The Application of Action Recognition Based on MPP-YOLOv3 Algorithm in Posture Correction

Zhongwei Wang[1*], Shujuan Dong[2]
[1]Department of Public Studies, Henan Vocational College of Nursing, Anyang 455000, China
[2]Information Engineering Institute, Yellow River Conservancy Technical Institute, Kaifeng 475004, China
Email of corresponding author: wzw202312@126.com, 13707617998@126.com

Keywords: YOLOv3, key points, posture recognition, posture correction, lightweight

Received: March 15, 2024

Posture recognition, as a research hotspot, has been widely applied. A recognition model based on bone key point detection is proposed for the posture correction application module. Firstly, the lightweight You Only Look Once v3 tiny network was chosen as the infrastructure, and the OpenPose algorithm in the bottom-up strategy was chosen to implement posture recognition. To reduce the computational burden of the model, the Media Pipe Blaze Pose algorithm was introduced for improvement. At the same time, by refining more bone key points, the accuracy of the model has been improved. The experiment outcomes revealed the recognition accuracy of Cross View in the NTU60 RGB+D dataset and NTU120 RGB+D dataset was 94.7% and 82.7%, respectively. Compared to graph Transformer networks and semantic posture recognition models, the Cross-Subject metric improved by an average of 3.5%. Therefore, the research and design model has shown better robustness in the field of posture recognition, which can help complete pose correction more efficiently.

Povzetek: Predstavljen je nov sistem za prepoznavanje in popravljanje drže. Uporablja MPP-YOLOv3, OpenPose in Media Pipe Blaze Pose.

1 Introduction

Computer vision has gradually integrated into people's daily lives. Among them, posture recognition has always been one of the research hotspots in this field. Posture recognition is applied in multiple realms including home monitoring, posture correction, and rehabilitation training [1-2]. The demand for pose correction has gradually increased, and it is mostly used to correct improper movement postures. This is especially important in professional sports training, physical therapy, and personal fitness scenarios, where accurately identifying and correcting incorrect exercise postures can greatly improve exercise efficiency, reduce the risk of injury, and promote physical health. Deep learning technology is one of the commonly used methods in this field, playing an important role in the design of motion pose correction systems [3]. Traditional motion recognition techniques often rely on wearable sensors or complex labeling systems, which are often limited, costly, and may interfere with the natural movements of athletes. The deep learning network model provides a non-invasive solution for it, which can directly recognize human posture by analyzing image or video data obtained from cameras. In the field of deep learning, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and others are widely used for feature extraction and modeling of visual data, thereby achieving recognition and analysis of human actions. Posture recognition has two parts to compose its full function, namely identifying bone keypoints and keypoint connections [4-6]. However, in practical applications, models need to learn data from different actions and perspectives, which will inevitably impose a greater computational burden on them. Therefore, the research aims to further enhance the computational ability of the model while ensuring its accuracy. Based on this, a You Only Look Once version 3 (YOLOv3) posture recognition network was studied and designed, which achieved a balance between recognition accuracy and speed through lightweight and bone key point refinement. It is based on Media Pipe Blaze Pose and is therefore known as Media Pipe Blaze Pose YOLOv3 (MPP-YOLOv3). The research has 4 parts. There are two innovations in research technology. The first is the use of lightweight YOLOv3 Tiny to build the basic framework, and the second is the implementation of the OpenPose algorithm to enhance the system and optimize model calculations. Additionally, the model extracts more skeletal details to capture motion details and improve technical reliability. As a result, the research model accurately recognizes the posture and movements of the human body and provides precise posture correction suggestions. The first gives the status of posture recognition, the second part designs deep learning network models, the third part conducts experimental analysis on the performance of the design model, and the last part summarizes the experimental data.

2 Related works

The application of posture recognition is very extensive and has received academic research on prevention. Zhou et al. proposed a new semantic posture recognition method that classifies actions in videos by learning multiple visual pose models and pose dictionaries associated with body parts. The researchers identified hidden poses in video frames and mapped them to actions in semantic instructions. The experiment outcomes indicated that their solution was effective on multiple datasets [7]. Liu et al. learned and put forward a Transformer network for skeleton based human posture recognition. They utilized multi head self attention and temporal kernel attention to capture high-order dependencies of joints in the skeleton and enhance the temporal correlation of actions. The experimental results indicated that their model outperforms the baseline models [8]. Alemu et al. proposed a method for generating human actions using an auxiliary conditional genetic neural network. The aim was to overcome the limitations of single-view action generation by creating samples from new perspectives and expanding the view range. Additionally, they introduced a view domain generalization model to improve posture recognition performance from different perspectives. Tests on multiple RGB+D skeleton datasets showed that their method effectively improves the accuracy of posture recognition [9]. Fang et al. learned and put forward a spatiotemporal slow fast graph convolutional network, which effectively captured the spatiotemporal joint relationships of long and short distances in skeleton data by designing specific adjacency matrices. They used fast and slow paths to process action information at different time scales. Tests on multiple datasets showed that STSF-GCN achieves leading recognition performance at lower computational costs [10].

Coskun et al. designed a minimal transfer learning method. By independently training local visual cues and using a meta learning-based framework, the action classification model was transferred with only a few samples. The experiment outcomes indicated that their solution was effective [11]. Li et al. proposed an posture recognition method supported by wavelet transform, which enhanced the sensitivity and discriminative ability of graph convolutional networks to local movements through an innovative three attention module. By aggregating global statistical information and integrating multidimensional features, the perception of significant changes was strengthened. The experiment outcomes indicated that their solution achieved comparable performance on multiple datasets [12]. Hao et al. proposed the use of hypergraph neural networks to enhance human motion recognition based on machine vision. By constructing hypergraphs, attention mechanisms, and residual modules to obtain discriminative features, the three-stream fusion architecture further improved recognition accuracy. Their method achieved optimal performance on two benchmark datasets [13]. Table 1 depicts the specific literature content.

Table 1: Main contents of literature review

Reference	Research method	Research dataset and results	Limitations
Reference [7]	A new semantic pose recognition method has been proposed in the study	Using a pose dictionary dataset for classification proves the effectiveness of this technique	This technology is computationally complex
Reference [8]	A Transformer network based on skeleton for human pose recognition has been proposed in the study	Tested on a universal human image dataset, this model outperforms the baseline model	This technology does not take into account changes in complex motion data
Reference [9]	Research proposes a human action generation method supported by auxiliary conditional genetic neural networks	Tested on the RGB+D skeleton dataset, this method effectively improves the accuracy of pose recognition	This model requires complex parameter calculations
Reference [10]	A spatiotemporal slow and fast graph convolutional network has been proposed in the study	STSF-GCN has excellent recognition performance in both self-made and universal skeleton data	This method is susceptible to noise drying
Reference [11]	Researched and designed a minimum transfer learning method	The model classification performance is tested on a linear sample dataset, and the scheme is reliable	This method has poor stability
Reference [12]	A pose recognition method supported by wavelet transform has been proposed in the study	Tested on NTU-RGB+D-skeletons data, the research method has shown excellent performance	The model performs poorly on lower samples

Current posture recognition models primarily rely on extracting bone key points to improve recognition accuracy, but often neglect the improvement of detection speed. Therefore, the study proposes an posture recognition model based on lightweight YOLOv3 Tiny, which not only enhances the refinement of key points but also improves the detection efficiency of the model.

3 Design of a posture recognition system based on YOLOv3 tiny network architecture and mpp optimization

A deep learning based action recognition network is proposed for the application field of posture correction. Firstly, the lightweight YOLOv3 Tiny network is chosen as the infrastructure and optimized using the OpenPose algorithm. Subsequently, Media Pipe Pose is introduced to achieve deep convolutional separation and lightweight design, achieving an improvement in the accuracy of the network model.

3.1 Design of action recognition module based on improved YOLOv3 tiny network architecture

With the artificial intelligence growth, machine vision are applied in various realms such as motion, medicine, and industry. Deep learning has been applied to the direction of posture correction, and a human action recognition algorithm has been developed. CNN are a commonly used technology in deep learning and have matured significantly. The research selects the most representative YOLO algorithm, which is based on the feature network of the image, performs grid partitioning, generates corresponding prior boxes, and finally performs target recognition through regression tasks. Based on the performance comparison of various versions of the YOLO algorithm, a lightweight YOLOv3 Tiny algorithm is chosen for research, which has better flexibility and agility. However, its lightweight characteristics can also lead to a decrease in model accuracy. Therefore, the study further strengthened the accuracy of algorithm recognition by reducing the number of predicted classifications to lower network operating parameters, as shown in Figure 1.

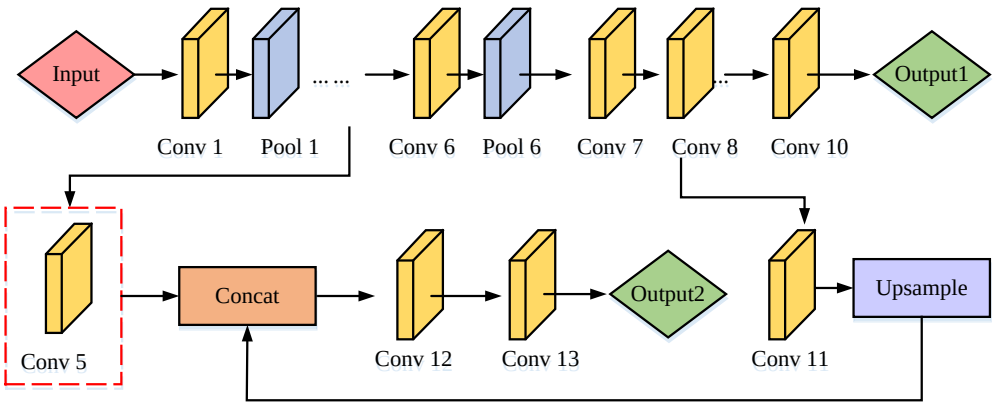


Figure 1: YOLOv3-Tiny network infrastructure

The prerequisite for achieving posture correction is action recognition, which involves identifying key points in the human body and making optimal connections. Deep learning-based recognition techniques include two forms: two-dimensional and three-dimensional. The running process of the two is roughly the same, but 3D recognition requires a transformation of 3D space after mapping key points. According to the number of target recognition, it contains two categories: single person and multi person recognition [14-15]. The study investigated more complex scenarios for recognizing multiple individuals, using common detection strategies such as bottom-up and top-down approaches. In the top-down approach, only target parameters need to be detected before action estimation. While the recognition accuracy is high, it does not meet the real-time requirements for multiple targets. From top to bottom, it is necessary to detect and group key human points, and then perform matching recognition. This approach is more suitable for recognizing multi-target actions. To study the commonly used OpenPose algorithm in bottom-up strategies, it is necessary to first perform feature extraction and generate corresponding feature maps. Then input it into the Visual Geometry Group 19 (VGG19) network for feature extraction. Estimate the Part Confidence Maps (PCMs) in the upper branch, and estimate the Part Affinity Fields

(PAFs) in the lower branch. Among them, PAF is a 2D limb annotation technique that can preserve the position and directional parameter information of limb intervals, as shown in Figure 2.

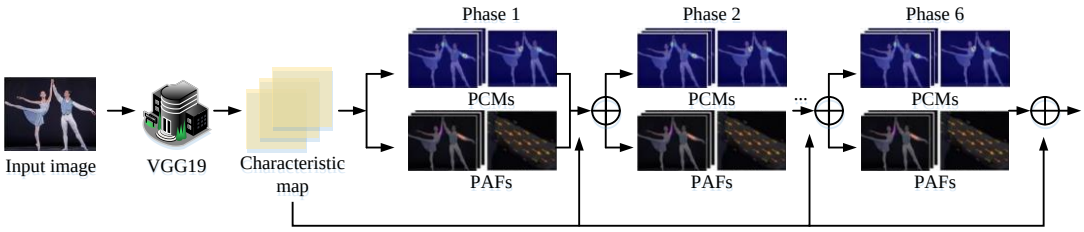


Figure 2: OpenPose module combination diagram

PAFs can represent the practice of each key point, providing support for subsequent key point matching. The joint confidence plot S_j and PAFs plot L_c are shown in formula (1).

$$\begin{cases} S_j \in R^{w*h}, j \in \{1,2,...,J\} \\ L_c \in R^{w*h*2}, c \in \{1,2,...,C\} \end{cases} \quad (1)$$

In formula (1), $w*h$ represents the input image size, and J/C represent the total number of human keypoints and bone connections, respectively. The confidence plots S^l and PAFs L^l for the production cost period are shown in formula (2).

$$\begin{cases} S^l = \rho^{'}\left(F, S^{l-1}, L^{l-1}\right), \forall t \geq 2 \\ L^l = \phi^{'}\left(F, S^{l-1}, L^{l-1}\right), \forall t \geq 2 \end{cases} \quad (2)$$

In formula (2), $\rho^{'}/\phi^{'}$ represent the stage inference of the confidence map and PAFs map, F represents the feature map, and represents the confidence map and PAFs map of the previous stage, respectively. The study introduces L2 loss functions to calculate the losses of each branch separately, aiming to ensure the correct expansion of the training direction. Subsequently, the study aimed to address the phenomenon of missing labels in annotated samples and optimized the loss function through mask operation. The loss function of the entire model is the sum of two branches, and the improved loss function f_S^t / f_L^t corresponding to the upper branch of PCMs and the lower branch of PAFs is shown in formula (3) [16].

$$\begin{cases} f_S^t = \sum_{j=1}^{J} \sum_p W(p) \cdot \left\| S_j^t(p) - S_j^*(p) \right\|_2^2 \\ f_L^t = \sum_{c=1}^{C} \sum_p W(p) \cdot \left\| L_c^t(p) - L_c^*(p) \right\|_2^2 \end{cases} \quad (3)$$

In formula (3), t represents the corresponding number of stages, $S_j^t(p)/L_c^t(p)$ represent the pixel PCMs and PAFs of each stage, $S_j^*(p)/L_c^*(p)$ represent the confidence map and partial affinity domain of the annotated points, and $W(p)$ represents the binary mask function of the pixel points p. $W(p)$ value of 1 or 0 corresponds to the cases where pixels are labeled and unlabeled, respectively. There are a total of 18 key points in the human body, as Figure 3.

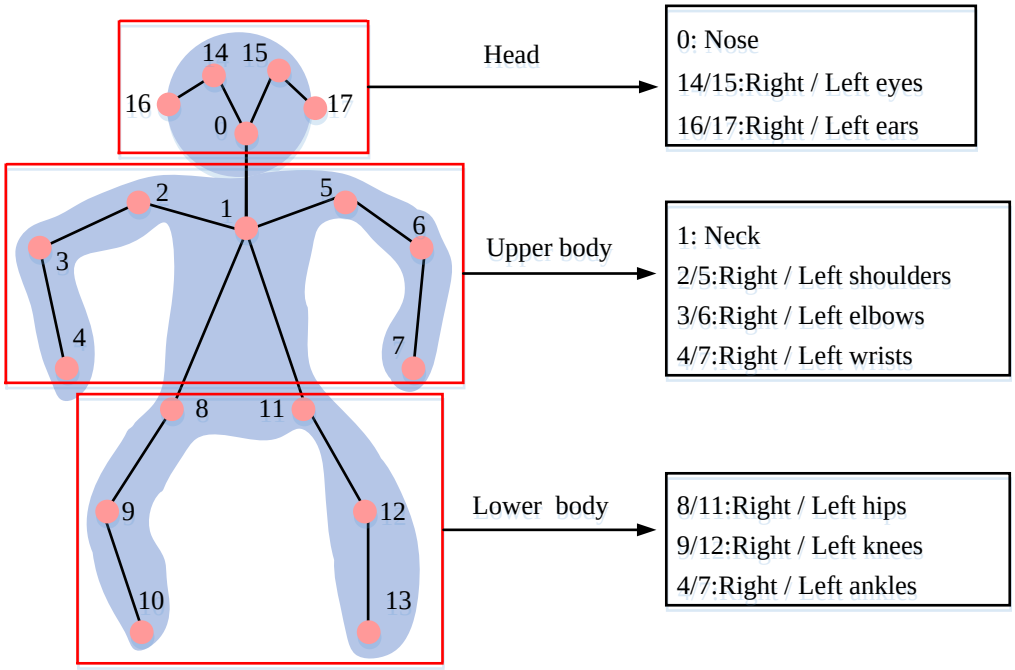


Figure 3: Key points of human bones

The human body is considered as three parts: the head, upper body, and lower body. Except for the nose and neck, all other key points are symmetrical on both sides. When there is no occlusion in the human body image, there is a unique maximum value among its corresponding PCMs. When there are k human targets in the image, there will be k peaks in the corresponding keypoint PCMs, represented as $S_{j,k}^{*}(p)$. By using the max method, multi-objective PCMs aggregation $S_{j}^{*}(p)$ can be achieved, as formula (4).

$$\begin{aligned} S_{j}^{*}(p) &= \max_{k} S_{j,k}^{*}(p) \\ &= \max_{k}\left[\exp\left(-\frac{\left\|p-x_{j,k}\right\|_{2}^{2}}{\sigma^{2}}\right)\right] \end{aligned} \tag{4}$$

In formula (4), σ represents the standard deviation, p represents the position of $S_{j,k}^{*}(p)$, and $x_{j,k}$ represents the pixel position of k people in the corresponding joint point j. After the key points are extracted, they need to be grouped and connected to ultimately achieve human pose recognition. When there are multiple targets in the image, the fully connected form can lead to redundant connections, while the midpoint detection method can lead to errors in connecting multiple targets. Therefore, the study aims to use the corresponding partial affinity domain values for key point connections through two-dimensional vector field PAFs, as shown in formula (5) [17].

$$L_{c,k}^{*}(p)=\begin{cases} v, p\in k \\ 0, p\notin k \end{cases} \tag{5}$$

In formula (5), $L_{c,k}^{*}(p)$ represents the PAFs values of the corresponding limb target keypoints, c represents the truncation of the target limb, and v represents the unit vector. The calculation of the unit vector is given as formula (6).

$$v=\frac{x_{j_2.k}-x_{j_1.k}}{\left\|x_{j_2.k}-x_{j_1.k}\right\|_{2}} \tag{6}$$

In formula (6), j_1 / j_2 represent different key points, and $x_{j_2.k} / x_{j_1.k}$ represents the coordinates of target k in different feature points.

3.2 Design of posture correction system based on Media Pipe Blaze Pose

To reduce the computational burden of the model, the study introduces the Depth Separable Convolution (DSC) strategy for improvement. Among them, DSC improves an algorithm model for annotated convolution by separating the correlation between the spatial dimension and the channel (depth) dimension.

This reduces the number of parameters required for convolution calculation and improves the computational efficiency of the model [18]. Although the recognition accuracy of this network may slightly decrease, the decrease in accuracy can be negligible when the parameters are significantly reduced. The DSC strategy aims to decompose standard convolution into deep convolution kernel 1 * 1 convolution, corresponding to the combination of spatial dimension data and channel data. Between each depth convolutional layer and 1 * 1 convolutional layer, a BN layer and ReLU layer are set, consistent with the standard convolution. The DSC decomposition structure is given in Figure 4.

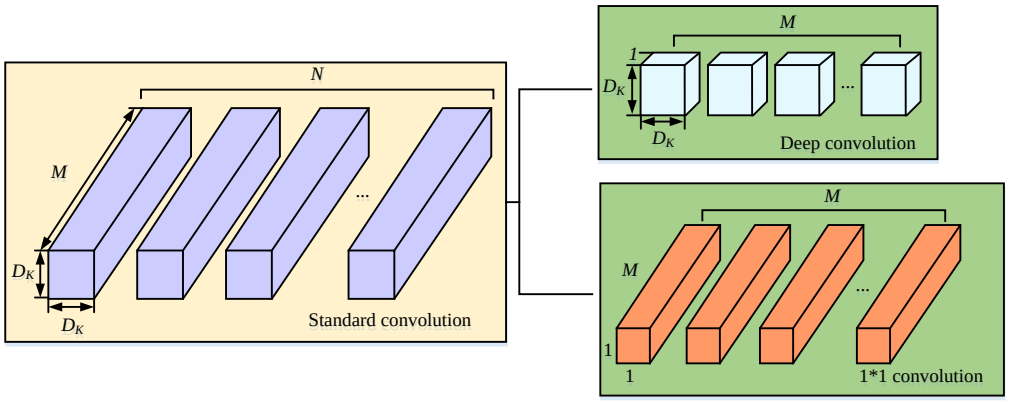


Figure 4: Depthwise separable convolution structure

In Figure 4, the convolution sum size is D_K * D_K, $D_F \times D_F \times M$ is the symbol that denoting the input feature map size, $D_F \times D_F \times N$ is the symbol that denoting the output feature map size, and N convolution kernels number. The required computational cost C_S is given in formula (7) [19-20].

$$C_S = D_K \times D_K \times M \times D_F \times D_F \times N \quad (7)$$

The DSC convolution computation C_D divides the standard convolution computation into two modules: depth convolution and 1 * 1 convolution, as shown in formula (8).

$$C_D = D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \quad (8)$$

By comparing the standard convolution computation with the DSC convolution computation, the corresponding computational ratio of the two can be obtained, as shown in formula (9).

$$P_C = \frac{C_D}{C_S} = \frac{1}{N} + \frac{1}{D_K^2} \quad (9)$$

From formula (9), the calculation ratio is related to the number and size of convolution kernels. In practical networks, the number of convolutional kernels is usually large, so the impact on computational complexity can be ignored in calculations. In the study of using network models, the kernel size is set to 3 * 3, which means D_K is equal to 3. Therefore, the ratio of DSC convolution computation to standard convolution computation is approximately 1:9, which also proves that DSC can help cut the computational parameters and burden of the model greatly. Based on this, a graphic cross platform architecture Media Pipe is introduced and applied to the target post recognition. This model can achieve end-to-end acceleration, with a built-in fast ML inference and processing framework that enables it to run on servers such as mobile devices and workstations. In addition, the Media Pipe model can also simultaneously build multiple learning channels such as videos and sensors. The Media Pipe infrastructure is shown in Figure 5.

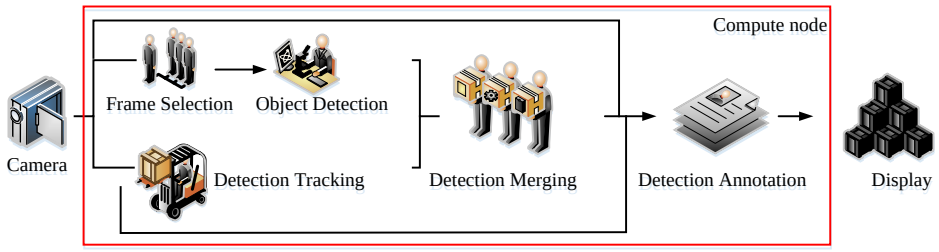


Figure 5: Media Pipe infrastructure

In addition, the model also needs to further implement a module for real-time inference, and a lightweight CNN network Blaze Pose is studied and introduced, which is specifically designed for mobile device applications. When using the Blaze Pose network for inference, 33 bone key points can be extracted through the detector tracker module. The tracker can predict the position of key points, as well as predict whether there is a target in the image frame and the pose interest region of the image frame. If the presence of human targets is not detected in the corresponding image frame, continue to re detect and predict tracking in the next frame. This working mode determines that the dependency between adjacent frames is strong. Embed the Blaze Pose network into the Media Pipe framework to obtain the Media Pipe Pose model. The Blaze Pose network is a stacked encoder-decoder heatmap network, along with a regression encoder network. The model training applies heat map and offset loss, and deletes the corresponding output layer during inference. Jump connections are used between different stages to balance high and low-level functions. However, the regression encoder gradient stops the connection. This strategy can effectively improve prediction accuracy and coordinate regression accuracy. The number of human bone keypoints in the Blaze Pose network has increased to 33, and the head, hand, and foot keypoints have been refined, as shown in Figure 6.

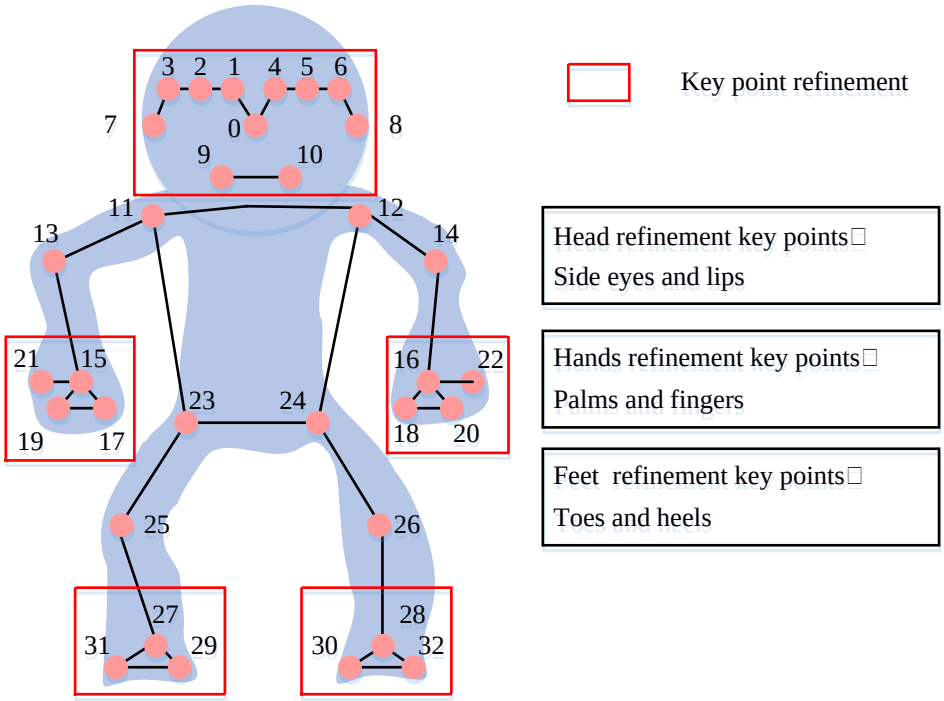


Figure 6: Detailed analysis of key points

In Figure 6, the newly added outer and inner sides of the left and right eyes on the head are 3/6 and 1/4, respectively, and the left and right lips are 9/10, respectively. In the newly added key points of the hand, the left and right little thumbs are 17/18, 19/20 respectively for the left and right index fingers, and 21/22 respectively for the left and right thumbs. The feet have been added with 29/30 on the left and right heels, and 31/32 on the left and right toes. The remaining key points in the model are consistent with the original. The human

body posture can be divided into 8 types: standing, walking, squatting, waving, bending, kicking, as well as side pressing and sitting. After obtaining attitude data, normalization processing is required, which means that all key points are mapped to the (0,1) interval. In the normalization of the horizontal axis in a two-dimensional coordinate system, the normalized horizontal axis is shown in formula (10).

$$X_1 = \frac{X}{W} \tag{10}$$

In formula (10), X / X_1 respectively represent the horizontal coordinates of the key points before and after normalization, while W is the symbol that denoting the display camera width. Correspondingly, the vertical coordinates in the two-dimensional coordinate system are normalized, as shown in formula (11).

$$Y_1 = \frac{Y}{H} \tag{11}$$

In formula (11), Y / Y_1 represent the vertical coordinates of the key points before and after normalization, and H represents the length of the display camera. However, some key points in the image may be obscured. To avoid changes in the dimensionality of the feature vector, its coordinates should be supplemented with (0,0).

4 Performance and practical application analysis of action recognition network based on MPP-YOLOv3 tiny

To assess the reliability of the proposed deep learning network for action recognition, the study conducted experiments on the model's performance, including its training and validation error, detection speed, and weight file size. Subsequently, practical application analysis is conducted to understand the specific recognition performance for different actions, with the demand to verify its superiority in action recognition.

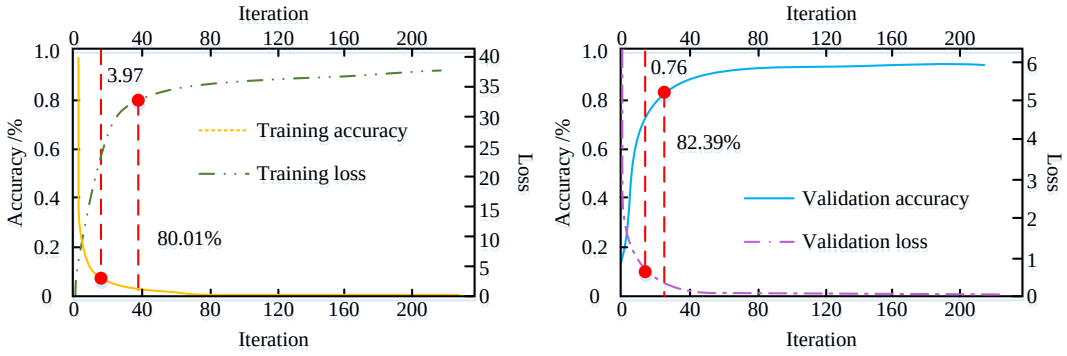
4.1 Performance verification of action recognition model based on MPP-YOLOv3 Tiny network

The study first investigated the performance of each module in the MPP-YOLOv3 Tiny network. The specific experimental environment and parameter selection are shown in Table 2.

Table 2: Experimental environment and parameter settings

Name	Settings
Operating system	Ubuntu 18.04
GPU	NVIDIA Quadro M2200
CPU	Intel Xeon CPU E3-1505M v6 @3.00GHz
RAM	16GB
CUDA	10.1
Programming language	Python 3.6
Epochs	220
Batch size	64
Learning_rate	0.0001
Optimizer function	Adam

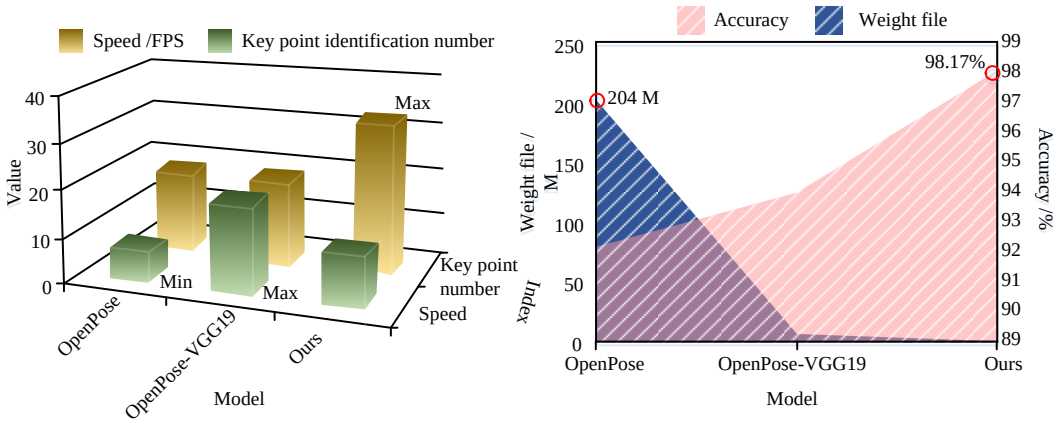
The study selects the CityPersons public dataset for experimentation, which contained a total of 3475 image data. Among them, the number of test sets is 500. The accuracy and loss curve of the model for training and validation are shown in Figure 7.



(a) Model training performance (b) Model validation test performance

Figure 7: Model training verification performance analysis

In Figure 7 (a), as iterations increased, the curve of the training data rapidly improved and eventually stabilized. As the model approached 40 iterations, the accuracy curve became smoother, and ultimately reached 93.01% in the 230 th iteration. The loss value during training began to flatten out around the 20 th iteration, and at this point, the training loss decreased to 0.085. At the 230 th iteration, the training loss of the model is only 0.0462. In Figure 7 (b), the model validation data quickly drops to a flat state around the 22 nd iteration. As the number of iterations increases, its accuracy gradually improves. At the 230 th iteration, the accuracy of the validation data reaches 96.97%. Moreover, its validation loss curve also rapidly decreases to a steady state after around the 15 th iteration. In the 230 th iteration, the validation loss value is only 0.0417. Overall, the model performs well in both training and validation. Further research compares the designed MPP-YOLOv3 Tiny model with the pre-optimized OpenPose algorithm model and OpenPose-VGG19 algorithm model, as shown in Figure 8.



(a) Comparison of speed and key points before and after optimization

(b) Comparison of file size and accuracy before and after optimization

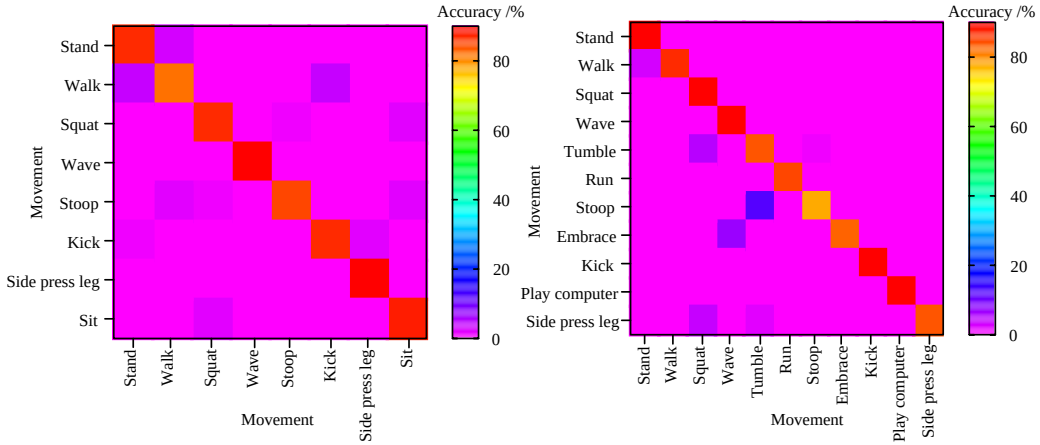
Figure 8: Comparison of model performance before and after optimization

Figure 8 (a) shows the comparison of detection speed and number of key point recognition before and after optimization of the model. The MPP-YOLOv3 Tiny network designs for research could recognize up to 33 key points of human bones, while the two models before optimization could only recognize 18 key points of human bones. Additional key points may improve the recognition accuracy of human body movements in the image, but may also result in slower recognition speed. Therefore, the detection speed of the research design model is 11 FPS, which is 36.36% lower than the 7 FPS of the OpenPose algorithm model. However, compared to the OpenPose-VGG19 algorithm, the detection speed is still 42.11% higher. In Figure 8 (b), the accuracy of the research design model reaches 98.17%, while the accuracy of the two models before optimization is below 95%. The accuracy of the MPP-YOLOv3 Tiny network is improved by an average of 5.01%. The weight file for the

research and design model is 1.6 M, which is 99.22% lower than the OpenPose algorithm. Compared to the OpenPose-VGG19 algorithm, it reduces by 79.94%. In summary, although the detection speed of the research and design model is slightly lower than that of the OpenPose algorithm, its overall performance is the best. Therefore, studying the optimization of initial action recognition models is effective and reliable.

4.2 Practical analysis of posture correction model based on MPP-YOLOv3 tiny network

With the demand to do recognition performance verification for 8 common actions, a pose dataset is constructed using MediaPipe Pose. In addition, the study also introduces running, hugging, computer operation, and falling movements that are easily confused with other movements for further comparison. The total number of images is 17980, and the experiment randomly divides them into a training set and a validation set in an 8:2 ratio. The experiment outcomes are given as Figure 9.



(a) Common attituderecognition results (b) Confusing attitude recognition results

Figure 9: Different action recognition accuracy analysis

Figure 9 (a) shows the visualization of the recognition effect of the model on common actions. Among them, the recognition accuracy of the model for side leg pressing and hand waving movements is 100%, and the recognition accuracy for other movements is also above 90%. The recognition accuracy of walking movements is 91.1%, because walking movements are easily confused with standing and kicking movements, with recognition errors of 4.4% for both movements. The recognition error for squatting and sitting movements is 2.2%, and the recognition error for kicking and lateral pressure is also 2.2%. Bending, squatting, walking, and sitting movements are easily confused, but the errors are all within 2.2%. Overall, the model achieves an average recognition accuracy of 96.7% for common actions. In Figure 9 (b), four movements including running have been added, while sitting and standing movements have been removed. The model has achieved 100% recognition accuracy for five movements: standing, squatting, waving, kicking, and playing computer games. Except for the bending motion, the recognition accuracy for all other movements has also reached over 92%. The recognition accuracy of bending and falling movements is only 86.7%, because bending and falling movements are easily confused, with a recognition error of 12.3%. Further research will apply the MPP-YOLOv3 Tiny action recognition network to posture correction. Taking squatting as an example, the model's recognition effect on incorrect squatting movements is shown in Figure 10.

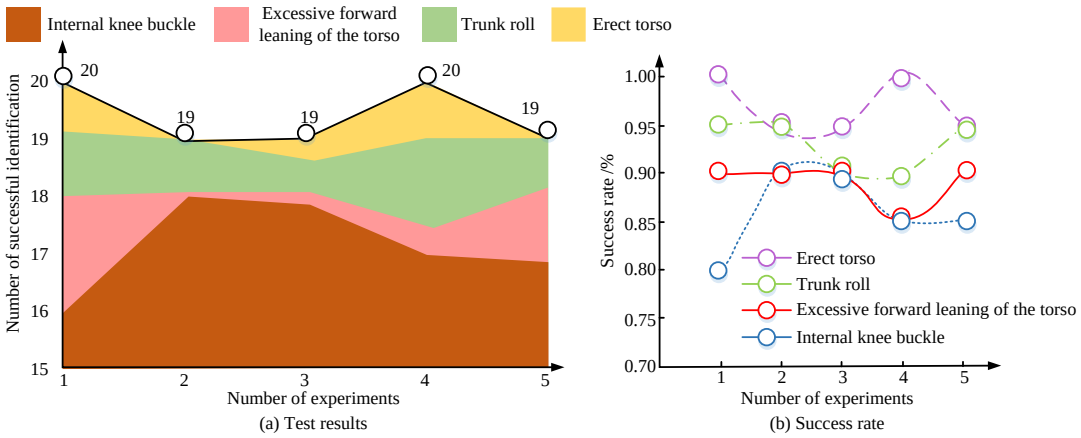


Figure 10: Recognition results of squat posture correction

Figure 10(a) shows that the model's overall recognition performance for the knee joint buckle is poor, with a recognition accuracy of only 87.4%. This is mainly due to recognition errors that occur during the motion amplitude of this action. The recognition accuracy for excessive forward leaning of the trunk is relatively high, reaching 92.6%. However, the main reason affecting its recognition is the obstruction of the excessive forward leaning surface. The recognition rate for trunk tilt and excessive upright posture is 96.5%. Figure 10(b) indicates that more than 19 individuals are successful, primarily due to significant changes in their movements. Next, the study compares the semantic action recognition model based on Pose Lexicon proposed by reference [7], and the action recognition model based on graph transformer network proposed by reference [8]. The experimental datasets used are the NTU60 RGB+D dataset and the NTU120 RGB+D dataset, respectively, with the latter being an extension of the former. The experiment outcomes are given as Table 3.

Table 3: Comparative analysis of the performance of each model

Models	Index	Data sets	
		NTU60 RGB+D	NTU120 RGB+D
Zhou. et al [7]	CV	93.6%	82.6%
	CS	86.1%	78.8%
Liu. et al [8]	CV	92.5%	82.1%
	CS	85.2%	74.8%
Ours	CV	94.7%	82.7%
	CS	89.3%	80.6%

The indicators in Table 3 represent the recognition accuracy of the model from the perspectives of Cross View (CV) and Cross Subject (CS), respectively. It could be concluded that all models performed better in the NTU60 RGB+D dataset. However, the research design model is still 1.7% better than the other two models in terms of CV index, while the CS index is 3.2% better than the other two models. In the NTU120 RGB+D dataset, the recognition accuracy of each model has decreased to below 90%. The CV index of the research design model in this dataset is increased by 0.35% compared to the other models. The CS index has relatively increased by 3.8%. In summary, the MPP-YOLOv3 Tiny network proposed in the study can better achieve pose correction through action recognition. Additionally, eight tests are conducted to compare the performance of various models in terms of recall, precision, and F1 values, as depicted in Figure 11.

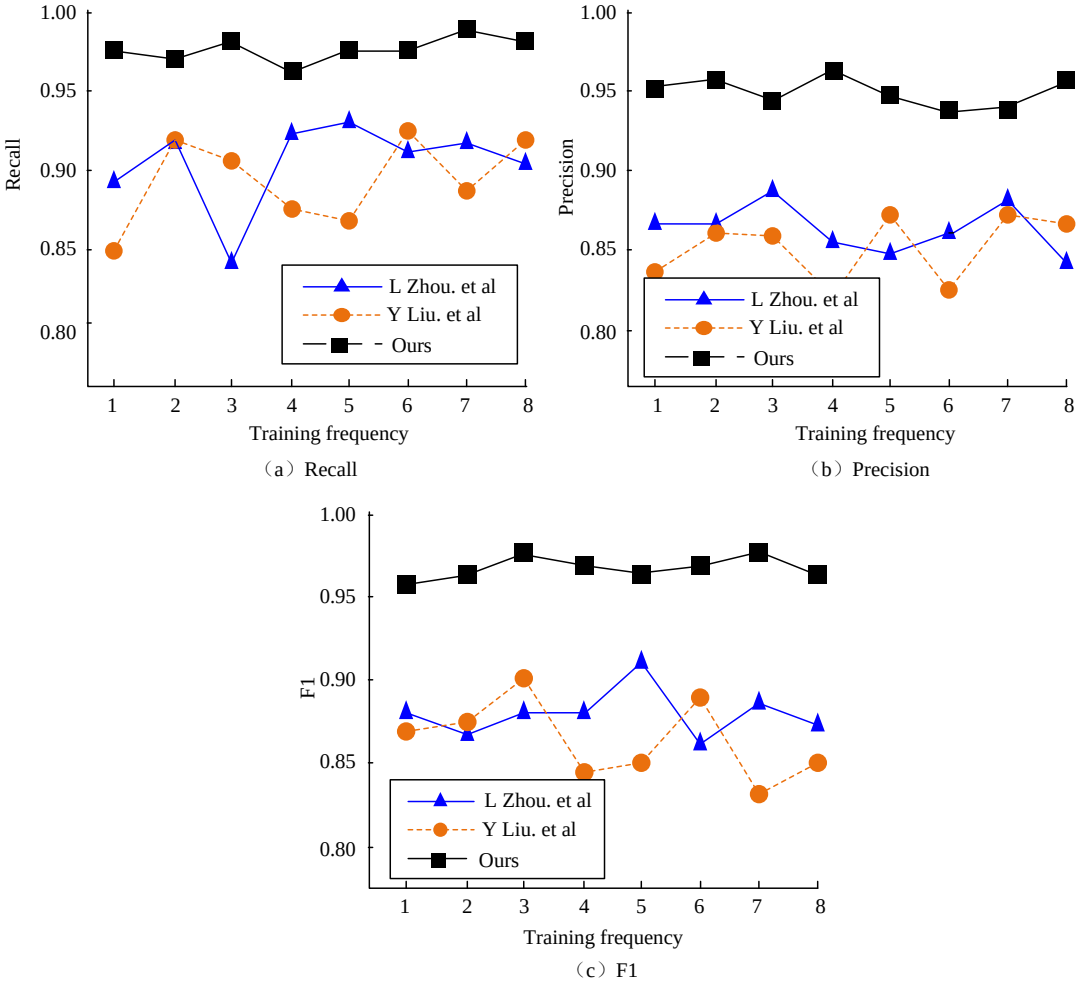


Figure 11: Test results of recall, accuracy, and F1 value for different models

Figures 11 (a) to 11 (c) display the results of the recall, accuracy, and F1 value tests, respectively. The research design model maintained a recall rate of 0.95 or higher in 8 rounds of testing, outperforming other models. However, the model proposed by L Zhou et al performed poorly, with a recall rate below 0.85 in the third round of testing and poor stability. When comparing accuracy, the design model consistently performed excellently with a stability of 0.93 or above, while the other two models had an accuracy range of 0.85 and poor overall stability. In terms of the F1 value indicator, the model proposed by Y Liu et al showed significant fluctuations in the 4th and 7th rounds, performing worse than the design model overall. However, the F1 values of the design model are all above 0.94 with excellent stability. To test the recognition performance of the design model, conventional action data are collected in real environments as a self-generated training dataset for simple scenes, with a total of 1565 entries. Additionally, 1756 complex scene data, such as squatting, kicking, and occlusion, are selected as a self-generated complex scene dataset. These two datasets are used to produce recognition results for real-world scenarios, as shown in Figure 12.

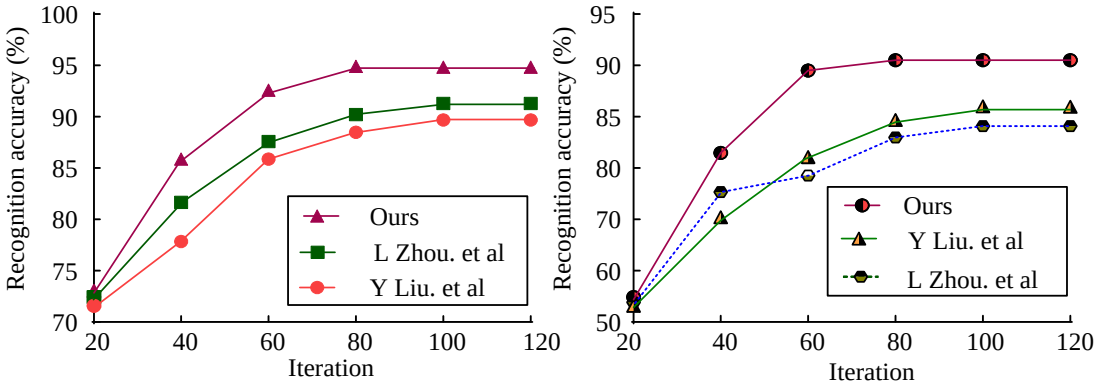


Figure 12: Comparison of recognition effects of different models in real scenarios

In Figure 12 (a), the results of action recognition in a simple scenario are presented. The designed model achieved convergence after 80 iterations, with an accuracy of 95.25%. Both the models proposed by L. Zhou et al. and Y. Liu et al. achieved convergence after 100 iterations. However, the overall recognition accuracy of the model proposed by L. Zhou et al. was 90.25%, while that of Y. Liu et al. was 88.21%. In complex scenarios, the recognition accuracy of all three models significantly decreased. According to Zhou et al, their proposed model exhibited poor overall stability and lower recognition accuracy than Liu's model in the later stage. The main reason for this may be the model's inability to handle the extraction of multivariate features in complex scenes, which affects the accuracy of later recognition. In comparison, the overall design model outperformed the other two models with a recognition accuracy of 90.95% when converging. In summary, the study found that the MPP-YOLOv3 algorithm has better practical application results. This was due to its lightweight design and enhanced feature extraction of upper and lower limbs, which ensured accuracy and stability of the technology. In complex recognition scenarios, similar models with less feature extraction will significantly decrease accuracy. However, research models optimized the recognition process through lightweight design and DSC strategy to ensure recognition accuracy.

4.3 Discussion

The application of action recognition technology in correcting posture holds significant research value. This technology allows for real-time monitoring and analysis of human body posture, aiding in the evaluation and correction of poor posture, and ultimately improving overall health and motor skills. Reference [10] proposed an action recognition method based on spatiotemporal slow fast graph convolutional networks. However, this method was susceptible to external noise during data collection and action recognition processes, which can decrease the quality of the collected data and affect the recognition effect of the final action.

Reference [11] proposed a technique for action recognition with minimal transfer learning. However, this technology exhibited high stability and recognition accuracy during the training process. As the training progresses, the stability gradually decreased and cannot meet the needs of practical applications. Reference [12] proposed a motion recognition technique based on wavelet transform. This technology relied on a large amount of collected data. The overall recognition accuracy of this method significantly decreased with fewer data samples, limiting its feasibility and accuracy in practical applications. Therefore, this study examined the application of action recognition in attitude correction using the MPP-YOLOv3 algorithm. The study compared and analyzed the techniques proposed in references [10], [11], and [12], and found that these methods have limitations. In contrast, the research method has advantages. Firstly, it can effectively overcome external noise during data collection and action recognition. The main focus of this study is to explore methods for reducing the number of predicted classifications in order to lower network operating parameters and improve data quality and action recognition performance. Secondly, the research method exhibited high stability and recognition accuracy during the training process. This was especially true with the introduction of the Blaze Pose network, which significantly improved the inference process and makes the research technology more suitable for practical applications. Additionally, when compared to literature [12], the research method demonstrated lower dependence on data samples while maintaining high recognition accuracy even with fewer data samples.

In summary, the study found that the action recognition method based on the MPP-YOLOv3 algorithm has better application effects in attitude correction, higher recognition accuracy, and stronger stability. However, further experiments and research are needed to validate and improve the research methods, in order to enhance their accuracy and stability, and to promote their widespread application in practical settings.

5 Conclusion

To correct posture and apply it to various fields, such as sports and medicine, research proposes using deep learning for action recognition. First, a lightweight network infrastructure based on YOLOv3 Tiny was built. Then, the model's accuracy was improved by refining bone key points through the MPP module. In the experimental analysis on the CityPersons public dataset, the results showed that the curves of model accuracy and loss values tended to stabilize with increasing iterations, and reached a training accuracy of 93.01% at 230 iterations. The validation accuracy and loss values were 96.97% and 0.0417, respectively. Although its detection speed was slightly lower at 11 FPS, compared to the pre optimized OpenPose algorithm and OpenPose-VGG19 algorithm, the accuracy had improved by 3.2% and the model size has increased by 83.1%. In practical application analysis, the MPP-YOLOv3 Tiny network achieved a recognition accuracy of 96.7% for common actions, and can accurately recognize confusing actions such as running, hugging, operating a computer, and falling in datasets, demonstrating good generalization ability. When applied to pose correction for squatting movements, the success rate of identifying knee joint buckles was 87.4%, and the average success rate of identifying other erroneous movements reached 94.7%. This indicated that the model had practical value in pose correction. Compared to the action recognition model proposed by Zhou et al. [7] and the model proposed by Y Liu et al. [8], in the NTU60 RGB+D and NTU120 RGB+D datasets, the average CV index improved by 1.02% and the average CS index improved by 3.5%. In summary, the action recognition model based on MPP-YOLOv3 Tiny network proposed in the study has significant application value in posture correction. However, the detection speed of the action recognition model designed for research still needs to be improved. In the future, further efforts should be made to enhance the recognition speed of this module while ensuring lightweight and accuracy.

References

[1] Y. Bai, Q. Zou, X. Chen, L. Li, Z. Ding, and L. Chen, "Extreme Low-Resolution Action Recognition with Confident Spatial-Temporal Attention Transfer," International Journal of Computer Vision, vol. 131, no. 6, pp. 1550-1565, 2023. https://doi.org/10.1007/s11263-023-01771-4

[2] K. Bhosle, and V. Musande, "Evaluation of deep learning CNN model for recognition of devanagari digit," Artificial Intelligence Applications, vol. 1, no. 2, pp. 114-118, 2023. https://doi.org/10.47852/bonviewAIA3202441

[3] H. Liu, Y. Chen, W. Zhao, and S. Zhang, "Human pose recognition via adaptive distribution encoding for action perception in the self-regulated learning process," Infrared Physics And Technology, vol. 114, no. 1, pp. 103660-103669, 2021. https://doi.org/10.1016/j.infrared.2021.103660

[4] R. Kumar, and S. Kumar, "Survey on artificial intelligence-based human action recognition in video sequences," Optical Engineering, vol. 62, no. 2, pp. 23102-23123, 2023. https://doi.org/10.1117/1.OE.62.2.023102

[5] C. Bian, W. Feng, L. Wan, and S. Wang, "Structural knowledge distillation for efficient Skeleton-Based action recognition," IEEE Transactions on Image Processing, vol. 30, no. 1, pp. 2963-2976, 2021. https://doi.org/10.1109/TIP.2021.3056895

[6] A. Kumar, S. Majee, and S. Jain, "CDM: A coupled deformable model for image segmentation with speckle noise and severe intensity inhomogeneity," Chaos, Solitons & Fractals, vol. 173, no. 3, 104385-104396, 2023. https://doi.org/10.1016/j.chaos.2023.113551

[7] L. Zhou, and T. Jiang, "Learning body part-based pose lexicons for semantic action recognition," IET Computer Vision, vol. 17, no. 2, pp. 135-155, 2023. https://doi.org/10.1049/cvi2.12143

[8] Y. Liu, H. Zhang, D. Xu, and K. He, "Graph transformer network with temporal kernel attention for skeleton-based action recognition," Knowledge-Based Systems, vol. 240, no. 3, pp. 108146-10862, 2022. https://doi.org/10.1016/j.knosys.2022.108146

[9] G. Alemu, Y. Ji, Y. Yang, L. L. Gao, and H. T. Shen, "Arbitrary-View human action recognition via Novel-view action generation," Pattern Recognition, vol. 118, no. 10, pp. 108043-108052, 2021. https://doi.org/10.1016/j.patcog.2021.108043

[10] Z. Fang, X. Zhang, T. Cao, Y. Zheng, and M. Sun, "Spatial-Temporal slowfast graph convolutional network for Skeleton-Based action recognition," IET Computer Vision, vol. 16, no. 3, pp. 205-217, 2021. https://doi.org/10.1049/cvi2.12080

[11] H. Coskun, M. Z. Zia, B. Tekin, F. Bogo, N. Navab, F. Tombari, and H. S. Sawhney, "Domain-Specific priors and meta learning for Few-Shot First-Person action recognition," IEEE Transactions on Software Engineering, vol. 45, no. 6, pp. 6659-6673, 2021. https://doi.org/10.48550/arXiv.1907.09382

[12] X. Li, W. Zhai, and Y. Cao, "A tri‐attention enhanced graph convolutional network for Skeleton‐Based action recognition," IET Computer Vision, vol. 15, no. 2, pp. 110-121, 2021. https://doi.org/10.1049/cvi2.12017

[13] X. Hao, J. Li, Y. Guo, T. Jiang, and M. Yu, "Hypergraph neural network for Skeleton-Based action recognition," IEEE Transactions on Image Processing, vol. 30, no. 1, pp. 2263-2275, 2021. https://doi.org/10.1109/TIP.2021.3051495

[14] T. Le, N. Huynh-Duc, C. T. Nguyen, and M. T. Tran, "Motion embedded images: an approach to capture spatial and temporal features for action recognition,"

Informatica, vol. 47, no. 3, pp. 327-329, 2023. https://doi.org/10.31449/inf.v47i3.4755

[15] P. Climent-Pérez, and F. Florez-Revuelta, "Improved action recognition with separable Spatio-Temporal attention using alternative skeletal and video Pre-Processing," Sensors, vol. 21, no. 3, pp. 1005-1005, 2021. https://doi.org/10.3390/s21031005

[16] Z. Liu, J. Cheng, L. Liu, Z. Ren, Q. Zhang, and C. Song. "Dual-stream cross-modality fusion transformer for RGB-D action recognition," Knowledge-based Systems, vol. 255, no. 11, pp. 109741-109752, 2022. https://doi.org/10.1016/j.knosys.2022.109741

[17] M. Gutoski, A. E. Lazzaretti, and H. S. Lopes, "Deep metric learning for open-set human action recognition in videos," Neural Computing & Applications, vol. 33, no. 4, pp. 1207-1220, 2021.

[18] M. Smith, and R. Toumi, "Using video recognition to identify tropical cyclone positions," Geophysical Research Letters, vol. 48, no. 7, pp. 1-9, 2021. https://doi.org/10.1029/2020GL091912

[19] D. Lee, D. Wang, Y. Yang, L. Deng, and G. Li, "QTTNet: Quantized tensor train neural networks for 3D object and video recognition," Neural Networks, vol. 141, no. 5, pp. 420-432, 2021. https://doi.org/10.1016/j.neunet.2021.05.034

[20] 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