Hindawi
Mobile Information Systems
Volume 2023, Article ID 3649217, 15 pages
https://doi.org/10.1155/2023/3649217

Research Article

A Novel Model for Intelligent Pull-Ups Test Based on Key Point Estimation of Human Body and Equipment

Guozhong Liu,[1] **Jian Wang**,[2] **ZhiBo Zhang**,[2] **Qingyi Liu**,[2] **Yande Ren**,[3] **Mengjiao Zhang**,[2] **Shan Chen**,[2] **and Peirui Bai**[2]

[1]*College of Physical Education, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[3]*Department of Radiology, The Affiliated Hospital of Qingdao University, Qingdao 266555, China*

Correspondence should be addressed to Peirui Bai; bprbjd@163.com

Received 27 September 2022; Revised 13 December 2022; Accepted 31 December 2022; Published 23 January 2023

Academic Editor: Mirco Marchetti

Applying computer vision and machine learning techniques into sport tests is an effective way to realize "intelligent sports." Facing practical application, we design a real-time and lightweight deep learning network to realize intelligent pull-ups test in this study. The main contributions are as follows: (1) a new self-produced pull-ups dataset is established under the requirement of including a human body and horizontal bar. In addition, a semiautomatic annotating software is developed to enhance annotation efficiency and increase labeling accuracy. (2) A novel lightweight deep network named PEPoseNet is designed to estimate and analyze a human pose in real time. The backbone of the network is made up of the heatmap network and key point network, which conduct human pose estimation based on the key points extracted from a human body and horizontal bar. The depth-wise separable convolution is adopted to speed up the training and convergence. (3) An evaluation criterion of intelligent pull-ups test is defined based on action quality assessment (AQA). The action quality of five states, i.e., ready or end, hang, pull, achieved, and resume in one pull-ups test cycle is automatically graded using a random forest classifier. A mobile application is developed to realize intelligent pull-ups test in real time. The performance of the proposed model and software is confirmed by verification and ablation experiments. The experimental results demonstrated that the proposed PEPoseNet has competitive performance to the state of the art. Its PCK@0.2 and frames per second (FPS) achieved were 83.8 and 30 fps, respectively. The mobile application has promising application prospects in pull-ups test under complex scenarios.

1. Introduction

Classical physical tests such as pull-ups test in university or middle and primary school are routine examinations in physical teaching. However, the current examination is usually conducted manually. The manual test always leads to low efficiency, inconsistent standards, and subjectivity. More recently, China and other countries pay increasing attention to ensure the fairness and objectivity of the sports tests. The vision-based AI technologies provide an efficient way to enhance the test efficiency and fairness. In addition, it will also reduce the work burden of physical education teachers.

With the development of next generation of information technology such as the Internet of Things (IoT), cloud computing, wearable devices, big data, and machine learning, "intelligent sports" has become a hot area attracting much attention of domain expert in the information and sports field [1, 2]. Automatic human pose estimation (HPE) is a common and critical task in an intelligent physical fitness test. The common practice is to analyze images or videos of the examinee's actions online or offline by employing computer vision and machine learning techniques.

More recently, deep learning is widely applied to many fields, e.g., predicting malfunctions of sensor and machinery [3, 4], or detecting intracranial aneurysms [5], owing to its powerful self-learning ability and adaptability of visual processing tasks. The deep learning networks are also introduced in the field of HPE [6]. Most of the HPE algorithms identify human body joints based on the capture of key point graph of the human body. We adopted this strategy to realize intelligent physical testing function in our first application software, i.e., the video stream captured by a camera was converted into a stream of a key point graph based on the traditional HPE algorithm. The size of a key point graph was normalized according to the distance from the nose to the hips. However, it was found that not all the captured key points were useful. So, several sets of key points (including wrists and shoulders, elbows and buttocks, left and right ankle, and left and right knee) were designated to reduce the data redundancy and computational cost. However, there still exists limitations of the traditional HPE algorithms when applying them to practical physical testing. For example, in the test of pull-ups or sit-ups that require equipment assistance, the negligence of key points of the auxiliary equipment may lead to misjudgment or cheating actions (If an examinee just stands on the ground and imitates the test actions, it is hard to discern the cheating actions in terms of vision-based technology).

Therefore, comprehensive utilization of key points of both human body and equipment is a promising strategy to improve the performance of HPE. Inspired by this idea, we design a lightweight deep learning network and a mobile application to estimate and analyze human poses in a pull-ups test. The main contributions are as follows: (1) We establish a benchmark dataset containing more than 2,000 images for pull-ups test and develop a semiautomatic annotating software for labeling key points of human body and equipment. (2) A novel deep learning network named PEPoseNet is designed to jointly estimate the key points of human body and equipment. The network adopted depthwise separable convolution (double encoder-decoder) to improve estimation accuracy and speeded up the training by pretraining and freezing the gradient backpropagation of the heatmap branch. (3) An AQA algorithm for key point estimation of human body and equipment generated by PEPoseNet is designed, and an intelligent pull-ups test mobile application is developed. The application can realize real-time assessment of five states, i.e., ready or end, hang, pull, achieved and resume in one pull-ups test cycle, and rate each cycle and total movement.

The remainder of the paper is organized as follows. The related works of HPE in sports are reviewed in Section 2. The proposed model, i.e., PEPoseNet and related dataset are described in Section 3. Section 4 presents the experiments and results, as well as comparisons of the state of the art methods. The concluding remarks are drawn in Section 5.

2. The Related Work

Human pose estimation (HPE) refers to determine or judge human body posture by processing and analyzing images or videos. Currently, HPE has wide applications in many fields such as virtual reality (VR) [7], human health [8, 9], motion capture systems [10], and human computer interaction (HCI) [11]. Originally, the traditional machine learning methods were employed to estimate human posture. For example, Eichner et al. applied conditional random fields (CRF) to learn potential relationships between appearance of different body parts and annotated images [12]. Shakhnarovich et al. utilized a parameter-sensitive hashing function to estimate the joints of human body. However, since the first popular CNN model, i.e., AlexNet emerged, the deep learning methods displayed abrupt developments in HPE, owing to their powerful self-learning ability and remarkable performance [13, 14]. For instance, Newell et al. proposed the classical stacked hourglass networks architecture which provided inspiration for many subsequent works [15]. Cao et al. proposed a convolutional pose machine to find the position of each joint and adopted a part affinity field to assemble the joints [16]. Bazarevsky et al. proposed BlazePose which estimates human pose by means of a set of code with high FPS [17].

Based on the research of HPE, some studies have begun to pay attention to the model of action quality assessment (AQA) [18]. The AQA task aims to design a system that can automatically and objectively evaluate some specific human actions through video or images. AQA is currently being developed in many practical application scenarios, such as surgical skill rating, medical rehabilitation test, athlete posture correction, coaching system, operation compliance analysis, and dangerous action monitoring. The evaluation module can be classified into three types, i.e., regression scoring, grading, and pairwise sorting. In this study, following the human pose estimate, we adopt grading to assess action quality of the five states in one pull-ups test cycle.

In sports field, many actions are often different from daily movements. It is usually difficult to track complicated movements and high speed actions, just like the explosive actions in fencing and challenging body postures in yoga. In addition, the occlusion and interference of sport equipment also raise difficulty in localizing the targets. Many efforts have been made to improve the performance of HPE and AQA in sports. Zecha et al. proposed a method of posture correction for underwater training [19]. Neher et al. improved a stacked hourglass network to predict the attitude of hockey players and stick at the same time [20]. Trejo and Yuan developed an interactive system which adopted Adaboost to perceive several postures of learning yoga and provided users with the function of posture correction [21]. Promrit and Waijanya proposed video posture embedding by adopting the triplet-loss technique and applied one-shot learning to detect a badminton player's posture [22]. Suda et al. presented a method that predicts the ball trajectory of a volleyball toss 0.3 s before the actual toss by observing the motion of setter player [23]. Xu et al. proposed self-attentive LSTM and multiscale convolutional skip LSTM to predict total element score (TES) and total program component score (PCS) in figure skating [24]. Xiang et al. [25] divided the diving process into four stages: beginning, jumping, dropping, and entering into the water and adopted four independent P3D models [26] to complete feature extraction.

For the pull-ups test, the horizontal bar is needed just like the equipment to fix one's feet in sit-ups. However, the influence of auxiliary equipment is often ignored in posture estimation. Therefore, detecting and localizing the key points of equipment may provide complementary information for HPE and AQA. In this work, we train a deep learning network by feeding the key points of human body and equipment. Then, a grading assessment of action quality is carried out using a random forest classifier. The well-trained network is ported to embedded platforms for confirming its practicality. The intelligent pull-ups test can be carried out with satisfactory performance.

3. The Proposed Method

The workflow of realizing intelligent pull-ups test based on PEPoseNet is shown in Figure 1. There are three modules that are marked using dotted boxes, i.e., dataset module, PEPoseNet module, and assessment module. The dataset module completes data collection and labeling. The PEPoseNet module trains and tests samples with a lightweight network architecture. The assessment module is responsible for actions quality assessment of pull-ups test.

3.1. Data Collection and Annotation. In the field of sport and physical exercise (SPE), there exist several popular data sets, e.g., Leeds Sports Pose (LSP) [27], Frames Labeled In Cinema (FLIC) [28], and Penn Action [29]. However, to the best of our knowledge, there are no available public pull-ups datasets till now. Therefore, we need to establish a self-produced pull-ups dataset for research.

The self-produced pull-ups dataset is named SDUST-PUT, which includes 263 images extracted from online videos and 1,737 images taken from volunteers. The images should contain a subject and horizontal bar at the same time. Different postures were considered, e.g., standing under the horizontal bar, preparing to jump, and various stages of doing pull-ups.

Figure 2 demonstrates four example images of different states in SDUST-PUT. Figures 2(a) and 2(b) are taken from volunteers and represent ready or end and hang, respectively, while Figures 2(c) and 2(d) are extracted from online videos and represent pull or resume and achieved, respectively. To annotate the images efficiently, we develop a piece of software running on a Windows or Mac system based on the flutter framework [30]. The annotator can label human joints and key points of equipment in a semiautomatic style. At first, the OpenPifPaf algorithm [31] is employed to label human joints automatically. Then, the users only need to annotate a small number of key points of the equipment and correct a smaller number of inaccurate points annotated by OpenPifPaf. The efficiency of data annotating enhances greatly using the developed software. The annotating results are saved as .json format. In order to avoid inaccurate annotation for compressed images, the annotator records the distance ratio instead of direct distance.

The distance ratio of the key point and the left border is recorded as the value of horizontal axis, and the distance ratio of the key point and the upper border is recorded as the value of vertical axis. The annotator is publicly available for download at https://github.com/PEPoseNet/PEPoseNet. The annotator can also be used to label similar datasets related to human posture estimation. Figure 3(a) illustrates the running interface of the annotator, while Figures 3(b) and 3(c) illustrate two example labeling results.

3.2. The Proposed PEPoseNet. Figure 4 illustrates the overall architecture of the proposed PEPoseNet, along with the structure of different blocks. As shown in Figure 4(a), the backbone is inspired by Google's BlazePose [17]. It consists of two hierarchical networks, i.e., the heatmap network and the key point network. The backbone adopts three types of convolution layer structure, i.e., Block 1, Block 2, and Block 3 as illustrated in Figures 4(b)–4(d), respectively. The three blocks are filled with different colors for easy discrimination. The Block 1 combines simple depth-wise separable convolution and regular convolution as shown in Figure 4(b). Except for the similar depth-wise separable convolution and regular convolution layers, the Block 2 in Figure 4(c) adopts a Maxpool layer to hierarchically reduce image scale, whereas the Block 3 in Figure 4(d) adopts an upsampling layer to hierarchically increase image scale. The design of these blocks is to facilitate them running on mobile platforms or embedded devices.

3.2.1. The Heatmap Network and the Key Point Network. The basic structure of the heatmap network is illustrated at right side in Figure 4(a), which is similar to the stacked hourglass networks proposed by Newell et al. [15]. The encoder receives original image with size of 512×512. A series of depth-wise separable convolutions followed by the maximum pool layer are carried out in sequence. In the implementation procedure, the number of channels increases step by step to extract potential information with different scales. Then, the encoder output $8 \times 8 \times 288$ heatmaps. In the connection of encoder and decoder, the residual structure [35] is adopted to reduce information loss. The decoder adopts multilayer upsampling operation to continuously increase the size of heatmap. The depth-wise separable convolution is utilized to further decode information at different scales.

In the training procedure, how to capture the key points of the original images effectively is an important issue. Here, we employ a 2D Gaussian kernel in the loss function of the heatmap network to extract rough center of key points as close as possible. The loss function of the heatmap network is as follows:

$$l_{HMN} = C_P \left\| M^P \odot \left(P - P^*\right) \right\|_2^2 + C_E \left\| M^E \odot \left(E - E^*\right) \right\|_2^2, \quad (1)$$

where P and E represent the predicted heatmap of human joints and key points of equipment, respectively, P^* and E^* represent the corresponding ground truths. M^P and M^E represent the mask of human joints and equipment key points, respectively, which are utilized to assign different weights to positive and negative samples. That is, the weight of positive samples is 1, while negative samples is 0.1. The symbol $\odot$ denotes element-wise product. C^P and C^E

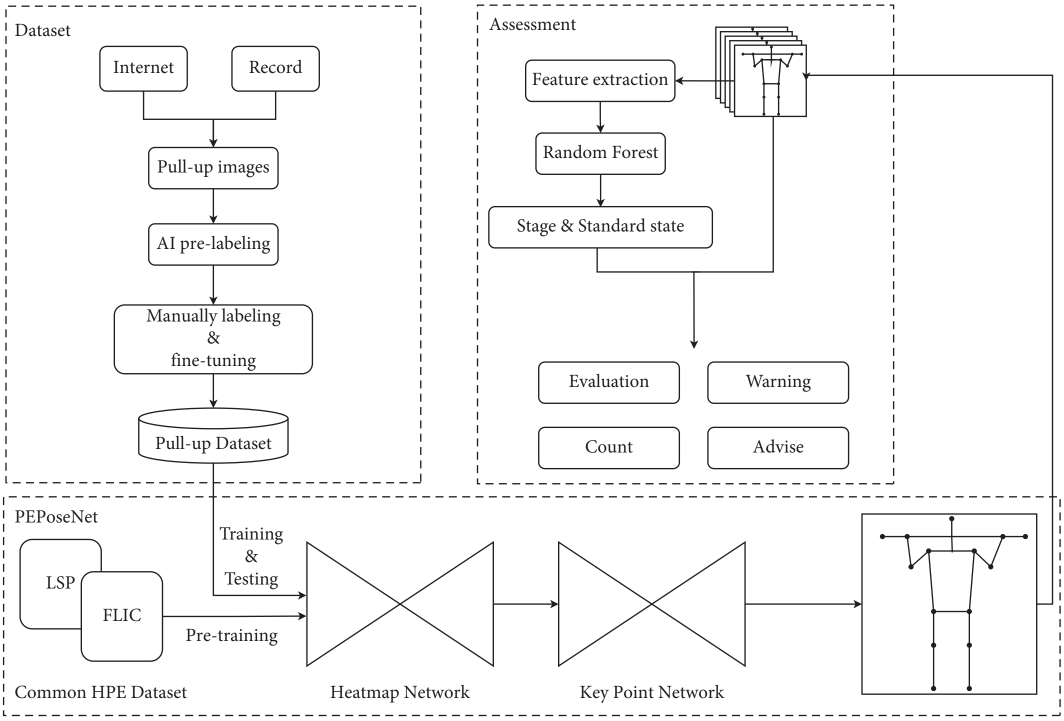


Figure 1: The workflow of intelligent pull-ups test based on PEPoseNet.

represent the training weight of human joints and equipment joints, respectively.

As shown in Figure 5, the center of each heatmap is searched in terms of the label data (coordinates of the key point), and a heatmap is formed by setting the nearest pixel to be closer to 1, while the pixels away from the center are set to 0. In such a way, each original image produced 15 heatmaps with size of 128 × 128 that are corresponding to 15 key points.

The heatmap network output abundant information of key points that will feed to the key point network for accurate localizing key points. From Figure 4(a), we can see that the front layers of the heatmap network are connected to the decoder of a key point network. Two specific modifications are made in the construction of the key point network. First, we exploit intermediate data of the heatmap decoder instead of the final output. It is observed that a large amount of effective information exists in the intermediate data, rather than in the final convolution layer. Second, only forward propagations are retained in the training procedure of the key point network (denoted as dotted arrows in Figure 4(a)), the gradient backpropagations between the key point decoding and the output heatmaps are frozen. This modification can effectively avoid affecting the generation of heatmap in the training of a key point decoder. The loss function of the key point network is divided into two parts, i.e., classification loss and regression loss. It is defined as follows:

$$l_{KPN} = l_{HMN} + \lambda l_{REG}, \tag{2}$$

where l_{HMN} represents the classification loss which is same as the loss function of the heatmap network. l_{REG} represents the regression loss function which indicates the position difference between the predicted point and the label point. λ is a constant for balancing the two kinds of losses. It is set to 0.05 in this work.

The regression loss function is defined as follows:

$$l_{REG} = \sum_{i\in G} \frac{1}{z_i} \mathrm{Smooth}_{L1}\left(O_i - O_i^{\ *}\right), \tag{3}$$

where O and O^* represent the predicted position and the corresponding label of key points, respectively. G is a set of label points type. Z represents the area of human body, which can be estimated as follows:

$$Z = \sqrt{\Delta x^2 + \Delta y^2}, \tag{4}$$

where Δx and Δy are the maximum distance between the horizontal and vertical coordinates in the real label, respectively. Smooth_{L1} is a thresholding function defined as follows:

$$\mathrm{Smooth}_{L_1} = \begin{cases} 0.5x^2, & |x| < 1, \\ |x| - 0.5, & \text{Otherwise.} \end{cases} \tag{5}$$

Figure 6 illustrates a demo result of the heatmap network and the key point network. In Figure 6(a), the original image and the 15 predicted heatmaps are presented. It can be seen that the centers of the heatmaps are very close to the real positions of human body joints and key points of horizontal bar. In Figure 6(b), the refined key points are output by the key point network. It is obvious that the two key points of horizontal bar can provide a good positioning reference for pull-ups test.

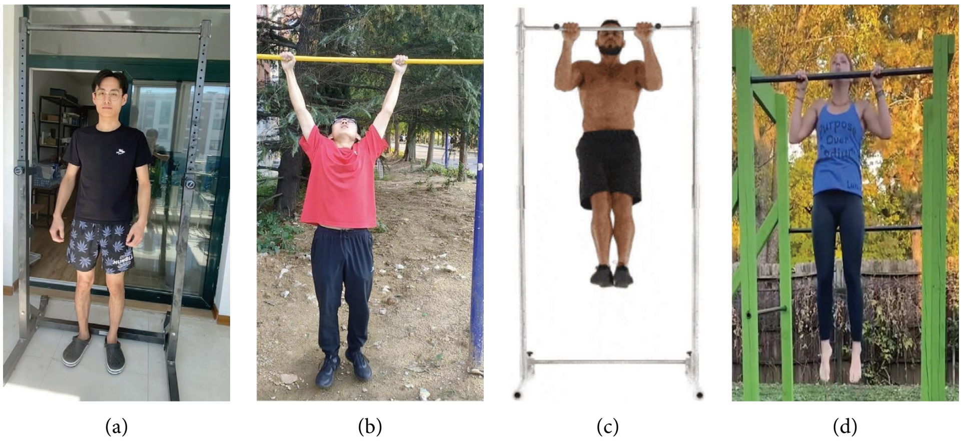
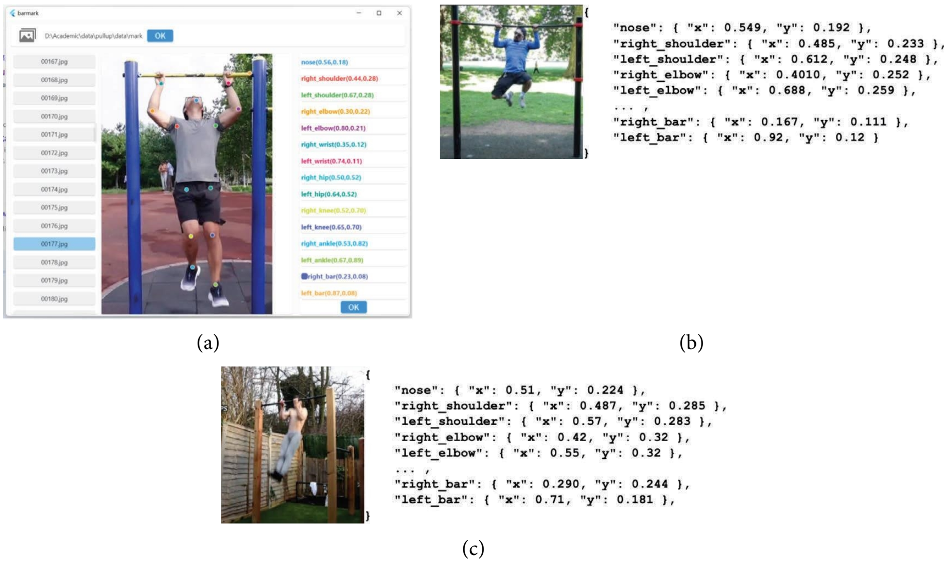


Figure 2: Example of pull-ups images at different states in SDUST-PUT. (a) Ready or end and (b) hang are taken from volunteers; (c) pull/resume; (d) achieved are extracted from online videos.



```
nose(0.56,0.18)
right_shoulder(0.44,0.28)
left_shoulder(0.67,0.28)
right_elbow(0.30,0.22)
left_elbow(0.80,0.21)
right_wrist(0.35,0.12)
left_wrist(0.74,0.11)
right_hip(0.50,0.52)
left_hip(0.64,0.52)
right_knee(0.52,0.70)
left_knee(0.65,0.70)
right_ankle(0.53,0.82)
left_ankle(0.67,0.89)
right_bar(0.23,0.08)
left_bar(0.87,0.08)
```

```
{
    "nose": { "x": 0.549, "y": 0.192 },
    "right_shoulder": { "x": 0.485, "y": 0.233 },
    "left_shoulder": { "x": 0.612, "y": 0.248 },
    "right_elbow": { "x": 0.4010, "y": 0.252 },
    "left_elbow": { "x": 0.688, "y": 0.259 },
    ... ,
    "right_bar": { "x": 0.167, "y": 0.111 },
    "left_bar": { "x": 0.92, "y": 0.12 }
}
```

```
{
    "nose": { "x": 0.51, "y": 0.224 },
    "right_shoulder": { "x": 0.487, "y": 0.285 },
    "left_shoulder": { "x": 0.57, "y": 0.283 },
    "right_elbow": { "x": 0.42, "y": 0.32 },
    "left_elbow": { "x": 0.55, "y": 0.32 },
    ... ,
    "right_bar": { "x": 0.290, "y": 0.244 },
    "left_bar": { "x": 0.71, "y": 0.181 },
}
```

Figure 3: (a) The running interface of the annotator. The label symbols are small circle filled with different colors. (b) Labeling results of a straight front shot. (c) Labeling results of an oblique front shot.

3.2.2. The Pretrained Network. In the training procedure of the deep learning network, overfitting will occur if the amount of data is small. There are two solutions to overcome this problem, i.e., data augmentation and transfer learning based on the pretrained network. In this work, we are prone to adopt pretrained scheme as the LSP [27] and FLIC [28] dataset and are suitable for pretraining the network. Since the LSP and FLIC dataset did not require containing a horizontal bar, the images with consistent labels of SDUST-PUT are screened out for pretraining the deep network. In the pretraining procedure, we first train the heatmap network. Then, the key point network is trained by fixing the parameters of the heatmap network. The pretraining is satisfactory because percentage of correct key-points (PCK@ 0.2) can achieve 85.1 after 200 epochs. Therefore, the parameters of pretraining on the LSP and FLIC dataset are adopted as initialization parameters when training the PEPoseNet on SDUST-PUT. It is noted that the parameters of the heatmap network output layer and key point network output layer should be replaced by random values. Benefiting from the transfer learning, the training efficiency and generalization ability of the PEPoseNet improved obviously.

3.3. Action Quality Assessment of Pull-Ups Test. The pull-ups test is an important physical test item in many fields such as in school and troops, which consists of a series of complex actions. It is a challenging task to realize intelligent pull-ups test based on the analysis of images or videos. Two visual processing tasks should be conducted in intelligent pull-ups test, i.e., human pose estimation (HPE) and action quality assessment (AQA). The aforementioned PEPoseNet is capable of conducting human

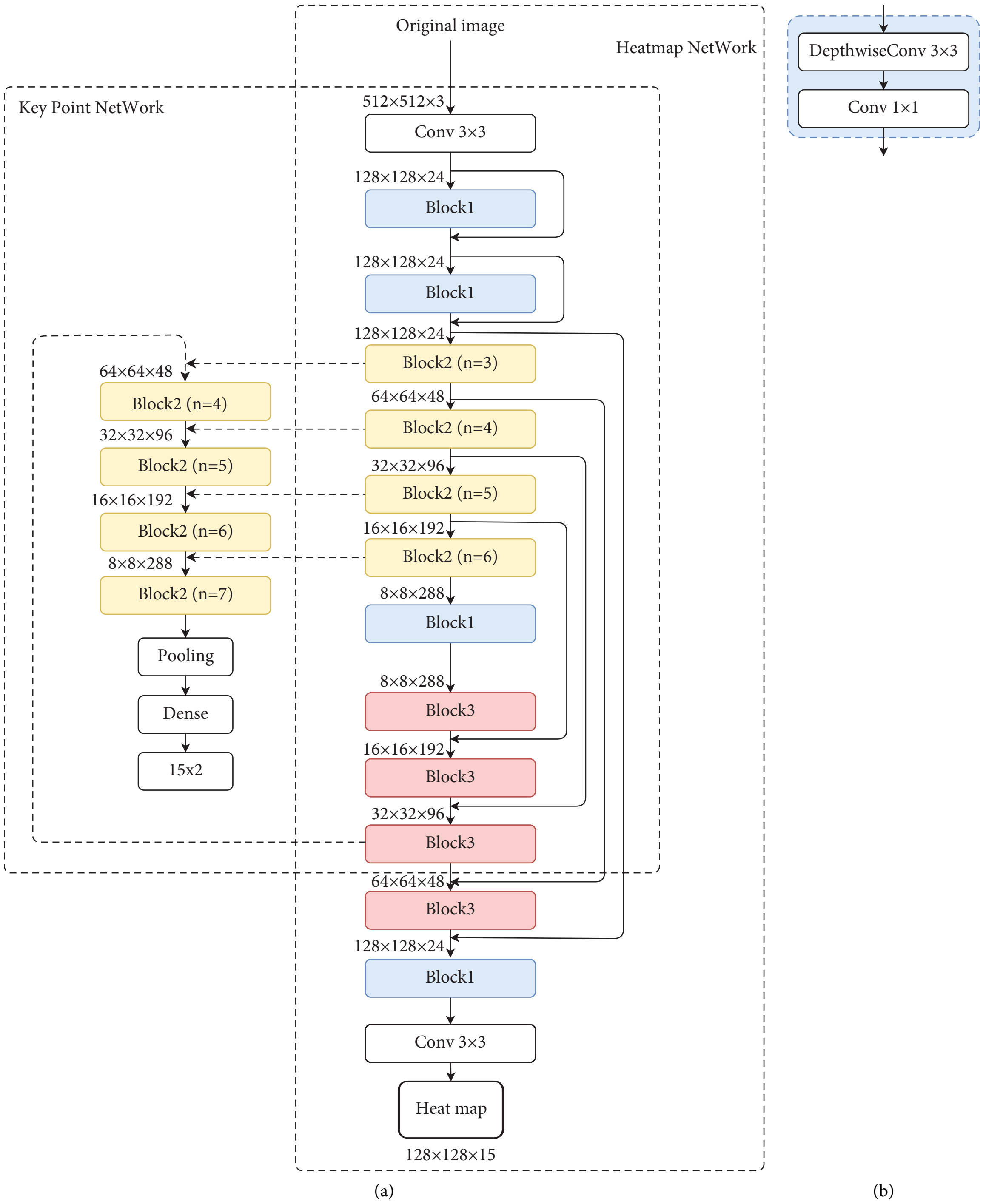


Figure 4: Continued.

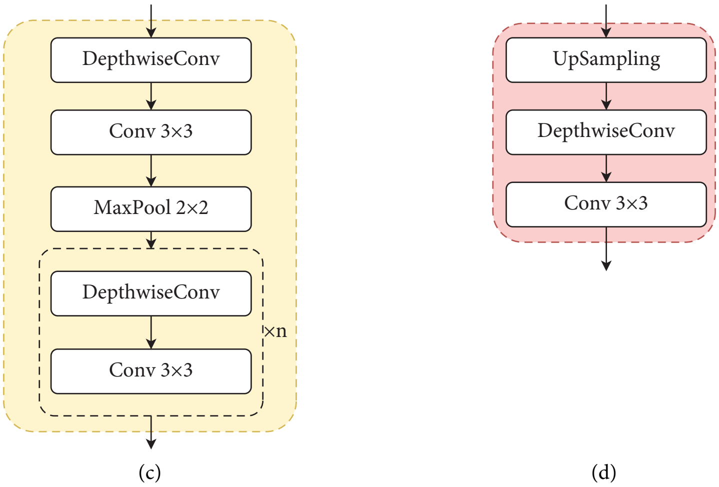


Figure 4: The architecture of the proposed PEPoseNet. (a) The backbone being made of two functional network, i.e., heatmap network and key point network. (b) The structure of block 1, (c) the structure of block 2, and (d) the structure of block 3. The channels denoted by dotted arrow refer to that they have forward propagation only and have no gradient backward propagation. The "Conv" refers to standard convolution layer, while the depthwiseConv refers to depth-wise separable convolution [32–34].

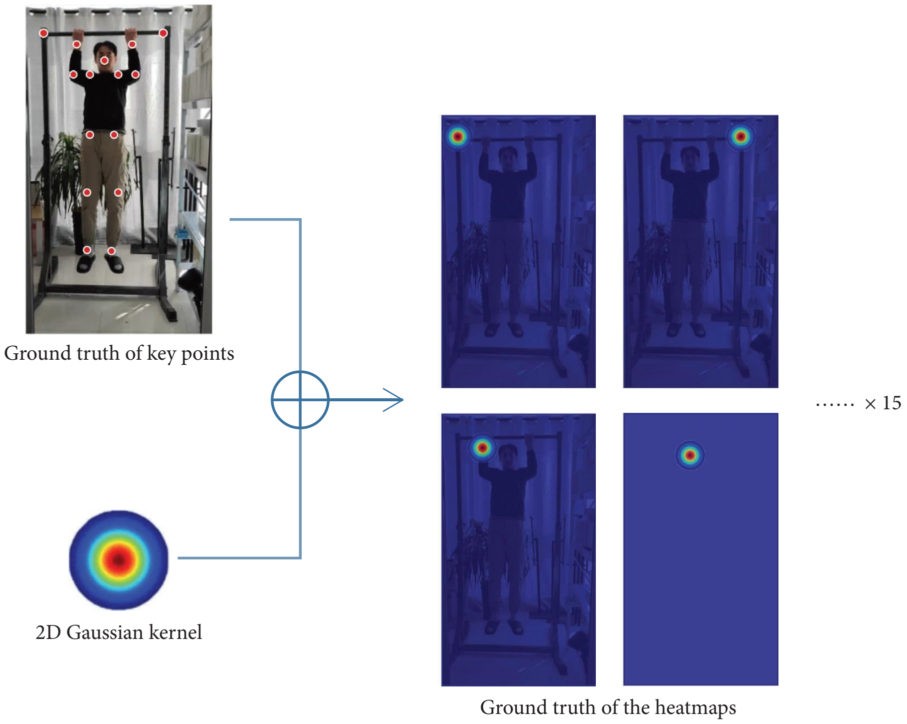


Figure 5: The mechanism of producing ground truth of the heatmaps by employing a 2D Gaussian kernel.

pose estimation. However, it still has no reports about automatic action quality assessment for pull-ups test. In this study, we present a complete process for intelligent evaluation scheme of pull-ups test. First, we divide the movements in one pull-ups cycle into five states, i.e., ready or end, hang, pull, achieved, and resume, as listed in Table 1. The division is presented by experienced teachers who have been occupied in physical education and pull-ups test for more than 20 years.

Figure 7 illustrates the five states and the sequence relations. It can be seen that the ready or end state represents the start and stop of a set of pull-ups. The hang state refers to the body is hanging from the horizontal bar (arms fully extend is required). The pull state refers to lift one's body with his or her arms. The attained state refers to keep the head above the horizontal bar. The resume state refers to relax one's arms and return to the hang state. In one pull-ups cycle, it is required that no obvious bending and swinging of the body or legs.

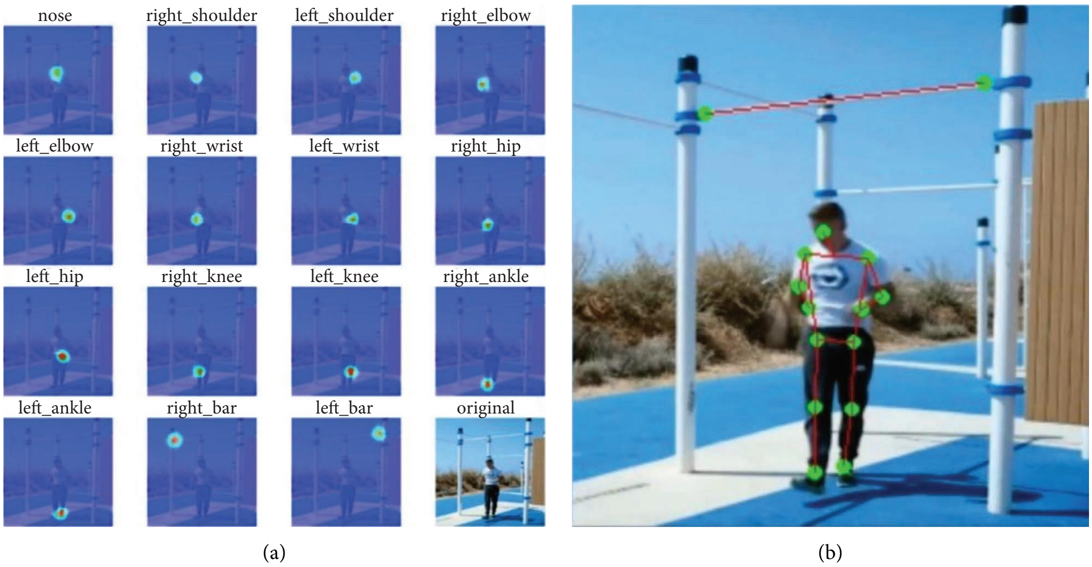


Figure 6: Demo heatmaps and key points extracted from one subject using the heatmap network and the key point network, respectively. (a) The original image and 15 predicted heatmap nodes and (b) the extracted key points of human joints and horizontal bar.

Second, we design a grading assessment solution for each state in one pull-ups cycle. As show in Figure 8, the standard action and nonstandard action of each state are illustrated. In order to make automatic grading, we adopt a random forest classifier in the assessment module. 21 videos (containing 8718 frames) are collected from volunteers for training the classifier. In order to ensure the robustness of action evaluation, the distances and angles of all key points in $n-4$th, $n-2$th, nth, $n+2$th, and $n+4$th frames are considered, as shown in Figure 9. The angle refers to the one between the horizontal direction and line connected by the two key points. The other four frames are selected to obtain more obvious features in time dimensionality. In addition, several angles between the lines with obvious changes were also selected as features. For each frame, there are 2,270 features that can be used for making the assessment. The coordinate values of each set of the key points are divided by the distance between two hip joints to normalize the data. The spatial-temporal features of the key points of human body and equipment, output from PEPoseNet, are fed to the classifier to obtain the state of the nth frame.

In practical application, the state streams coming out of the random forest are filtered by a mode filter. Then, the software counts the number of pull-ups using the cycle of states, and grades each cycle using action evaluation. Figure 10 illustrates the automatic scoring scheme of practical pull-ups test. We assume a complete pull-ups test has N cycles, and each cycle has M frame. Then, the total score of this pull-ups test can be calculated as follows: (1) Calculating the cumulative scores in each cycle. For each frame in one cycle, if the action is standard, the grading value 1 is assigned. Otherwise, the grading value is assigned to 0.5. Then, the score of each cycle is obtained by cumulatively summing the score of each frame divided by the number of frame (M). (2) Calculating the cumulative scores in one test. That is, the score of each cycle is summed directly to obtain the total score.

4. Experiments and Results

4.1. Dataset and Evaluation Metrics. Three datasets including the self-produced SDUST-PUT and two public dataset, i.e., LSP [27] and FLIC [28] were used to train and generalize the proposed PEPoseNet. The LSP dataset contains 2,000 images of a single player who are doing actions in badminton, baseball, and gymnastics. The FLIC dataset contains more than 5,000 tagged frame images extracted from the movies. Among these datasets, the .png images that have same label points as the images in the SDUST-PUT dataset are selected and scaled to 512 × 512. These images are utilized for pretraining the deep network. The SDUST-PUT dataset contains 2,000 pull-ups images that are required to include human body and horizontal bar at the same time. The ratio of training set and test set is 7 : 3.

The accuracy of key point detection is measured by percentage of correct key-points (PCK), which refers to the percentage of detections that fall within a normalized distance of the ground truth [28]. The PCK is defined as follows:

$$PCK@T = \frac{\sum_i \delta\left(d_i/d \le T\right)}{\sum_i 1}, \tag{6}$$

where i represents the number of the joint points, d_i represents Euclidean distance between the ith predicted point and its ground truth, and d represents the normalization scale factor. In this work, we adopt the Euclidean distance between left shoulder and right hip. T denotes the threshold value ($T = 0.2$ in this work).

In order to verify the practicability of the model and software proposed in this study, 21 pull-ups videos collected from volunteers were utilized to test the PEPoseNet and the traditional HPE algorithms. The key points from the PEPoseNet and the traditional HPE algorithms are extracted, respectively. Then, action quality assessment is carried out by implementing the random forest classifier.

Table 1: The five states in one pull-ups cycle and their assessment criteria.

States	Assessment criteria
Ready or end	Stand below the horizontal bar
Hang	Body hanging from the horizontal bar, arms are required stretching completely
Pull	Lift one's body with his/her arms
Attained	Keep the head above the horizontal bar
Resume	Relax one's arms and return to the hang state

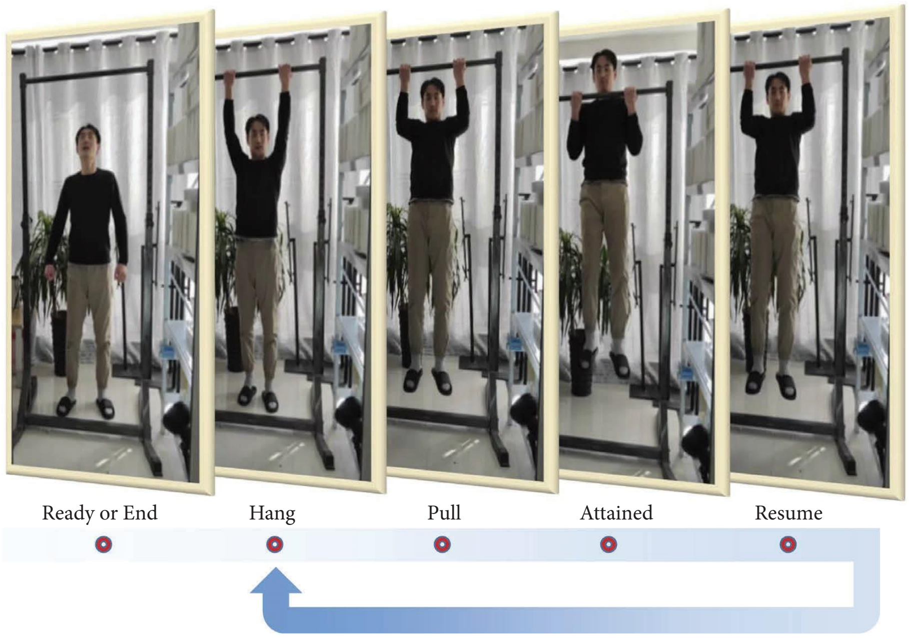


Figure 7: Illustration of the five states in one pull-ups cycle along with the sequence order.

Four quantitative metrics, i.e., accuracy, precision, recall, and F_1 are adopted to evaluate classification performance of PEPoseNet and traditional HPE algorithms. The definitions are as follows:

$$\begin{aligned}\text{Accuracy}(s) &= \frac{TP_s + TN_s}{TP_s + FN_s + FP_s + TN_s},\\ \text{Precision}(s) &= \frac{TP_s}{TP_s + FP_s},\\ \text{Recall}(s) &= \frac{TP_s}{TP_s + FN_s},\\ F_1(s) &= \frac{2 \times \text{Precision}(s) \times \text{Recall}(s)}{\text{Precision}(s) + \text{Recall}(s)},\end{aligned} \tag{7}$$

where the subscript s stands for a state or an action. TP_s represents the number of correctly classified s frame. TN_s represents the number of correctly classified non-s frame. FP_s represents the number of wrongly classified s frame, and FN_s represents the number of wrongly classified non-s frame.

4.2. Implementation Details. In the training of the PEPoseNet, TensorFlow 2.0 python library [36] was called. The input color images were resized to $512 \times 512 \times 3$. The output was coordinates of 15 key points. Adam optimizer [37] was employed to speed up the training. The learning rate was 0.001. The initial weights adopted the results of the pre-trained model training on the LSP and FLIC datasets. 200 epochs were implemented on Tesla P100 16G Nvidia GPU. Considering the limited computing power of the embedded or mobile platform, we also tested the performance on AMD Ryzen 7 3700X CPU without GPU.

To the best of our knowledge, there are no related pull-ups test deep networks to make comparison. Thus, we compare with two latest OpenPifPaf [31] and MediaPipe [38] (optimized by BlazePose) as they also carry out human posture estimations. The model parameters are provided by their official webs.

4.3. Ablation Experiment Schemes. We design four ablation experiments to evaluate the effects of key modules of the proposed method. The PEPoseNet-A architecture is designed to evaluate the effect of the heatmap network, i.e., directly input the heatmap output to the key point network or input the intermediate layer information of the heatmap network instead. The PEPoseNet-B architecture is designed to evaluate the stability of the output of the heatmap network. In the baseline architecture of PEPoseNet, the heatmap network is trained independently. Then, the key point network is trained

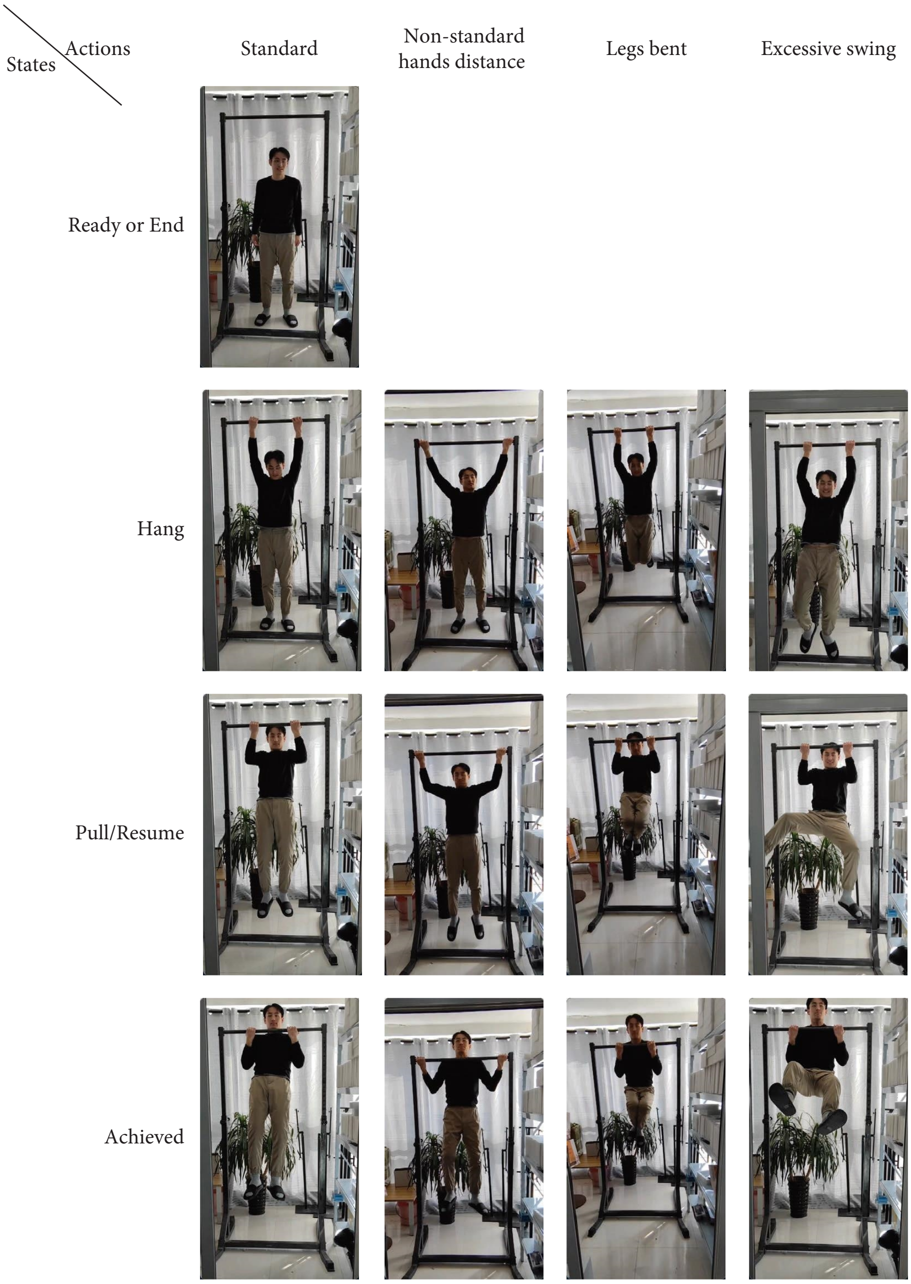


Figure 8: Illustration of standard and nonstandard actions in each state.

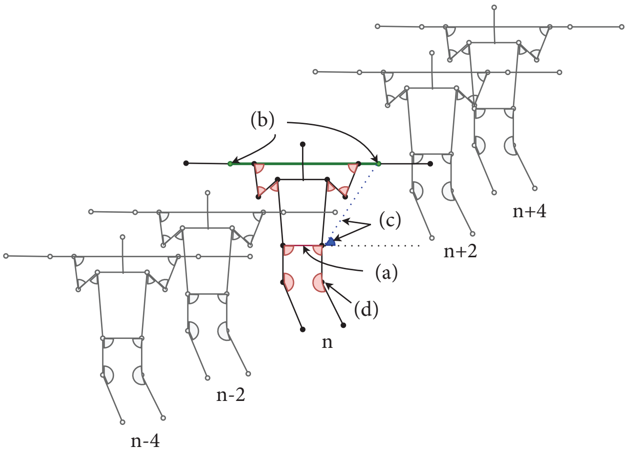


Figure 9: The schematic diagram of selected features in action quality assessment of pull-ups test. (a) The distance between two hip joints is employed as a standard, (b) additional bar key points, (c) the distance and angle between two key points, and (d) auxiliary reference angle.

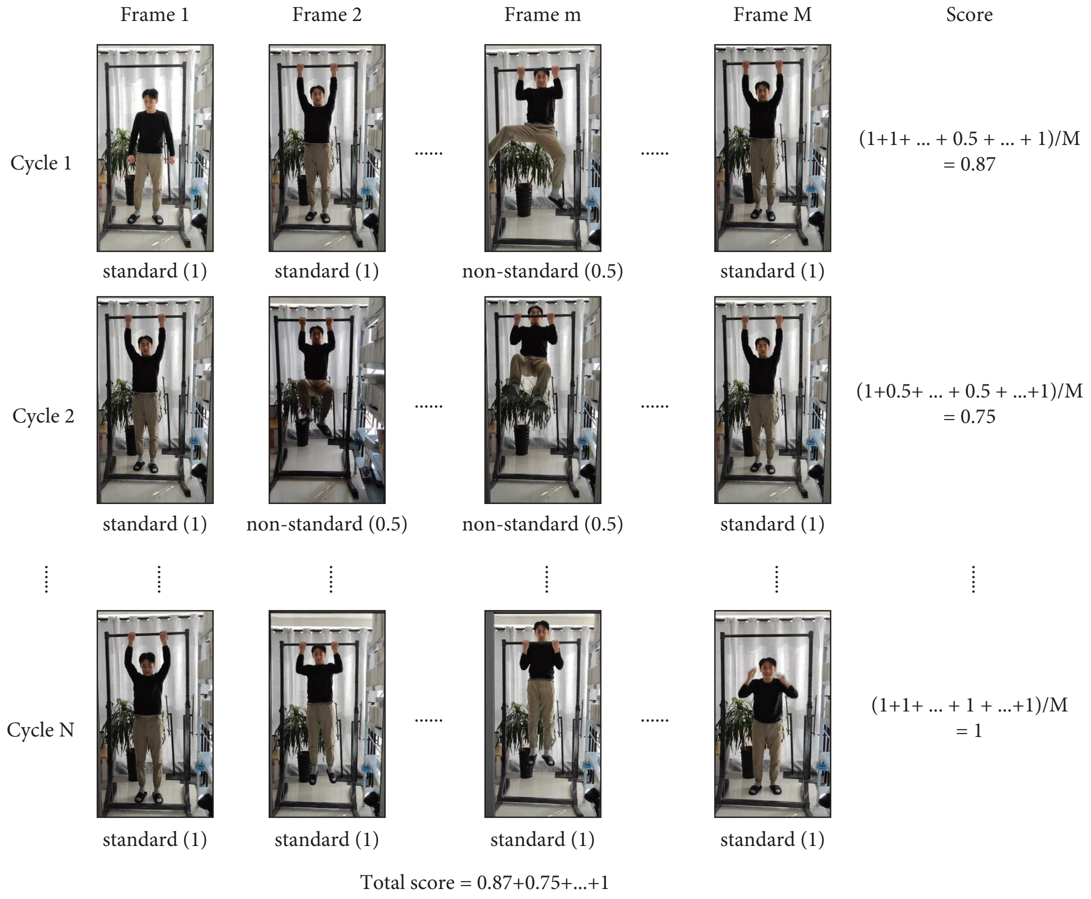


Figure 10: The automatic scoring scheme of practical pull-ups test. For each cycle, the grading value 1 or 0.5 is assigned to each frame in terms of whether the action is standard. The score of standard action is 1 and that of nonstandard is 0.5. The total score is obtained by cumulatively summing the cumulative scores of each cycle.

based on the trained heatmaps in the condition of freezing the channel of heatmaps. The PEPoseNet-B architecture removes the freezing of the heatmap network and enables it to be adjusted in the training of the key point network. The PEPoseNet-C architecture is designed to evaluate the role of pretraining. As the ground truth of OpenPifPaf or MediaPipe is not consistent with the final output of the PEPoseNet, it is uncertain to determine whether pretraining brings positive or negative effects. Therefore, the PEPoseNet-C architecture is trained only on the SDUST-PUT dataset without adopting pretraining. The PEPoseNet-D architecture is designed to evaluate the effect of the two key points of the bar. That is, in the baseline of the PEPoseNet, the two key points of the bar are considered and attend to make decision in the following AQA algorithm, while the PEPoseNet-D architecture removes the two key points. Thus, it can be determined the role of the two key points of the bar by comparing the results of the PEPoseNet-D and baseline PEPoseNet.

Table 2: The estimation accuracy of 15 key points using the PEPoseNet on SDUST-PUT.

Key points	PCK@0.2
Nose	85.2
Right shoulder	84.2
Left shoulder	83.9
Right elbow	84.1
Left elbow	84.0
Right wrist	83.6
Left wrist	83.8
Right hip	82.0
Left hip	82.6
Right knee	84.2
Left knee	84.1
Right ankle	83.5
Left ankle	83.8
Right bar	81.0
Left bar	80.2

4.4. Experiment Results

4.4.1. The Performance of the Baseline PEPoseNet. Table 2 lists the estimation accuracy of 15 key points. The right bar and left bar refer to two sides of the horizontal bar in pull-ups test. The other 13 key points refer to the critical positions of human body for posture estimation. It can be seen that all the PCK values are larger than 80. It reflects that the key points can be captured accurately and efficiently by introducing the cascaded operations of the heatmap and key

Table 3: Average PCK and FPS of different models.

Models/methods	PCK@0.2	FPS
OpenPifPaf	**88.7**	0.4
MediaPipe	84.2	27
PEPoseNet-baseline	83.8	**32**
PEPoseNet-A	58.4	27
PEPoseNet-B	80.1	31
PEPoseNet-C	76.1	31

The values in bold represent the best data for this metrics.

Table 4: Comparison of state classification conducted by PEPoseNet and MediaPipe in pull-ups test.

Metrics	Models	*States*				
		Ready or end	Hang	Pull	Achieved	Resume
Accuracy	PEPoseNet	**0.998**	**0.992**	**0.992**	**0.994**	**0.991**
	PEPoseNet-D	0.952	0.988	0.986	0.985	0.983
	MediaPipe	0.951	0.986	0.985	0.985	0.982
Precision	PEPoseNet	**0.995**	**0.976**	**0.989**	**0.968**	**0.984**
	PEPoseNet-D	0.875	0.954	0.973	0.901	0.957
	MediaPipe	0.871	0.950	0.971	0.904	0.953
Recall	PEPoseNet	**0.990**	**0.978**	**0.988**	**0.979**	**0.981**
	PEPoseNet-D	0.721	0.976	0.987	0.972	0.974
	MediaPipe	0.709	0.974	0.985	0.969	0.974
F_1 score	PEPoseNet	**0.993**	**0.977**	**0.988**	**0.973**	**0.982**
	PEPoseNet-D	0.791	0.965	0.980	0.935	0.966
	MediaPipe	0.782	0.962	0.978	0.935	0.963

The values in bold represent the best data for this metrics.

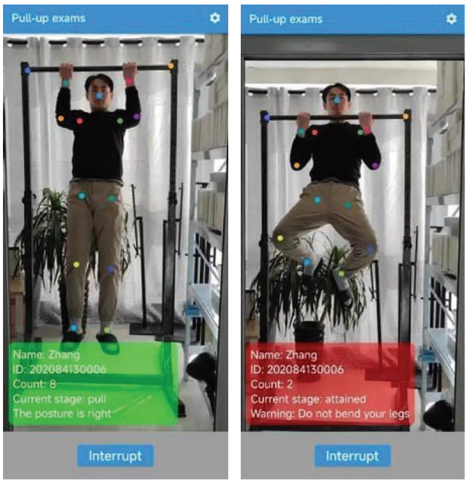
Table 5: Comparison of action classification conducted by PEPoseNet and MediaPipe in pull-ups test.

Metrics	Models	Standard actions	*Nonstandard action*		
			Legs bent	Excessive swing	Nonstandard hands distance
Accuracy	PEPoseNet	**0.980**	**0.990**	**0.993**	**0.994**
	PEPoseNet-D	0.981	0.991	0.984	0.990
	MediaPipe	0.980	0.983	0.990	0.991
Precision	PEPoseNet	0.965	**0.990**	**0.988**	**0.980**
	PEPoseNet-D	0.974	0.969	0.969	0.978
	MediaPipe	**0.974**	0.967	0.977	0.968
Recall	PEPoseNet	**0.982**	**0.972**	**0.978**	**0.982**
	PEPoseNet-D	0.973	0.974	0.968	0.977
	MediaPipe	0.972	0.966	0.976	0.974
F_1 score	PEPoseNet	**0.974**	**0.981**	**0.983**	**0.981**
	PEPoseNet-D	0.974	0.971	0.968	0.978
	MediaPipe	0.973	0.967	0.976	0.971

The values in bold represent this best data for this metrics.

point network and depth-wise separable convolution. The FPS of the PEPoseNet achieved 32. The results indicated that the test accuracy and speed of the proposed model is acceptable for practical application.

4.4.2. The Results of the Comparative and Ablation Experiments. Table 3 lists the comparison results in terms of PCK and frame per second (FPS). It can be seen that the PCK of OpenPifPaf achieved 88.7, but its FPS was only 0.4. It means that the computational cost of OpenPifPaf is expensive that will limit its transplant to mobile or embedded devices. The PCK of MediaPipe was 84.2 that was slightly higher than 83.8 of the PEPoseNet. However, its FPS 27 was smaller than 32 of the PEPoseNet. The slightly higher PCK of MediaPipe may become from the larger training set and optimization tricks supported by Google engineers. In contrast, the PEPoseNet achieved the best FPS owing to the depth-wise separable convolution. In practice, high FPS is the most attractive characteristics for transplanting the



Name: Zhang
ID: 202084130006
Count: 8
Current stage: pull
The posture is right

Name: Zhang
ID: 202084130006
Count: 2
Current stage: attained
Warning: Do not bend your legs

Figure 11: The interface of pull-ups test application in practical test.

model into mobile or embedded devices. In addition, the PEPoseNet has the advantage of being able to locate the key points of equipment. The PCK and FPS of the PEPoseNet-A decreased obviously compared to the baseline model. It indicated that a large amount of feature data lost in the heatmap. Direct usage of the heatmap is not conducive to the key points. The PCK reduction of the PEPoseNet-B indicates that the scheme of frozen back-propagation routes is effective. The heatmap network is freed from the interference of the key point network. The results of PEPoseNet-C demonstrated that the effectiveness of a pretraining model based on the common HPE datasets.

Tables 4 and 5 list the performance of action quality assessment conducted by the PEPoseNet and the MediaPipe, respectively. The four quantitative metrics of the PEPoseNet were obviously superior to that of the MediaPipe. It reflects the effectiveness of introducing the information of the key points of the equipment. The key points extracted from the horizontal bar are helpful to provide reference localization information that are crucial for determining the movement states more accurately and robustly. For example, it is hard to identify and distinguish the Ready or End state in terms of only the key points of human body. If we take relative position of the examinee and the horizontal bar into consideration, it is easy to make correct determination by judging whether his or her hands hold the bar.

After successful training and testing of the PEPoseNet, we transplant the model into Android and iOS mobile platforms. The TFLite of TensorFlow and the cross-platform of Flutter are adopted. The developed mobile App can perform intelligent pull-ups test with friend interface and efficient implementation. Figure 11 illustrates the App interface in practical pull-ups test. The application is tested on more than 100 volunteer students. The results indicate that the application is suitable for practical pull-ups test with satisfactory accuracy and robustness. It provides the function of grading assessment and count of the pull-ups that is beneficial to avoid the cheating actions or false scores.

5. Conclusions

In this work, we proposed a novel deep learning model named PEPoseNet for intelligent pull-ups test based on the key point estimation of human body and horizontal bar. A self-produced pull-ups dataset containing 2,000 color images collected from volunteers and Internet was established (SDUST-PUT). The data were normalized and annotated semiautomatically. The lightweight deep network adopted backbone containing the heatmap network and the key point network. The depth-wise separable convolution was adopted to speed up the training and convergence. A grading assessment standard of 5 states in one pull-ups cycle was defined and implemented in the framework. A simple automatic grading score scheme was designed. A robust and friendly mobile application was developed for practical pull-ups test. The validation, comparison, and ablation experiments were carried out to evaluate the proposed model and software. The experimental results demonstrated that the proposed PEPoseNet and the mobile application can improve the efficiency, practicability, and fairness of pull-ups test. In the following work, we will continue to expand the size of dataset, investigate more efficient schemes to speed up the deep network, and explore more elaborate scoring scheme. Furthermore, the extension of the network and software to other sport projects will be explored and realized.

Data Availability

Some of the data that support the findings of this study are openly available in https://github.com/PEPoseNet/PEPoseNet. Other data are available from the corresponding author upon reasonable request.

Disclosure

Guozhong Liu and Jian Wang share co-first authorship.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Guozhong Liu and Jian Wang contributed equally.

Acknowledgments

This work was supported in part by the Medical and Health Research Projects of Qingdao (Grant no. 2021-WJZD192).

References

[1] D. Zhou and K. Peng, "Research on the development situation and countermeasures of intelligent equestrian sport in China," in *Proceedings of the 2021 International Conference on Information Technology and Contemporary Sports (TCS)*, pp. 592–595, Guangzhou, China, January 2021.

[2] N. Xiao, W. Yu, and X. Han, "Wearable heart rate monitoring intelligent sports bracelet based on Internet of things," *Measurement*, vol. 164, Article ID 108102, 2020.

[3] W. Zhang, Z. Wang, and X. Li, "Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis," *Reliability Engineering & System Safety*, vol. 229, Article ID 108885, 2023.

[4] X. Li, Y. Xu, N. Li, B. Yang, and Y. Lei, "Remaining useful life prediction with partial sensor malfunctions using deep adversarial networks," *IEEE/CAA Journal of Automatica Sinica*, pp. 1–14, 2022.

[5] P. Bai, J. Liu, T. Tang et al., "A 3D multi-domain U-net model for intracranial aneurysms detecting," in *Proceedings of the 2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, pp. 529–533, Nanjing, China, September 2022.

[6] A. Badiola-Bengoa and A. Mendez-Zorrilla, "A systematic review of the application of camera-based human pose estimation in the field of sport and physical Exercise," *Sensors*, vol. 21, no. 18, 2021.

[7] H. Y. Lin and T. W. Chen, "Augmented reality with human body interaction based on monocular 3D pose estimation," in *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 321–331, Berlin, Germany, December 2010.

[8] J. Stenum, K. M. Cherry-Allen, C. O. Pyles, R. D. Reetzke, M. F. Vignos, and R. T. Roemmich, "Applications of pose estimation in human health and performance across the lifespan," *Sensors*, vol. 21, no. 21, 2021.

[9] Y. Li, C. Wang, Y. Cao, B. Liu, J. Tan, and Y. Luo, "Human pose estimation based in-home lower body rehabilitation system," in *Proceedings of the 2020 International Joint Conference on Neural Networks*, pp. 1–8, Glasgow, UK, July 2020.

[10] M. Li, Z. Zhou, and X. Liu, "Cross refinement techniques for markerless human motion capture," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 1–18, 2020.

[11] S. Salti, O. Schreer, and L. Di Stefano, "Real-time 3d arm pose estimation from monocular video for enhanced HCI," in *Proceedings of the 1st ACM Workshop on Vision Networks for Behavior Analysis*, pp. 1–8, Canada, October 2008.

[12] M. Eichner, V. Ferrari, and S. Zurich, "Better appearance models for pictorial structures," in *Proceedings of the British Machine Vision Conference*, pp. 3.1–3.11, London, UK, September 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[14] A. Toshev and C. Szegedy, "Deeppose: human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, Piscataway, NJ, USA, January 2014.

[15] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision*, pp. 483–499, Berlin, Germany, September 2016.

[16] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, Piscataway, NJ, USA, November 2017.

[17] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: on-device real-time body pose tracking," 2020, https://arxiv.org/abs/2006.10204.

[18] S. Wang, D. Yang, P. Zhai et al., "A survey of video-based action quality assessment," in *Proceedings of the 2021 International Conference on Networking Systems of AI (INSAI)*, pp. 1–9, Piscataway, NJ, USA, January 2021.

[19] D. Zecha, M. Einfalt, C. Eggert, and R. Lienhart, "Kinematic pose rectification for performance analysis and retrieval in sports," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1791–1799, Piscataway, NJ, USA, December 2018.

[20] H. Neher, K. Vats, A. Wong, and D. A. Clausi, "Hyperstacknet: a hyper stacked hourglass deep convolutional neural network architecture for joint player and stick pose estimation in hockey," in *Proceedings of the 2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 313–320, Piscataway, NJ, USA, May 2018.

[21] E. W. Trejo and P. Yuan, "Recognition of Yoga poses through an interactive system with Kinect device," in *Proceedings of the 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 1–5, Piscataway, NJ, USA, June 2018.

[22] N. Promrit and S. Waijanya, "Model for practice badminton basic skills by using motion posture detection from video posture embedding and one-shot learning technique," in *Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference*, pp. 117–124, Kobe, Japan, December 2019.

[23] S. Suda, Y. Makino, and H. Shinoda, "Prediction of volleyball trajectory using skeletal motions of setter player," in *Proceedings of the 10th Augmented Human International Conference 2019*, pp. 1–8, Reims, France, March 2019.

[24] C. Xu, Y. Fu, B. Zhang, Z. Chen, Y.-G. Jiang, and X. Xue, "Learning to score figure skating sport videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4578–4590, 2020.

[25] X. Xiang, Y. Tian, A. Reiter, G. D. Hager, and T. D. Tran, "S3d: stacking segmental p3d for action quality assessment," in *Proceedings of the 2018 25th IEEE International conference on image processing (ICIP)*, pp. 928–932, Piscataway, NJ, USA, September 2018.

[26] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5533–5541, Piscataway, NJ, USA, November 2017.

[27] S. Johnson and M. Everingham, "Clustered pose and non-linear appearance models for human pose estimation," in *Proceedings of the British Machine Vision Conference*, pp. 12.1–12.11, England, UK, September 2010.

[28] B. Sapp and B. Taskar, "Modec: multimodal decomposable models for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3681, Piscataway, NJ, USA, November 2013.

[29] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: a strongly-supervised representation for detailed action understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2248–2255, Portland, OR, USA, June 2013.

[30] G. Llc, "Flutter: build apps for any screen," 2022, https://flutter.dev.

[31] S. Kreiss, L. Bertoni, and A. Alahi, "OpenPifPaf: composite fields for semantic keypoint detection and spatio-temporal association," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13498–13511, 2022.

[32] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," 2014, https://arxiv.org/abs/1403.1687.

[33] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[34] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, vol. 37, pp. 448–456, Lille, France, July 2015.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Piscataway, NJ, USA, January 2016.

[36] M. Abadi, A. Agarwal, P. Barham et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," 2016, https://arxiv.org/abs/1603.04467.

[37] L. Huang, X. Liu, and X. Hao, "The power of online learning in stochastic network optimization," in *Proceedings of the 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, pp. 153–165, Austin, Texas, USA, June 2014.

[38] C. Lugaresi, J. Tang, H. Nash et al., "Mediapipe: a framework for building perception pipelines," 2019, https://arxiv.org/abs/1906.08172.