Research Article

Long Liu*, Yuxin Dai, and Zhihao Liu

Real-time pose estimation and motion tracking for motion performance using deep learning models

https://doi.org/10.1515/jisys-2023-0288
received December 01, 2023; accepted January 05, 2024

Abstract: With the refinement and scientificization of sports training, the demand for sports performance analysis in the field of sports has gradually become prominent. In response to the problem of low accuracy and poor real-time performance in human pose estimation during sports, this article focused on volleyball sports and used a combination model of OpenPose and DeepSORT to perform real-time pose estimation and tracking on volleyball videos. First, the OpenPose algorithm was adopted to estimate the posture of the human body region, accurately estimating the coordinates of key points, and assisting the model in understanding the posture. Then, the DeepSORT model target tracking algorithm was utilized to track the detected human pose information in real-time, ensuring consistency of identification and continuity of position between different frames. Finally, using unmanned aerial vehicles as carriers, the YOLOv4 object detection model was used to perform real-time human pose detection on standardized images. The experimental results on the Volleyball Activity Dataset showed that the OpenPose model had a pose estimation accuracy of 98.23%, which was 6.17% higher than the PoseNet model. The overall processing speed reached 16.7 frames/s. It has good pose recognition accuracy and real-time performance and can adapt to various volleyball match scenes.

Keywords: motion performance, pose estimation, motion tracking, deep learning models, real-time performance

1 Introduction

With the rapid development of social information technology, sports performance analysis has gradually become a focus of attention. The rise of the intelligent era has profoundly changed people's understanding and analysis of sports. Nowadays, in sports, models are difficult to adapt to posture estimation in multiple human bodies and different environments, resulting in low accuracy and poor real-time performance of human posture estimation. They cannot track athletes' movements and postures in a timely manner, especially for fast sports types. Precise pose estimation and tracking of sports videos can provide reference for athlete training and improvement, thus promoting the integration of technology and sports, so as to further respond to the sports publicity work.

With the continuous advancement of deep learning technology, a large number of research results have been achieved in posture estimation and motion tracking in sports. Zheng and other scholars conducted research and analysis on pose estimation solutions and challenges, providing references for future

*** Corresponding author: Long Liu,** School of Health Care, Chongqing Preschool Education College, Chongqing 404047, China, e-mail: liulong@cqyz.edu.cn
Yuxin Dai: School of Physical Education, Chongqing Preschool Education College, Chongqing 404047, China, e-mail: dyx_953190@163.com
Zhihao Liu: School of Physical Education, Chongqing Preschool Education College, Chongqing 404047, China, e-mail: 806863291@qq.com

experiments [1]. Rohan and other scholars used Convolutional Neural Network (CNN) models to estimate and classify human postures in order to solve the problem of pose classification accuracy [2–4]. Scholars such as Dong et al. used multi-directional matching algorithms based on convex optimization to recognize human postures, improving robustness [5]. In order to track the movements of football players in real-time, scholars including Felipe et al. developed a multi-camera tracking system (Mediacoach), which improved the tracking and recognition accuracy [6]. Scholars such as Luvizon et al. proposed a multitasking framework for jointly estimating 3D (three-dimensional) human posture from monocular color images, resulting in improved recognition accuracy of human actions [7]. Scholars including Li et al. used bounding box constraints and long short-term memory methods for multi-person pose estimation, which were robust to bounding box displacement and compactness [8]. Sengupta and other scholars used a millimeter wave radar for real-time detection and tracking of human bones to improve robustness, achieving excellent robustness and detection results [9,10]. The above scholars have improved the accuracy of human pose estimation to some extent, but their real-time performance is poor.

In order to meet the current social reality needs, many researchers have conducted research on improving the real-time performance of human posture tracking. Scholars such as Yi et al. used the TransPose method for real-time human pose estimation, achieving a real-time speed of 90 fps [11]. Scholars including Xu-Wei et al. proposed a YOLOv4 (You Only Look Once version 4) model combined with a Kalman filter real-time hand tracking method to address the shortcomings of gesture tracking in terms of accuracy and speed. The real-time tracking achieved a speed of 41.822 frames per second (fps) [12]. Scholars such as Wu et al. introduced the OpenPose pose algorithm for posture estimation of sports athletes, achieving good performance [13–15]. Naik and other scholars, in order to improve the real-time performance of human pose tracking, used Kalman filtering and SORT (Simple Online and Real-time Tracking) algorithm with overlapping bounding boxes to achieve tracking, with a tracking speed of only 11.3 fps [16]. Scholars such as Razzok et al. used the DeepSORT tracking algorithm for real-time pedestrian tracking, at 46 fps per second [17]. Scholars including Sajina and Ivasic-Kos compared different tracking algorithms and found that the DeepSORT method performed the best in tracking bones in dynamic motion scenes [18]. In summary, scholars know that the combination of OpenPose and DeepSORT models used in this article for real-time pose estimation and tracking of volleyball videos is feasible and can solve current practical problems.

In order to solve the problem of low accuracy and poor real-time performance in human pose estimation during sports, this article used the OpenPose and DeepSORT combination model to perform real-time pose estimation and tracking on volleyball videos. First, the original video was uniformly sampled by frame; pose recognition regions were annotated, and image enhancement operations such as Gaussian noise and random rotation were added. Then, the OpenPose algorithm was adopted to estimate the pose of the human body region, and the DeepSORT model target tracking algorithm was utilized to track the detected human pose information in real-time. Finally, using UAVs as carriers, the YOLOv4 object detection model was adopted to perform real-time human pose detection on standardized images. The experimental results on the Volleyball Activity Dataset showed that the OpenPose model had a pose estimation accuracy of 98.23%, a precision rate of 99.3%, a recall rate of 97.31%, and an overall processing speed of 16.7 frames/s, achieving high pose recognition accuracy and good real-time performance. At the same time, it has strong robustness and can adapt to Gaussian noise and multi-person occlusion.

The innovation of this article lies in the creative integration of deep learning models OpenPose and DeepSORT, and their application in volleyball sports scenes. The challenge of analyzing sports performance in the field of sports has been solved through real-time pose estimation and motion tracking, providing an advanced and comprehensive solution for flexible sports performance analysis in actual competitions.

2 OpenPose algorithm for pose estimation

The combination of OpenPose algorithm and DeepSORT tracking algorithm can fully address the challenges of real-time pose estimation and motion tracking in volleyball sports scenes. OpenPose and DeepSORT have

complementarity in handling information at different levels. OpenPose focuses on detailed analysis of poses, while DeepSORT focuses on continuous tracking of targets. Combining them can achieve more comprehensive and accurate motion performance analysis in pose estimation and target tracking. The complementary advantages of these two factors enable the combination model in this article to achieve good real-time performance while maintaining high accuracy, providing a comprehensive and effective solution for performance analysis in volleyball sports. Compared with the pose estimation framework PoseNet, OpenPose exhibits superior multi-body keypoint detection and pose estimation capabilities in terms of multi-body recognition ability.

OpenPose has a wide variety of practical application scenarios in the field of pose estimation, especially in sports, demonstrating significant application value. By capturing the key postures of athletes in real-time, precise data support is provided to coaches during training and competitions, which can be used for personalized training, tactical analysis, and prevention of sports injuries. OpenPose has also demonstrated its potential in sports medicine research by monitoring athlete postures and gaining a deeper understanding of the impact of different sports on the body, providing important support for developing scientific sports training programs. These practical application scenarios highlight the potential application value of OpenPose in improving athlete skills, optimizing tactical strategies, and promoting sports medicine research.

OpenPose is a top-down detection algorithm [19–21] that mainly relies on convolutional neural networks and supervised learning for human pose estimation. Scholars such as Lee M F R constructed a human skeleton position map and used the OpenPose algorithm to detect multiple postures from a single image using CNN. Through CMN, the heat map format is used to predict the key points of each body part, and through PAF (Part Affinity Fields), vector maps are used to reflect the interaction probability between these key points [22]. These display the corresponding positions of the eyes, ears, neck, shoulders, elbows, wrists, buttocks, knees, and ankles.

PAF is the core of OpenPose and an important feature that distinguishes it from other key point detection frameworks, which is mainly used to represent the affinity between different joint points. Among them, the affinity for different joints of the same person is often higher, while the affinity for joints between different people is lower. The first level of the partial affinity network uses a 3 × 3 convolutional kernel, and the second level uses a 7 × 7 convolutional kernel. In addition, the predicted values S^t and L^t of the two branches are connected at the end of each level with the original feature map F as the input of the next level, resulting in a larger perceptual field. The specific expressions are shown in formulas (1) and (2) [23].

$$S^t = p^t(F, S^{t-1}, L^{t-1}), \tag{1}$$

$$L^t = \phi^t(F, S^{t-1}, L^{t-1}). \tag{2}$$

Among them, t in formulas (1) and (2) are both greater than or equal to 2.

3 DeepSORT tracking algorithm

DeepSORT is a target-tracking algorithm with extensive applications in video surveillance, traffic management, and sports tracking. DeepSORT is an improvement on SORT, introducing a new data association measurement method based on target motion information and appearance information. The processing steps of DeepSORT include trajectory processing and state estimation, data association, and cascading matching [24–26]. The DeepSORT algorithm defines an eight-dimensional space vector $(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}$, and $\dot{h})$ to represent the trajectory condition at a certain time, which includes the center position of the bounding box (u, v) and the aspect ratio of γ. h represents height, and $\dot{x}$, $\dot{y}$, $\dot{\gamma}$, and $\dot{h}$ represent the velocity of u, v, γ, and h relative to the image coordinates.

For the data association part, DeepSORT achieves the association of motion information by detecting the Markov distance $d^{(1)}(i, j)$ between the results and the tracker prediction results. Scholars such as Yuemeng used the DeepSORT algorithm to predict the real-time position of unmanned aerial vehicles [27]. The relevant calculations cited in this article are shown in formula (3). Among them, d_j represents the predicted position of

the jth detection box; y_i represents the predicted position of the ith tracker on the target; S_i represents the covariance matrix between the detection position and the average tracking position.

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1}(d_j - y_i). \tag{3}$$

To address the significant mismatch of Markov distance caused by the uncertainty of motion caused by Markov distance, the minimum cosine distance $d^{(2)}(i,j)$ is introduced to associate the appearance information of the target to adapt to motion uncertainty. The calculation formula is shown in formula (4). Among them, r_j represents the surface feature descriptor of the jth detection box; $\|r_j\| = 1$; $R_k = \{r_k^{(i)}\}_{k=1}^{L_k}$ is used to store the latest L_k descriptors for each track.

$$d^{(2)}(i,j) = \min\{1 - r_j^T r_k^{(i)} \mid r_k^{(i)} \epsilon R_i\}. \tag{4}$$

On the basis of the above, the final correlation metric is calculated by combining the weighted Markov distance and the minimum cosine distance. The specific calculation formula is shown in formula (5). When $c_{i,j}$ is within the intersection of two metric set thresholds, it is considered that the information association is successful. The DeepSORT algorithm combines deep appearance information to improve the accuracy of target tracking under occlusion, enabling it to achieve real-time online target tracking.

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1 - \lambda) d^{(2)}(i,j). \tag{5}$$

In this article, in order to further improve the model's detection of athlete posture, YOLOv4 is introduced, and a four-scale object detection layer is established. The specific steps are as follows. A volleyball match video is input and first preprocessed into an image sequence. The OpenPose network is utilized for real-time pose estimation, and each frame is input into the YOLOv4 network to perform human pose detection, obtaining a set of detection box coordinates, confidence scores, and pose categories for the image. Then, these values are used as inputs to the DeepSORT algorithm and an identifier is created for the input target. The human keypoint information provided by OpenPose is associated with the target ID provided by DeepSORT. The DeepSORT algorithm creates a tracking list, writes the above detection results to the tracking queue for real-time human posture tracking, and introduces a cascade matching strategy to match the detection target with the tracking target. Finally, the detected bounding boxes and human posture are utilized to update the target state and output the target center position.

4 Experimental data

4.1 Experimental data set

This article uses the Volleyball Activity Dataset, which is a collection of annotated volleyball video sequences captured from professional competitions. This is highly adaptable to this study, and the data are authentic and widely used by other scholars. The data set consists of six videos with a resolution of 1,920 × 1,080 and 25 frames per second, including serving, receiving, attacking, blocking, and standing. Six actions are set. The tenfold cross-validation method [28] is used to divide all six videos into training and testing sets, and all data are randomly divided into 10 subsets, with 30% being the testing set and 70% being the training set. The experiment is conducted in turns and the average of the results is taken as the final evaluation criterion.

4.2 Data preprocessing

(1) Splitting by frame

For the original video, first, the volleyball video is extracted per frame, extracting multiple consecutive time periods of actions, each corresponding to an action.

(2) Image annotation processing

Image annotation adds labels to specific areas or objects in an image in order to enable computer vision models to understand the content of the image. In the field of pose recognition, image annotation is used to label key points, body parts, or motion actions in the image, making the model more focused on the pose recognition of the annotated area.

(3) Image enhancement

To balance the data set, images are uniformly subjected to random rotation and noise addition operations. The athlete's action image is uniformly rotated counterclockwise by 30 degrees, and Gaussian noise is added to the original image. The first row is the image processed by adding Gaussian noise, and the second row is the image rotated randomly.

5 Real-time pose estimation and tracking experiment for athletes

5.1 Experimental environment

This experiment is based on a Windows 10 system, implemented using the TensorFlow framework in Python, using an Intel Core i7-6800k CPU (central processing unit), NvidiaTITAN Xp (12GB) graphics card, and 16GB of memory.

5.2 Experimental process

In this experiment, first, the video is input to perform uniform sampling of video frames, image annotation, image enhancement operations such as adding Gaussian noise and rotation, and pre training is performed using ImageNet. After pre-training, the training set of the volleyball data set is added for model training and fine-tuning parameters. First, all samples in the training set are trained 30 times to better test the effectiveness of the model (epoch = 30), and when adjusting parameters, the number of training times is dynamically expanded and stacked in multiples of 10. Then, in the loss function of the model, cross entropy is used to calculate the error between the predicted value and the true value, and the model is optimized using the Adam (Adaptive Moment) gradient descent algorithm. The learning rate is initialized to 0.0001, with betas set to (0.9, 0.999). Finally, the performance of different pose estimation models is compared on the volleyball data validation set to verify the model.

6 Real-time pose estimation experimental results

6.1 Evaluation indicators

In this study, accuracy, precision, recall, and comprehensive evaluation index $F1$-score are used to evaluate the results. Below, based on the combination of OpenPose and DeepSORT models, estimation and classification in human posture are evaluated [29,30].

Accuracy: the proportion of all correct judgments made by the model to the total.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

Precision: the proportion of all predictions that are truly correct to positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

Recall rate: the proportion of true correctness to all actual positives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

*F*1-score: The *F*1-score is the arithmetic mean divided by the geometric mean, and the larger the result, the better it is. The *F*1-score is weighted for both precision and recall, and the *F*1-score belongs to 0–1. In this model, 1 represents the best recognition and classification results, while 0 represents the worst.

$$f1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{9}$$

In this study, multiple classifications are used as a whole. Among them, the actual positive class prediction is True Positive (TP), which predicts blocking actions as blocking actions; the actual positive class is predicted to be False Negative (FN), and the blocking action is predicted to be a standing action; the actual negative class is predicted to be False Positive (FP), which predicts standing movements as blocking movements; the actual negative class prediction is True Negative (TN), which predicts standing movements as standing movements.

6.2 Experimental results

After the above experimental process, the detection pose estimation and tracking system of the UAV is constructed. The results of single-person pose estimation are shown in Table 1, and the experimental results of multi-person pose estimation are shown in Table 2.

Table 1: Experimental results of single-person posture estimation

Serial number	Actual posture	Predict pose	Serial number	Actual posture	Predict pose
1	Attack	Attack	7	Serve	Serve
2	Attack	Attack	8	Serve	Serve
3	Attack	Attack	9	Setting	Setting
4	Block	Block	10	Setting	Setting
5	Reception	Serve	11	Stand	Stand
6	Serve	Serve	12	Stand	Setting

Table 2: Experimental results of multi-person posture estimation

Serial number	Actual posture	Predict pose	Serial number	Actual posture	Predict pose
1	Attack, Attack, Stand	Setting, Attack, Stand	7	Reception, Setting	Reception, Setting
2	Block, Block	Block, Block	8	Reception, Reception, Setting	Reception, Reception, Setting
3	Block, Block	Block, Block	9	Reception, Stand, Stand	Reception, Stand, Stand
4	Block, Block	Block, Block	10	Serve, Stand, Stand	Serve, Stand, Stand
5	Setting, Block	Setting, Attack	11	Setting, Attack	Setting, Attack
6	Stand, Stand, Stand	Stand, Stand, Stand	12	Stand, Stand, Stand	Stand, Stand, Stand

7 Experimental discussion

7.1 Posture recognition error

The experimental results of single-person posture estimation are shown in Table 1. The position recognition of the overall individual posture recognition is relatively precise. In sequence 1, the athlete approaches the volleyball net line in a posture where their feet are spread apart; their left-hand swings backward; and their right-hand swings forward. The system recognizes it as an attacking posture, but in reality, it is also an attacking posture. In sequence 2, the athlete is located on the side of the volleyball net line, with his left foot on the ground and his right foot in a jumping position. The overall center of gravity tilts back, and the system recognizes it as an attacking position, which is correct. In sequence 4, the athlete is located on the side of the volleyball net line, with both feet off the ground and jumping up as a whole, with both hands facing up in a blocking position. The system recognizes it as a blocking position, indicating correct recognition. In sequence 7, the player is located in the serving position, leaning forward as a whole, with the upper hand in a forward-pushing position. The system recognizes it as serving, but the actual serving position is. However, there are some errors in individual pose recognition. In sequence 5, the player is located on the right back of the court, with their feet in a squat position and their hands raised upwards. The system recognizes it as the serving position, but in reality, it is the receiving position. This is because the serving and receiving positions exhibit certain similar movements, and there is some error in the system. In sequence 12, the athlete is located on the left back of the court, with feet forked and hands down. The system recognizes it as a set action, but in reality, it is a standing action. However, compared to sequence 11, it is found that the athlete can better recognize standing actions in this scenario. Overall, it can be seen that the estimation of single-person posture is quite good.

The results of the multi-person pose estimation experiment are shown in Table 2. Overall, the system model can adapt to the situation of multi-person pose estimation. In sequence 2, both athletes in the detection area are located next to the volleyball net line, with similar postures. Both of them have feet in the air, and their hands are extended forward and above, presenting a posture of blocking the ball. The system recognizes both as blocking postures, and the recognition is correct. In sequence 8, the recognition object is the posture of three athletes, with their first left foot forked, their center of gravity tilted forward, and their hands facing down to present the set posture. The other two athletes, with their feet half squatted and forked, and their hands flat, are ready to receive the ball. The system sequentially recognizes the set and receiving posture. In sequence 11, the recognition area consists of two athletes. The left athlete is located on the side of the volleyball field, with their feet bent and their hands extended towards the ball to prepare for the set position, and the system recognizes it as the set position. The other athlete is located in the middle of the tennis ball near the net, with their feet bent open and their hands swinging to present an attacking posture. The system recognizes it as the attacking posture, but there is some error for multi-person recognition. In sequence 1, the recognition area is the posture of three people. The athlete on one side of the court presents a posture of normal feet apart and hands akimbo, and the system recognizes it as a standing motion. There are two athletes on the other side of the field, one of whom is in a running position with both hands open and feet striding forward, and the system recognizes it as an attacking position; the other athlete is standing normally with both feet and hands extended, and the system recognizes it as a set action. In fact, this athlete is in an attacking position with a small amplitude, which is sometimes difficult for the system to distinguish in the current posture. In sequence 5, both athletes are located in the net position. One athlete rises into the air with his hands waving upwards, which is recognized as an attacking position by the system but actually as a blocking position. The other athlete bends his feet, moves his hands forward, and leans back as a whole, presenting a set posture, which is recognized as a set posture by the system. This is mainly because there are some occlusions or incomplete features in the recognition area, which omit the athlete's posture and result in failure to recognize. Overall, the system has achieved good recognition results.

7.2 Confusion matrix of estimation results for different postures of the human body

For different pose types, the confusion matrix is shown in Figure 1. The horizontal axis represents the predicted type, followed by Attack, Block, Stand, Serve, Reception, and Setting from left to right. The vertical axis represents the actual category, and from top to bottom, it is the same as the horizontal axis. From the analysis in Figure 1, it can be seen that the highest proportion of correctly predicted actions was concentrated in Block, and the proportion of correctly predicted actions in the sample reached 98%, indicating a very impressive classification effect. The lowest concentration was in the Setting, and the proportion of correctly predicted samples reached 90%, indicating a higher number of classification errors. Among them, 1% was incorrectly predicted as Block; 3% was incorrectly predicted as Stand; 2% was predicted as Attack; 3% was predicted as Reception; 1% was predicted as Serve. The prediction accuracy of both Serve and Reception was 95%. For the Serve posture, 1% was predicted as Stand; 3% was predicted as Reception; 1% was predicted as Setting. For the Reception posture, 1% was predicted as Attack; 1% was predicted as Stand; 2% was predicted as Serve; 1% was predicted as Setting. Among them, Block is the easiest to detect because it is more pronounced than other human postures, resulting in higher estimation accuracy. Other categories have good classification performance and can meet practical application requirements.

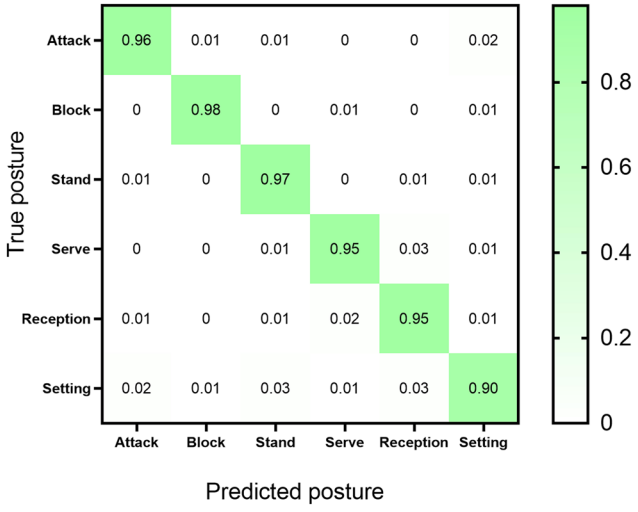

Figure 1: Confusion matrix for estimating results of different postures of the human body.

7.3 Accuracy and *F*1-score of pose estimation for different models

In order to explore the accuracy of pose estimation for different models, PoseNet, HRNet, AlphaPose, and OpenPose models were compared and analyzed for their performance in pose estimation, as shown in Figure 2. Overall, the OpenPose model had the highest performance in both bar and line plots. From Figure 2, it can be seen that the accuracy of the OpenPose model reached 98.23%, which was an improvement of 0.72% compared to the AlphaPose model. In addition, the HRNet model reached 95.37%, with the worst being the PoseNet model. Due to its lightweight design posture estimation accuracy, which was only 92.06%, it decreased by 6.17% compared to the OpenPose model. For the *F*1-score of the model, the OpenPose model reached 0.97, an improvement of 0.04 compared to the HRNet model; the AlphaPose model reached 0.96; the PoseNet model only reached 0.84. Overall, the pose estimation models OpenPose and AlphaPose have shown good performance in terms of accuracy and *F*1-score.

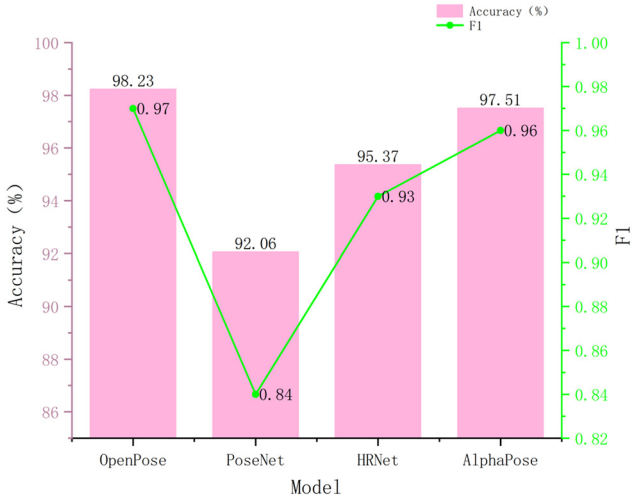

Figure 2: Accuracy and *F*1-score of pose estimation for different models.

7.4 Recall rate and precise value of pose estimation for different models

The comparison of recall and precise values for different pose estimation models is shown in Figure 3. In Figure 3, the red dot position represents precision, and the gray triangle position represents the recall rate. From the perspective of recall rate, the OpenPose model reached the highest, at 97.31%; the PoseNet model had the lowest recall rate, only 89.92%; the HRNet model achieved 94.20%, an increase of 4.28% compared to the PoseNet model. In addition, the AlphaPose model achieved 95.59%, a decrease of 1.72% compared to the OpenPose model. In terms of precision, the OpenPose model had the highest position of red dots in the figure, reaching 99.30%, which was 13.98% higher than the PoseNet model; the AlphaPose model achieved 98.91%, while the HRNet model achieved 96.74%. Overall, the pose estimation models OpenPose and AlphaPose perform better in terms of precision and recall.

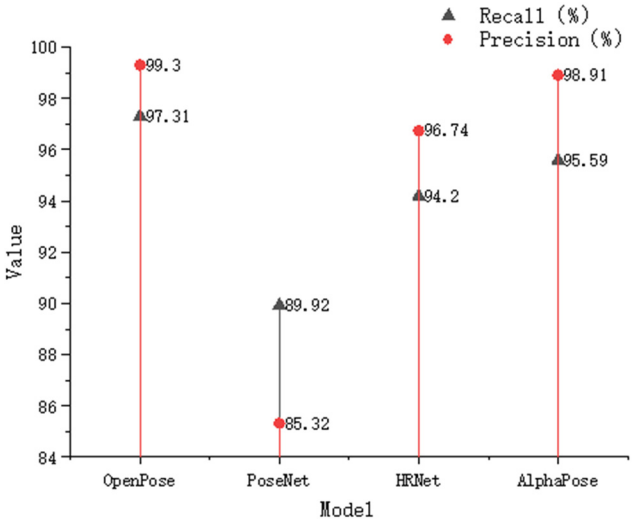

Figure 3: Recall rate and precise value of pose estimation for different models.

7.5 Real-time detection results of athlete posture

To explore the real-time performance of the model, a video was randomly selected for real-time pose recognition, and divided into two segments. The real-time specific posture recognition results of athletes are presented, with the corresponding segmented time periods of posture detailed in Table 3. In the first row, the time points recorded were 16:18:16, 16:18:19, and 16:18:20, respectively, corresponding to athletes numbered 1, 8, and 9. This segment, extracted from a volleyball match interval, was identified by the system as a standing posture. Moreover, it can be seen from the figure that the posture varied over time, indicating that real-time posture estimation can be achieved. In the second row, the time points were 16:40:53, 16:40:54, and 16:40:55. In the first image, the posture recognition is as Reception, and the estimated posture is correct. The second image is estimated to be in the Reception position, where the athlete steps forward with both feet in a semi-squat position and holds down with both hands in a receiving position. In the third picture, the athlete's posture is standing with both feet normal, and their hands are lowered and placed flat to present the set posture. The system recognizes it as Setting, indicating that the posture recognition is correct. In summary, it can be seen that the system can meet the requirements for real-time recognition of athletes' postures.

Table 3: Real-time pose estimation results

Time	Athlete number	Actual posture	Predict pose	Time	Athlete number	Actual posture	Predict pose
16:18:16	Number 1	Stand	Stand	16:40:53	Number 11	Reception	Reception
	Number 8	Stand	Stand		Number 12	Reception	Reception
	Number 9	Stand	Stand				
16:18:19	Number 1	Stand	Stand	16:40:54	Number 11	Reception	Reception
	Number 8	Stand	Stand		Number 12	—	
	Number 9	Stand	Stand				
16:18:20	Number 1	Stand	Stand	16:40:55	Number 11	Setting	Setting
	Number 8	Stand	Stand		Number 12	—	
	Number 9	Stand	Stand				

7.6 Comparison of processing speed and parameter quantity between different models

In order to delve deeper into the overall real-time performance of the model, the processing speed and parameter quantity of the OpenPose-DeepSORT, PoseNet-DeepSORT, HRNet-DeepSORT, and AlphaPose-DeepSORT models were compared and analyzed, as shown in Figure 4. The gray line in the figure represents the processing speed, while the red line represents the size of the parameter quantity. In terms of processing speed, the OpenPose-DeepSORT model reached 16.7 frames/s; the processing speed of the HRNet-DeepSORT model reached 11.3 frames/s; the slowest processing speed was the AlphaPose-DeepSORT model, which was only 8.6 frame/s and had the worst effect; the PoseNet-DeepSORT had the fastest processing speed, reaching up to 19.9 frame/s. This is because the PoseNet pose estimation model adopts a lightweight setting, which shows excellent performance in processing speed despite a decrease in accuracy. From the perspective of parameter quantity, the PoseNet-DeepSORT model had the least parameter quantity, only requiring 14.5 MB, which was 21.5 MB less than the OpenPose-DeepSORT model. Although the OpenPose-DeepSORT model had fewer parameters than the PoseNet-DeepSORT model, it reduced the number of parameters by 16.7MB compared to the HRNet-DeepSORT model; the highest was the AlphaPose-DeepSORT model, with the worst performance and a parameter count of up to 61.8 MB. Overall, the PoseNet-DeepSORT model performs best in terms of processing speed.

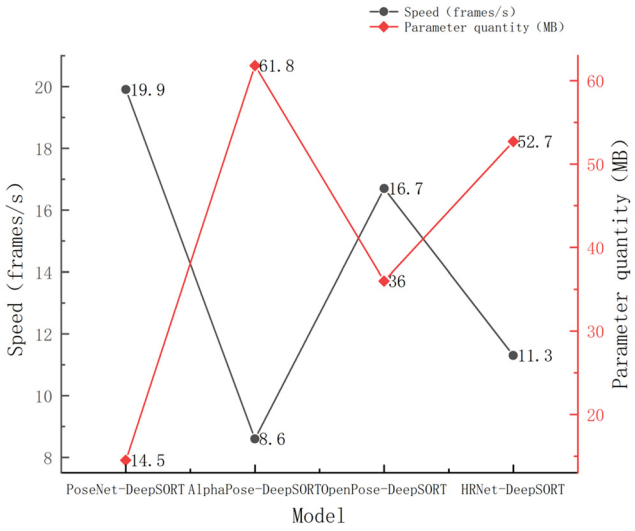

Figure 4: Comparative analysis of processing speed and parameter quantity of different models.

7.7 Model robustness

In real volleyball matches, there are various UAV shooting angles and situations where multiple people are obstructing. In order to explore the robustness of the model, a comparative analysis was conducted on the pose recognition performance of the model under Gaussian noise, rotation of 30 degrees, random cropping, and multiple occlusion, as shown in Figure 5. In Figure 5, light yellow represents estimated accuracy, and light blue represents processing speed. Overall, the estimated accuracy of the four scenarios was not significantly different, reaching over 95%, and the processing speed fluctuated slightly. From the perspective of pose estimation accuracy, the pose recognition accuracy in the presence of Gaussian noise reached 95.65%, a decrease of 0.82% compared to the case of multiple occlusion. However, in the case of rotation and cropping, the prediction accuracy of the model reached over 97%, with a relatively small impact. In terms of processing speed, the fastest was achieved at 15.2 frames/s under Gaussian noise and 12.5 frames/s under multiple occlusion. In addition, the model performed well under rotation and cropping, reaching over 13.0 frames/s. In summary, it can be seen that this model can better adapt to posture recognition in different situations.

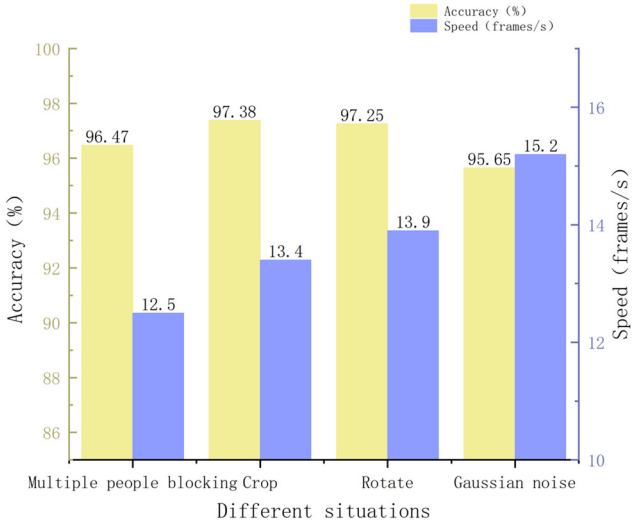

Figure 5: Comparison of model robustness.

8 Conclusions

This article used a combination of OpenPose and DeepSORT models for real-time pose estimation and tracking of volleyball videos. The OpenPose algorithm is employed for accurate pose estimation of human body regions, while the DeepSORT model facilitates real-time tracking of the detected human pose information. Utilizing Unmanned Aerial Vehicles (UAVs) as carriers, the YOLOv4 object detection model is integrated to achieve real-time human pose detection on standardized images. The experimental results demonstrate the model's commendable pose recognition accuracy and real-time performance. However, the study acknowledges certain shortcomings, primarily related to the complexity of the combined model, necessitating further improvements in its real-time capabilities. Future endeavors aim to enhance the model's performance by developing a lightweight structure. The integration of OpenPose, DeepSORT, and YOLOv4 models demonstrates promising results in real-time pose estimation and tracking of volleyball players. Acknowledging the identified shortcomings, future efforts will focus on refining the model's structure to enhance real-time performance, ensuring its applicability in dynamic sports environments.

Funding information: This work was supported by The Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJZD-K202302901); Optimization of key technology of visual object detection based on complex moving image and its application in teaching and training; The Chongqing Preschool Education College High-level Talent Research Workstation Project: Children's physical health and sports ability promotion workstation. (Grant Number: 2023GZZ-001).

Author contributions: Long Liu was responsible for writing papers, analyzing data, checking language, and implementing research projects; Yuxin Dai was responsible for language proofreading, data organization, image organization, and table processing of the paper; and Zhihao Liu was responsible for the image organization, model analysis, and language polishing of the paper. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The author(s) declare(s) that there is no conflict of interest regarding the publication of this article.

Data availability statement: The data used to support the findings of this study are available from the corresponding author upon request.

References

[1] Zheng C, Wu W, Chen C, Yang T, Zhu S, Shen J, et al. Deep learning-based human pose estimation: A survey. ACM Comput Surv. 2023;56(1):1–37. doi: 10.1145/3603618.

[2] Rohan A, Rabah M, Hosny T, Kim SH. Human pose estimation-based real-time gait analysis using convolutional neural network. IEEE Access. 2020;8:191542–50. doi: 10.1109/ACCESS.2020.3030086.

[3] Xu W, Chatterjee A, Zollhoefer M, Rhodin H, Fua P, Seidel HP, et al. Mo 2 cap 2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. IEEE Trans Vis Comput Graph. 2019;25(5):2093–101. doi: 10.1109/TVCG.2019.2898650.

[4] Kamel A, Liu B, Li P, Sheng B. An investigation of 3D human pose estimation for learning Tai Chi: A human factor perspective. Int J Hum Comput Interact. 2019;35(4–5):427–39. doi: 10.1080/10447318.2018.1543081.

[5] Dong J, Fang Q, Jiang W, Yang Y, Huang Q, Bao H, et al. Fast and robust multi-person 3d pose estimation and tracking from multiple views. IEEE Trans Pattern Anal Mach Intell. 2021;44(10):6981–92. doi: 10.1109/TPAMI.2021.3098052.

[6] Felipe JL, Garcia-Unanue J, Viejo-Romero D, Navandar A, Sanchez-Sanchez J. Validation of a video-based performance analysis system (Mediacoach®) to analyze the physical demands during matches in LaLiga. Sensors. 2019;19(19):4113–22. doi: 10.3390/s19194113.

[7] Luvizon DC, Picard D, Tabia H. Multi-task deep learning for real-time 3D human pose estimation and action recognition. IEEE Trans Pattern Anal Mach Intell. 2020;43(8):2752–64. doi: 10.1109/TPAMI.2020.2976014

[8] Li M, Zhou Z, Liu X. Multi-person pose estimation using bounding box constraint and LSTM. IEEE Trans Multimed. 2019;21(10):2653–63. doi: 10.1109/TMM.2019.2903455.

[9] Sengupta A, Jin F, Zhang R, Cao S. mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs. IEEE Sens J. 2020;20(17):10032–44. doi: 10.1109/JSEN.2020.2991741.
[10] Cui H, Dahnoun N. High precision human detection and tracking using millimeter-wave radars. IEEE Aerosp Electron Syst Mag. 2021;36(1):22–32. doi: 10.1109/MAES.2020.3021322.
[11] Yi X, Zhou Y, Xu F. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. ACM Trans Graph. 2021;40(4):1–13. doi: 10.1145/3450626.3459786.
[12] Xu-Wei DU, Dong C, Hua-Jiang LIU, Zhaokun M, Qianqian Y. Real-time hand tracking based on YOLOv4 model and Kalman filter. J China Univ Posts Telecommun. 2021;28(3):86–94. doi: 10.19682/j.cnki.1005-8885.2021.0011.
[13] Wu CH, Wu TC, Lin WB. Exploration of applying pose estimation techniques in table tennis. Appl Sci. 2023;13(3):1896–909. doi: 10.3390/app13031896.
[14] Echeverria J, Santos OC. Toward modeling psychomotor performance in karate combats using computer vision pose estimation. Sensors. 2021;21(24):8378–404. doi: 10.3390/s21248378.
[15] Xu J, Tasaka K. Keep your eye on the ball: detection of kicking motions in multi-view 4K soccer videos. ITE Trans Media Technol Appl. 2020;8(2):81–8. doi: 10.3169/mta.8.81.
[16] Naik BT, Hashmi MF. YOLOv3-SORT: detection and tracking player/ball in soccer sport. J Electron Imaging. 2023;32(1):011003. doi: 10.1117/1.JEI.32.1.011003.
[17] Razzok M, Badri A, El Mourabit I, Ruichek Y, Sahel A. Pedestrian detection and tracking system based on Deep-SORT, YOLOv5, and new data association metrics. Information. 2023;14(4):218–33. doi: 10.3390/info14040218.
[18] Sajina R, Ivasic-Kos M. 3D pose estimation and tracking in handball actions using a monocular camera. J Imaging. 2022;8(11):308–41. doi: 10.3390/jimaging8110308.
[19] Chen W, Jiang Z, Guo H, Ni X. Fall detection based on key points of human-skeleton using openpose. Symmetry. 2020;12(5):744–60. doi: 10.3390/sym12050744.
[20] Tsai YS, Hsu LH, Hsieh YZ, Lin SS. The real-time depth estimation for an occluded person based on a single image and OpenPose method. Mathematics. 2020;8(8):1333–52. doi: 10.3390/math8081333.
[21] Kim IH, Jung IH. A study on korea sign language motion recognition using openpose based on deep learning. J Digit Contents Soc. 2021;22(4):681–7. doi: 10.9728/dcs.2021.22.4.681.
[22] Lee MFR, Chen YC, Tsai CY. Deep learning-based human body posture recognition and tracking for unmanned aerial vehicles. Processes. 2022;10(11):2295–317. doi: 10.3390/pr10112295.
[23] Xiang H. Lightweight open pose based body posture estimation for badminton players. For Chem Rev. 2022;339–50.
[24] Meimetis D, Daramouskas I, Perikos I, Hatzilygeroudis I. Real-time multiple object tracking using deep learning methods. Neural Comput Appl. 2023;35(1):89–118. doi: 10.1007/s00521-021-06391-y.
[25] Zhang G, Yin J, Deng P, Sun Y, Zhou L, Zhang K. Achieving adaptive visual multi-object tracking with unscented kalman filter. Sensors. 2022;22(23):9106–23. doi: 10.3390/s22239106.
[26] Gong X, Le Z, Wu Y, Wang H. Real-time multiobject tracking based on multiway concurrency. Sensors. 2021;21(3):685–702. doi: 10.3390/s21030685.
[27] Z Yuemeng L Huigang. Low altitude unmanned aerial vehicle detection and tracking based on optimized YOLOv4 algorithm. Laser Optoelectron Prog. 2022;59(12):1215017. doi: 10.3788/LOP202259.1215017.
[28] Wainwright R, Shenfield A. Human activity recognition making use of long short-term memory techniques. Athens J Sci. 2019;6(1):19–34. doi: 10.30958/ajs.
[29] Franco A, Magnani A, Maio D. A multimodal approach for human activity recognition based on skeleton and RGB data. Pattern Recognit Lett. 2020;131:293–9. doi: 10.1016/j.patrec.2020.01.010.
[30] Tufek N, Yalcin M, Altintas M, Kalaoglu F, Li Y, Bahadir SK. Human action recognition using deep learning methods on limited sensory data. IEEE Sens J. 2019;20(6):3101–12. doi: 10.1109/JSEN.2019.2956901.