Journal of Computers Vol. 35 No. 4, August 2024, pp. 291-301
doi: 10.53106/199115992024083504020

Application of Neural Network-based Intelligent Refereeing Technology in Volleyball

Xu Guang[1] and Xing-Chen Wu[2*]

[1] Sports Department, Shenyang Aerospace University, Shenyang 110136, China
2677675362@qq.com

[2] Jiushao Institute of AI Algorithm, Jihua Laboratory, Foshan 528200, China
wuxingchen3687@sina.com

Received 4 December 2023; Revised 5 May 2024; Accepted 23 June 2024

Abstract. Advance in AI technology is revolutionizing sports officiating, yet volleyball has seen limited application of such innovations. This paper introduces a novel neural network-based approach for real-time intelligent refereeing in volleyball, utilizing an advanced multi-scale object detection network and a dynamic adaptive sampling method to enhance real-time performance. Our contributions include a unique method for integrating human-object interaction detection using Transformers, significantly improving detection accuracy and real-time processing capabilities compared to existing technologies. Experimental results demonstrate superior performance, with marked improvements in accuracy and real-time applicability. This work not only advances the application of intelligent refereeing in volleyball but also sets a foundation for broader adoption in other fast-paced sports.

Keywords: intelligent refereeing, multi-scale object detection, human-object interaction, transformer

1 Introduction

In recent years, more and more sports events have begun to consider the introduction of technology for intelligent refereeing [1, 2]. In the field of soccer, the Video Assistant Referee (VAR) has been widely used in major competitions. The 2022 Qatar World Cup even integrated AI into this technology, adopting a semi-automatic approach for rulings on offside, and it is foreseeable that intelligent refereeing will have broader applications in sports events in the future. Currently, AI refereeing is in the development stage, and its feasibility has been preliminarily demonstrated. The current status and future application scenarios of intelligent refereeing in sports such as basketball and soccer have been explored. However, we still need to continuously explore the technical challenges and barriers in the application of intelligent refereeing.

For intelligent refereeing, real-time capability is a crucial feature. In fast-paced sports events, we need to capture the required information from a continuous stream of video frames. This requires an appropriate method for sampling from the video stream. A too small sampling frequency can increase system processing time and affect real-time performance, while a too large sampling frequency may impact the accuracy of system results. The continuous video frames provide contextual information, which can be used to calculate the motion information of individuals or objects and interpolate position information between two frames to fit object motion trajectories.

In ball sports, rulings can be categorized as pertaining to objects and individuals. Rulings on the ball mainly focus on position detection, such as touches on the boundary or in/out judgments. On fields with prominent sidelines, edge detection algorithms such as monochromatic value processing with Gaussian filtering can effectively determine the coordinates of the boundary lines, aiding in out-of-bounds ball judgments. However, traditional methods such as color transformation and Hough circular detection are not suitable for ball object detection. The high-speed changing scene and other objects on the field can interfere with the detection of ball objects. Neural network-based detection methods will become the mainstream algorithm for ball object detection in sports arenas.

The judgment of human behavior can be seen as a group activity recognition problem in the volleyball domain. Group activity recognition work and related datasets have been gradually developed in recent years.

* Corresponding Author

Waltner et al. [3] proposed a preliminary and relatively comprehensive solution that generates local spatial information feature subspaces using a class-Bayesian detection method and then generates global spatial descriptors to model group activities. Spatial descriptors for activities are represented using SVM. However, traditional machine learning methods have relatively low recognition accuracy. SVM classifiers based on features like HOG and HOF can only achieve around 70% accuracy.

Recently, group activity recognition methods based on deep learning have been developed. Capturing and modeling temporal dynamics is a starting point [4]. A hierarchical architecture considers modeling the temporal information of each individual using Long Short-Term Memory networks (LSTM) and aggregates individual-level information to understand group behavior. However, high-speed changes in the background can easily disrupt the temporal dynamics of the target, which requires precise target labeling and dynamic sampling detection methods.

Another key task for intelligent refereeing is human-object interaction detection, which involves recognizing and understanding the actions and relationships between individuals and objects. In volleyball matches, we need to detect and determine in real-time the interactions between individuals and the ball and net. High-dimensional semantic understanding of the entire scene is crucial [5]. Currently, methods for human-object interaction detection can be categorized as follows:

1. Two-Stage Detection Methods: Two-stage human-object interaction detection typically first detect individuals and objects in the first stage, outputting target detection boxes. Then, an interaction classifier is used to determine the interaction category. Two-stage detection methods are direct but often require a large number of candidate pairs, increasing model computational complexity. They also have difficulty incorporating global contextual information and are less suitable for video applications, especially in complex scenarios with multiple people and objects. Methods like FCMNet [6] and PDNet [7] have worked on this basis and used word embedding layers to capture spatial semantic information better.
   However, two-stage detection methods face several disadvantages including high computational complexity due to the need for generating and processing numerous region proposals, which makes them slower and less suitable for real-time applications like live sports refereeing. Additionally, their larger model sizes and higher memory usage limit their use in environments with restricted computational resources. The training of these systems is also more challenging because it involves learning both region proposal and object classification tasks, and the overall performance heavily relies on the quality of the region proposals—if these are poor, even a highly accurate classifier cannot compensate, leading to suboptimal performance. Moreover, managing a large number of candidate regions can be inefficient, particularly in scenes with complex backgrounds or multiple overlapping objects, and two-stage methods often struggle to integrate contextual information effectively, which is crucial in scenarios where a broader scene understanding is necessary for accurate detection.
2. One-Stage Detection Methods: One-stage human-object interaction detection methods directly output all possible human-object interactions, corresponding detection boxes, and categories on a given image. This approach can reduce redundancy and information loss compared to two-stage detection methods. However, in scenarios with multiple target recognitions, the computational complexity becomes high, and dealing with long-distance dependencies is challenging. InteractNet [8] first proposed this approach, which relies on cascaded human-object interaction reasoning. PPDM [9] improved upon this by unifying target detection and human-object interaction detection into a single model, introducing the concept of interaction points and converting interaction detection into point detection problems.
   One-stage detection methods like YOLO and SSD also present several disadvantages, including lower accuracy, especially in detecting small or overlapping objects, due to their design that prioritizes speed over precision. They are also prone to higher rates of false positives, particularly in cluttered scenes, and struggle with handling objects of varying sizes despite recent improvements like feature pyramids. Additionally, these methods may lack the necessary contextual understanding for complex scenarios where interactions or subtle distinctions are critical, and their performance is highly dependent on the design of anchor boxes, with inappropriate scales and aspect ratios significantly impacting detection quality. Moreover, the training of one-stage detectors can be less stable, sensitive to initial parameters and learning rate settings, and while they are faster, any added complexity to improve accuracy can diminish their speed advantage over two-stage detectors.
3. End-to-End Detection Methods: These methods typically use network structures like Transformers and do not require specific candidate pair generation steps for target detection or human-object interaction. They directly obtain human-object interactions, positions, and categories from image features. End-to-end solutions simplify the entire system's workflow and generally capture global contextual information

and long-distance dependencies better. DETR [10] uses a Long Short-Term Memory network with a Transformer structure and can parallelly detect and output sequences of human-object interactions. These methods usually use the Hungarian algorithm to match predicted values with ground truth.

Due to reliance on self-attention mechanisms, which necessitate powerful hardware for efficient operation, end-to-end methods also require large, well-annotated datasets to perform optimally and avoid overfitting, leading to long training times that can hinder rapid development and deployment. Additionally, these methods often struggle to incorporate specific domain knowledge or constraints due to their automated learning process, and their black-box nature can lead to challenges in interpretability, making it difficult to understand decision-making processes. Moreover, end-to-end systems are sensitive to noise and variability in data, which can degrade performance if the training data is not representative of real-world conditions, and they may face generalization challenges, struggling to adapt to new or slightly different scenarios than those encountered during training.

The remainder of this paper is organized as follows: Section 2 describes the methodologies employed, including our improved multi-scale network structure for ball detection and the dynamic adaptive sampling method for real-time video analysis. We also detail our novel approach for detecting human-object interactions using an end-to-end Transformer-based solution. Section 3 presents the experimental setup, the datasets used, and the results obtained. This section provides a comprehensive analysis comparing our system's performance against other existing methods. It includes detailed metrics and discussion on the validation of our intelligent refereeing system. Section 4 discusses the implications of our findings, limitations of the current study, and potential areas for further research. This section contextualizes our results within the broader field of AI in sports refereeing. Section 5 summarizes the key findings and contributions of our research, reiterating the impact on the field of sports technology and intelligent systems. We also outline future work and how our approach can be extended to other applications.

2 Methods

2.1 Object Recognition and Classification

We combine object detection and traditional boundary judgment methods to achieve real-time ball rulings in volleyball events. The image input is sampled from the real-time video stream captured by a camera.

A challenge in detecting the ball is that it occupies a relatively small number of pixels in the entire frame. To address this, we intend to use a multi-scale architecture neural network. We enhance the image resolution through deconvolution operations to improve the network's performance in detecting small objects. In contrast to CNN, where the final output is generated after all convolution layers, we refer to the network structure of SSD [11] and directly perform detection on the feature maps produced by each convolution layer. Convolution can be seen as an up-sampling and cropping operation on the image, and each operation is a scale transformation process. This approach ensures that the original object is detected at multiple scales. Unlike SSD, we use residual modules [12] as the convolution layers at the bottom of the network. Residual modules ensure that the model has better generalization for detecting small objects at different scales. Regarding the loss function, we use a weighted average of the CIoU [13] location loss and confidence loss [14].

$$E_{IoU} = \sum_{i \in P}^{N} (1 - \frac{|A \cap A_{gt}|}{|A \cup A_{gt}|} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{d^2} + \alpha v). \tag{1}$$

In this equation, A represents the corresponding region, and $\frac{|A \cap A_{gt}|}{|A \cup A_{gt}|}$ represents the calculation of IoU (Intersection over Union). Here, b and b^{gt} respectively denote the centers of the predicted region and the ground-truth region. ρ represents the distance between the centers of the two regions (using Euclidean distance). d represents the diagonal length of the minimum enclosing region that simultaneously contains the predicted region and the ground-truth region. α represents the weight factor for CIoU (Complete Intersection over Union), and

$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$ is used to measure the similarity in aspect ratios of the two regions.

$$E_{conf}(x,c) = -\sum_{i\in P} x_{ij}^{p} \log(\hat{c}_i^{p}) - \sum_{i\in N} \log(c_i^{0}). \tag{2}$$

In this equation, c represents the confidence prediction value for a specific class, which depends on the output of the classifier. i represents the i-th prior prediction box in the sample, and j represents the j-th ground-truth region. We use softmax as the classifier error, where $\hat{c}_i^{p} = \frac{\exp(c_i^{p})}{\sum_{p}\exp(c_i^{p})}$, and p represent the corresponding class.

The final loss function is obtained by taking the weighted average of the two.

$$E = \frac{1}{N}(E_{IoU} + \beta E_{conf}). \tag{3}$$

In this equation, β represents the weight factor between the position error and the confidence error.

The CIoU loss is used to measure the similarity between the detected target region and the ground-truth region, considering factors such as the distance between their centers and the overlapping area. The confidence loss, to some extent, represents the confidence that the outlined region can contain the target object. For small object detection, errors in relatively larger objects are somewhat amplified. The choice of this loss function allows for considering both the similarity in aspect ratios and the distance between the two regions as indicators of consistency. Additionally, it quantifies the requirement that the region must encompass the target object.

The entire network structure is as shown in Fig. 1.

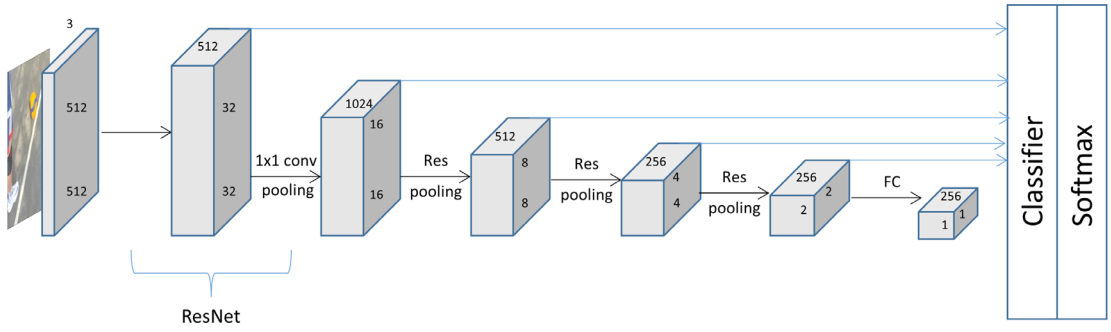

Fig. 1. The structure of the object detection network

2.2 Optimization of Real-time Video Streaming

In our application, our detection is not based on individual still images but is performed in real-time. In our algorithm implementation, we have made certain optimizations for this real-time scenario. Each frame of the image first undergoes an edge detection process, using the Canny operator [15], to extract prominent edges. Subsequently, we utilize the Hough gradient method [16] to calculate the gradient of all non-zero points in the image obtained from the Canny edge detection. By accumulating these gradients, we identify all possible centers and calculate the distances between them to obtain radii.

Circle detection may yield multiple potential circles, and at this stage, we cannot confirm whether a sphere is our target object. Other parts such as the human head might also be detected. Therefore, the results at this stage serve as the input for the ROI (Region of Interest) for the target detection algorithm and as a confidence consideration rather than the direct target detection output. During the prediction phase, we consider the motion trajectory of the sphere, combining the previous frame's position, motion direction, and circle detection results to assist in target detection assessment.

For a continuous sequence of video frames $F = \{f1, f2, ..., fn\}$, if the previous frame's predicted position and estimated velocity are known, they can be used as a prediction for the next frame's position, weighted and incorporated into the confidence. If only the predicted position is known, the object's velocity can be estimated based on the positions between two frames. If both are unknown, we predict the current frame's position and use it in conjunction with the next frame's position to determine the estimated velocity.

Considering the real-time nature of the algorithm, we introduce a dynamic sampling method to strike a balance between reducing computational complexity and improving algorithm accuracy. In the initial phase, we run the full algorithm to obtain the positions of the ball for the first few frames while simultaneously predicting the ball's velocity. Afterward, we set a sampling interval period γ, with an initial value of 1 second. During non-keyframes in the video stream, we only perform real-time circle detection and determine the objects that meet the conditions based on the previous frame's position and velocity. These objects are then updated accordingly. This process involves assessing whether there has been a significant change in velocity, and we have

$$\theta = \arccos \frac{v_n \times v_{n-1}}{|v_n| \times |v_{n-1}|}. \tag{4}$$

When θ exceeds the preset threshold, it is necessary to forcibly trigger one target detection operation.

This is because such situations can be challenging for traditional methods to accurately determine, and we require a timely offset correction. Building upon this, we can outline the motion trajectory of the ball.

2.3 Detection of Human-object Interaction

In this application, we need to implement detection and assessment of human-object interaction with key objects such as balls, boundaries, and nets to enable automated intelligent judgment in sports events.

The problem of predicting the interaction between humans and key objects is based on continuous video frames $F = \{f^1, f^2, ..., f^t\}$, masks of the human body $H = \{h^1, h^2, ..., h^t\}$, and masks of the key objects $M = \{m^1, m^2, ..., m^t\}$. The goal is to predict the contact status between humans and key objects to make judgments on humans in consecutive video frames, such as whether they stepped on a line, touched the net, or touched the ball, among other contact states.

For detecting interactions like touching the net, touching the ball, and stepping on a line, we treat it as a Human-Object Interaction (HOI) task. Compared to traditional convolutional neural network methods, we additionally extract context information of humans and objects from the video to assist in learning. We define human interaction as a quintuple (C_human, C_interaction, C_object, P_human, P_object), where P represents the position for each corresponding category, and C represents the confidence for each corresponding category.

Our network starts with a multi-layer convolutional neural network (CNN) structure for feature extraction from the input images. After passing through the CNN, the input image is transformed into a feature map with high-dimensional semantic information, with dimensions (H_f, W_f, C_f). Subsequently, a 1×1 convolutional layer is used to reduce the number of channels to d_f. This dimension reduction helps in reducing the number of parameters, enhancing interaction between different channels, and improving the network's non-linear fitting capability. After dimension reduction, we obtain a feature map with dimensions (H_f, W_f, C_f). The Transformer encoder typically requires a sequence input, and to preserve spatial information, we flatten the feature map into a sequence of length $H_f \times W_f$ [17], where each element is a vector of length d. Instead of traditional convolutional kernels, we use residual modules (ResBlocks) as our feature extraction network structure.

The obtained feature maps can be used as input to the Transformer encoder. We employ a feedforward neural network with multi-head attention mechanism [18]. In each attention layer of the Transformer, we introduce positional encoding to help the attention layers capture the relative positional information of the flattened sequence. We use the sinusoidal positional encoding method:

$$PE_{(pos,2i)} = \sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}), PE_{(pos,2i+1)} = \cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}). \quad (5)$$

The positional encoding is added to the feature sequence and weighted summation, serving as the input to the Transformer encoder. This helps the encoder's output to contain sufficient global information.

The Transformer decoder comprises multi-head cross-attention layers. The key, value, and query vectors in the Transformer, on the decoder side, are combinations of the feature sequence vector containing positional encoding, the serialized vector, positional encoding, and the human-object interaction query vector, respectively.

The output of the Transformer decoder serves as the input to a multi-layer perceptron (MLP) embedding layer, which is responsible for decoding the human interaction quintuple. This MLP includes three single-layer perceptrons, each responsible for detecting and outputting confidence scores for humans, interaction categories, and object categories. Additionally, it includes two three-layer perceptrons for recognizing human body object detection boxes and object detection boxes. The output layer of the confidence perceptron uses softmax classification. The output vector lengths for human and interaction confidence perception are 2, representing foreground/background and interaction occurring/not occurring (in our application, interaction includes only one type). The output vector length for object category confidence perception is 4, representing confidence scores for the ball, net, line, and background. The output vector length for human and target detection box from the multi-layer perceptron is 4, representing the four coordinates of the target detection box.

The final network structure is as shown in Fig. 2.

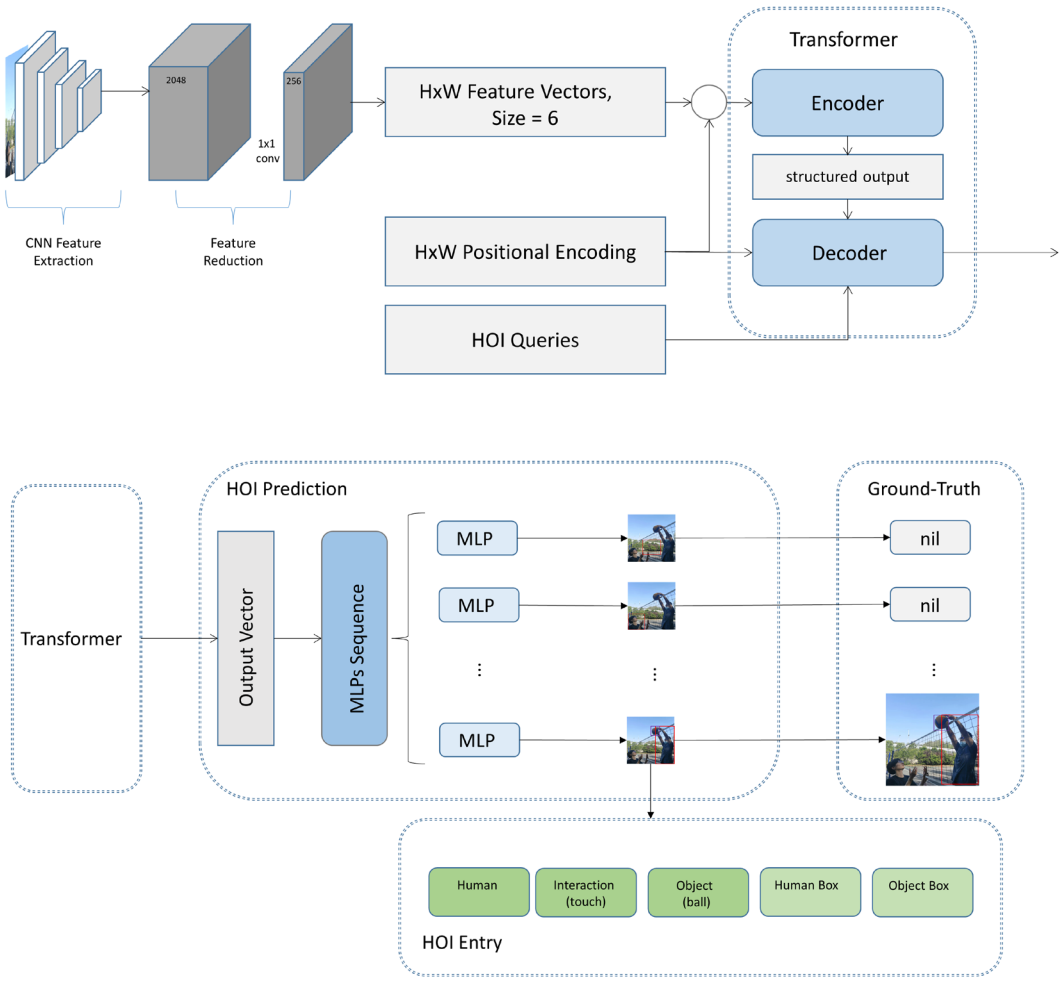

Fig. 2. A Transformer-based solution for human-object interaction detection

In the model training phase, after obtaining predicted human interaction (HOI) tuples, we need to match them with the ground truth. We define the model's output HOI tuples as $O = o^i$, $i = 1, 2, ..., N$, and the true HOI tuples as $T = t^i$, $i = 1, 2, ..., M$. Since the model might make incorrect judgments about interactions, the lengths of these two sets may not be equal. To make them equal, we will pad the true HOI tuples with $N - M$ elements.

Based on this, we can define a mapping function $\epsilon_{O,T}$ that maps the sequence index of model output to the sequence index of the ground truth, where $\epsilon(i)$ corresponds to the i-th ground truth.

The model's matching loss function can be defined as follows:

$$L = \sum_{i}^{N} l(t^i, o^{\epsilon(i)}). \tag{6}$$

The term $l\,(t^i, o^{\epsilon(i)})$ represents the specific item matching loss between t^i and $o^{\epsilon(i)}$.

$$l(t^i, o^{\epsilon(i)}) = a_1 \sum_{j \in human, object, interaction} b_j l^j_{class} + a_2 \sum_{k \in human, object} l^k_{box}. \tag{7}$$

The term l^j_{class} represents the classification loss for humans, objects, and interactions, and it is defined as $l^j_{class} = l_{class}(t^i_j, o^{\epsilon(i)}_j)$. This uses a standard softmax classification loss function.

The term l^k_{box} represents the distance measurement between detection boxes, and here we refer to the method mentioned earlier in object detection, using the CIoU loss function.

After deriving the matching loss function, we utilize the Kuhn-Munkres (KM) algorithm to solve the bipartite matching problem, identifying the optimal match between the predicted human interaction sequence and the ground truth values.

$$\hat{\epsilon} = \arg\min_{\epsilon \in \Theta_N} L. \tag{8}$$

Where Θ_N represents the solution space for the entire bipartite matching problem. Once each matching pair is obtained, the network's loss can be calculated for training. Through iterative optimization, the final model is obtained by fitting the data continuously.

This Transformer-based approach serves as a comprehensive end-to-end solution for human interaction. In contrast to traditional one-stage or two-stage methods for human interaction detection, this model can directly complete the process from input image to output human interaction detection sequences. Traditional approaches often treat confidence score output and detection box output as two independent tasks, whereas in our implementation, these tasks are performed simultaneously.

3 Experiment

3.1 Ball localization Quality

We used a publicly available volleyball detection dataset \footnote{https://towardsdatascience.com/ball-tracking-in-volleyball-with-opencv-and-tensorflow-3d6e857bd2e7} and a custom dataset, and applied data augmentation techniques as listed in Table 1.

Table 1. Data augmentation techniques

Methods	Value
Rotation	40°
Translation (width)	20%
Translation (height)	20%
Shear	20%
Scale	[80%-120%]
Horizontal flip	

The input image is first preprocessed, scaled to (512, 512), and then color-normalized based on the mean RGB values of the dataset before being input into the network. As shown in Fig. 3, the output of ball object detection includes the four coordinates of the ball detection box and their corresponding confidence scores.

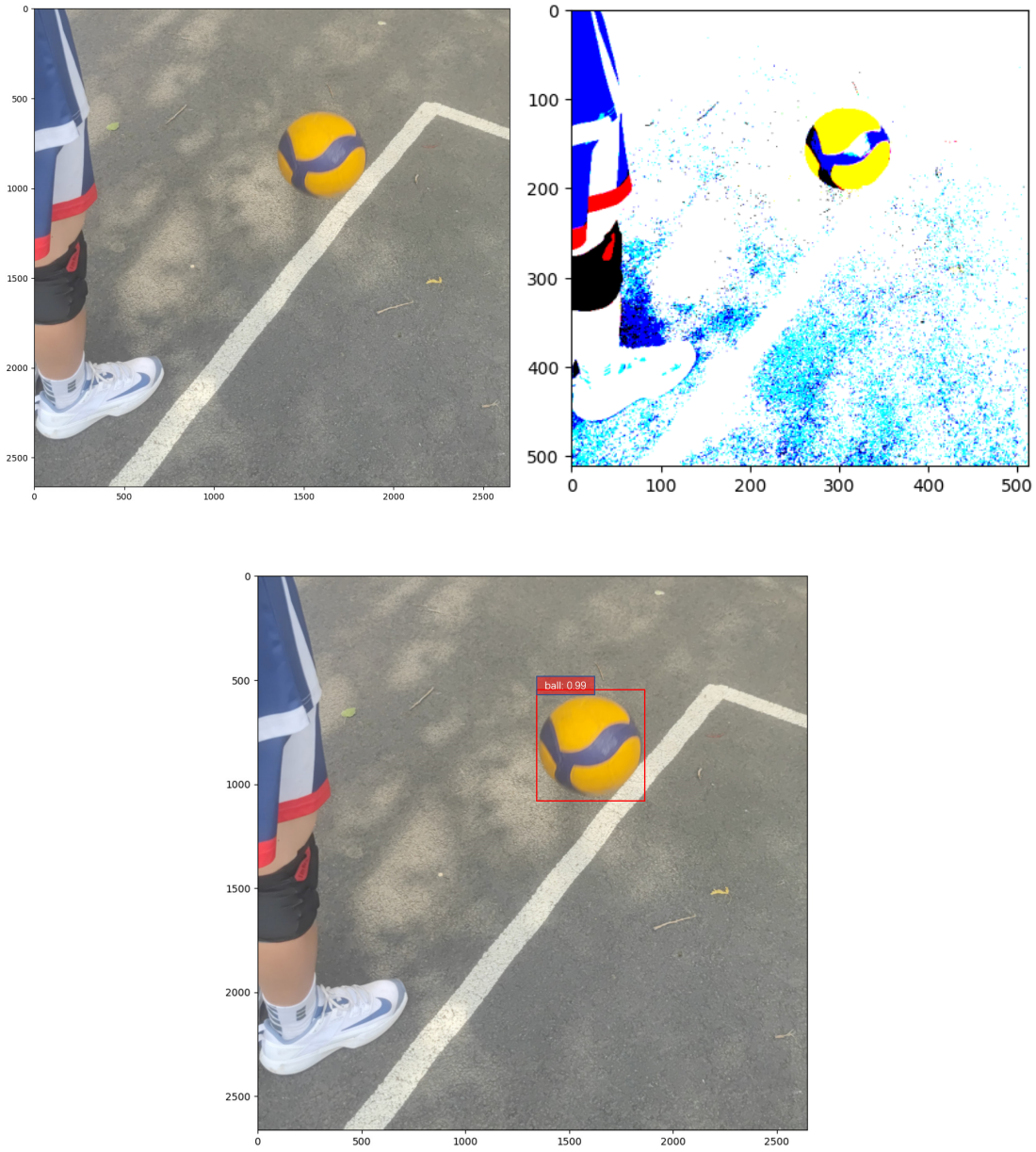

Fig. 3. The image processing for one frame

(The original input image will be preprocessed and output the real-time detection results.)

We compared the improved multiscale architecture neural network with the results of R-CNN and SSD networks and found an increase in the mean average precision (mAP) metric. Data augmentation also showed a noticeable improvement in network performance. Taking into account the high-frequency movements and deformations that volleyball might exhibit in a real-time sports environment, data augmentation is crucial during model training. This object detection task focuses on detecting volleyball in a relatively limited scene space, resulting in higher detection accuracy compared to all-class detection. The final results are as shown in Table 2.

Table 2. The volleyball object detection results

Methods	Datasets	mAP	aug
R-CNN	Custom dataset	63.2	66.3
R-CNN	Vd dataset	68.1	70.8
SSD	Custom dataset	67.0	71.1
SSD	Vd dataset	71.2	72.9
This paper	Custom dataset	69.7	73.5
This paper	Vd dataset	71.8	74.0

For real-time video stream optimization, we set the sampling interval to 1 second to strike a balance between the operational performance in actual applications and algorithm accuracy.

3.2 Human-object Interaction Judgment

We conducted experiments on human interaction detection using the HICO-DET dataset [19], V-COCO dataset [20], and a custom dataset. We used the mean average precision (mAP) as the evaluation metric for the model. This metric is more relevant in real volleyball applications, focusing on a smaller number of interaction categories. We consider an output sample as a true positive (TP) only when the predicted HOI output's human and object detection boxes have an IoU (Intersection over Union) value greater than 0.5 with the ground truth and when the model correctly predicts the interaction category (i.e., contact).

The image's feature extraction layer employs the classical ResNet [21] processing, followed by channel dimension reduction. We adopt a transfer learning approach, initializing the weight parameters of ResNet, the Transformer encoder, and decoder using the pre-trained parameters from DETR [10]. The image input and output for human interaction detection are as shown in Fig. 4.

Fig. 4. The output of the human interaction tuple includes human and object detection boxes and confidence scores for human, object, and interaction categories

The final evaluation results for human interaction detection are as shown in Table 3.

Table 3. The mAP for human interaction detection

Methods	HICO-DET	V-COCO	Custom dataset
FCMNet	29.55	60.63	69.0
PDNet	28.53	58.64	62.41
InteractNet	18.44	43.37	44.24
PPDM	32.92	41.19	67.27
This paper	36.79	62.24	71.78

In the object detection phase of human-object interaction detection, in practice, the output of the multiscale object detection network serves as a reference for the detection boxes in the multi-layer perceptron. The detection results are compared with the output of the multi-layer perceptron, and the CIoU is computed to readjust the confidence scores for ball category detection. Experimental validation has shown that this auxiliary input can improve the accuracy of ball object detection in the multi-layer perceptron.

3.3 The Sideline Detection Solution

In practical applications, for out-of-bounds ball detection, we utilize a common sideline detection solution.

As mentioned earlier, after edge detection processing of the image, we apply Hough line detection [22] to identify the boundary lines of the volleyball court. In the image, we obtain a collection of lines, denoted as $Y = \{y_1, y_2, ..., y_N\}$, $y_i = k_i x + b_i$.

After obtaining the object detection box for the volleyball, as mentioned earlier, we continuously monitor for a sudden change in the vertical direction of the ball's speed (determined based on the camera's relative coordinates). Additionally, we wait for the next 5 frames to check for any signs of human-object contact. If such contact is detected, the check is skipped. Otherwise, we conclude that the ball has made contact with the ground. At this point, the bottom coordinate $P_b = (x_b, y_b)$ of the object detection box represents the contact point. Subsequently, we use the same-side rule to determine whether this point is contained within the boundary line. If it is not contained, we consider the ball to be out of bounds.

4 Conclusion

This paper divides real-time officiating in volleyball matches into two tasks: small object detection for the volleyball and human interaction detection for people, the ball, net, and boundary lines. The object detection task uses an improved residual module multiscale detection network, while the human interaction task employs an end-to-end Transformer module. In the context of real-time volleyball video streams, there has been an improvement in mean average precision for all classes, and experiments have shown that this approach can assist in automating and semi-automating officiating tasks.

In terms of object detection, this paper employs an improved multiscale neural network approach, allowing the model to possess good object detection capabilities at different scales, with particular optimization for the small object detection problem of the volleyball. The use of residual connections instead of traditional convolutional modules ensures the model's generalization ability. Additionally, for real-time volleyball video stream scenarios, a dynamic adaptive sampling method is provided to alleviate the performance pressure on algorithms in real-time settings.

For human interaction detection, this paper adopts an end-to-end solution based on Transformer. It begins by extracting image features through multiple convolutions, reducing the dimensionality of the feature channels, and then flattening them to serve as input to the Transformer encoder. Position encoding is introduced to learn spatial relative position information. The output of the Transformer decoder passes through a multi-layer perceptron to eventually obtain the predicted human interaction quintuple. A specific matching loss function is defined, and the training phase solves the bipartite matching problem between predicted and real human interaction tuples using the KM algorithm.

For automating the judgment of ball hitting the sideline, traditional methods still have some limitations. It may be worthwhile to consider using multiple cameras to reconstruct the three-dimensional coordinates of the ball and boundary lines for a more precise judgment. The results of object detection can aid in more accurate three-dimensional reconstruction.

Increasing the number of cameras and making a comprehensive judgment from multiple video sources is indeed a valuable improvement strategy. Whether in object detection or human interaction detection, strong spatial semantic information support is essential, including spatial context information and relative positional information. Adding video sources from different angles can not only reduce error rates from a sample size perspective but also provide additional spatial semantic information, enhancing the overall officiating capability.

References

[1] X. Shi, J. Zhang, RETRACTED: Analysis of Application of Tennis Electronic Referee Based on Artificial Intelligence in Tennis Matches, in: Proc. 2021 Journal of Physics: Conference Series, 2021.

[2] X. Chen, Research on the application of artificial intelligence technology in the field of sports refereeing, Journal of Physics: Conference Series 1952(3)(2021) 042048.

[3] G. Waltner, T. Mauthner, H. Bischof, Indoor activity detection and recognition for sport games analysis. <https://doi.org/10.48550/arXiv.1404.6413> , 2014 (accessed 25.04.2021).

[4] M.S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, G. Mori, A hierarchical deep temporal model for group activity recognition, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] M. Antoun, D. Asmar, Human object interaction detection: Design and survey, Image and Vision Computing 130(2023) 104617.

[6] Y. Liu, Q. Chen, A. Zisserman, Amplifying key cues for human-object-interaction detection, in: Proc. 2020 European Conference on Computer Vision, 2020.

[7] X. Zhong, C. Ding, X. Qu, D. Tao, Polysemy deciphering network for human-object interaction detection, in: Proc. 2020 European Conference on Computer Vision, 2020.

[8] G. Gkioxari, R.B. Girshick, P. Dollar, K. He, Detecting and recognizing human-object interactions, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[9] Y. Liao, S. Liu, F. Wang, Y. Chen, C. Qian, J. Feng, PPDM: Parallel point detection and matching for real-time human-object interaction detection, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[10] C. Zou, B. Wang, Y. Hu, J. Liu, Q. Wu, Y. Zhao, B. Li, C. Zhang, C. Zhang, Y. Wei, J. Sun, End-to-end human object interaction detection with HOI transformer, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single shot multibox detector, in: Proc. 2016 European Conference on Computer Vision, 2016.

[12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[13] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, in: Proc. of the AAAI Conference on Artificial Intelligence, 2020.

[14] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[15] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(6)(1986) 679-698.

[16] J. Illingworth, J. Kittler, A survey of the Hough transform, Computer Vision, Graphics, and Image Processing 44(1)(1988) 87-116.

[17] J. Hu, L. Cao, Y. Lu, S. Zhang, Y. Wang, K. Li, F. Huang, L. Shao, R. Ji, ISTR: End-to-end instance segmentation with transformers. <https://arxiv.org/abs/2105.00637> , 2021 (accessed 03.05.2021).

[18] A. Vaswani, N.M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. <https://arxiv.org/abs/1706.03762> , 2017 (accessed 12.06.2017).

[19] Y.-W. Chao, Y. Liu, M. X. Liu, H. Zeng, J. Deng, Learning to detect human-object interactions, in: Proc. 2018 IEEE Winter Conference on Applications of Computer Vision, 2018.

[20] S. Gupta, J. Malik, Visual semantic role labeling. < https://arxiv.org/abs/1505.04474> , 2015 (accessed 17.05.2015).

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning. < https://arxiv.org/abs/1602.07261> , 2016 (accessed 23.02.2016).

[22] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15(1)(1972) 11-15.