INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Improving Badminton Player Detection using YOLOv3 with Various Training Heuristics

Muhammad Abdul Haq [a,*], Norio Tagawa [a]

[a] *Department of Electrical Engineering and Computer Science, Tokyo Metropolitan University, Hachioji-shi, Tokyo, Japan*
Corresponding author: [*]muhabdulhaq@gmail.com

***Abstract*—There has been a considerable rise in the amount of research and development focused on computer vision over the previous two decades. One of the most critical processes in computer vision is "visual tracking," which involves following objects with a camera. Tracking objects is the practice of following an individual moving object or group of moving things over time. Identifying or connecting target elements in consecutive video frames of a badminton match requires visual object tracking. The aim of this study is to identify badminton players using the You Only Look Once (YOLO) technique in conjunction with a variety of training heuristics. This methodology has a few advantages over other approaches to detecting objects. The convolutional neural network and Fast convolutional neural network are two examples of the many algorithmic approaches that are available. In this study, a neural network is used to produce predictions about the bounding boxes and the class probabilities for these boxes.. The results demonstrated that it was far faster than other methods in terms of its ability to recognize the image. The performance of image classification networks significantly improved as a result of the implementation of a variety of training strategies for the detection of objects. The mean average precision score for YOLOv3 with various training heuristics increased from 32.0 to 36.0 as a direct result of these adjustments. In comparison to YOLOv3, our future study might examine the performance of alternative models like Faster R-CNN or RetinaNet.**

***Keywords*— Multiple object tracking; convolutional neural network; various training heuristics.**

Manuscript received 3 Sep. 2022; revised 27 Oct. 2022; accepted 10 Nov. 2022. Date of publication 30 Jun. 2023.

I. INTRODUCTION

The development of computer programs capable of performing tasks that normally require some amount of intelligence is one of the most significant aspects of the branch of computer science known as artificial intelligence (AI); that is, the goal is to create robots that can think and act like people and make their own decisions. The field of computer vision is a subfield of AI that concentrates on the processing of visual data, such as images and video files, in a manner that is analogous to how people perform this task, glean information from it, and interpret the data's subject matter. The field of computer vision is always growing because a large quantity of visual data is being acquired from many sources, such as security systems, traffic cameras, and routine uploads that users make to the internet. More than three billion images are published on social media platforms daily, with Facebook and Instagram leading the pack. Additionally, hundreds of hours of new video content are added every day on YouTube, possibly the most popular online video search engine. Using these data, it is possible to create a variety of datasets for use in computer vision implementations. The exponential growth of computer vision caused by a large quantity of visual data may be ascribed to improvements in hardware, which has resulted in increased processing power, in addition to open-source machine learning approaches that are openly accessible and have been validated in various applications [1].

In some manner, the life of every person is centered around sports. Everyone has a good time participating in or watching sports and video games. Neural networks can function with large amounts of data in addition to small amounts of data or difficult data. In machine learning, there is a concept called a model, which is a file that has been trained to recognize certain types of patterns. Machine learning technique is different in the two models because they are like the human brain in certain respects. It has a very high neuronal density, which allows it to process and transmit information [2].

Object detection is a technique for identifying significant subjects inside digital still images and moving pictures. In reality, object detection is a computer system that is responsible for the creation and exploration of virtual

environments [3]–[5]. One application in real time is player detection during badminton matches using machine learning. According to several specialists' opinions, deep learning is the most promising form of machine learning [6]. In badminton competitions, the aim is to single out one player for recognition. Within the context of such a program, the player is the thing that can be found the most frequently. Most of a sports analytics presentation's audience often consists of community organizations and coaching staff [7]. Researchers are increasingly turning to video-based activity detection because of technological developments that enable live sports broadcasts to be streamed online [8]. Because real-time systems require significant object localization, we use object localization to determine the identity of objects. Both classification-based algorithms and methods that do not rely on classification can be used to identify things. When compared with the detection of pedestrians, the recognition of players is a greater challenge because of the body deformations and camera movements involved [9]. For example, badminton players are required to twist their bodies, jump to great heights, and stretch their hands. Traditional deep neural networks also tend to ignore smaller players by pixel size because their activation is lost toward the end of a deep network. This is because player heights (i.e., their bounding boxes) can range from a few pixels to hundreds of pixels. One of the most interesting aspects of team sports is the way in which the players in each squad must work together to compete against the opposing squad. An example of a challenging scenario is that the badminton net may block the player's view of the opponent. Therefore, player identification and labeling is needed in sport analysis. Because this is a functional requirement, we were compelled to devise a standardized method for player identification and player labeling categorization.

Fig. 1 Badminton broadcast video used for testing in this study. Badminton player position data are crucial for an evaluation by the badminton coach.

To achieve this, we must first locate the relevant portions of the frame and then classify them using a convolutional neural network (CNN). This approach takes a large amount of time because we must predict every location. Regression-based algorithms are included in this neural network. Thus, the You Only Look Once (YOLO) philosophy becomes relevant. In this scenario, we do not choose the parts of the frame that interest us. Instead, in a single application of the approach, we use a single neural network to identify many objects, and forecast the classes and bounding boxes of the entire image. In comparison with other classification algorithms, the YOLOv3 method is relatively quick [10]. The YOLOv3 algorithm has certain localization issues; however, it also predicts fewer false positives than other algorithms.

The "object detection" application is dynamic. The fundamental problem with the present system of region-based CNNs (R-CNNs) is that, for detection, we must categorize many areas. The current system is obsolete in terms of item detection and resource use. The requirements for storing the feature map of the region suggestions are significantly bigger than the alternatives available. As a result, training takes a long time. The system must be replaced with a better system to eliminate all these restrictions and allow it to be implemented in badminton matches.

All existing techniques for detecting objects rely on using regions to locate the object within the image. The network pays attention to only a portion of the image. Hence, it is necessary to concentrate efforts on the parts of the image that have the best chance of containing the object. Region-based object identification techniques are quite different from YOLO, but their structure and implementation are straightforward to understand. The system makes only a moderate demand on the available system resources and is adaptable to virtually any environment [11]. We determine the application's operation using YOLOv3 object detection and the OpenCV library. A researcher proposed an inference environment with minimal overhead as a set of training changes that dramatically enhanced model performance [12].

The remainder of this paper is structured as follows: In Section 2, we present the literature on CNNs with various heuristic training approaches and the method used to attain the results. In Section 3, we present the results of this study. Finally, in Section 4, we present the conclusion of this study.

II. Materials and Method

In this section, we examine some pertinent studies, in addition to the contributions they made. The study of video abstraction or summary has garnered the interest and resources of many academics. The production of sports video highlights is an example of a subcategory of video summarizing. In several studies, researchers investigated various approaches to extract highlights from sports footage [13]. Some researchers concentrated on following the players in a particular sports game, such as developing a comprehensive player tracking system for basketball videos [14]–[21]. For many applications, player recognition from photos and videos is crucial [22]. For instance, intelligent broadcast systems take advantage of player positions to direct the perspectives of broadcast cameras [23]–[25]. Player detection, which falls under the umbrella of pedestrian detection, has received a significant amount of attention from researchers. A method such as subtracting the backdrop is one example [26].

According to the literature and data, academic studies on identifying moving targets have progressively focused on the detection of multi-target, intricate motion, and complex object tracking. CNNs have been more prevalent during the last several years in the field of player tracking research. CNN is also particularly well suited to monitoring players with advanced results similiar to action detection[27].

A CNN is a subtype of deep neural networks that was developed primarily for image processing and object identification[28]. It is possible to use CNN algorithms

without requiring many established significant parameters. As a result of the abundance of data and simplicity with which models can be trained, CNN algorithms have become a reality. CNN algorithms, which are solely based on mathematical principles, express and extract the properties of incoming data. This system uses a weight-sharing mechanism to determine and identify information that has comparable features. This method enables networks to process enormous amounts of data and produce precise classifications at the end of the process. The processing power and parameter scope in datasets that are now available present a clear obstacle to the process of improving results using CNN models. These are two of the most important aspects of the models.

There are some layers in a typical CNN structure: one for input data, one for convolution, one for activation[29], one for Batch Normalization, one for pooling, and one for output data. Additional layers can be added to some CNN models to accomplish a variety of goals. A multi-layered architecture that makes use of forward pass and error backpropagation calculations allows for the proficiency of the target to be attained. For this architecture to be taught to become a model in a certain manner, images and the labels that correspond to them are required. Following the completion of training, the most acceptable weights for use during testing are established. It is possible to describe these levels further as follows.

The image data are initialized in the input layer and all possible dimensions are zero-centered before the layer is used. Additionally, this layer oversees the process of lowering the size of all incoming data to a value between zero and one, which helps to speed up the convergence process. Whitening the data is another benefit of this normalization.

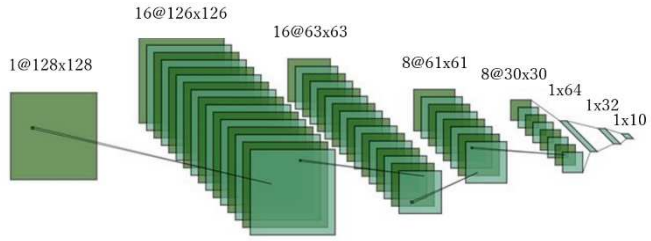

Fig. 2 Convolutional neural network has many layers, including convolution and max pooling.

Because it functions as a layer, the CNN's convolutional layer is the most significant structural layer. Using several element maps and many neurons inside each of them, each neuron is constructed to decipher the nearby characteristics of various locations in the layer underneath it. When a user uploads an image into the conv kernel, the original image slides across the filter's surface, which results in the creation of a new image. The local correlated data values of each pixel are multiplied and added together by the convolution kernel before it performs the convolution to compute the picture's component portrayal. Because of the so-called "law of convolution," the conv kernel may be used to derive the image's characteristics.

The weights being the same across all the filtered areas of a picture is the primary justification for using the same conv kernel for all filtering. By using the same weights in this manner, it is possible to identify neutral cells with comparable qualities and place them into the same item category. A variety of kernel and filter-related parameters may be entered into this.

Underfitting results in a fading gradient, which the active layer works to fix. The underfitting and nonlinear problem may be traced back to the convolutional layer that came before it. Even though the rectified linear unit function is preferred because of its convergence speed, the sigmoid and tanh functions are still used extensively because of the ease and efficacy with which they may be calculated. The goal of the convolutional layer is to reduce the amount of data that is sent into the pooling layer as efficiently and effectively as is feasible. When the outputs of several neurons in one layer are merged into a single neuron in the next layer, the number of items in the component maps decreases and the strength of the selected extractions increases. There are three unique pooling layers, that is, overlapping pooling, general pooling, and spatial pyramid pooling, which are all located between two convolutional layers. The width of a pooling layer must at least be equivalent to its stride length for the layer to be referred to as "universal." Maximum pooling and average pooling are the two most frequently used variations of the general pooling method. The term "max pooling" refers to using the most severe rewards available to each set of neurons in the preceding layer. Average pooling computes the average of the elements present in the filter's feature map region. Thus, while max pooling returns the most prominent feature in a given patch of the feature map, average pooling returns the average of all features in that patch. It is possible to obtain aberrant state features from the input layer using two convolutional layers in combination with a final pooling layer.

The fully connected layer acts as a conduit for the flow of data to and from the output layer, which makes it a vital component of the construction of a CNN. Using each neuron from the layer below it and linking it to each neuron on its own, the process of computing data may be made more straightforward and accomplished more quickly than without linking each neuron. Because an entirely related layer is always followed by a yield layer, none of the geographical data entered is stored.

In addition to the layers required to construct a CNN model, specific CNN models require extra layers to provide the desired results. Among these layers are those that execute dropout and regression. By updating the weights of the neural cell knot with a given frequency, dropout layers are often used to resolve overfitting by avoiding substantially subjective weights (which is decided by the stochastic policy). By contrast, the regression layer is used to classify features using methods such as logistic regression, Bayesian linear regression, and Gaussian processes for regression. Using a regression layer, the probabilities for any object can be obtained.

Convolutional layers are responsible for extracting information, whereas fully connected layers calculate predictions and probabilities for the bounding boxes. Because bounding box predictions and class probabilities are associated with grid cells, the center cell is used to convey the forecast for an item that spans many cells. In the training phase, the associated confidence value for a bounding box prediction is zero if no item is present. If there is an item in the scene and detection is anticipated, the confidence value may be calculated as the intersection-over-union (IoU) score

of the boxes containing the prediction and ground truth. Although YOLOv3 has been a popular and quick object identification technique, it is not the best. YOLOv3's performance can be improved by implementing a few basic training strategies and architectural tweaks. With the use of enhanced training processes, image classification networks' performance has significantly improved [30]. We describe each of these heuristics individually to better explain their basic concept.

A. Image Mixup for Object Detection

Picture mixup is simply the linear interpolation of the pixels of two pictures in image classification networks. The mixup approach creates its distribution of blending ratios for picture classification from a beta distribution using the single blending ratio. However, this would require resizing all the bounding boxes of objects in the photos to accomplish the mixup, which is not always possible. As a result, the picture is more aesthetically appealing and has a higher mAP score than it would without the application of picture mixup.

B. Data Preprocessing

Data preprocessing seems to enhance object detection models, although it appears to have a more significant impact on single-stage detectors than multi-stage models. In multi-stage detectors, such as Faster R-CNN, detection results are produced by repeatedly clipping important areas in feature maps, where a specified number of candidate object suggestions are sampled from a vast pool of generated ROIs. Multi-stage models can replace the random cropping of input pictures because of this cropping action. Hence, these networks do not require significant geometric augmentation during training.

C. Rate Scheduler

Most object detection networks use a learning rate scheduler (e.g., Fast R-CNN, YOLO). Therefore, because of the sudden change in the learning rate, subsequent iterations may be more stable. The validation accuracy may be improved even further using a cosine scheduler with a good warmup, as illustrated in the example below.

D. Synchronized Batch Normalization

Batch normalization is a critical component of contemporary deep convolutional designs. However, before the batch normalization process, there is a process called label smoothing. The encoded labels are converted to a smooth probability distribution using

$$q_i = \begin{cases} 1 - \varepsilon & if\ i = y, \\ \varepsilon/(\mathrm{K} - 1) & otherwise, \end{cases} \qquad (1)$$

where K is the number of classes, ε is a tiny constant, and q is the distribution of ground-truth values. Regularizer is obtained by lowering the model's confidence. By normalizing hidden layer activations, the training process is speeded up and makes the network less sensitive to weight initialization. Only so many candidate object proposals can fit on a single GPU when dealing with substantial input images, feature pyramid designs, and a high number of candidate object proposals. The hidden activations are normalized inside each GPU in the distributed training paradigm. As a result, inaccurate mean and variance values are calculated, which impedes the overall batch normalization process.

E. Detecting the Object

In this section, we describe the player detection system and an essential component of the deep learning-based model proposed in this paper; the difficulties that arise during badminton video analysis; the concepts of multiple object detection and tracking, which serve as the cornerstone of our study; and, finally, a variety of performance metrics that can be applied to the generation of player analytics.

Visual object tracking in movies is accomplished using YOLO and SORT for object recognition and tracking. Python is used for the full implementation. This approach is used in a one-stage network architecture, as shown in Figure 2, to anticipate class probabilities and match bounding boxes in a single step.

Fig. 3 Example image of a badminton match that we used for evaluation.

Fig. 4 This is a challenging scenario for image processing because the players are positioned closely; hence, there will be an error detection.

When an item fills more than one grid cell, the prediction for that object is maintained for the center cell because grid cells are tied to bounding box forecasts and class probabilities in this manner. The equivalent confidence value for a prediction that no item is present is set to zero during the training phase while bounding box predictions are being produced. When an object is known to be in the vicinity and its detection is predicted, the confidence value is derived using the IoU score of the prediction and ground-truth boxes.

The video used for this experiment contained a number of women's singles matches, for example, in Figure 3, An Se Yong from South Korea versus Akane Yamaguchi from Japan. Additionally, we also conducted experiments using other matches. Figure 4 shows that the players are in positions that

are close. This is a challenge for the detector to detect because the objects are very close together.

Regarding making predictions for the bounding boxes of all objects, YOLOv3 just uses a single neural network. YOLOv3 can also accurately predict the boundaries of each video frame in real time, which is a significant improvement over the previous version. The fundamental purpose of YOLOv3 is to divide the video frame into an S × S grid. If the object's center is inside one of the grid cells, then the grid cells will detect the item.

A confidence score is assigned to each cell to establish how probable it is that an item will be located inside the bounding box. This may be performed by computing the score. A confidence score is assigned to each cell to obtain an idea of how probable it is that a particular item will be discovered inside the bounding box. The x, y, w, and h predictions and the confidence score are all included in each bounding box. For example, the top left corner (x, y) and its width and height (wh) are represented by the coordinates (x, y).

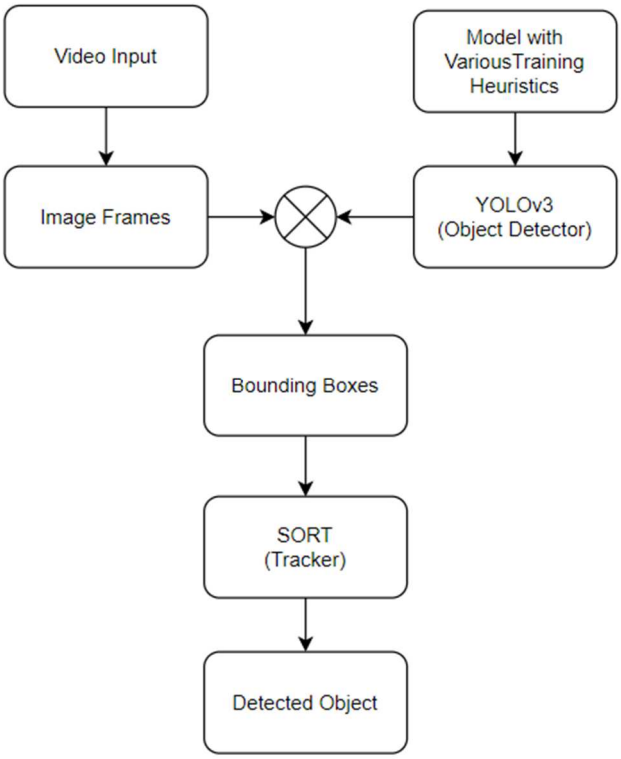

Fig. 5 Overall flow chart of the working system from input to output.

Confidence is measured by the IoU. To locate people inside video frames, the YOLOv3 algorithm is applied during the processing of the frames. A bounding box appropriate for person detection is produced as a direct consequence of following these procedures in the appropriate order. The important point of this study is the use of various training heuristic methods for the model: image mixup, data preprocessing, learning rate scheduler, and synchronized batch normalization. YOLOv3's mAP score increased because of these changes, with no more computation required during inference and just a little increase in analysis required during training.

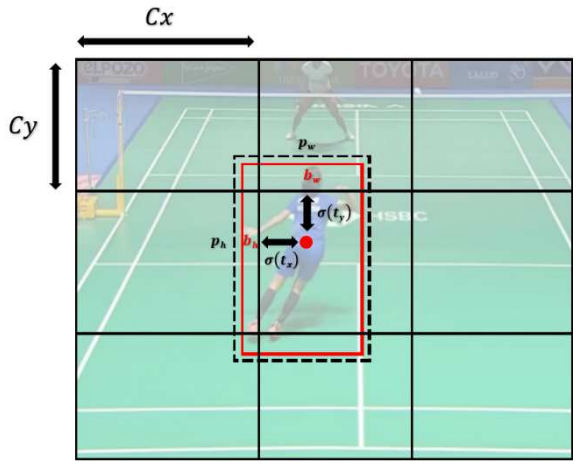

Fig. 6 This is how YOLOv3 works. It uses the width and height of the identified object.

When training the YOLOv3 model using the standard technique, it is feasible that overfitting can be avoided by training the model on several scales in the same manner in which it is trained using the standard YOLOv3 training method. . The entire performance of YOLOv3 has been significantly improved as a direct consequence of these enhancements that have had an effect, which has led to the accomplishment of a significant step forward.

III. Results and Discussion

The YOLO algorithm was run after a picture was captured, which came first in the process. In this particular instance, the image was divided into grid matrices. The picture's complexity level directed the number of grids that were used. Following image segmentation, the object was categorized before it was positioned within each grid. The objectivity of each grid, in addition to its level of trustworthiness, was evaluated. If no valid object could be found within the grid, both the objectness and the bounding box value of the grid were set to zero. If an object could be located within the grid, the objectness was set to one and the bounding box value was the identified item's matching bounding values. More details are provided about the bounding box prediction in the following. Anchor boxes were also used to improve object detection.

Every time it was essential to extract relatively accurate bounding boxes from a picture, the YOLOv3 method was applied. Each image grid was subjected to picture categorization and object localization procedures, and when the process was complete, a label was assigned to each grid. The algorithm then proceeded to iterate over each grid in order, marking the labels that included objects, in addition to the bounding boxes for those items.

The evaluation of the suggested approach was performed in extensive detail throughout numerous videos. After that stage, object recognition in the video clips was performed using weight files. After an object inside a video was recognized by a trained object detector, the information on the item's bounding box was passed to the SORT tracking algorithm and object tracking was performed.

Fig. 7 Badminton player was successfully detected and tracked.

The video was divided into frames for the purposes of visual object recognition and tracking. Information regarding the detection and tracking of visible objects was received for each input video after YOLOv3 and SORT were applied for object detection and tracking. This information was stored for each frame, in addition to the video output. It was possible to significantly improve the YOLOv3 baseline using training approaches that focused on object detection at its most fundamental level. The screenshot in Figure 7 is an illustration of one possible output screen from a video test.

Fig. 8 Badminton player detection was tracked stably from the first frame until last frame

For these solutions to be effective, only minor architectural adjustments and straightforward integration were necessary. Using the training principles, additional adjustments were made to the pre-trained YOLOv3 models. The higher speed-accuracy ratio of YOLOv3 was undeniably more advantageous. Figures 7 and 8 show how this methodology was used to track badminton players.

The video used for testing was a badminton match with a single match type obtained from the Badminton World Federation YouTube channel. Table I shows that the video used for this study consisted of two games from a women's single (WS) match and three games from a men's single (MS) match.

TABLE I
VIDEO TESTED

No	Players (Win) vs (Lost)	Round Stage
1	TAI Tzu Ying vs Kirsty GILMOUR	WS, Round 16, First game
2	TAI Tzu Ying vs Evgeniya KOSETSKAYA	WS, Round 16, First game
3	Ygor COELHO vs Maxime MOREELS	MS, Round 64, First game
4	Ygor COELHO vs Maxime MOREELS	MS, Round 64, Second game
5	Rasmus GEMKE vs Brice LEVERDEZ	MS, Round 64, First game

TABLE II
COMPARISON OF VARIOUS HEURISTICS WITH THE ORIGINAL YOLOV3

Model	Original mAP	Using Various Heuristics	Improvement
YOLOv3-320	29	33	+4.0
YOLOv3-416	32	36	+4.0
YOLOv3-608	34	37	+3.0

Table II shows that the badminton players could be tracked with a mAP score of 33.0 using YOLOv3-320, 36.0 using YOLOv3-416, and 37.0 using YOLOv3-608. Thus, the methodology can be applied to identify badminton players.

IV. CONCLUSION

In this study, we effectively constructed the YOLOv3 detector using a variety of training heuristics. Even when other things were in the way, the model could still identify the badminton player. Because of these modifications, YOLOv3's mAP score improved. Although no more computing was necessary during inference, there was a little increase in the amount of analysis that was required during training. As a result of its adaptability to a wide variety of video formats and domains, the system has the potential to be trained in the not-too-distant future to detect and track a wide variety of objects. It is possible to install and test several object detectors and trackers to accomplish the specified object detection and tracking, and as a result, a number of different results will be gathered that may be assessed in a future investigation.

ACKNOWLEDGMENT

We thank Edanz (https://jp.edanz.com/ac) for editing a draft of this manuscript.

REFERENCES

[1] K. Host and M. Ivašić-Kos, "An overview of Human Action Recognition in sports based on Computer Vision," *Heliyon*, vol. 8, no. 6, p. e09633, Jun. 2022, doi: 10.1016/J.HELIYON.2022.E09633.

[2] K. Joshi, V. Tripathi, C. Bose, and C. Bhardwaj, "Robust Sports Image Classification Using InceptionV3 and Neural Networks," *Procedia Comput Sci*, vol. 167, pp. 2374–2381, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.290.

[3] N. E. Miner, "Interactive virtual reality simulation system for robot control and operator training," *Proc IEEE Int Conf Robot Autom*, no. pt 2, pp. 1428–1435, 1994, doi: 10.1109/robot.1994.351289.

[4] S. A. Stansfield, "A Distributed Virtual Reality Simulation System for Situational Training," *Presence: Teleoperators and Virtual Environments*, vol. 3, no. 4, pp. 360–366, Nov. 1994, doi: 10.1162/PRES.1994.3.4.360.

[5] S. Li and J. Sun, "Application of virtual reality technology in the field of sport," *Proceedings of the 1st International Workshop on Education Technology and Computer Science, ETCS 2009*, vol. 2, pp. 455–458, 2009, doi: 10.1109/ETCS.2009.363.

[6] N. A. Rahmad, N. A. J. Sufri, N. H. Muzamil, and M. A. As'ari, "Badminton player detection using faster region convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1330–1335, Jun. 2019, doi: 10.11591/IJEECS.V14.I3.PP1330-1335.

[7] W.-Y. Wang, H.-H. Shuai, K.-S. Chang, and W.-C. Peng, "ShuttleNet: Position-aware Fusion of Rally Progress and Player Styles for Stroke

Forecasting in Badminton," Dec. 2021, doi: 10.48550/arxiv.2112.01044.

[8] M. Firdhaus *et al.*, "The new Convolutional Neural Network (CNN) local feature extractor for automated badminton action recognition on vision based data," *J Phys Conf Ser*, vol. 1529, no. 2, p. 022021, Apr. 2020, doi: 10.1088/1742-6596/1529/2/022021.

[9] M. Manafifard, H. Ebadi, and H. Abrishami Moghaddam, "A survey on player tracking in soccer videos," *Computer Vision and Image Understanding*, vol. 159, pp. 19–46, Jun. 2017, doi: 10.1016/J.CVIU.2017.02.002.

[10] B. Thulasya Naik, M. Farukh Hashmi, C. Author, and M. Farukh Hashmi mdfarukh, "Ball and Player Detection & Tracking in Soccer Videos Using Improved YOLOV3 Model," 2021, doi: 10.21203/rs.3.rs-438886/v1.

[11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018, doi: 10.48550/arxiv.1804.02767.

[12] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Freebies for Training Object Detection Neural Networks," Feb. 2019, doi: 10.48550/arxiv.1902.04103.

[13] A. A. Khan and J. Shao, "SPNet: A deep network for broadcast sports video highlight generation," *Computers and Electrical Engineering*, vol. 99, p. 107779, Apr. 2022, doi: 10.1016/J.COMPELECENG.2022.107779.

[14] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, "Multi-camera multi-player tracking with deep player identification in sports video," *Pattern Recognit*, vol. 102, Jun. 2020, doi: 10.1016/J.PATCOG.2020.107260.

[15] G. Quanan and X. Yunjian, "Kalman Filter Algorithm for Sports Video Moving Target Tracking," *Proceedings - 2020 International Conference on Advance in Ambient Computing and Intelligence, ICAACI 2020*, pp. 26–30, Sep. 2020, doi: 10.1109/ICAACI50733.2020.00010.

[16] H. Kim and K. S. Hong, "Soccer video mosaicing using self-calibration and line tracking," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, pp. 592–595 vol.1. doi: 10.1109/ICPR.2000.905407.

[17] M. Archana and M. K. Geetha, "Object Detection and Tracking Based on Trajectory in Broadcast Tennis Video," *Procedia Comput Sci*, vol. 58, pp. 225–232, Jan. 2015, doi: 10.1016/J.PROCS.2015.08.060.

[18] T. Watanabe, M. Haseyama, and H. Kitajima, "A soccer field tracking method with wire frame model from TV images," in *2004 International Conference on Image Processing, 2004. ICIP '04.*, 2004, pp. 1633-1636 Vol. 3. doi: 10.1109/ICIP.2004.1421382.

[19] Y. Lyu and S. Zhang, "Badminton Path Tracking Algorithm Based on Computer Vision and Ball Speed Analysis," *J Sens*, vol. 2021, 2021, doi: 10.1155/2021/3803387.

[20] S. Yang, F. Ding, P. Li, and S. Hu, "Distributed multi-camera multi-target association for real-time tracking," *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1–13, Jun. 2022, doi: 10.1038/s41598-022-15000-4.

[21] A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games," in *2002 International Conference on Pattern Recognition*, 2002, pp. 303–306 vol.1. doi: 10.1109/ICPR.2002.1044697.

[22] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, "Computer vision for sports: Current applications and research topics," *Computer Vision and Image Understanding*, vol. 159, pp. 3–18, Jun. 2017, doi: 10.1016/J.CVIU.2017.04.011.

[23] J. Chen and J. J. Little, "Where should cameras look at soccer games: Improving smoothness using the overlapped hidden Markov model," *Computer Vision and Image Understanding*, vol. 159, pp. 59–73, Jun. 2017, doi: 10.1016/J.CVIU.2016.10.017.

[24] J. Chen, H. M. Le, P. Carr, Y. Yue, and J. J. Little, "Learning online smooth predictors for realtime camera planning using recurrent decision trees," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 4688–4696, Dec. 2016, doi: 10.1109/CVPR.2016.507.

[25] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A Hierarchical Deep Temporal Model for Group Activity Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 1971–1980, Dec. 2016, doi: 10.1109/CVPR.2016.217.

[26] P. Parisot and C. De Vleeschouwer, "Scene-specific classifier for effective and efficient team sport players detection from a single calibrated camera," *Computer Vision and Image Understanding*, vol. 159, pp. 74–88, Jun. 2017, doi: 10.1016/J.CVIU.2017.01.001.

[27] L. Liu, "Objects detection toward complicated high remote basketball sports by leveraging deep CNN architecture," *Future Generation Computer Systems*, vol. 119, pp. 31–36, Jun. 2021, doi: 10.1016/J.FUTURE.2021.01.020.

[28] K. Lu, J. Chen, J. J. Little, and H. He, "Lightweight convolutional neural networks for player detection and classification," *Computer Vision and Image Understanding*, vol. 172, pp. 77–87, Jul. 2018, doi: 10.1016/J.CVIU.2018.02.008.

[29] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," Mar. 2018, doi: 10.48550/arxiv.1803.08375.

[30] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 558–567, Dec. 2018, doi: 10.48550/arxiv.1812.01187.