Viser Technology Pte. Ltd.
Mechanical Engineering Science
DOI: 10.33142/mes.v6i1.13221

Research Article Open Access

Computer Vision-Based Human Body Posture Correction System

Yangsen QIU, Yukun WANG, Yuchen WU, Xinyi QIANG, Yunzuo ZHANG *

School of Information science and Technology, Shijiazhuang Tiedao University, Shijiazhuang, Hebei, 050043, China

****Corresponding Author:*** *Yunzuo ZHANG, E-mail: zhangyunzuo888@sina.com*

Abstract

With the development of technology and the progress of life, more and more people, regardless of entertainment, learning, or work, cannot do without computer desks and cannot put down their mobile phones. Due to prolonged sitting and often neglecting the importance of posture, incorrect posture can often lead to health problems such as hunchback, lumbar muscle strain, and shoulder and neck pain over time. To address this issue, we designed a computer vision-based human body posture detection system. The system utilizes YOLOv8 technology to accurately locate key points of the human body skeleton, and then analyzes the coordinate positions and depth information of these key points to establish a criterion for distinguishing different postures. With the assistance of an SVM classifier, the system achieves an average recognition rate of 95%. Finally, we successfully deployed the posture detection system on Raspberry Pi hardware and conducted extensive testing. The test results demonstrate that the system can effectively detect various postures and provide real-time reminders to users to correct poor posture, demonstrating good practicality and stability.

Keywords: *computer vision; human posture; deep learning; image processing*

1 Introduction

With the development of modern society, the pressure of learning and work is increasing. People often need to sit for long periods to work and study, maintaining a correct sitting posture for an extended period is quite challenging, leading to various poor sitting postures such as hunching, leaning, and tilting. Prolonged incorrect sitting posture may lead to various health issues including hunchback, lumbar muscle strain, neck and shoulder pain, myopia, and strabismus, seriously affecting physical and mental health. Correct posture and posture are crucial for physical health. Therefore, we designed a computer vision-based human posture correction system aimed at timely reminding users of incorrect sitting postures through real-time posture recognition and voice broadcast technology, correcting erroneous sitting postures, and reducing the harm of poor sitting postures to the body. We hope to suppress health problems caused by incorrect posture from the root, help users improve poor posture, and prevent and reduce posture-related health problems. The system relies on deep learning training models, with high accuracy and real-time performance, while possessing good practicality and research significance, thereby helping users maintain the correct posture and ensure physical health.

2 System Architecture and Hardware Design

This system is mainly based on the Raspberry Pi 4B processor motherboard and Raspberry Pi Camera Module V2 camera. OpenCV is used as the platform for visual processing tasks, combined with YOLOv8 for object recognition and keypoint detection technology. The camera accurately locates the positions of the facial features and various skeletal points of the body, combined with SVM support vector machine model for posture judgment, and implements the function of correcting human sitting posture. If the posture is incorrect, an alarm and voice prompt will be issued. The system architecture is shown in Figure 1.

The system hardware mainly consists of the main processor Raspberry Pi 4B motherboard, camera module V2, LCD touchscreen, speaker module, power circuit, etc. The processor motherboard receives data from the LCD touchscreen to complete user-specified operations through parameter settings. Then, it processes the data from the camera module to judge and recognize the sitting posture, and the speaker module provides reminders. This system is compact and supports both Windows and Linux operating systems, suitable for small and portable designs overall. The circuit connection diagram is shown in Figure 2, and the physical

Received on May 15, 2024; Accepted on June 11, 2024

appearance of the system is shown in Figure 3.

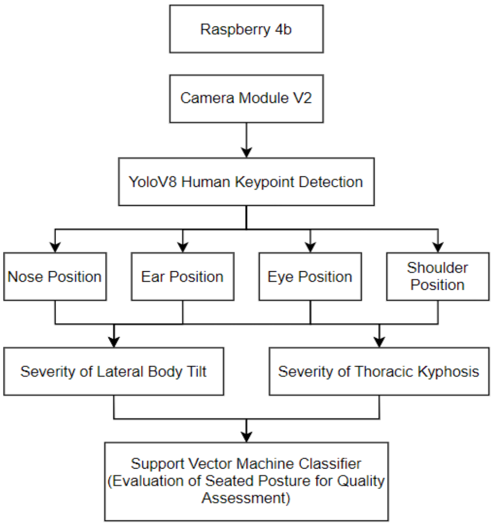

Figure 1 **System Architecture Diagram**

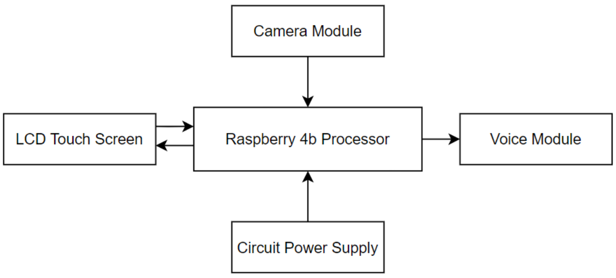

Figure 2 Circuit Connection Schematic Diagram

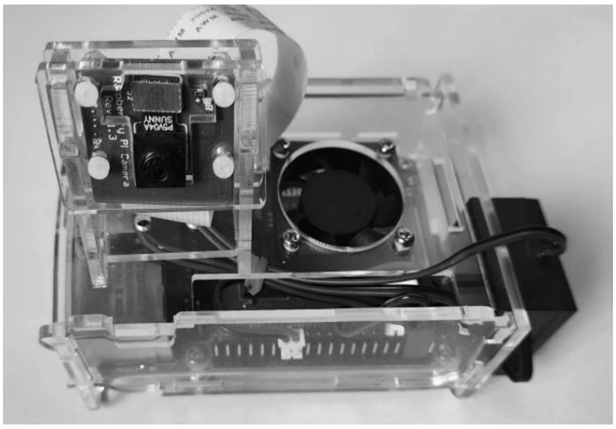

Figure 3 System Physical Appearance

3 Program and Algorithm Design

3.1 Program design

The main program flowchart is shown in Figure 4. After successful login and user operations, the model is first loaded. Simultaneously, the camera module is invoked, and upon collecting data, the captured video frames are passed to the system for processing. The system identifies the positions of key points such as eyes, nose, ears, and shoulders in the video frames. After processing with our designed algorithm, the human body posture in the video frame can be accurately distinguished. If the user adopts poor postures such as hunching or leaning, it will be marked as "bad" and a voice reminder will be issued; otherwise, it will be marked as "good". In addition to correcting poor postures, the system also sets reminders for prolonged sitting. If the user reaches the preset sitting time, the system will also issue a voice reminder.

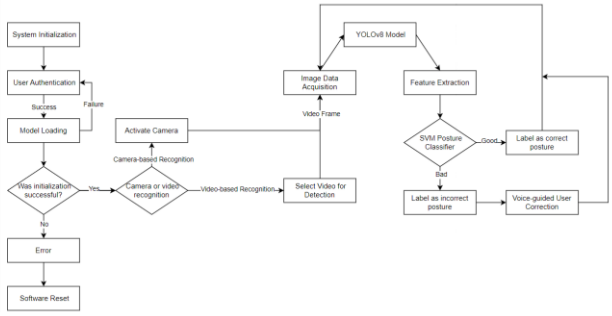

Figure 4 Main Program Flowchart

3.2 Algorithm design

3.2.1 Key-point detection and localization

In the human posture correction system, keypoint detection and localization are crucial stages based on computer vision technology. The neural network structure of this system is implemented based on the state-of-the-art object detection model - YOLOv8n, which integrates target detection and keypoint localization of the human body. The model network structure is divided into three parts: feature extraction network, feature fusion network, and detection head. Among them, the feature extraction network adopts CSPDarknet53, which includes a series of convolutional layers and residual blocks to extract high-level features from input images. The feature fusion network includes convolutional layers and upsampling layers to fuse feature maps of different resolutions, in order to better adapt to targets of different scales. Finally, the detection head consists of a series of convolutional layers and the final output layer, used to predict the positions of detection boxes and keypoint coordinates.

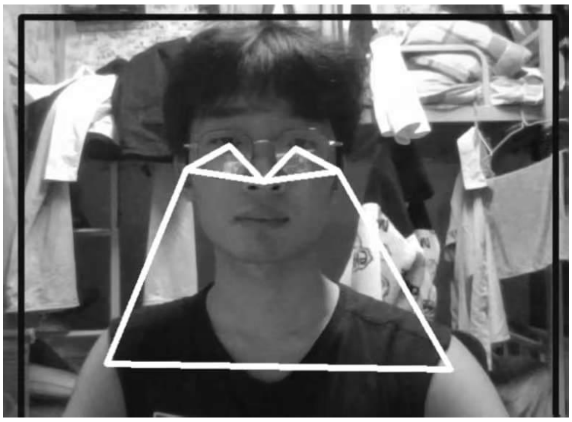

Figure 5 Key-point Connection Diagram

Additionally, we use two-dimensional arrays created with Numpy to store the keypoint coordinates output by the YOLOv8 model, including the positions of eyes, ears, nose, and shoulders. This provides reliable data support for subsequent posture analysis in the system, thereby achieving precise monitoring and analysis of user posture. The schematic diagram of keypoint connections is shown in Figure 5.

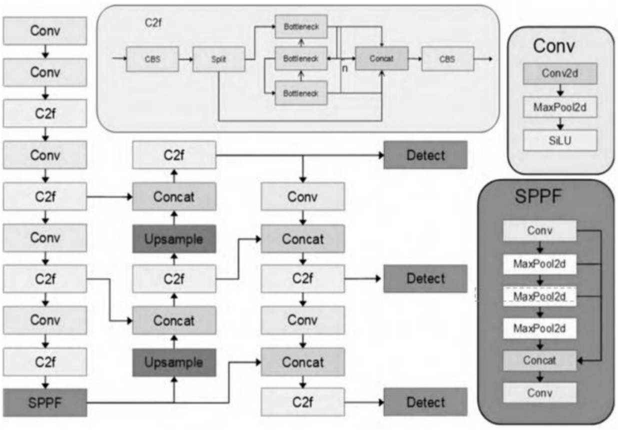

Figure 6 YOLOv8 Network Architecture

The YOLOv8 network architecture is depicted in Figure 6. It consists of an input end, Backbone, and Neck, describing the overall network structure. In YOLOv8, the input end adopts adaptive anchor calculation and adaptive grayscale padding. The backbone network includes Conv, C2f, and SPPF structures. The C2f module is mainly used to learn residual features and is inspired by the ELAN structure of YOLOv7. By introducing more branch cross-layer connections, it enhances the gradient flow of the model, forming a neural network module with stronger feature representation capability. Additionally, the Neck module adopts the PAN (path aggregation network) structure to enhance the network's ability to fuse features of objects at different scales.

3.2.2 Pose analysis and classification

In the stage of pose analysis and classification, this paper proposes to fuse the keypoint data mentioned above into two feature values, f1 and f2. These features are then inputted into an SVM classifier to accurately assess and classify the user's pose. The specific steps are as follows:

(1) Definition of Data to be Calculated

We first define the data to be calculated, including the coordinates of the left ear, right ear, left shoulder, and right shoulder, as well as the distances between these points. Specifically, the distance from the left ear to the left shoulder is defined as s1, the distance from the right ear to the right shoulder is defined as s2, and the distance from the left shoulder to the right shoulder is defined as s3. These are illustrated in Table 1 below.

To extract a representative midpoint for the positions of the left (right) eye, left (right) ear, and nose, and thus represent the center point of the body position, we choose to calculate the incenter coordinates of the triangle formed by these three points. The incenter is equidistant from the three sides of the triangle, making it a good representation of the central position of these three points. Using Equation 1 and Equation 2, we can obtain the coordinates of the left face midpoint, respectively. Similarly, we can derive the coordinates of the right face midpoint.

Table 1 Data Quantity Table

Data	Definition	Value
The position of the left ear	LeftEar	{x1,y1}
The position of the right ear	RightEar	{x2,y2}
The position of the left eye	LeftEye	{x3,y3}
The position of the right eye	RightEye	{x4,y4}
The position of the nose	Nose	{x5,y5}
The position of the left shoulder	LeftShoulder	{x6,y6}
The position of the right shoulder	RightShoulder	{x7,y7}
The position of the midpoint of the left side of the face	LFaceCenter	{x8,y8}
The position of the midpoint of the right side of the face	RFaceCenter	{x9,y9}
The distance from the midpoint of the left side of the face to the left shoulder	Left_face_Shoulder	s1
The distance from the midpoint of the right side of the face to the right shoulder	Right_face_Shoulder	s2
The distance from the left shoulder to the right shoulder	Shoulder_Shoulder	s3

$$x_8 = \frac{x_1\sqrt{(x3-x1}+x2\sqrt{(x5-x3)^2+(y_5-y_3)^2}+x3\sqrt{(x5-x1)^2+(y_5-y_1)^2}}{\sqrt{(x3-x1)^2+(y_3-y_1)^2}+\sqrt{(x5-x3)^2+(y_5-y_3)^2}+\sqrt{(x5-x1)^2+(y_5-y_1)^2}} \quad (1)$$

$$y_8 = \frac{y_1\sqrt{(x3-x1}+y2\sqrt{(x5-x3)^2+(y_5-y_3)^2}+y3\sqrt{(x5-x1)^2+(y_5-y_1)^2}}{\sqrt{(x3-x1)^2+(y_3-y_1)^2}+\sqrt{(x5-x3)^2+(y_5-y_3)^2}+\sqrt{(x5-x1)^2+(y_5-y_1)^2}} \quad (2)$$

Equation (3) represents the value of s_1, Equation (4) represents the value of s_2, and Equation (5) represents the value of .

$$s_1 = \sqrt{(x_6 - x_8)^2 + (y_6 - y_8)^2} \tag{3}$$

$$s_2 = \sqrt{(x_7 - x_9)^2 + (y_7 - y_9)^2} \tag{4}$$

$$s_3 = \sqrt{(x_6 - x_7)^2 + (y_6 - y_7)^2} \tag{5}$$

(2) Calculation of Feature Values

Let feature value 1 be f_1(to exclude the influence of the distance between the subject and the camera, normalization is required), as shown in Equation (6).

$$f_1 = \frac{|(s_1 - s_2) \times 10|}{s_3} \tag{6}$$

Feature value f_1 is used to measure the tendency of the user to lean to the left or right. Based on anatomical observations and experimental results, when the body is not leaning in a seated position, from a frontal perspective, the torso and neck should maintain a relatively stable vertical position, with the body's left and right sides being relatively symmetrical and the head positioned above the centerline. Therefore, when the user maintains good posture, the $f_1$value should be close to 0, indicating that the user's body is not leaning to the left or right.

From experimental results, it can be observed that when the user leans to the left or right, the torso and neck will exhibit a tilting state on one side, causing noticeable displacement of the torso and head relative to the other side. This will cause the f_1 value to deviate from 0, leaning towards the left or right. Specifically, when the user leans to the left or right, the value of f_1 will be larger.

Let feature value 2 be f_2(to exclude the influence of the distance between the subject and the camera, normalization is required), as shown in Equation (7).

$$f_2 = \frac{(s_1 + s_2)}{s_3} \tag{7}$$

This feature value $f_2$is primarily used to determine whether the user exhibits a tendency to hunch or slouch. Based on anatomical observations and experimental results, when the body is in an upright seated position, the torso and neck are in a natural upright state, the angle between the body and the ground is relatively stable, and the head position relative to the shoulders is also normal. Therefore, when the user maintains good posture, the f_2 value should be close to an ideal baseline value, typically between 1.2 and 1.4.

From experimental results, it can be observed that when the user hunches or slouches, the torso and neck will exhibit a certain degree of forward inclination, and the head position relative to the shoulders will be lower. This results in a decrease in the f_2 value, deviating from the baseline value of 1.33. Specifically, when the user hunches or slouches, the $f_2$value may be below 1.1, or even below 0.9.

(3) Error Threshold Analysis

Physiological Differences: There are inherent physiological differences in the human body's structure and form, leading to minor tilting or asymmetry in different individuals when seated. These physiological differences may stem from factors such as skeletal structure, muscle development, body proportions, etc., making it difficult to completely eliminate fluctuations in feature values even in a good seated posture.

Environmental Factors: In real-world usage scenarios, users may be in different environmental conditions, such as different chair heights, desk heights, lighting conditions, etc. These factors may affect the user's posture performance. Even if users try to maintain correct posture, changes in environmental factors may still cause slight fluctuations in feature values.

Postural Adjustments: Users may make minor adjustments to their posture during use, such as slight body rotation or adjusting the comfort of their sitting position. Although these minor postural adjustments do not affect the overall quality of the seated posture, they may lead to slight changes in feature values.

Considering the above factors, it is reasonable to set a certain error range for feature values in practical applications. This error range can tolerate minor variations caused by physiological differences, environmental factors, and postural adjustments, while ensuring the stability of the system and user experience. This design not only reduces false positives but also better adapts to individual differences among users and actual usage environments, thereby improving the reliability and practicality of the system.

3.2.3 Classification based on support vector machine

In this study, we employ the powerful machine learning algorithm of Support Vector Machine (SVM) for target classification. SVM shows significant advantages in addressing problems associated with small datasets, non-linearity (by introducing slack variables), and high-dimensional pattern recognition (through kernel functions). Based on the sample data, an SVM model is constructed, and an appropriate kernel function is selected. The kernel function is shown in Equation (8).

$$k(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{8}$$

This paper opts for the Gaussian kernel due to its good adaptability and the ability to control model complexity by adjusting the kernel width. The SVM model from the scikit-learn library is used for data training, validation, and testing. First, samples in the training set, validation set, and test set are labeled. Then, using a grid search method, the penalty parameter C and kernel parameter σ^2 are each assigned N and M values, respectively. Different SVMs are trained for each of the N×M combinations of (C,σ^2), and their generalization accuracy is estimated. The combination that yields the highest generalization accuracy among the N×M combinations of (C, σ^2) is selected as the optimal parameter set. To estimate the generalization accuracy, this paper employs a 5-fold cross-validation method.

4 System Implementation and Testing

4.1 System implementation and results display

Based on the above functional requirements and design plans, we deployed the system on Raspberry Pi hardware. The software developed in this paper was written using PyCharm software. The system interface is shown in Figure 7, enabling the system to receive human target keypoint data and provide intuitive graphical display and posture anomaly identification.

Figure 7 System Interface

For each type of posture including normal sitting posture, hunchback, head tilt to the left, head tilt to the right, body lean to the left, and body lean to the right, the system provides a health assessment. The assessment results are shown in Figure 8: normal sitting posture is labeled as "good" with no voice reminder triggered, while hunchback, head tilt to the left, head tilt to the right, body lean to the left, and body lean to the right are labeled as "bad" with voice reminders triggered.

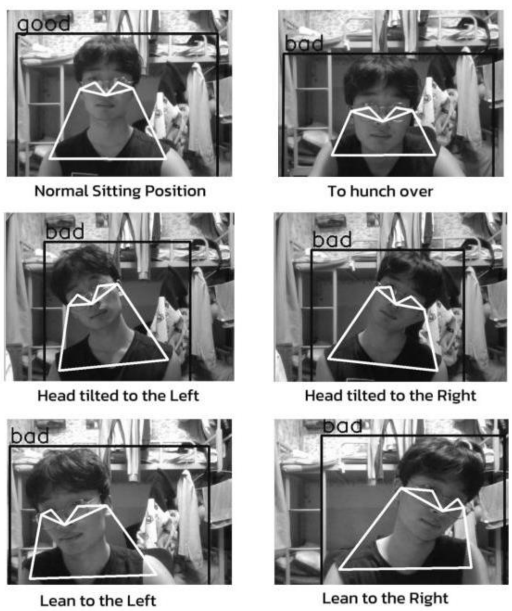

Figure 8 Test Results Diagram

4.2 Testing method and performance estimation

To accurately evaluate the performance of the system in real-world applications, we set up a testing environment that meets the requirements. We positioned the device directly in front of the subject to ensure that the camera can fully capture the upper body of the test subject, with a depth of 1 meter. An example of the setup is shown in Figure 9.

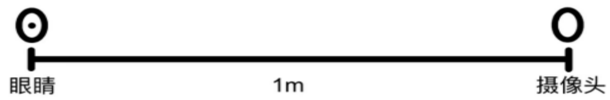

Figure 9 Example Setup for Positioning and Display

We first established criteria to distinguish between different sitting postures, and then conducted health assessments based on these criteria. In the study, we had 10 volunteers (both male and female) sit directly facing the camera and naturally adopt 6 different sitting postures. Each posture needed to be varied continuously, and 100 different images were captured for each posture. In the end, we established the following sitting posture database: including 6 postures, with 100 images for each posture, totaling 6000 images. Through YOLOv8 keypoint extraction, we calculated the f_1 and f_2 values proposed in this paper, and then the SVM classifier made judgments on the health of sitting postures.

For performance evaluation, the data obtained from our volunteers were divided into ten groups per person, and the test results are shown in Table 2. The performance of the SVM classifier is shown in Table 3.

Table 2 Average Feature Values of Volunteers.

Feature Value	Normal Sitting Posture	Hunchback	Head Tilt Left	Head Tilt Right	Body Lean Left	Body Lean Right
f_1	0.031	0.027	1.407	1.361	2.054	1.912
f_2	1.352	0.971	1.191	1.208	1.115	1.186

Table 3 SVM Classifier Test Results.

Index	Recall	Precision
1	0.964	0.964
2	0.929	0.945
3	0.964	0.982
4	0.929	0.945
5	0.964	0.964
6	0.929	0.929
7	0.951	0.982
8	0.975	0.929
9	0.964	0.942
10	0.926	0.934

To validate the overall accuracy of the system's judgments, we invited an additional 10 volunteers for testing. Similarly, each person performed 6 different sitting postures, and 10 frames were captured for each posture. Finally, the accuracy of the judgments was evaluated based on this standard.

Table 4 shows that the detection accuracy based on this standard is relatively high (with an average accuracy of around 95%). Through testing, we found that detecting one frame per second in the actual process is sufficient to meet the requirements. Therefore, this algorithm can basically meet the requirements for accuracy and real-time performance of the sitting posture detection system.

If the judgment method of "if joint in range, good; else, bad" is adopted, the test results are suboptimal due to variations in body proportions among individuals. This stands in stark contrast to the method employed in the previous discussion.

Table 4 Sitting Posture Recognition Rate

Sitting Posture Type	Total Count	Accuracy_svm	Accuracy_In a range
Normal Sitting Posture	100	99%	90%
Hunchback	100	96%	87%
Head Tilt Left	100	94%	91%
Head Tilt Right	100	93%	92%
Body Lean Left	100	92%	89%
Body Lean Right	100	93%	91%

5 Conclusion

This paper proposes a computer vision-based human posture correction system with the aim of correcting poor sitting postures. The camera module v2 combined with YOLOv8 human keypoint detection technology effectively captures the positions of key points on the user's body. The algorithm proposed in this paper preprocesses the position information of human key points into feature values and automatically judges the sitting posture situation based on the SVM method. Experimental results show that the average accuracy of judging 6 types of sitting postures reaches 95%. However, this design has certain limitations; it cannot detect positions arbitrarily. If the human body is not within the range of the camera, the required position information of the human key points cannot be obtained. This limitation is related to the placement of the camera. According to the positioning proposed in this paper, most scene requirements can be fulfilled.

Fund Projects: This work is funded by the Science and Technology Project of Hebei Education Department (No. ZD2022100).

References

[1] Moreira Rayele, et al. A computer vision-based mobile tool for assessing human posture: A validation study[J]. Computer Methods and Programs in Biomedicine, 2022 (214):106565.

[2] Debnath B, O'Brien M, Yamaguchi M, et al. A review of computer vision-based approaches for physical rehabilitation and assessment[J]. Multimedia Systems, 2022,28(1):209-239.

[3] Hellsten T, Karlsson J, Shamsuzzaman M, et al. The potential of computer vision-based marker-less human motion analysis for rehabilitation[J]. Rehabilitation Process and Outcome, 2021(10):1-12.

[4] Andriluka M, Iqbal U, Insafutdinov E, et al. PoseTrack: A benchmark for human pose estimation and tracking[C]. Salt Lake City: IEEE, 2018.

[5] Sun Ke. Hierarchical occlusion reasoning for human pose estimation with incomplete data[J]. IEEE Computer Vision and Pattern Recognition (CVPR), 2021(11):55-59.

[6] Gao Y, Ren Y, Zhang D, et al. Multi-level attention network for fine-grained action recognition[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[7] Zhao Y, Xiong Y, Wang L, et al. Weakly supervised action recognition through contrastive spatiotemporal regions[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[8] Zhang Y, Cui W, Cao Y, et al. Multimodality human action recognition with deep attention mechanism[J]. Computers & Electrical Engineering, 2020(80):221-239.

[9] Tang Y, Liu R. Skeleton Embedding of multiple granularity attention network for human action recognition[C]. International Conference on Articulated Motion and Deformable Objects Springer, 2020.

[10] Liang ZJ, Wang XL, Huang R, et al. An Expressive deep model for human action parsing from a single image[C]. 2014 IEEE International Conference on Multimedia and Expo (ICME). Chengdu: IEEE, 2014.

[11] Zhai W, Zhang Y, Cheng H, et al. Cooperative eye tracking: a gaze-aware interface for 3D object manipulation[C]. Proceedings of the ACM International Conference on Interactive Surfaces and Spaces, 2019.

[12] Yang X, Chen Y, Liu J, et al. Rapid prototyping of tangible augmented reality interfaces: towards exploratory learning for science education[J]. Interactive Learning Environments, 2019,27(4):469-483.

[13] Zhang D, Peng Y, Yang W, et al. Multi-viewpoint interaction with social robots: a case study of speech therapy for children with autism[J]. Journal of Intelligent & Robotic Systems, 2018,92(3-4):359-372.

[14] Zhang R, Li J, Xiao T, et al. BodyPoseNet: Body pose estimation driven by deep neural networks[J]. Signal Processing: Image Communication, 2021(99):116290.

[15] Chen L, Papandreou G, Kokkinos I, et al. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE

Transactions on Pattern Analysis and Machine Intelligence, 2018,40(4):834-848.
[16] Zeng W, Gao Y, Zheng Y, et al. DenseReg: Fully convolutional dense regression for accurate 3D human pose estimation[J]. IEEE Transactions on Image Processing, 2021(30):2830-2842.
[17] Li J, Chen L, Wei Y, et al. CrowdPose: Efficient crowded scenes pose estimation and a new benchmark[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[18] Du W, Wang X, Wang X, et al. PoseFix: Model-agnostic general human pose refinement network[C]. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques, 2020.
[19] Zhu L, Zhou C, Li S, et al. CTSegNet: A context-transformed segmentation network for brain tumor segmentation[C]. Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, 2019.
[20] Ji Xiaodong, Yang Qiaoning, Yang Xiuhui, et al. Human pose estimation: multi-stage network based on HRNet [J]. Journal of Physics: Conference Series, 2022,(1):2400.
[21] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015,37(9):1904-1916.
[22] Papandreou G, Zhu T, Kanazawa N, et al. Towards accurate multi-person pose estimation in the wild[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[23] Wei L, Zhang S, Dai J, et al. ST-GCN: Spatial temporal graph convolutional networks for skeleton-based action recognition[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[24] Sun M, He X, Yang S. U^2-Net (RE) for Human Pose Estimation[J]. arXiv preprint arXiv, 2021(2102):380.
[25] Zhou F, Zhu M, Bai J, et al. Deformable ConvNets v2: More deformable, better results[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[26] Carreira J, Agrawal P, Fragkiadaki K, et al. Associative embedding: End-to-end learning for joint detection and grouping[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[27] Papandreou G, Zhu T, Kanazawa N, et al. PersonLab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model[C]. Proceedings of the European Conference on Computer Vision, 2018.
[28] Kreiss S, Bertoni A, Alahi A. PifPaf: Composite fields for human pose estimation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.
[29] Insafutdinov M, Pishchulin L, Andres B, et al. DeepCut: Joint subset partition and labeling for multi person pose estimation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[30] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation[C]. Proceedings of the European Conference on Computer Vision (ECCV), 2016.
[31] Chen Y, Wang Z, Peng Y, et al. Cascaded pyramid network for multi-person pose estimation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[32] Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[33] Xiao B, Wu H, Wei Y. Simple baseline for human pose estimation and tracking[C]. Proceedings of the European Conference on Computer Vision (ECCV), 2018.
[34] Wei L, Zhang S, Yin W, et al. Convolutional pose machines[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.