Dynamic Gesture Recognition Based on iCPM and RNN

CHEN Weihui\textsuperscript{1, a}, FAN Yichao\textsuperscript{2, b}, ZHANG Ye\textsuperscript{3, c}\textsuperscript{*}

\textsuperscript{1}College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, China
\textsuperscript{2}College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, China
\textsuperscript{3}College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, China
\textsuperscript{a}2595570531@qq.com, \textsuperscript{b}1260178344@qq.com
\textsuperscript{c*} Corresponding author: zhangye@zjut.edu.cn

Abstract—Aim to solve the problem that continuous complex actions are difficult to be recognized by computer vision technology in actual production, this paper collects gesture video from single-and double-viewpoint, constructs gesture joint point coordinate recognition network model by improved Convolutional Pose Machine (iCPM) model, obtains gesture Gauss heat map and joint point coordinates in each frame of video, and then inputs joint point coordinates with time series into gesture sequence recognition network. Finally, the gesture sequence is obtained. The experimental results show that the recognition accuracy of gesture recognition model based on double-viewpoint is 4.18% ~ 6.92% higher than that of the single-viewpoint gesture recognition model, which verifies that the gesture recognition model based on dual-viewpoint has better recognition effect. In addition, the video samples of Tin-Lead Soldering including continuous and complex actions verify that the model has ideal recognition ability for dynamic gesture in actual production process.

1. INTRODUCTION

With the development of computer vision, deep learning has become a new research hotspot. Zhong Huiying [1] in order to improve the recognition accuracy of dynamic gesture, designed a series network model, which combined convolutional neural network (CNN) and BiLSTM, and introduced connective time classification (CTC) as the loss function. Gu Chennan et al. [2] simultaneously integrated multiple 3D-CNN models for dynamic gesture recognition. Sun Jinghao [3] established a Gaussian mixture model according to the skin color of human hands, and used CamShift algorithm to track target gestures. Wang Haoyu et al. [4] proposed using trajectory template matching method to recognize dynamic gesture. Chen Pengzhan et al. [5] proposed a continuous dynamic gesture recognition method based on acceleration sensor according to the difference and instability of individual gesture signal. However, the above dynamic gesture recognition method based on deep learning is only to identify which kind of gesture the dynamic gesture belongs to, which belongs to the recognition of single gesture, which is difficult to play a role in practical production. In this paper, based on computer vision technology, a dynamic gesture recognition method based on iCPM and RNN is proposed to obtain gesture sequence,
improve the recognition accuracy of continuous action, reduce the complexity of the algorithm, and then realize the recognition of complex dynamic hand of employees in the production process.

2. REDEFINITION OF BASIC GESTURE ELEMENTS

The 18 kinds of Therbligs proposed by the Frank and Lillian Gilbreth, the Predetermined Time System (PTS), Modular Arrangement of Pre-determined Time Standard (MOD) and Method Time Measurement (MTM) have a strong scientific nature in the division of basic motion elements of human hands [6, 7]. And it is widely used in the field of industrial engineering. But if this kind of action division method is directly applied to the field of computer vision, it will lead to the following shortcomings: (1) different motion elements are difficult to be recognized due to their strong visual similarity, such as Transport Empty and Transport Loaded; (2) some action elements with small displacement are difficult to be recognized, such as Press, Pre-Position and Position; (3) no actual action cannot be recognized, such as Hold, Think, Unavoidable Delay, Avoidable Delay, etc.

Therefore, it is necessary to optimize and screen the basic action elements of visual recognition on the basis of PTS, MOD and MTM. So, in this paper we define five kinds of basic action elements that can be recognized by computer, and they are called basic gesture elements, i.e. Empty Move (EM), With Move (WM), Turn (TN), Grasp (GP) and Release Load (RL), as shown in TABLE I.

| Number | The name of Basic gesture element | Symbol | Definition | Relationship with PTS, MOD and MTM |
|--------|----------------------------------|--------|------------|-----------------------------------|
| 1      | Empty Move                       | EM     | Hand movement without objects | All |
| 2      | With Move                        | WM     | Hand movement with objects    | MOD, MTM |
| 3      | Turn                             | TN     | Wrist based rotation          | MTM |
| 4      | Grasp                            | GP     | The act of grasping objects by hand | All |
| 5      | Release Load                     | RL     | The act of releasing or placing an object | All |

3. CONSTRUCT GESTURE JOINT POINT COORDINATE RECOGNITION NETWORK

3.1. Select Gesture Joint Points

When the finger bends, it will present three segments with different degrees of curvature. The connection point between the three segments happens to be the joint point of the finger. Therefore, the point at the fingertip of each finger is selected as the starting joint point of the finger, and then the joint point between the three segments of the respective finger is connected. Finally, the end joint point of each finger is connected with a joint point at the wrist. In other words, a total of 21 gesture joint points can be selected. Moreover, the joint points of the model are labeled and connected in a certain order. Because the joint point at the wrist is the final connection point of each finger, this point is taken as the starting point of the gesture joint point, labeled as No.1. Then, according to the space distance of the joint points, the four joint points of the thumb are marked as 2, 3, 4 and 5 from the bottom to the top, that is, the fingertip is the end of each finger, and so on. The result is shown in Figure 1.
3.2. Construct Basic Network Model of Gesture Joint Point

VGG-13 is composed of five convolution layers, five pooling layers, three fully connected layers and one softmax classification layer [8,9]. Choosing VGG-13 as the basic network structure can realize the utilization of receptive field structure, that is, replacing a $6 \times 6$ convolution layer with two convolution layers and a pooling layer cascade structure. At the same time, it has the following advantages: (1) reducing the network parameters; (2) strengthening the nonlinear structure of the network.

3.2.1. Receptive Field

The size of receptive field is related to the sliding window of convolution layer or pooling layer, which can be regarded as a mapping. The pixel value of the range of $k \times k$ on the $n$-th layer feature map is compressed into a pixel on the $n+1$-th layer feature map, which is represented by $f_{{k}s}$, where $s$ is the step size of sliding window and $k$ is the size of convolution kernel or pooling kernel. Then their mapping relationship is as follows:

$$x^{n+1} = f_{{k}s}(x^n)$$  \hspace{1cm} (1)

Where $x^n, x^{n+1}$ are the feature maps of the $n$-th layer and the $n+1$-th layer respectively.

The parameters of receptive field and convolution kernel or pooling kernel of each link are shown in TABLE II, and the receptive field of the original image to itself is $1 \times 1$.

| Image          | Receptive field | The size of convolution kernel or pooling kernel | Stride |
|----------------|-----------------|-----------------------------------------------|--------|
| Original image | $1 \times 1$    | /                                             | /      |
| The first layer| $3 \times 3$    | $3 \times 3$                                  | 1      |
| The second layer| $5 \times 5$   | $3 \times 3$                                  | 1      |
| The third layer| $6 \times 6$   | $2 \times 2$                                  | 2      |
3.2.2. Extraction of Feature

Drawing on the idea of CPM for human posture estimation [10], VGG-13 is used to extract the image features. Firstly, define the position coordinate of the \( p \)-th gesture joint in the image pixel as \( Y_p \), then:

\[
Y_p \in Z \subset \mathbb{R}^2
\]

Where set \( Z \) represents the position of all pixels in the image.

Then, we set \( P \) joint points to be predicted, and the goal is to get the coordinates(\( Y \)) of all the joint points:

\[
Y = \{Y_1, Y_2, \ldots, Y_P\}
\]

And then, a multi-stage prediction classifier \( g_n(x) \) is defined to predict the position of each joint point in each stage. In each stage \( t \in \{1, 2, \ldots, T\} \), the prediction classifier will assign a point \( z \) in the image to \( Y_p \), and generates a heat map of each gesture joint point in each stage:

\[
\left\{\begin{array}{l}
Y_p = z \\
\forall z \in Z, t \in \{1, 2, \ldots, T\}
\end{array}\right.
\]

When the classifier predicts the position of gesture joint points in the first stage \( (t = 1) \), it will generate a heat map of gesture joint points and the corresponding belief values:

\[
g_t(x) \longrightarrow \{b^t_r(Y_p = z)\}_p(0 \ldots P)
\]

Where \( b^t_r(Y_p = z) \) is the score of the gesture joint point predicted by the classifier in the first stage when the \( p \)-th gesture joint point is at the position \( z \). For each next stage, the score of the \( p \)-th gesture joint point at position \( z \) can be expressed as follows:

\[
b^r_t(u, v) = b^t_r(Y_p = z)
\]

Where \( u \) and \( v \) represent the coordinate values of a certain position \( z \) in the image.

Similarly, the prediction classifier of each stage \( (t \geq 2) \) can still generate a heat map of gesture joint points and the corresponding belief values in each stage:

\[
g_t(X'_p, \psi_t(z, b_{t-1})) \rightarrow \{b^t_r(Y_p = z)\}_p(0 \ldots P)
\]

Where \( \psi_t(z, b_{t-1}) \) represents a mapping between belief value and image context information, \( X'_p \) represents the image features extracted in the previous stage around position \( z \).

By repeating the above process, in each stage the position of the \( p \)-th gesture joint point is modified based on the image context information of the previous stage and the image features extracted in the first stage, so that the model can finally estimate the more accurate position of the gesture joint point.

3.2.3. Output Gesture Gauss Heat Map and Hand Joint Point Coordinates

The position of gesture joint points in the image is the coordinate value of a certain pixel in the image. After feature extraction of gesture joint points in the image by using convolutional neural network, the output of the coordinate value of gesture joint point is a mapping process from image to coordinate value. This process increases the learning complexity of the model, and reduces the performance of the model in different degrees, which leads to its performance in high-precision area detection is insufficient. The heat map is used to calibrate the position of the joint point instead of the coordinate value, and the heat map of the joint point is taken as the output of the network, which can avoid the weakening of the generalization ability of the model caused by the transformation between the image and the coordinate value [11,12].

The heat map of gesture joint points is formed based on the coordinates of joint points. The color depth of each point represents the possibility of a gesture joint point. The darker the color, the greater the probability that the point is a joint point; the lighter the color, the less likely the point is to be a joint point. However, each joint point occupies a pixel area in the image, rather than only one pixel, that is, the generated heat map should be a regional heat map, not a single point heat map. Based on
this, we take a certain point in the pixel area where the relevant node is located as the center, and take the specific number of pixel points as the radius, a circular region is drawn, and the region where the joint points are located is divided as the probability region of joint points. The color of the center position of the region is the deepest, which means that the probability of the joint point in this position is the largest, while the color of other areas gets shallower outward from the center. The color will peak in the center and gradually lighten around the image form, which is similar to the image of Gaussian function. Therefore, we can use the Gaussian function to generate the heat map of each joint point area:

$$G(x, y) = \frac{1}{2\pi \sigma^2} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}$$

(8)

Where, \(x_0\) and \(y_0\) are the real coordinate values of the gesture joint points; \(x\) and \(y\) are the coordinate values of the pixel points in the heat map area of the gesture joint points; \(\frac{1}{2\pi \sigma^2}\) is the amplitude value of the two-dimensional Gaussian function; and \(\sigma^2\) is the standard deviation of \(x\) and \(y\).

Thus, based on the original image, a two-dimensional Gaussian function distribution heat map can be generated. The heat map generates a Gaussian distribution probability region based on the center point coordinates of the gesture joint area. The probability value at the center point of this area is the maximum, and the more it spreads around, the smaller the probability value. In the Gaussian probability region centered on the peak point with the largest probability value, the total probability value of all points exceeds one. However, in this probability area, the total probability of gesture joint points at all pixel positions should be one. Therefore, the function values of all pixels in the region can be addition processing to ensure that the probability sum of all points is one. The processing method is as follows:

$$p(x, y) = \frac{\sum f(x, y)}{\sum f(x, y)}$$

(9)

Among them, \(p(x, y)\) represents the probability of joint points existing in the processed pixels; \(f(x, y)\) represents the two-dimensional Gaussian function value corresponding to the pixels in the probability area; and \(\sum f(x, y)\) represents the sum of the function values of all pixels.

In each stage of the model, the Gaussian heat map of 21 joint points will be output, that is, each joint point corresponds to a Gaussian heat map. As shown in Figure 2, the Gaussian heat map of the four-hand gesture joint points No.1, No.6, No.10 and No.18 is shown.
4. CONSTRUCT GESTURE SEQUENCE RECOGNITION NETWORK

The gesture sequence recognition network is constructed based on RNN [8,13], and its input is a data with time series, that is, the coordinates of 21 gesture joint points in each frame in the video; the output is a gesture sequence.

4.1. Activation Function

When the network level is not deep, the problem of gradient vanishing is relatively minor, so Tanh can be used as the activation function in the recurrent neural network [9,14], namely:

\[ \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \] (10)

\[ \tanh'(x) = \frac{4e^{-x}}{(1 + e^{-2x})^2} \] (11)

4.2. Loss Function

Since the gesture sequence need to be output at the last layer of the network, the Softmax loss function of multi-classification is used to calculate the probability that the gestures in the input video frame belong to each category. Finally, the prediction result of gesture in the video is the category with the highest probability in each category [15].

Assuming that \( x \) is a set of eigenvectors input to Softmax layer by RNN, W and \( b \) are parameters of Softmax, then the first step of Softmax is to score each category and calculate the score of each category:

\[ \text{Logit} = W^T x + b \] (12)

Then, Softmax converts the scores of each category into their respective probability values. Assuming that the scores of five gesture categories are \( c, d, e, f, g \), the formula for Softmax to convert them into probability values can be expressed as follows:

\[ p_{i\rightarrow\text{pro}}^i = \frac{e^i}{\sum_{j=i}^5 e^j} \] (13)

Where \( i \) is the \( i \)-th gesture category and \( e^i \) is the probability score of the \( i \)-th gesture, \( i = c, d, e, f, g \).

So far, the model outputs the probability distribution of five gesture categories. This probability distribution is a predicted value, which is called \( q(x) \), and the gesture also has an actual label, that is, a real probability distribution, called \( p(x) \). Since Softmax function is also called Cross Entropy Loss function, and cross entropy describes the distance between two probability distributions:

\[ H(p, q) = -\sum p(x) \log_q q(x) \] (14)

Suppose \( p(x) = (A, B, C) \), \( q(x) = (u, v, w) \), and \( p(x) \) is the real value and \( q(x) \) is the predicted value, then the cross entropy of \( p(x) \) is expressed by \( q(x) \):

\[ H((A, B, C), (u, v, w)) = -(A \log u + B \log v + C \log w) \] (15)

When the positions of \( q(x) \) and \( p(x) \) are exchanged, their cross entropy is different. The closer the predicted probability distribution \( q(x) \) is to the true value \( p(x) \), the smaller the cross entropy of them will be, which means that the closer the output of the model is to the real value, the more accurate the prediction of the model will be.
4.3. Modeling
The five basic gesture recognition network models belong to a five-classification model, so it is necessary to design the depth of the network into two layers and output five different classification results, as shown in Figure 3.

Figure 3. Schematic diagram of five-classification recurrent network structure

In the model, \( X=(x_1, x_2, x_3, ..., x_T) \) are the video frames expanded according to the time series. These time series frames are used as the input of the RNN. The information contained in each frame is the coordinate value of the joint points of each gesture, and the length of the time series is set to \( T \). If the hidden state of the first hidden layer is \( H=(h_1^{(1)}, h_2^{(1)}, ..., h_T^{(1)}) \), the hidden state of the first hidden layer is as follows:

\[
h_t^{(1)} = f \left( U^{(1)} x_t + W^{(1)} h_{t-1} + b^{(1)} \right)
\]  

(16)

The hidden state of the first sequence of the first hidden layer is:

\[
h_t^{(1)} = f \left( U^{(1)} x_t + b^{(1)} \right)
\]  

(17)

For the second hidden layer, the input is determined by the hidden state of the previous time and the hidden state of the previous hidden layer which is also in the current time. Then the hidden state of the second hidden layer can be expressed as follows:

\[
h_t^{(2)} = f \left( U^{(2)} h_t^{(1)} + W^{(2)} h_t^{(2)} + b^{(2)} \right)
\]  

(18)

The hidden state of the first sequence of the second hidden layer is as follows:

\[
h_t^{(2)} = f \left( U^{(2)} h_t^{(1)} + b^{(2)} \right)
\]  

(19)

Finally, for the prediction results of five gestures, \( Y=(Y_1, Y_2, Y_3, Y_4, Y_5) \), there are:

\[
Y_i = \text{Softmax} \left( V h_t + k \right)
\]  

(20)

Where \( i=(1,2,3,4,5) \), \( U \), \( W \) and \( V \) are all parameter matrices, which are used for matrix transformation of the hidden state of input and hidden layer. And \( b \) and \( k \) are offset, and all parameters are shared in each stage of the network.

5. EXPERIMENT

5.1. Preparation of Training Sample Data Set
Firstly, five kinds of basic gesture elements were collected in a single-viewpoint. Each gesture collected 500 short videos of 1 to 2 seconds, which were completed by 20 different people. Each gesture took 50 short videos, a total of 5000 short gestures videos, to establish a database of basic gesture elements in a single-viewpoint.
Then, considering that there may be self-occlusion of joints in the process of motion change, we collected 5000 short hand gestures videos from two-viewpoint. Among them, the gesture video samples were collected in the form of two cameras with 90° angle, as shown in Figure 4.

![Figure 4. 90°angle between two perspective viewpoints](image)

In addition, based on the partition rules of the data set, under the condition that the samples are independent and identically distributed, 5000 video image samples from single-viewpoint and double-viewpoint are divided into training set, verification set and test set according to 8:1:1.

5.2. Pretreatment
Due to the influence of internal and external environment, the video image collected by the camera will have noise, which has unpredictable adverse effects on gesture recognition, so it must be filtered out. Before the start of video image filtering, first of all, select a pixel in the image randomly, and then carry out Kalman filtering according to the horizontal and vertical directions [16]. Other pixels are processed as the same method, and the image denoising is realized by predicting and updating the pixel value.

The "Grasp" gesture video image is compared before and after Kalman filter processing, as shown in Figure 5.

![Figure 5. The “Grasp” gesture video image processing using Kalman filter](image)

5.3. Evaluation

5.3.1. Evaluation of Model
The experimental environment of this paper is shown in TABLE III and TABLE IV. Among them, TABLE III shows the hardware environment of the experimental computer, TABLE IV shows the development environment of the experiment, and TABLE V lists the parameters of the model.

| TABLE III. COMPUTER CONFIGURATION FOR EXPERIENT |
|-----------------------------------------------|
| **Operating System** | Linux |
| CPU                | i5-7300HQ |
| GPU                | Nvidia GeForce GTX 1080 Ti |
| Memory             | DDR4 8G |
### TABLE IV. DEVELOPMENT ENVIRONMENT FOR EXPERIMENT

| Language Environment | Python3.5.2 |
|----------------------|-------------|
| Development Framework| Tensorflow-gpu-1.2 |
| CUDA                 | CUDA8.0 |
| CUDNN                | CUDNN8.0 |
| OpenCV               | Opencv_python-3.4.2 |
| Compiler             | PyCharm |

### TABLE V. TRAINING PARAMETER

| Training Parameters               | Value Setting |
|-----------------------------------|---------------|
| Weight decay                      | 0.0005        |
| Optimizer                         | Adam          |
| Learning rate                     | 0.001         |
| Learning rate decay factor        | 0.94          |
| End learning rate                 | 0.0001        |
| Batch size                        | 8             |
| Max step                          | 30000         |

First of all, the video samples collected from single-viewpoint and double-viewpoint are trained. The initial learning rate is 0.001, and the Learning rate decay factor is 0.94, and the minimum learning rate after attenuation is 0.0001. The size of the video frame read during training is 408×720, and the video length is between 1 and 2 seconds. Therefore, the length of the video frame read each time is uncertain. After each iteration, the value of the loss function will be output once. After a total of 30000 iterations, the convergence of the loss function of five basic gesture elements under single-and double-viewpoint can be obtained after 30000 steps of training, as shown in Figure 6.

(a) Single-Viewpoint
Through comparative analysis, it can be seen that training the model by using the video samples collected from dual-viewpoint is helpful to the convergence of the model. Although the loss function of the two methods does not decrease significantly after 5000 steps, the loss function convergence of the double-viewpoint is between 10 and 30, which is improved compared with the results of 20 to 40 convergence in single-viewpoint.

Furthermore, the verification set and test set are used to compare and evaluate the trained models in single-and double-viewpoint. The evaluation results are shown in TABLE VI.

**TABLE VI. ACCURACY OF FIVE GESTURES IN VERIFICATION SET AND TEST SET IN SINGLE ANF DOUBLE VIEWPOIINT**

| Basic Gesture Elements | Accuracy of verification set | Accuracy of test set |
|------------------------|-----------------------------|---------------------|
|                        | Single-Viewpoint | Double-Viewpoint | Single-Viewpoint | Double-Viewpoint |
| EM                     | 82.99%           | 89.23%           | 83.69%           | 88.34%           |
| WM                     | 81.89%           | 86.75%           | 81.03%           | 85.21%           |
| TN                     | 72.24%           | 79.16%           | 72.47%           | 79.32%           |
| EM                     | 72.68%           | 78.03%           | 73.02%           | 77.89%           |
| RL                     | 74.02%           | 78.27%           | 73.78%           | 78.33%           |

It can be seen that the accuracy of verification set of five basic gesture elements in dual-viewpoint is improved compared with that in single-viewpoint, in which EM, WM, TN, EM and RL are improved by 6.24%, 4.86%, 6.92%, 5.35% and 4.25%, respectively. The accuracy of the test set of the five basic gesture elements in the dual view was also improved compared with that in the single view. Among them, EM, WM, TN, EM and RL were improved by 4.65%, 4.18%, 6.85%, 4.87% and 4.55%, respectively. Therefore, the effectiveness of the video samples collected in dual-viewpoint is verified to solve the three difficulties of gesture self-occlusion, gesture ambiguity and gesture normal motion relative to the viewpoint.

### 5.3.2. Verification of Model

The video of *Tin-Lead Soldering* is collected in two-viewpoint which is to verify the gesture estimation results of the basic gesture elements. Some estimation results are shown in Figure 7.
Figure 7. Partial estimation results of five basic gesture elements in double-viewpoint

Then, the sequential arrangement of the basic gesture elements of Tin-Lead Soldering is verified to verify whether the model can realize the recognition of continuous and complex hand gestures.

Taking a video sample, the video gesture content is: the gesture of left hand is EM, GP, WM, WM and RL; and the gesture of right hand is EM, GP, WM, RL, EM, GP, WM, WM, RL. The size of the video frame is 408×720, and the gestures in the video are continuous actions, and the duration of each gesture is about 2 seconds. In order to better illustrate the experimental results, the time arrangement results of left and right hands are compared, as shown in Figure 8 and Figure 9.

In Figure 8 (b), the ordinate is the accuracy rate of basic gesture elements recognition, and the abscissa is the time when each basic gesture element occurs, and each action lasts about two seconds. Compare Figure 8 (a) with Figure 8 (b), we can find that during a whole set of continuous action process of Tin-Lead Soldering, the temporal logic of the basic gesture elements of the left hand in Figure 8 (b) is basically consistent with that of the basic gesture elements in Figure 8 (a). According to
the time sequence, EM, GP, WM, WM and RL are identified in turn. And there are time overlaps between EM and GP, as well as WM and RL, and the actions are completed almost at the same time. It should be noted that the sequence arrangement of the relative order basic gesture elements shown in Figure 8 (a) can end at the same time in the action completion time, but when the machine recognizes the gesture, it has a sequential recognition order, which will lead to a certain delay in Figure 8 (b), while the basic gesture elements with absolute order roughly meet the end-to-end basic trend.

Similarly, compared with Figure 9 (a) and Figure 9 (b), the nine basic gesture elements basically conform to the rules of temporal logic, and the basic gesture elements in accordance with the relative order have certain overlap on the recognition curve. There are four parts in Figure 9 (b) overlap, and the basic gesture elements conforming to absolute sequence meet the end-to-end basic trend.

Through the above verification, it can better show that the model in this paper has the ideal recognition ability for the continuous complex gestures in the actual production, and the verification results well reflect the sequence arrangement of the basic hand gesture elements of the left and right hands, which verifies the feasibility of the model.

6. CONCLUSIONS
Based on computer vision technology, a gesture recognition method based on iCPM and RNN is proposed in this paper. In order to solve the problem that continuous and complex actions are difficult to be recognized by computer vision technology in actual production, the CPM model is improved based on redefining the basic gesture elements, and the gesture joint point coordinate recognition network model is established to obtain the gesture Gauss heat map and joint point coordinates of gesture video samples collected from single-viewpoint and double-viewpoint and pre-processed by normalization and Kalman filtering. And then we input them into the corrected standard gesture
sequence recognition network to get the gesture action sequence and the category of gesture. At the same time, the recognition results of five basic gesture elements in single-viewpoint and dual-viewpoint are compared, and the recognition accuracy of gesture recognition model based on dual-viewpoint is 4.18% ~ 6.92% higher than that of single-viewpoint gesture recognition model. In addition, through the use of Tin-Lead Soldering video samples including continuous and complex movements, the model has an ideal recognition ability for dynamic gestures in the actual production process.

Dynamic gesture recognition using RNN is a new gesture recognition method, which has a certain application prospect. However, RNN may not be able to learn the motion law of gestures in video sequences with long time span due to the problem of vanishing gradient. Therefore, it is necessary to enhance the ability of the model to deal with the "long-range problem".

REFERENCES
[1] Zhong, H.Y. (2020) Dynamic Gesture Recognition Method Based on Depth Learning. Electronic Measurement Technology,43:128-132.
[2] Gu, C.N., Zeng, X.Q. (2019) Dynamic Gesture Recognition Based on 3D Convolutional Neural Network. Computer and Modernization, 11:75-80.
[3] Sun, J.H. (2018) Gesture Recognition Based on Deep Learning. Computer Products and Circulation, 09:177.
[4] Wang, H.Y., Qi, J., Fang, T.E., Liu, D.Q. (2017) Dynamic Hand Gesture Recognition Based on Track Template Matching. Application of Single Chip Microcomputer and Embedded System, 07:39-43+46.
[5] Chen, P.Z., Luo, M., Li, J.(2016) Continuous Dynamic Gesture Recognition Based on Acceleration Sensor. Transducer and Microsystem Technologies, 01:39-42.
[6] Yi, S.P., Guo, F.(2013) Foundation of Industrial Engineering. China Machine Press, Beijing.
[7] Guo, F., Qian, X.S. (2013) Human Factors Engineering. China Machine Press, Beijing.
[8] Francois, C.(2017) Deep Learning with Python. Manning Publications, Greenwich.
[9] Xie, Y.(2017) Deep Learning & Computer Vision: Algorithms and Examples. China Machine Press, Beijing.
[10] Wei, S.E., Ramakrishna, V., Kanade, T.(2016) Convolutional Pose Machines. In: IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas. pp: 4724-4732.
[11] Tompson, J., Goroshin, R., Jain, A. (2015)Efficient Object Localization Using Convolutional Networks. In: IEEE Conference on Computer Vision and Pattern Recognition. Boston. pp: 648-656.
[12] Tompson, J., Jain,A. , LeCun, Y.(2014)Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In: Neural Information Processing Systems. Montreal. pp: 1799-1807.
[13] Li, J. (2019)Dynamic Gesture Recognition Based on Deep Convolution Neural Network. Shandong University. Jinan.
[14] Zhang, M.Y. (2019) Multimodal Feature Analysis Method of Gesture Interaction Based on Monocular Vision. Lanzhou University of Technology. Lanzhou.
[15] Luo, H.F. (2018)Research on Dynamic Gesture Recognition Technology Based on Video Similarity. Huazhong University of Science and Technology. Wuhan.
[16] Kalman, R.E. (1960)A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering, 82: 35-45.