Sports Action Detection and Counting Algorithm Based on Pose Estimation and Its Application in Physical Education Teaching

Zhengyuan Song[1], Zhonghai Chen[2*]
[1]College of Physical Education, Chongqing Technology and Business University, Chongqing 400067, China
[2]Physical Education Institute, Yanching Institute of Technology, Sanhe 065201, China
E-mail: sszhyy163@163.com
[*]Corresponding author

Keywords: pose estimation, sports, action recognition, counting, ViBE

Received: March 15, 2024

Accurate motion detection and counting are important for improving training effectiveness and preventing sports injuries in physical education teaching and training. Traditional analysis of sports movements mainly relies on observing coaches and the subjective feelings of athletes. This method is not only time-consuming and labor-intensive, but also susceptible to personal experience and judgment biases. A 3D bone keypoint detection algorithm based on the pose estimation model Visual Background Extractor was proposed to address this issue. Each repeated action was divided and scored using a penalty function by analyzing the continuous action information on the time series. Meanwhile, the lightweight application of this function was proposed based on OpenPose. The performance test confirmed that the detection accuracy of the playground and outdoor space was the same, both at 96%, which was the highest among all environments. Although the height, gender, and clothing of the participants varied, these factors did not significantly affect the performance of the algorithm from an individual performance perspective, with accuracy ranging from 92% to 94%. These experiments confirm that the proposed motion detection and counting model based on pose estimation has high robustness and reliability under different environmental conditions.

Povzetek: Algoritem za zaznavanje in štetje športnih akcij temelji na oceni drže in uporablja model Visual Background Extractor za izboljšanje vadbe in preprečevanje poškodb.

1 Introduction

Accurate action detection and counting are important for improving training effectiveness and preventing sports injuries in physical education teaching and training. As technology advances, action detection and counting algorithms based on pose estimation are research hotspots. These algorithms can provide real-time feedback on the quality of exercise execution by analyzing the posture and movements of athletes. These algorithms can help coaches and athletes optimize training methods and improve athletic performance [1]. Traditional analysis of sports movements mainly relies on observing coaches and the subjective feelings of athletes. This method is not only time-consuming and labor-intensive, but also susceptible to personal experience and judgment bias [2]. Pose estimation-based algorithms provide new possibilities for automatic detection and analysis of motion actions with the advancement of computer vision and machine learning technology. These algorithms can accurately identify and count specific movements in real-time, such as jumping, weightlifting, and yoga postures, by analyzing video or sensor data [3]. However, there are still many challenges when applying these algorithms to practical physical education teaching and training. For example, how to ensure that the algorithm can maintain high accuracy and robustness in different environments and conditions, how to handle the recognition and counting of complex actions, and how to make the algorithm adapt to the individual differences of different athletes. In addition, there is an important research topic on how to effectively integrate the output of algorithms into teaching and training to improve teaching quality and athlete learning efficiency. In view of this, the research aims to explore sports action detection and counting algorithms based on pose estimation and study their application in physical education teaching. This study develops an algorithm that can accurately recognize and count multiple motion actions by combining advanced computer vision technology and machine learning methods.

The study consists of five parts. Firstly, an introduction is given to physical education teaching, sports training, action detection, and counting algorithms. Secondly, an action detection model and counting algorithm are constructed based on pose estimation. Then the performance of the model and algorithm is tested and analyzed. Furthermore, the results obtained from the research are discussed. Meanwhile, the expectations and challenges in practical applications are pointed out. Finally, a discussion is made on the above content.

2 Related works

Research on motion recognition is gradually becoming popular with the popularization of image algorithms and the popularity of miniaturized wearable devices. Bingzhu et al. used different feature analysis and combined the feature signals of lying and sitting positions to improve the classifying performance of robots for lower limb movements. A trained motion decoder was obtained through sEMG feature extraction and pattern recognition. Control commands were sent to the robot to drive the lower limbs for corresponding rehabilitation training. These experiments confirmed the effectiveness of control methods with sEMG signals [4]. Pengyun et al. believed that human motion recognition based on ultra wideband through wall radar faced limitations in terms of samples and perspectives. Therefore, they proposed a multi-radar cooperative human motion recognition model based on transfer learning ResNeXt network and ensemble learning. These experiments confirmed that ResNeXt networks based on set learning could achieve higher recognition accuracy compared to fusion models based on single-view radar [5]. Neural network is a mathematical model that mimics the structure and function of biological neural networks. Neural networks have simple decision-making and judgment abilities similar to humans, which can provide better results in image and speech recognition. The neural network is divided into Convolutional Neural Network (CNN), Generative Adversarial Network (GAN), recurrent neural networks, etc. according to different connection methods. The applications of neural networks have become increasingly widespread with the development of artificial intelligence. Nemani et al. investigated the application of deep learning in predicting the remaining life of bearings. They determined the bearing fault threshold based on ISO standards and proposed a two-stage Long Short-term Memory (LSTM) model for extracting fault feature signals of bearings. Gaussian layers were embedded in LSTM for parameter optimization. These experiments confirmed that this model had good accuracy in predicting bearing life [6]. Chen, Beijing et al. proposed a novel Xception-LSTM by integrating spatiotemporal attention mechanism with ConvLSTM to improve the accuracy of fake face detection. ConvLSTM was introduced to consider frame structure information and modeled temporal information. These excellent performance results confirmed that this algorithm performed better than existing algorithms [7].

Kaadoud et al. used an internal state clustering algorithm in LSTM to address the transparency and interpretability in machine learning algorithms and understand the results from simple clues and rules. They studied the hidden states of spatial extraction knowledge and established and validated automatic sequences for extraction based on basic syntax. These experiments confirmed that sequences extracted from the original syntax had high recognition rates [8]. Le et al. used LSTM to study the channel access of vehicles in wireless tram networks, transforming the access control into a non-Markov problem. LSTM was integrated into Q-learning networks, and a vehicle connecting method was put forward using deep recursion in Q-learning networks. Simulation experiments confirmed that this algorithm had higher stability and efficiency compared to the benchmark scheme [9]. Wu et al. combined recurrent neural networks and LSTM to construct a system for predicting dynamic gestures through joint coordinate features. This model achieved the highest accuracy of 99.31%, indicating the superior recognition performance [10]. Jia et al. proposed a motorcycle helmet detection method based on YOLOv5 for detecting motorcycle driver helmet. The soft-NMS was used instead of NMS to fuse the YOLOv5 detector. The experiment achieved 97.7% mAP, 92.7% F1 score, and 63 Frames Per Second (FPS), which was superior to other state-of-the-art detection methods [11]. Li and Ye constructed a model for predicting network wireless traffic by integrating LSTM and recurrent neural networks. These experiments confirmed that this model had good prediction accuracy and training speed, met the needs of wireless network traffic prediction, and had good application prospects [12]. Dudi and Rajesh proposed a CNN-based plant leaf classification model to facilitate the classification of plant leaves and identify plant types. They improved classification accuracy and validated the effectiveness of the model by introducing hybrid whale optimization algorithm based on shark odor [13]. In recent years, the combination of sports and computer teaching becomes a hot research field. Mcdonough et al. designed a control experiment to improve the intervention effect of the school dance sports game education model and enhance the enjoyment and self-efficacy of urban minority students. Urban minority students had a higher happiness in the group exercise mode, which was an effective dance sports game intervention mode [14]. Liu et al. investigated the effectiveness of the Small Private Online Course (SPOC) teaching model and conducted experiments using embryology courses as an example. These experiments confirmed that SPOC teaching improved students' average professional grades and enhanced their learning motivation. This indicated that the SPOC teaching model was scientifically reasonable and could be promoted and applied in medical courses [15]. The literature summary is shown as follows Appendix 1.

In summary, current research has further improved the effectiveness of action recognition and training through image recognition and wearable devices. The advancement of action detection technology is achieved through models such as neural networks. Meanwhile, the application of different educational intervention modes improves the effectiveness of physical education teaching. However, the current method lacks specificity and efficiency in identifying actions in sports testing. As a result, it is difficult to meet the needs of physical education teaching to a certain extent. A motion posture estimation algorithm based on bilateral filtering and ViBE is developed to address this issue, and a penalty function is used to complete the scoring. Finally, this algorithm is applied in physical education teaching. Therefore, the effectiveness of sports detection can be further enhanced in physical education teaching, the efficiency of student sports learning can be improved, sports posture can be corrected, and sports injuries can be reduced, thus promoting the development of physical education teaching.

3 Construction of sports action detection and counting algorithm based on pose estimation

Current research regards sports action detection as an image classification problem. The action detection has limitations in recognizing repetitive actions and cannot accurately distinguish the frame intervals of a single action in the image. A 3D bone keypoint detection algorithm based on the pose estimation model, namely Visual Background Extractor (ViBE), is proposed to deal with this challenge. Each repeated action is divided and scored using a penalty function by analyzing the continuous action information in the time series.

3.1 Construction of motion pose estimation algorithm based on bilateral filtering and ViBE

There is a significant difference between motion recognition and detection counting. Action recognition mainly focuses on recognizing specific postures. Action detection counting should recognize posture and accurately divide each action’s beginning and end, requiring higher accuracy. For example, algorithmic inaccuracies may lead to incorrect division of the action process and repetitive counting in repetitive movements such as sit ups [16]. Motion is a continuous process in terms of timing, which is composed of a combination of behaviors from multiple frames of images. Therefore, whether a motion is qualified is a continuous cumulative process should be determined, and temporal information cannot be ignored. There is also a challenge of algorithm identification and screening of non-conforming actions in continuous processes. If external force is used to support the ground during sit ups, there are unqualified actions in certain frames of images during a certain movement process composed of multiple frames of images. Although this behavior may only exist in a few short frames of the image during the motion process, the algorithm should accurately recognize and filter out the motion process. Exercise is different from conventional behaviors such as walking, running, and jumping. The human body is almost lying flat on the ground during physical exercises such as sit ups. Meanwhile, the pixel range occupied by the human body in the picture is very small. There is also a lot of body overlap and occlusion during the exercise, which is a great challenge for posture estimation algorithms. The research is based on pose estimation technology as the algorithm framework and designed for sports detection and counting scenarios. Figure 1 shows the overall algorithm process.

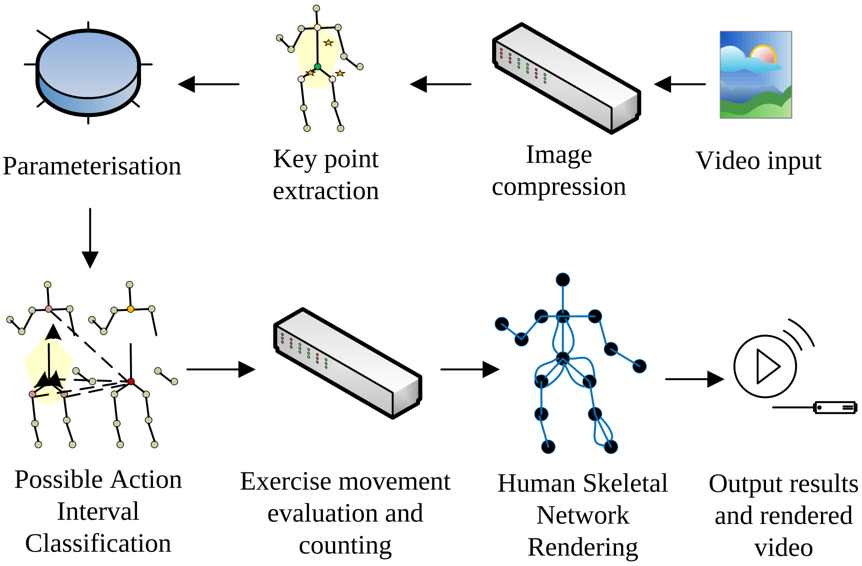

Parameterisation
Key point extraction
Image compression
Video input
Possible Action Interval Classification
Exercise movement evaluation and counting
Human Skeletal Network Rendering
Output results and rendered video

Figure 1: Overall algorithm process

The proposed algorithm is based on pose estimation technology, which is designed for the detection and counting of sports movements. Firstly, the input video is preprocessed, including encoding format adjustment, image denoising, and scaling, to ensure efficient subsequent processing. Then a pose estimation model is used to extract the coordinates of key points (such as elbows, wrists, etc.). Afterwards, the skeleton keypoint coordinates of each frame is used to calculate parameters such as body angle, arm angle, knee Euclidean distance, angle change curve, etc. Then the possible frame rate range for each motion process is accurately divided based on the main parameters. A penalty function is used to evaluate the quality of actions and select qualified actions. Here, the penalty function is defined by the algorithm that can evaluate the undesirable or unreasonable behavior and generate corresponding penalty scores. At the same time, the penalty function will be added to each possible action frame set to influence the model. The penalty weight adjusts the parameter that affects the influence of the penalty function. The higher the value, the more severe the punishment for the model's bad behavior. The threshold parameter defines when to trigger the penalty function and how to determine the threshold for the penalty score. That is, when the model's error exceeds this threshold, the penalty function will be triggered. Finally, a human skeleton network is generated using the Skinned Multi-Person Linear Model (SMPL) model to achieve visual representation of actions [17-18]. It is necessary to preprocess the images in advance to ensure uniform input image size and to ensure the performance of key point detection in 3D models. The study first modifies the encoding format of sports videos by using bilateral filtering to remove noise and interference from the images [19-22]. Figure 2 is the schematic diagram of bilateral filtering.

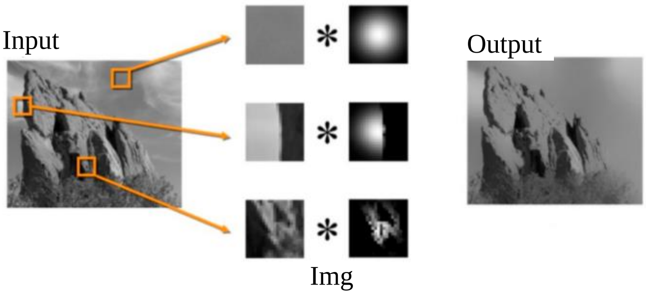


Figure 2: Schematic diagram of bilateral filtering

In Figure 2, bilateral filtering is an efficient image filtering technique that considers both spatial proximity and pixel value similarity, which can preserve edges while smoothing the image. This filter processes each pixel, weighting the pixels within its neighborhood based on spatial distance and pixel value differences. Therefore, bilateral filtering can effectively remove noise without blurring edges, which is widely used in fields such as image denoising, texture smoothing, and detail enhancement. The calculation of bilateral filtering is represented by equation (1).

$$I_{filtered}(x,y)=\frac{1}{W_p}\sum_{(i,j)\in S} I(i,j)w_{i,j}(x,y) \quad (1)$$

In equation (1), $I(i,j)$ represents the image to be filtered. (x,y) refers to the current pixel position. S means the size of the filter. W_p is the normalization factor. $w_{i,j}(x,y)$ is the weight of pixel (i,j), represented by equation (2).

$$w_{i,j}(x,y)=w_d(x,y)\square w_r(I(i,j),I(x,y)) \quad (2)$$

In equation (2), $w_d(x,y)$ represents the spatial domain weight. $w_r(I(i,j),I(x,y)$ is the pixel domain weight. The definition of $w_d(x,y)$ is represented by equation (3).

$$w_d(x,y)=\exp\left(-\frac{(i-x)^2+(j-y)^2}{2\sigma_d^2}\right) \quad (3)$$

In equation (3), σ_d^2 represents the square of the standard deviation of spatial distance. The calculation of $w_r(I(i,j),I(x,y)$ is represented by equation (4).

$$w_r(I(i,j),I(x,y)= \exp\left(-\frac{\left(I(i,j)-I(x,y)\right)^2}{2\sigma_r^2}\right) \quad (4)$$

In equation (4), σ_r^2 represents the standard deviation of the pixel value similarity.

Afterwards, the image is scaled and resolution adjusted using bilinear interpolation to adjust the image size without losing the original image information [23-25]. The basic idea of bilinear interpolation is to perform two linear interpolations separately to obtain pixel values in a new image. This increases the original image pixels to improve image clarity during image scaling operations. Two linear interpolations are represented by equation (5).

$$\begin{cases} f(x,y_1) \approx \dfrac{x_{2\_}x}{x_2 - x_1} f(Q_{11}) + \dfrac{x - x_1}{x_2 - x_1} f(Q_{21}) \\ f(x,y_2) \approx \dfrac{x_{2\_}x}{x_2 - x_1} f(Q_{12}) + \dfrac{x - x_1}{x_2 - x_1} f(Q_{22}) \end{cases} \quad (5)$$

In equation (5), the value of the unknown function at point P is (x, y). The function f is known to have four values of $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$, and $Q_{22} = (x_2, y_2)$. The linear interpolation in the y direction is represented by equation (6).

$$f(x,y) \approx \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(Q_{11}) & f(Q_{12}) \\ f(Q_{21}) & f(Q_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \quad (6)$$

In equation (6), the directions of two linear interpolations can be interchanged. The position of skeletal joints in the human body is the key to body activity recognition. Meanwhile, the relative positions are fixed in the human body, well reflecting the human body's moving status. At present, there are mainly two types of pose estimation methods: 2D and 3D pose estimation models. The calculation time of the 2D pose estimation model OpenPose is less than that of the 3D pose estimation model ViBE. Meanwhile, the calculation time for a one-minute motion action video is about twice as fast. However, the 2D pose estimation model ignores the depth information of the image throughout the entire activity due to the specific motion poses in most motion action videos. Meanwhile, the lack of depth information can affect the accuracy of feature extraction for bone key points due to changes in camera angle and the relationship between the camera and body position. This is mainly due to the fact that there may be mutual occlusion on both sides of the body in most physical education teaching and exercise processes when the image acquisition device is located on the side of the body. If there is a lack of image depth information, it is easy to encounter mutual occlusion between the body closer to the camera and the body farther away during the bone keypoint extraction. This phenomenon can affect the accuracy of model pose estimation. Therefore, the study uses a human pose estimation model to extract skeletal joints' key points. The 3D pose estimation model is based on ViBE. The 2D pose estimation model used in lightweight devices is based on OpenPose. Figure 3 is a schematic diagram of ViBE construction.

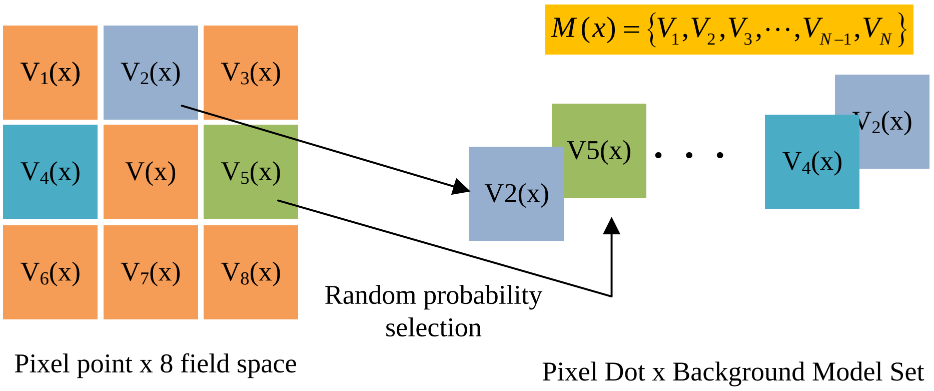

$M(x) = \{V_1, V_2, V_3, \cdots, V_{N-1}, V_N\}$
$V_1(x)$
$V_2(x)$
$V_3(x)$
$V_4(x)$
V(x)
$V_5(x)$
$V_6(x)$
$V_7(x)$
$V_8(x)$
V2(x)
V5(x)
$V_2(x)$
$V_4(x)$
Random probability selection
Pixel point x 8 field space
Pixel Dot x Background Model Set

Figure 3: Schematic diagram of ViBE model construction

In Figure 3, ViBE is an algorithm used for background extraction in videos. First, the background model is initialized through the initial frame of the video, and multiple sample values are randomly selected for each pixel. When processing subsequent frames, ViBE compares each pixel value with the sample in the background model and determines whether the pixel is foreground or background based on similarity. This algorithm regularly updates the background model and replaces sample values to adapt to changes in the scene. The advantages of ViBE lie in its efficiency and adaptability to dynamic scenes, but false positives may occur when dealing with extreme changes [26-28]. The VIBE network is trained using sports evaluation movements. 100 sets of sit up video data are extracted from the HMDB51 dataset for training taking sit ups as an example.

3.2 Construction of detection and counting module for sports projects based on skeletal key points

A motion detection and counting module is designed for the detection and counting of sports projects after obtaining the image coordinates of bone key points. Figure 4 shows the algorithm flow of sports action detection and counting module.

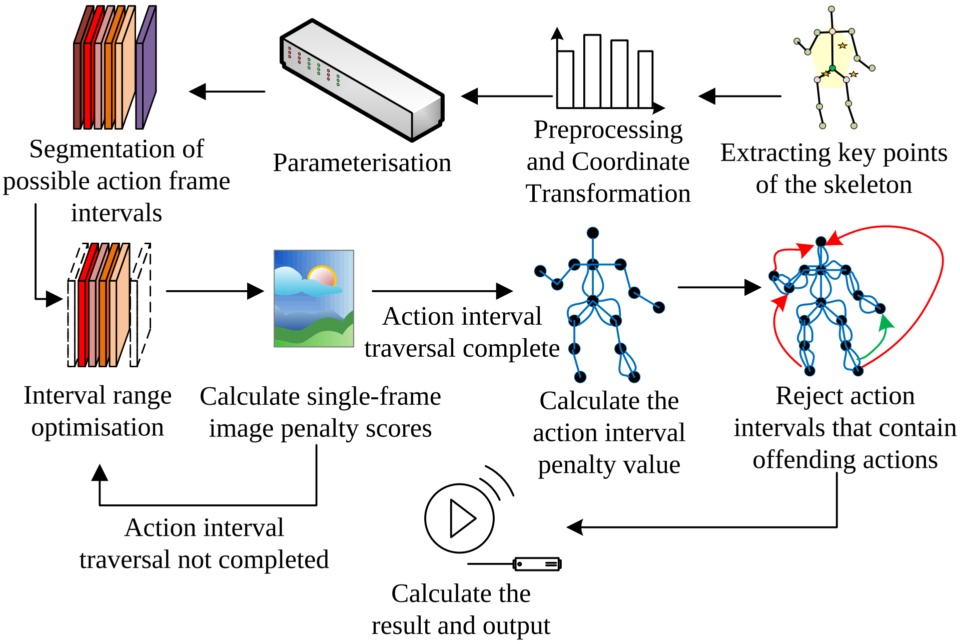

Segmentation of possible action frame intervals
Parameterisation
Preprocessing and Coordinate Transformation
Extracting key points of the skeleton
Interval range optimisation
Calculate single-frame image penalty scores
Action interval traversal complete
Calculate the action interval penalty value
Reject action intervals that contain offending actions
Action interval traversal not completed
Calculate the result and output

Figure 4: Algorithm flow of sports action detection and counting module

In Figure 4, the core of the motion detection and counting algorithm is to first capture the coordinates of bone keypoints through the pose estimation network. Subsequently, these coordinates are transformed and standardized to unify the processing of images at different scales. Next, the action related parameters of each frame of the image are calculated based on the coordinates of key points. Then the possible intervals for the actions are divided. These intervals are optimized to ensure the accuracy of the action intervals to improve accuracy. A penalty function is applied to evaluate each action interval due to the possibility of non-compliant actions. Therefore, penalty scores for multiple frames of images can be accumulated, and intervals that exceeded the set threshold can be eliminated. In the end, the most accurate action count and action improvement suggestions based on penalty scores are output. The joint coordinates are stored in the matrix after obtaining the 3D joint point data. This movement mainly revolves around the buttocks taking sit ups as an example. Therefore, the focus is on the coordinate points of the buttocks in each frame of the image. If a frame lacks hip coordinates, it is considered abnormal data and removed, which can reduce jitter during the rendering process and improve the accuracy of action detection and counting. Subsequently, a clear coordinate matrix following the image pixel coordinate system is obtained. This article converts the coordinate matrix into a normalized matrix based on human joints considering that differences in camera parameters and shooting angles may affect parameter calculations. Specifically, the coordinate system is adjusted to allow this algorithm to describe motion without being limited by shooting conditions with the buttocks as the pivot point. Actions can be described more accurately by transforming the coordinate matrix into a normalized matrix based on human joints. This process involves calculating the position and direction of each joint and describing them using a unified coordinate system. This normalization allows for comparison and analysis of data collected from different locations. In addition, the selection of this coordinate system makes it easier to classify and recognize different movements, thereby improving the performance of the system. Suitable calculation parameters are selected for different sports movements. Rotation is detected by calculating cosine similarity, represented by equation (7).

$$\theta_i = \cos^{-1}\left(\frac{\vec{t}_i \times \vec{y}_i}{\left\|\vec{t}_i\right\| \times \left\|\vec{y}_i\right\|}\right) \tag{7}$$

In equation (7), $\vec{t}_i$ and $\vec{y}_i$ represent the vector connecting the joint points. Euclidean distance is chosen to detect distance, represented by equation (8).

$$d(\vec{t}_i,\vec{y}_i)=\sqrt{\sum_{j=1}^{3}(\vec{t}_{i,j}-\vec{y}_{i,j})^2} \qquad (8)$$

In equation (8), $\vec{t}_{i,j}$ and $\vec{y}_{i,j}$ are the intercepts of two points in direction y, respectively. Figure 5 shows several key joint points and angles in this study.

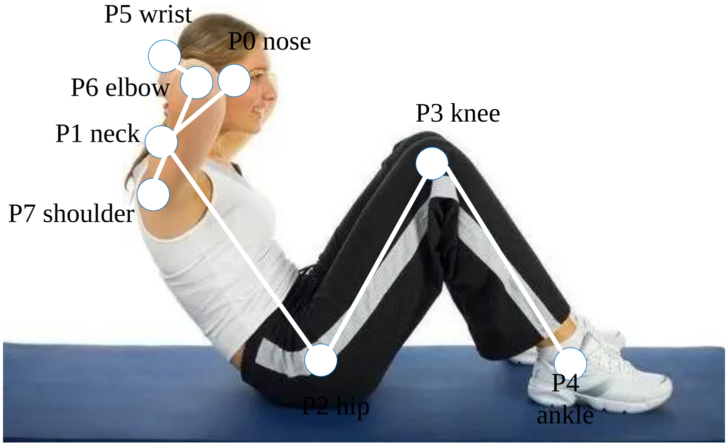


Figure 5: Joint key points and angles

In Figure 5, $\alpha_{body}=\angle p1p2p4$, $\alpha_{knee}=\angle p3p2p4$, $\alpha_{ankle}=\angle p4'p2p4$, and $\alpha_{wrist}=\angle p5p2p1$ represent the angles between the torso, knees and ground, ankles and ground, and wrists and upper body during exercise, respectively. The compliance of the movement is determined based on the above angles during exercise. The effective frames in the video are divided into multiple sets of candidate event frame datasets after calculating the key parameters for each frame. Each frame dataset contains a portion of frames pertaining to the target action event. In action events, it is also necessary to screen and count illegal actions. Taking sit ups as an example, assisting other parts of the body in completing actions is a violation, and the loss values and adjustment actions of other parts need to be calculated. Penalty factors are set for each loss value to form a penalty function, and weighted coefficients are set on the foundation of each joint's participation in sit ups. The penalty value is represented by equation (9).

$$S=\sum_{s_i}^{e_i}\left(\alpha\cdot loss_{wrist}+\beta\cdot loss_{knee}+\gamma\cdot loss_{ankle}\right) \qquad (9)$$

In equation (9), $loss_{wrist}$, $loss_{knee}$, and $loss_{ankle}$ represent the loss values of the wrist, knee, and ankle, respectively. e_i and s_i mean the end frame and start frame of the action, respectively. α, β, and γ are penalty factors. The loss values of various body parts are represented by equation (10).

$$loss_j=\sqrt{(\alpha_j-threshold)^2} \qquad (10)$$

In equation (10), α_j represents the current parameter values of each part. $threshold$ refers to the threshold. Table 1 shows the threshold values and penalty factors for various body parts parameters.

Table 1: Parameter thresholds and penalty factors for each part

Body parts	Penalty threshold	Penalty factor
Hand	35°	0.5
Knee	15°	0.2
Ankle	1.24	0.4

3.3 Lightweight network construction based on OpenPose

The study considers uploading videos to the cloud for processing. The powerful computing power of servers is utilized to parallelly process multiple action detection tasks. However, this also consumes bandwidth, and cloud services continue to consume server resources. An end-to-end cloud integration strategy is adopted to cope with short-term high concurrency pose estimation tasks. The research attempts to reduce the model size and accelerate the solution speed to deal with the high memory consumption and slow computing speed of the framework, and pre-training and other tasks are placed on the cloud for processing. The research will optimize algorithms and develop lightweight motion action detection and counting models to adapt to the limitations of computing power of edge devices. The pose estimation algorithm is optimized based on the OpenPose backbone network [29-30]. OpenPose is a bottom-up pose estimation method that first finds all the points of all people in an image. Then these points are matched and connected to connect the joint points of the same person. The input image will output two tensor data, namely the heat map of the key points and the corresponding connection relationship of the key points in the inference stage. These output heat maps are only one eighth of the original image, as shown in Figure 6.

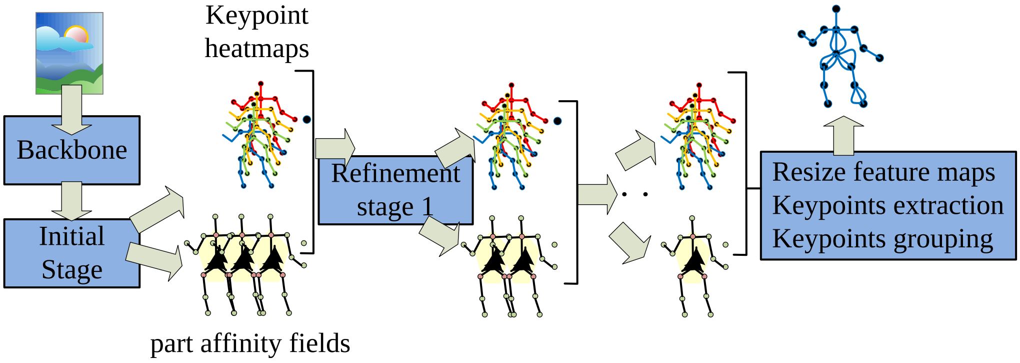


Figure 6: OpenPose process

In Figure 6, OpenPose is a deep learning based multiplayer pose estimation framework that first uses CNN to simultaneously predict the confidence map of body key points and the correlation map between parts from the input image. Next, the confidence map is processed using NonMaximum Value Suppression (NMS) to identify key point positions. Finally, bipartite graph matching and greedy algorithm are applied to pair and associate the obtained key points to construct the human pose. OpenPose can process video streams in real-time and recognize and track the poses of multiple people. The lightweight version of its OpenPose, Lightweight OpenPose, is selected as the baseline model. The parameter count of Lightweight OpenPose is only 15% of the original model, but its performance is almost the same. The INT8 quantization method is chosen for quantization considering the hardware characteristics and resource limitations of the devices to enable deep learning models to be deployed on low-power devices and quantify them. INT8 quantization is to convert model parameters from floating-point numbers to 8-bit integers to reduce model size and maintain accuracy.

4 Performance testing of sports action detection and counting algorithms based on pose estimation

The research aims to use the pose estimation algorithm for motion action detection and counting, thus selecting motion scene videos from various venues, scales, and angles, including offline shooting and network collection. A data augmentation method was adopted to enhance the training samples and enhance the model's generalization ability.

4.1 Tests of sports action detection and counting algorithm output accuracy

A test dataset consisting of 40 videos was constructed to verify the performance of the proposed algorithm in motion action detection and counting. 20 videos of this dataset were from the HMDB51 dataset. These videos were from movies and were characterized by complex and variable scenes. In these complex environments, HMDB51 videos' average detecting accuracy reached 74%. In addition, the experiment also included recording videos from 20 laboratories. 20 laboratory personnel wore different colored clothing to ensure recognizability in different environments. These videos were recorded under five different environmental conditions, each lasting 10 minutes. It was ensured that there was no interference from anyone other than the tester, and only the tester's own limbs were obstructed. Figure 7 shows the output results of action detection and computational algorithms.

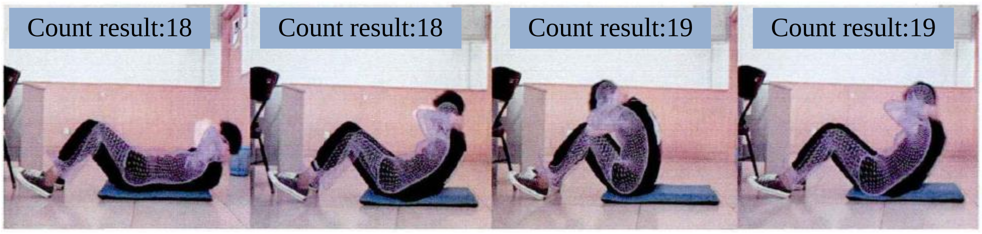


Figure 7: Output results of action detection and calculation algorithms

Figure 7 shows the recognition and counting of sit ups in video frames. The pink network in the figure indicates changes in human movement. Clearly, the count changed from 18 to 19 after completing one action, proving the accuracy of the proposed action detection model. The experimental personnel were kept dressed unchanged in different testing environments and not obstructed by other characters. Figure 8 shows the average counting accuracy results.

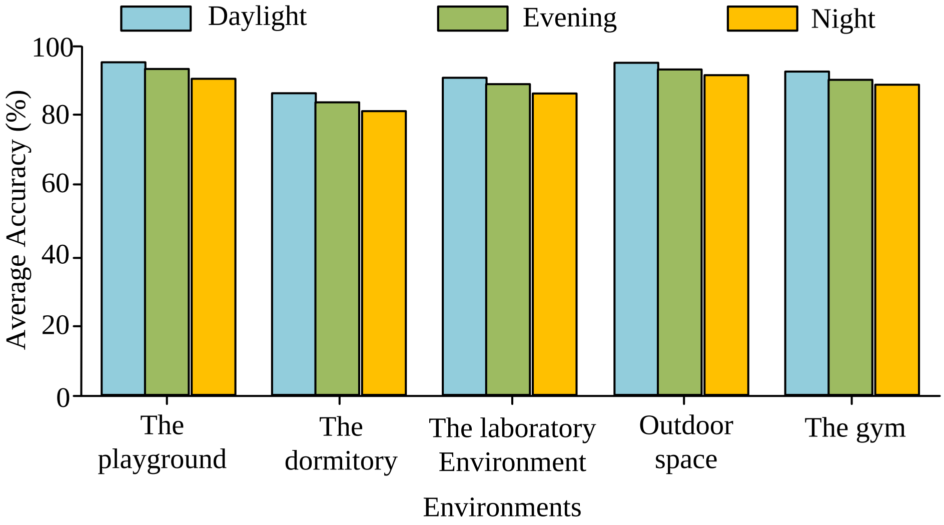


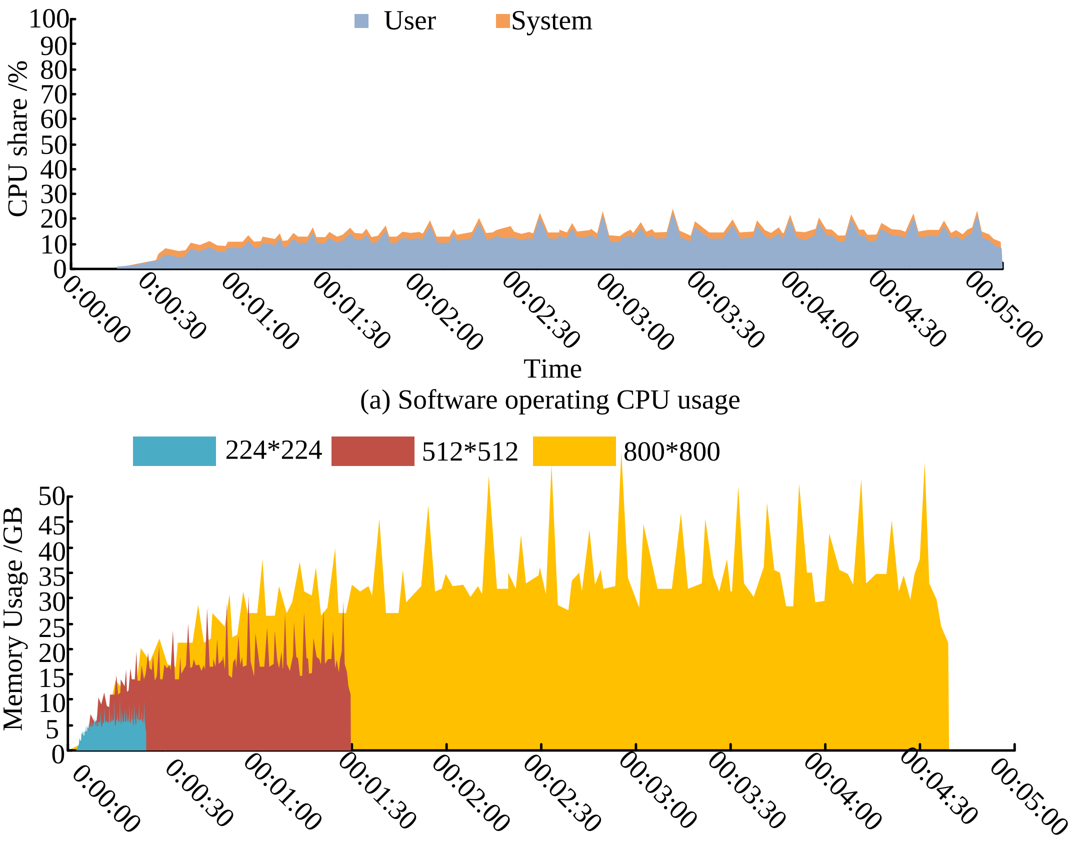
Figure 8: Average count accuracy results

In various testing environments, the average accuracy of the action detection algorithm varied in different locations when the testers were wearing fixed colored clothing and there were no other people obstructing them. Specifically, the detection accuracy of the playground and outdoor space was the same, both at 96%, which was the highest among all environments. The accuracy of the gym followed closely, at 94%. The accuracy in the laboratory environment was 92%. The accuracy in the dormitory environment was the lowest, at 88%. These data reflected the robustness and reliability of the algorithm under different environmental conditions. Table 2 shows the average counting accuracy of different testers in multiple scenarios.

Table 2: Average count accuracy of different testers in multiple scenarios

Person ID	1	2	3	4	5	6	7	8	9	10
ACC/%	93	92	92	94	92	93	93	94	92	93
Person ID	11	12	13	14	15	16	17	18	19	20
ACC/%	94	92	93	94	92	93	94	92	93	94

These experiments confirmed that the designed sports action recognition and counting algorithm exhibited good effectiveness. The changes in environmental background caused fluctuations in accuracy. Meanwhile, the algorithm achieved higher recognition accuracy in simple and open scenes. Participants had different heights, genders, and clothing for individual performance. However, these factors did not significantly affect the performance of the algorithm, with accuracy ranging from 92% to 94%. The fourth participant had the highest detection accuracy, at 94%. The accuracy of the second and third participants was 92%, which was the lowest among all participants. Figure 9 shows the test results of the calculation speed and memory usage.



(a) Software operating CPU usage

(b) Computation rate for different page sizes

Figure 9: Performance evaluation results

Figure 9 (a) shows the CPU usage test results of the system proposed model during runtime. The CPU usage of the server increased as the system continued to run, showing an overall upward trend, but with significant fluctuations. The CPU usage instantly increased and then decreased to normal levels when new processes joined. The highest CPU usage was 21.7%, with the lowest CPU usage at the beginning of the process. The CPU usage remained below 20% throughout the entire script runtime, and the server resource ratio was within a controllable range. The trend of system CPU usage over time was similar to that of server resources, slightly higher than that of server resources. The highest CPU usage was 27.6% during the entire system runtime, and the lowest CPU usage occurred at the beginning of the process. The CPU usage remained below 30% during the system runtime. Figure 9 (b) shows the comparison of the processing speed and resource consumption of video files with different Page sizes by the model, with file sizes of 224*224, 512*512, and 800*800, respectively. As the size of the model file changed, the processing speed of the system also changes accordingly, and the transaction processing efficiency of the system remained basically consistent. The model took 24.6ms to process files of size 224*224, 98.4ms to process files of size 512*512, and 274ms to process files of size 800*800. As the file size increased, Memory Usage gradually increased. Figure 10 shows the results of the ablation test.

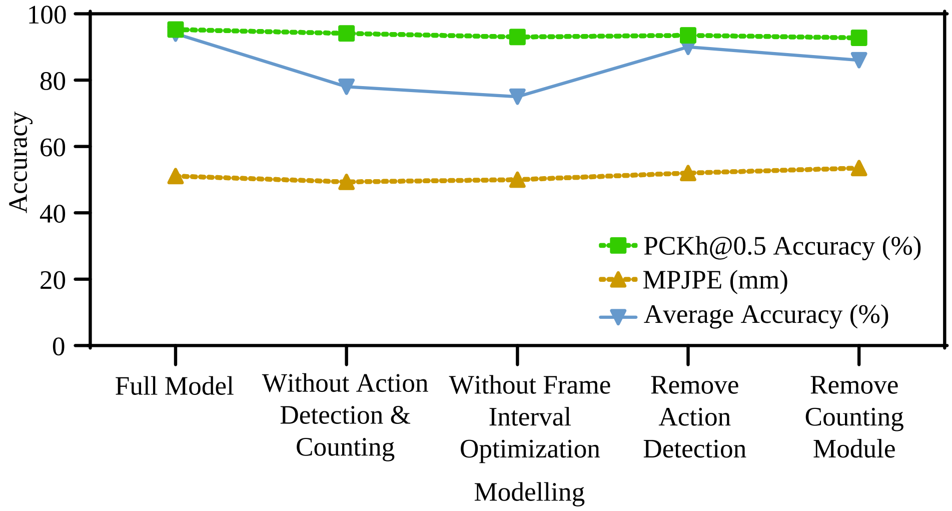


Figure 10: Results of ablation test

In the ablation experiment, the performance of the pose estimation model VIBE was evaluated by gradually removing the action detection and counting modules and optimizing the image frame intervals within them. On the Human3.6M dataset, the unoptimized ViBE achieved a joint accuracy of 94.1% under the PCKh@0.5 metric, slightly lower than the original model's 95.3%. It showed an average error of 49.4mm under the MPJ pose estimation index, which was better than the original model's 51.1mm. On the other hand, the complete model’s optimal average accuracy in action detection and counting was 78%, which was 16% lower than the 94% accuracy that included frame interval optimization steps. This indicated errors such as image noise, object occlusion, or keypoint jitter that occurred during sit up movements, emphasizing the importance of frame interval optimization in improving accuracy.

On-site videos of students from different grades performing different sports actions in actual physical education teaching were collected and classified according to different lighting conditions to further verify the robustness and reliability of the proposed algorithm. Subsequently, a fault injection testing method was adopted to test the detection ability of the proposed algorithm for violations, including hand support, foot off the ground, unilateral body roll up, and standing simulation sit ups during movement. The results are shown in Table 3.

Table 3: Test results of the proposed algorithm in actual physical education teaching

Category		ACC/%	Time/s
Violation actions	Hand support to the ground	95.15	2.16
	Feet off the ground	93.32	2.27
	Unilateral body mass	94.30	1.31
	Stand up simulation sit ups	91.28	2.32
Lighting conditions	Low	90.29	3.95
	Medium	94.34	2.44
	High	96.31	1.29

From Table 3, the motion recognition accuracy of the proposed algorithm was generally stable at over 90% under different lighting conditions, with a maximum of 96.31% and a minimum required time of only 1.29 seconds. Meanwhile, the fastest recognition time was only 1.31 seconds for violations of sports actions, and the highest accuracy reached 95.15%. This result verified that the proposed algorithm effectively identified and eliminated illegal actions, with good robustness and robustness.

4.2 Test of lightweight sports action detection and counting algorithm output accuracy

The action detection video with the primer pointing upwards was captured through a monocular RGB camera connected to the Jetson Nano device. All experiments were conducted for inference and acceleration on the Jetson Nano 2GB memory device developed by NVIDIA, using the default training parameters of Lightweight OpenPose and the COCO dataset as the training basis. The training was completed on an Intel CPU server equipped with 16 cores and 64GB of memory, operating system Ubuntu 18.04, and network input resolution set to 368×368. The model was trained in three stages and its accuracy was verified at the end of each stage. Figure 11 shows the comparison of Lightweight OpenPose accuracy and computational complexity.

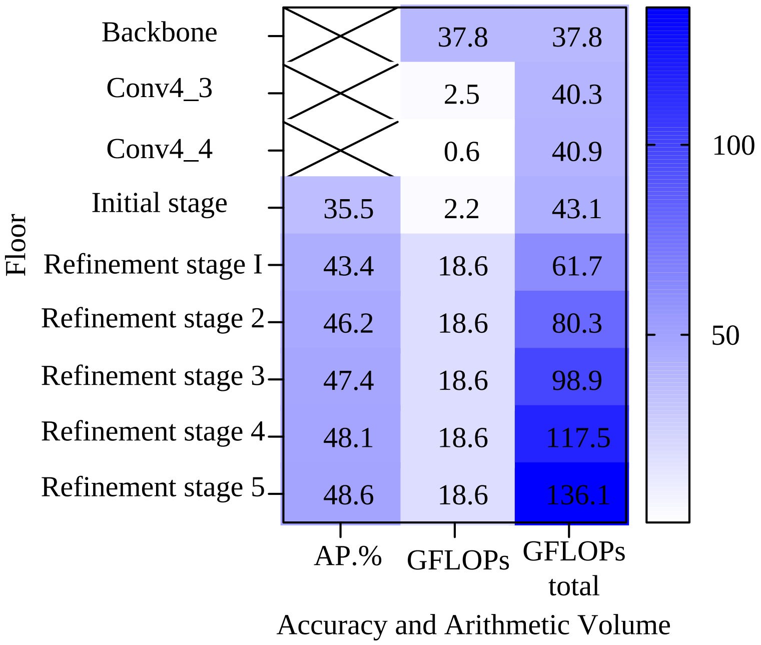


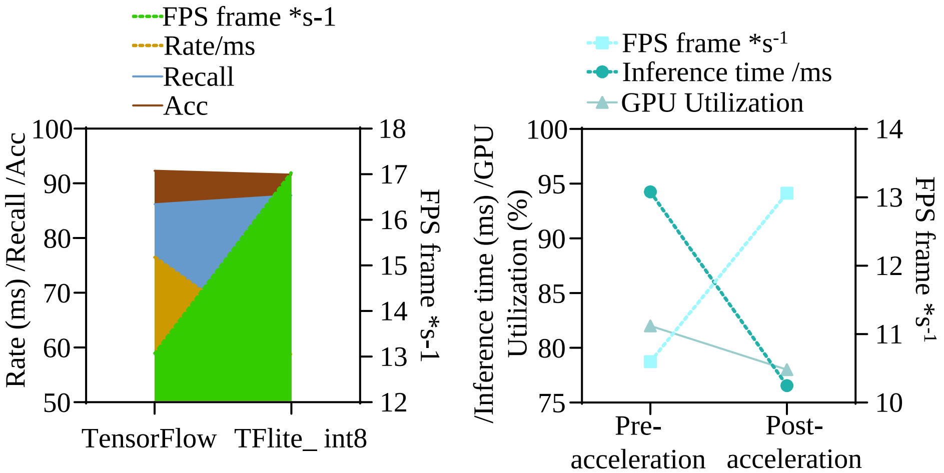
Figure 11: Comparison of OpenPose accuracy and computational complexity

The average accuracy (AP%) for Lightweight OpenPose increased from 35.5% in the initial stage to 43.4% when a Refinement stage was introduced. However, the AP% only increased to 48.6% despite increasing to five Refinement stages, indicating a small performance gain. Although the performance improvement was not significant, the computational load had almost doubled. Giga Floating-Point Operations Per Second (GFLOPs) increased from initial 43.1 to final 136.1. Therefore, this study attempted to use only one Refinement stage for pose estimation and tested its performance. The purpose of this method is to reduce the use of computing resources while maintaining reasonable accuracy, thereby optimizing the efficiency of the model. This work emphasizes the importance of balancing performance and computational efficiency in designing deep learning models. Table 4 shows the selected training accuracy and comparison results for Lightweight OpenPose.

Table 4: Training accuracy and comparative testing of Lightweight OpenPose

Step	APval	Epochs	Batch_size
1	0.3964	260	64
2	0.4196	260	64
3	0.4286	260	64
Model	Average FPS	Best FPS	/
OpenPose	1.32	2.2	/
Lightweight OpenPose	9.24	10.23	/
Model	File size/MB	ACC/%	FPS frame *s-1
OpenPose	200	96.13	1.52
Lightweight Openpose	8.2	95.26	17.04

In Table 4, the training accuracy and comparative test results of Lightweight OpenPose confirmed that the inference speed of Lightweight OpenPose deployed on Jetson Nano was about 9 times faster than the original model. The accuracy of Lightweight OpenPose was consistent with official data, with only a 6% decrease compared to OpenPose. The inference frame rate of the model increased by nearly 15 times while the accuracy decreased by less than 1% through pruning, modifying convolutional layers, and quantifying to Int8, using TensorRT acceleration. This demonstrated the lightweight of the pose estimation model and its efficiency in real-time motion action detection counting on edge devices. Lightweight OpenPose used MobileNet V1 instead of the original backbone network to study the rate, regression rate, and accuracy of quantized pre- and post motion action detection models on a local server. Meanwhile, the study also tested the inference time, FPS, and GPU utilization of Lightweight OpenPose without Int8 quantization before and after TensorRT acceleration to demonstrate the effectiveness of inference acceleration in Figure 12.



(a) Quantifying the performance of pre- and post-sport action detection models

(b) Performance change before and after model application inference acceleration

Figure 12: Inference acceleration test

In Figure 12, Int8 quantization significantly improved processing speed with an accuracy loss of less than 1% under the same model and input conditions. This study further validated the performance of unquantified Lightweight OpenPose after applying TensorRT acceleration. The inference time was shortened and the frame rate was increased by nearly 3 frames, while the GPU utilization rate remained unchanged. The addition of Int8 quantization further shortened the inference time by about 18 milliseconds and increased the frame rate by about 4 frames, verifying the effectiveness of Int8 quantization in inference acceleration.

5 Discussion

With the rapid development of computer vision applications, this technology used for online sports competitions and exercise to improve the quality of physical education teaching receives widespread attention. However, the current computer vision-based motion recognition framework focuses on classifying different actions during the motion. Meanwhile, there is still a lack of relevant research and practical application for further operations such as counting and filtering inappropriate actions. The research aims to explore motion detection frameworks and counting algorithms based on pose estimation, and they are applied to sports detection and recognition. Firstly, a motion detection framework and counting algorithm based on pose estimation were proposed, which improved the recognition accuracy of motion in sports evaluation scenarios. Secondly, a motion detection system combining end-to-end cloud technology was designed by using university sports testing as a practical application scenario. The results showed that although participants had different heights, genders, and clothing, the accuracy of the designed sports action recognition and counting algorithm varied between 92% and 94%, demonstrating good effectiveness. This method had stronger stability and wider applicability compared with the results obtained in three datasets in reference [7]. This is because the proposed method obtains a normalized matrix based on human joints. Therefore, actions can be more accurately described through a unified coordinate system, making classification and recognition of different movements easier. The inference speed of the Lightweight OpenPose model increased by about 9 times in terms of training accuracy and comparative testing. The inference frame rate increased by nearly 15 times with less than 1% accuracy decrease after using TensorRT acceleration. The proposed model achieved smaller improvements and better performance compared to the accuracy and efficiency in references [5] and [11]. The reason is that the study used INT8 quantization to convert floating-point parameters in deep learning models into fixed-point numbers. Therefore, the storage space and computational complexity of the model can be reduced, and the stable accuracy can be maintained while reducing the model size by four times.

Overall, the study successfully implemented a cloud-based training network and deployed lightweight models on edge devices. Therefore, real-time inference by optimizing network models and accelerating inference can be completed, further improving the efficiency of motion detection and counting tasks. The proposed technology can be integrated into existing physical education teaching environments in terms of practical application. More accurate motion detection and counting can be achieved based on sports motion data collected by sensors, cameras, etc. Therefore, personalized guidance and feedback to students can be provided, and movements can be timely adjusted to achieve better results, thus improving the physical education teaching. The proposed lightweight motion detection and counting algorithm improved the frame rate of the model on edge devices to a certain extent. However, this algorithm still cannot achieve the FPS of cloud-based inference. Meanwhile, this method saves computational resources. However, there is still room for improvement in detection performance. Therefore, continuous optimization is required to adapt to different sports movements and scenarios when applying this method to a wider range of physical education teaching. Meanwhile, how to improve the recognition ability for fast or complex movements should be explored.

6 Conclusion

Traditional sports action analysis mainly relies on the observation of coaches and the subjective feelings of athletes. This method is not only time-consuming and labor-intensive, but also easily influenced by personal experience and judgment bias. Therefore, the pose estimation-based algorithm automatically detected, analyzed, and counted motion actions. These performance tests confirmed that the detection accuracy of the playground and outdoor space was the same, both at 96%, which was the highest among all environments. On the Human 3.6M dataset, the unoptimized ViBE achieved a joint accuracy of 94.1% under the PCKh@0.5 metric, slightly lower than the original model's 95.3%. It showed an average error of 49.4mm under the MPJ pose estimation index, which was better than the original model's 51.1mm. The optimal average accuracy of the complete model in action detection and counting was 78%, which was 16% lower than the 94% accuracy that included frame interval optimization steps. The training accuracy and comparative testing of Lightweight OpenPose confirmed that the inference speed of Lightweight OpenPose deployed on Jetson Nano was about 9 times faster than the original model. Its accuracy was consistent with official data, with only a 6% decrease compared to OpenPose. After using TensorRT acceleration, the inference frame rate of the model increased by nearly 15 times while the accuracy decreased by less than 1%. The inference time was shortened and the frame rate was increased by nearly 3 frames after applying TensorRT acceleration, while the GPU utilization remained unchanged. The addition of Int8 quantization further shortened the inference time by about 18 milliseconds and increased the frame rate by about 4 frames, verifying the effectiveness of Int8 quantization in inference acceleration. These experiments confirmed that the proposed pose estimation-based motion action detection and counting model had high robustness and reliability under different environmental conditions. However, there are still some shortcomings in the research. Although the proposed pose estimation motion detection framework is effective in handling repetitive fitness movements, the universality of the detection and counting modules for specific movements is insufficient. In the future, more universal modules need to be developed to expand the applicability of the framework.

References

[1] J. Ge, J. Shi, Z. Zhou, Z. Wang, and Q. Qian, "A grasping posture estimation method based on 3D detection network," Computers and Electrical Engineering, vol. 132, no. 10, pp. 96-108, 2022. https://doi.org/10.1016/j.compeleceng.2022.107896.

[2] N. M. Ghahjaverestan, M. M. Kabir, S. Saha, B. Gavrilovic, K. Zhu, B. Taati, H. Alshaer, and A. Yadollahi, "Relative tidal volume and respiratory airflow estimation using tracheal sound and movement during sleep," Journal of Sleep Research, vol. 30, no. 4, pp. 79-80, 2021. https://doi.org/10.1111/jsr.13279.

[3] X. Li, S. Liu, Y. Chang, S. Li, Y. Fan, and H. Yu, "A human joint torque estimation method for elbow exoskeleton control," International Journal of Humanoid Robotics, vol. 17, no. 3, pp. 39-56, 2020. https://doi.org/10.1142/S0219843619500397.

[4] B. Wang, C. Ou, N. Xie, L. Wang, T. Yu, G. Fan, and J. Chu, "Lower limb motion recognition based on surface electromyography signals and its experimental verification on a novel multi-posture lower limb rehabilitation robot," Computers and Electrical Engineering, vol. 101, pp. 110-129, 2022. https://doi.org/10.1016/j.compeleceng.2022.108067.

[5] P. Chen, S. Guo, H. Li, X. Wang, G. Cui, C. Jiang, and L. Kong, "Through-wall human motion recognition based on transfer learning and ensemble learning," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 191-196, 2022. https://doi.org/10.1109/LGRS.2021.3070374.

[6] V. P. Nemani, H. Lu, A. Thelen, C. Hu, and A. T. Zimmerman, "Ensembles of probabilistic LSTM predictors and correctors for bearing prognostics using industrial standards," Neurocomputing, vol. 491, no. 6, pp. 575-596, 2022. https://doi.org/10.1016/j.neucom.2021.12.035.

[7] B. Chen, T. Li, and W. Ding, "Detecting deepfake videos based on spatiotemporal attention and convolutional LSTM," Information Sciences, vol. 60, no. 1, pp. 58-70, 2022. https://doi.org/10.1016/j.ins.2021.12.062.

[8] I. C. Kaadoud, N. P. Rougier, and F. Alexandre, "Knowledge extraction from the learning of sequences in a long short-term memory (LSTM) architecture," Knowledge-Based Systems, vol. 235, pp. 657-675, 2022. https://doi.org/10.1016/j.knosys.2021.107657.

[9] T. D. Le and G. Kaddoum, "LSTM-based channel

access scheme for vehicles in cognitive vehicular networks with multi-agent settings," IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 9132-9143, 2021. https://doi.org/10.1109/TVT.2021.3100591.

[10] B. Wu, J. Zhong, and C. Yang, "A visual-based gesture prediction framework applied in social robots," IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 3, pp. 510-519, 2022. https://doi.org/10.1109/JAS.2021.1004243.

[11] W. Jia, S. Xu, Z. Liang, Y. Zhao, H. Min, S. Li, and Y. Yu, "Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector," IET Image Processing, vol. 15, no. 14, pp. 3623-3637, 2021. https://doi.org/10.1049/ipr2.12295.

[12] L. Li and T. Ye, "Research on throughput prediction of 5G network based on LSTM," Intelligent and Converged Networks, vol. 3, no. 2, pp. 217-227, 2022. https://doi.org/10.23919/ICN.2022.0006.

[13] B. Dudi and V. Rajesh, "Optimized threshold-based convolutional neural network for plant leaf classification: a challenge towards untrained data," Journal of Combinatorial Optimization, vol. 43, no. 2, pp. 312-349, 2022. https://doi.org/10.1007/s10878-021-00770-w.

[14] D. Mcdonough, W. Liu, X. Su, and Z. Gao, "Small-groups versus full-class exergaming on urban minority adolescents' physical activity, enjoyment, and self-efficacy," Journal of Physical Activity and Health, vol. 18, no. 2, pp. 192-198, 2021. https://doi.org/10.1123/jpah.2020-0348.

[15] S. Liu, Y. Guo, H. Liu, A. Hao, X. Zhang, and H. Liu, "Blended learning model via small private online course improves active learning and academic performance of embryology," Clinical Anatomy, vol. 35, no. 2, pp. 211-221, 2022. https://doi.org/10.1002/ca.23818.

[16] M. Madadi, H. Bertiche, and S. Escalera, "SMPLR: Deep learning-based SMPL reverse for 3D human pose and shape recovery," Pattern Recognition, vol. 106, no. 7, pp. 72-78, 2020. https://doi.org/10.1016/j.patcog.2020.107472.

[17] M. Hasanvand, M. Nooshyar, E. Moharamkhani, and A. Selyari, "Machine learning methodology for identifying vehicles using image processing," AIA, vol. 1, no. 3, pp. 170-178, 2023. https://doi.org/10.47852/bonviewAIA3202833.

[18] P. Preethi and H. R. Mamatha, "Region-based convolutional neural network for segmenting text in epigraphical images," Artificial Intelligence and Applications, vol. 1, no. 2, pp. 119-127, 2023. https://doi.org/10.47852/bonviewAIA2202293.

[19] B. Liu, B. Li, J. Cao, W. Wang, and X. Liu, "Adaptive and propagated mesh filtering," Computer-Aided Design, vol. 154, no. 10, pp. 22-34, 2023. https://doi.org/10.1016/j.cad.2022.103422.

[20] Riya, B. Gupta, and S. S. Lamba, "Structure-aware adaptive bilateral texture filtering," Digital Signal Processing, vol. 123, no. 3, pp. 86-99, 2022. https://doi.org/10.1016/j.dsp.2022.103386.

[21] I. Gonzalez-Perez, P. L. Guirao-Saura, and A. Fuentes-Aznar, "Application of the bilateral filter for the reconstruction of spiral bevel gear tooth surfaces from point clouds," Journal of Mechanical Design, vol. 143, no. 5, pp. 24-34, 2021. https://doi.org/10.1115/1.4048219.

[22] C. Karam, K. Sugimoto, and K. Hirakawa, "Color-compressive bilateral filter and nonlocal means for high-dimensional images," Journal of Electronic Imaging, vol. 30, no. 2, pp. 23-44, 2021. https://doi.org/10.1117/1.JEI.30.2.023001.

[23] M. Redmann and I. P. Duff, "Model order reduction for bilinear systems with non-zero initial states - different approaches with error bounds," International Journal of Control, vol. 96, no. 4/6, pp. 1491-1504, 2023. https://doi.org/10.1080/00207179.2022.2053209.

[24] X. Zhao, C. Huang, X. Yu, S. Zou, and f. Qing, "An arbitrary Lagrangian-Eulerian discontinuous Galerkin method for two-dimensional compressible flows on adaptive quadrilateral meshes," International Journal for Numerical Methods in Fluids, vol. 95, no. 5, pp. 796-819, 2023. https://doi.org/10.1002/fld.5172.

[25] H. Wang, G. Xu, X. Pan, Z. Liu, N. Tang, R. Lan, and X. Luo, "Attention-inception-based U-Net for retinal vessel segmentation with advanced residual," Computers & Electrical Engineering, vol. 98, no. 3, pp. 92-110, 2022. https://doi.org/10.1016/j.compeleceng.2021.107670.

[26] A. M. Vukicevic, I. Macuzic, N. Mijailovic, A. Peulic, and M. Radović, "Assessment of the handcart pushing and pulling safety by using Deep Learning 3D pose estimation and IoT force sensors," Expert Systems with Application, vol. 183, no. 1, pp. 53-71, 2021. https://doi.org/10.1016/j.eswa.2021.115371.

[27] B. Li, Z. Xu, J. Zhang, X. Wang, and X. Fan, "Background modeling based on statistical clustering partitioning," Mathematical Problems in Engineering, vol. 2021, no. 2, pp. 1-28, 2021. https://doi.org/10.1155/2021/2346438.

[28] J. Coll-Font, O. Afacan, J. S. Chow, R. S. Lee, S. K. Warfield, and S. Kurugol, "Modeling dynamic radial contrast enhanced MRI with linear time invariant systems for motion correction in quantitative assessment of kidney function," Medical Image Analysis, vol. 67, no. 4, pp. 33-45, 2021. https://doi.org/10.1016/j.media.2020.101880.

[29] Y. Wang, "A study on the recognition of typical movement characteristics of ethic folk dances based on movement data," Informatica, vol. 48, no. 5, 2024. https://doi.org/10.31449/inf.v48i5.540.

[30] H. Jiang and S. B. Tsai, "An empirical study on sports combination training action recognition based

on SMO algorithm optimization model and artificial intelligence," Mathematical Problems in Engineering, vol. 2021, no. 31, pp. 83-94, 2021. https://doi.org/10.1155/2021/7217383.

Appendix

Appendix 1 Literature summary

Reference	Main content	Results
[4]	Feature extraction and pattern recognition through sEMG	Good control effect
[5]	ResNeXt network based on set learning	Higher recognition accuracy than fusion models based on single view radar
[6]	A two-stage LSTM model	The accuracy of bearing life prediction exceeded 95%
[7]	New Xception-LSTM algorithm	Excellent algorithm performance on three common datasets
[8]	Using internal state clustering algorithm to improve LSTM extraction algorithm	Extracting sequences from the original syntax had a higher recognition rate
[9]	Q-learning network vehicle connection algorithm based on deep recursion	Higher stability and efficiency
[10]	Combining recurrent neural networks and LSTM networks	Achieved the highest accuracy of 99.31%
[11]	Using Soft-NMS instead of NMS to fuse YOLOv5 detector	97.7% mAP, 92.7% F1 score, and 63 frames per second currents
[12]	A model for predicting wireless traffic in a predictive network that integrates LSTM algorithm and recurrent neural network	Good prediction accuracy and algorithm training speed
[13]	Introducing a hybrid whale optimization algorithm based on shark odor to optimize plant leaf classification models	Improved classification accuracy
[14]	Designing a control experiment to improve the intervention effect of school dance sports game education mode	Proved the effectiveness of group exercise mode
[15]	Using Embryology Course as an Example to Study the Effectiveness of SPOC Teaching Model	Improved the average professional grades of students and enhanced their learning enthusiasm