2007.
[23] A.-L. Barabási and Z. N. Oltvai, "Network biology: Understanding the cell's functional organization," *Nature Rev. Genet.*, vol. 5, no. 2, pp. 101–113, 2004.
[24] Y. Choi and Y. Yoon, "Moscato: A novel management tool for large-scale data and event processing platforms," in *Proc. 12th ACM Int. Conf. Distrib. Event-Based Syst.* New York, NY, USA: ACM, 2018, pp. 248–249.
[25] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[26] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 740–755.
[27] N. Q. Zhu, *Data Visualization With D3.js Cookbook*. Birmingham, U.K.: Packt, 2013.
[28] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw.*, vol. 56, no. 18, pp. 3825–3833, 2012.
[29] D. M. Gavrila and V. Philomin, "Real-time object detection for 'smart' vehicles," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, Sep. 1999, pp. 87–93.
[30] J. C. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 761–774, Aug. 2006.
[31] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4S: A real-time system for detecting and tracking people in 2 1/2D," in *Proc. Eur. Conf. Comput. Vis.* Freiburg, Germany: Springer, 1998, pp. 877–892.
[32] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.
[33] D. Maturana and S. Scherer, "Voxnet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.
[34] R. Shah and R. Romijnders. (2016). "Applying deep learning to basketball trajectories." [Online]. Available: https://arxiv.org/abs/1608.03793
[35] K.-C. Wang and R. Zemel, "Classifying NBA offensive plays using neural networks," in *Proc. MIT Sloan Sports Anal. Conf.*, 2016, pp. 1–9.
[36] S. Gangal and S. Raje, "The hawkeye technology," Department of Computer Science and Engineering, Tamil Nadu, Indian, Tech. Rep., 2007.
[37] R. Bartlett, C. Button, M. Robins, A. Dutt-Mazumder, and G. Kennedy, "Analysing team coordination patterns from player movement trajectories in soccer: Methodological considerations," *Int. J. Perform. Anal. Sport*, vol. 12, no. 2, pp. 398–424, 2012.
[38] Y. Granot, M. Tamir, and A. Sharir, "Tracking system for sports," U.S. Patent 6 567 038 B1, May 20, 2003.
[39] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 961–971.
[40] D. Acuna, "Towards real-time detection and tracking of basketball players using deep neural networks," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*. Long Beach, CA, USA: Google Scholar, 2017, pp. 1–6.
[41] X. Jiang, Z. Liu, and Y. Wang, "Tracking multiple players in beach volleyball videos," in *Proc. Chin. Conf. Intell. Vis. Surveill.* Beijing, China: Springer, 2016, pp. 65–71.
[42] L. Lamas, R. Drezner, G. Otranto, and J. Barrera, "Analytic method for evaluating players' decisions in team sports: Applications to the soccer goalkeeper," *PloS One*, vol. 13, no. 2, 2018, Art. no. e0191431.
[43] J. Sampaio, T. McGarry, J. Calleja-González, S. J. Sáiz, X. S. I del Alcázar, and M. Balciunas, "Exploring game performance in the national basketball association using player tracking data," *PloS One*, vol. 10, no. 7, 2015, Art. no. e0132894.
[44] B. Skinner and S. J. Guy, "A method for using player tracking data in basketball to learn player skills and predict team performance," *PloS One*, vol. 10, no. 9, 2015, Art. no. e0136393.
[45] S. Gerke, K. Muller, and R. Schafer, "Soccer jersey number recognition using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 17–24.

YOUNG YOON received the B.A. and M.S. degrees in computer sciences from The University of Texas at Austin, in 2003 and 2006, respectively, and the Ph.D. degree in computer engineering from the University of Toronto, in 2013. He is currently an Assistant Professor in computer engineering with Hongik University, Seoul, South Korea. Before joining Hongik University, he obtained various technical and research experiences with Samsung Electronics, the IBM T. J. Watson Research Center, Platform Computing, and Telus. He is the Director of the research position with NetcoreTech Inc. He currently conducts various research projects on distributed computing and artificial intelligence applications.

HEESU HWANG (M'87) received the B.S. and M.S. degrees from the Department of Materials Science and Engineering, Hongik University, Seoul, South Korea, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in material science and engineering. His main research interests include smart materials/systems, network science, and machine learning and their applications to materials science and engineering.



YONGJUN CHOI received the bachelor's and master's degrees in computer engineering from Hongik University, in 2016 and 2019, respectively. He is currently a Software Engineer with Wins and a former Research Assistant with the Application Platform Lab, Hongik University. He has conducted multiple rounds of internships with NetcoreTech Inc., Seoul, South Korea. At NetcoreTech Inc., he has contributed to the development of a large-scale distributed event-processing platform that is used by the Korean Ministry of Education for the realtime monitoring of various security issues with national education systems.

MINBEOM JOO received the bachelor's degree in computer engineering from the Department of Computer Engineering, Hongik University. He is currently a Research Engineer and the Team Lead with Cube AI Inc. Before joining Cube AI Inc., he has completed software engineering internships at NCSoft and Nexon, in 2017 and 2018, respectively. He is interested in computer visions, autonomous vehicles, and game developments.

HYEYOON OH received the bachelor's degree in computer engineering from the Department of Computer Engineering, Hongik University, in 2019. She is currently a Research Staff Member with Cube AI Inc., and a Former Undergraduate Research Assistant with Hongik University. She took a leave of absence from school, in 2017, and led Cross IT, an IT club, as a Developer and the President. Her research interests include the state-of-the-art technologies in artificial intelligence and software system integration.

INSUN PARK received the bachelor's degree in computer engineering from Hongik University, Seoul, South Korea, in 2019, where she was a former Research Assistant. From 2017 to 2018, she was an Intern with ALUX, a company that provides education contents for programming. During the internship, she has created mobile application course materials and wrote various technical documents. She is interested in targeted advertisements and the management of crowd-sourcing platforms.

KEON-HEE LEE is currently pursuing the Department of Materials Science and Engineering, Hongik University, where he is a Research Assistant. His research interests include oxide thin film transistors and the applications of deep learning to materials science.

JIN-HA HWANG received the B.S. and M.S. degrees in inorganic materials engineering from Seoul National University, in 1988 and 1990, respectively, and the M.S. and Ph.D. degrees in materials science and engineering from Northwestern University, in 1992 and 1996, respectively. He is currently a Professor in materials science and engineering with Hongik University, Seoul, South Korea. Before joining Hongik University, he has obtained academic and industrial research experiences in defect chemistry, transparent electronics, and low-temperature polycrystalline Si TFT technologies with Northwestern University and the Samsung SDI Central Research Center. His interests include 3D printing materials, smart materials, nano-electronics, and machine learning.