# Jointly Network: A Network Based on CNN and RBM for Gesture Recognition

**Wentao Cheng[1], Ying Sun[1, 2], Gongfa Li[1, 3*], Guozhang Jiang[2,4], Honghai Liu[5]**

[1] Key Laboratory of Metallurgical Equipment and Control Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China

[2] Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan 430081, China

[3] Research Center of Biologic Manipulator and Intelligent Measurement and Control, Wuhan University of Science and Technology, Wuhan 430081, China

[4] 3D Printing and Intelligent Manufacturing Engineering Institute, Wuhan University of Science and Technology, Wuhan 430081, China

[5] School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK

*Corresponding Author Email: ligongfa@wust.edu.cn

**Abstract:** Hand belongs to non-rigid objects and is rich in variety, making gesture recognition more difficult. The essence of dynamic gesture recognition is the classification and recognition of single-frame still images. Therefore, this paper mainly focuses on static gesture recognition. At present, there are some problems in gesture recognition, such as accuracy, real-time or poor robustness. To solve the above problems, in this paper, the Kinect sensor is used to obtain the color and depth gesture samples, and the gesture samples are processed. On this basis, a jointly network of CNN and RBM is proposed for gesture recognition. It mainly uses superposed network of multiple RBMs to carry out unsupervised feature extraction and combined with supervised feature extraction of CNN. Finally, these two features are combined to classify them. The simulation results show that the proposed jointly network has a better performance in identifying simple background gesture samples, and the recognition capability of gesture samples in complex background needs to be improved.

**Keywords:** Kinect Sensor; Jointly network; Static gesture recognition; Simulation experiment

## 1 Introduction

The traditional way of human-computer interaction is mainly through the keyboard, mouse, touch screen and other devices, which is still very different from the natural way of communication in daily life. In order to achieve more natural human-computer interaction, with the rapid development of computer science, scholars have done a lot of research on human motion capture and recognition, and gesture recognition technology has also become one of the cores of its research [1]. In recent years, gesture control has been increasingly applied to a variety of products. This more intuitive human-computer interaction method has enabled the application of gesture recognition technology to a wider range of applications such as virtual reality, entertainment game, industrial control and Aerospace and other fields. The goal of gesture recognition is to use manpower as a direct input device, eliminating the need for intermediate media and controlling the machine directly through defined gestures [2]. In interactive gestures of human-machine gestures, the purpose is to make the machine understand the meaning of gestures in front of it [3]. During this period are often required rivals in real-time detection, tracking, identification. Behind this series of complex things, its essence is to classify and recognize single-frame still images. The first two stages distinguish between hands and hands, and the last stage is the category distinction between gestures. Therefore, part of static gesture recognition is identifying the focus of the work [4].

Although there are many methods of gesture recognition, the vision-based gesture recognition still faces many serious problems in practice. It is mainly reflected in low recognition rate, poor robustness, insensitivity in real-time and poor practicability. At present, due to the repeated changes in gestures, its inaccuracy will be affected by the occlusion, and even its movement speed will have an impact on the gesture. When we actually perform gesture recognition simulation, the environment background, different illumination intensity and dynamic information are required to be high, so there is still room for improvement in gesture recognition research. CNN (Convolutional Neural Network, CNN) is mostly used in the field of image, but for one-dimensional signals, such as voice, RBM-based multi-layer network is mostly used. Each of them has its own shortcomings. RBM-based multi-layer network can be directly used for automatic feature extraction, automatic

coding, noise reduction, dimensionality reduction, dimensionality upgrade and so on. CNN is basically used for classification directly and the effect is very good. Therefore, in this paper, we propose a jointly network for gesture recognition based on them. Fully combine the advantages of two different networks, weaken its shortcomings, and improve the performance of the network as a whole, and solve the problems of low accuracy, real-time and poor robustness in current gesture recognition.

In the remainder of this paper, Section 2 reviews relevant literature on feature extraction and gesture recognition. In Section 3, we introduce the method of obtaining and processing gesture samples. And then, we introduce CNN and RBN networks and propose a jointly network based on them, and design its network structure in Section 4. Subsequently, we verified the recognition performance of several different networks under different conditions and compared the advantages of our proposed network in Section 5. In the last section, conclusions are presented.

## 2 Related Works

In this section, we review some work relevant to feature extraction and gesture recognition.

Previous approaches for feature extraction have mainly focused on a characteristic of supervision or unsupervised. In literature [5] puts forward an image feature set which contains three kinds of Haar features and applied it to pedestrian detection and face recognition in the environment, and achieved good results. However, due to the fewer features, the use of this feature requires a larger training set, which makes its practical application difficult. In literature [6], In order to improve the fineness of the description of the features of Hu invariant moments, three characterization vector formulas have been added for recognition research. In literature [7] fuses the information of the first four Hu moments and the contour of the gesture, which improves the accuracy of recognition. In the case of environmental impacts, the literature [8] extracted the HOG features from pedestrian detection and overcome the environmental change factors. The literature [9] uses LBP features to classify different texture images, and the operation is simple. In literature [10], twelve Fourier descriptors are used as feature vectors for 10 gesture types. In literature [11] uses the pixel histogram to represent the relationship between the number of fingers and it's corresponding, and distinguish the 1-9 gesture, the average recognition rate is about 90%. The literature [12] detected the outline information of the finger and judged the category by its specific number and direction.

Many previous approaches for gesture recognition, such as the literature [13], used Kinect's deep information to locate human hands, and detect the location of five fingers based on contour and convex concave points. The static gesture can be identified by calculating the angle characteristics of the three points. At present, 9 static gestures can be identified. The literature [14] combines depth and color information to recognize two kinds of gestures. Literature [15] used depth and extracted shape features to identify static gesture recognition of stone scissors cloth and algebraic operation, but did not give the recognition rate analysis. Literature [16] used Kinect to propose kernel learning algorithm to recognize 3 kinds of dynamic gestures. Literature [17] extracts the three-dimensional features of the hand locus and identifies 8 dynamic gestures using the hidden Markov model, but does not analyze the exclusion of undefined gestures. In practical applications, the elimination of undefined gestures is an important factor affecting the application of the undefined gestures. However, these methods have problems such as accuracy and robustness.

## 3 Acquisitions and Processing of Gesture Samples

In order to obtain both RGB and depth images at the same time, this paper uses a new type of 3D depth sensor Kinect, which is provides color data streams, raw audio data and depth data from infrared cameras with skeleton tracking, identification and speech recognition [5]. It offers the possibility of cheap, easy-to-use and real-time human-computer interaction. Because of the robustness of Kinect's infrared camera-based depth information to light changes and complex backgrounds, some researchers have used to Kinect sensors to study the segmentation and gesture recognition.

### 3.1 Kinect Sensor

Kinect sensor can get RGB and depth map at the same time, and it can also track the whole body and skeleton in real time. It can also accurately identify many complex movements of the human body. Fig. 1 is a Kinect appearance, in which the 3D depth sensor is made up of an infrared

emitter and receiving CMOS camera.


Fig. 1 Kinect Sensor

The biggest feature of Kinect is that it has a CMOS infrared sensor with no camera. It can use black and white spectrum to feel the external environment. White and black are replaced by infinite and infinite. There is a gray area between black and white, being the object and the real distance sensor. It can gather all points within the scope of the field of vision and create a depth map of the landscape, which represents the surrounding environment. The speed of the depth image flow is 30 frame / s, which can restore the surrounding environment in real time, online and 3D. Compared with some laser sensors, Kinect sensors cannot only get RGB and depth images at the same time, but also have strong anti-interference and low cost to the visible spectrum, so it is very suitable for some indoor navigation with low accuracy [6].

### 3.2 Gesture Image Preprocessing

Before the gesture recognition, the gesture needs to be segmented from the image. The background is removed to retain only the foreground gesture area, and the feature data of the gesture are extracted to provide sample data for subsequent recognition. Due to the variability of the background, the diversity of gestures, and the interference of the illumination, it makes the gesture segmentation difficult and affects the recognition rate [8-9]. Therefore, it is extremely important to accurately and completely segment the gesture. The gesture feature can reflect the similarity between similar gestures and the difference between different gestures. Therefore, extracting valid feature data is the basis for subsequent recognition.

At present, the representative gesture segmentation method is the skin color model method, because the skin color feature is not affected by rotation, scaling, and the like. The segmentation is not restricted by wear, the processing speed is fast, and the skin color has a clustering characteristic in a specific color space, which is the basis and focus of the segmentation, so it is often used for gesture or face image segmentation. The skin color based segmentation process is shown in Fig. 2.

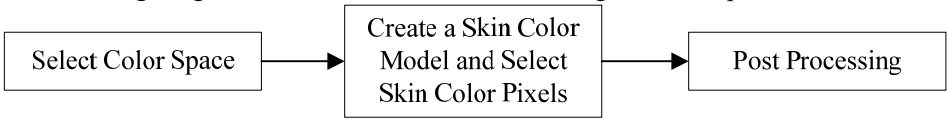


Fig. 2 Skin color segmentation process

The color space is represented by a three-dimensional coordinate system and subspaces of the system, where the colors are represented by a set of primary colors [10]. Each color has a unique position in the space, and the image is described differently in different color spaces. The color space are often involved in skin color segmentation include HSV color space and *YCbCr* color space. The gesture image is converted into two color space to analyze the clustering effect of skin color in two color space. Finally, compared with the HSV color space, the *Y* component in the *YCbCr* color space is also independent of the *Cb* and *Cr* components, so that the skin color segmentation is less affected by illumination, and the conversion to the RGB color space is linear, which is easier to calculate. The clustering effect of skin color in *YCbCr* color space is more compact than HSV color space and easy to segment. So this paper chooses to segment the gesture image in *YCbCr* color space.

After the color space is selected, experimental statistics are used to analyze the distribution of skin color. And establish a skin color model, set the judgment of the quasi-test to segment. The purpose of establishing a skin color model is to determine whether a pixel in the image is in the distribution area of the skin color, or to calculate the degree of similarity to the skin color. Common skin color models include threshold models, Gaussian models, and elliptical models.

In order to test the gesture segmentation effect of each skin color model in the *YCbCr* color

space, the test is performed using a simple background and a gesture image in a complex background, as shown in Fig. 3. In the simple background, the segmentation images of the three commonly used models are basically the same, and the segmentation effects are better. Under the complex background, the threshold model has the worst segmentation effect, and the ellipse model is the best. Since the threshold area contains a large number of non-skinned pixels, there are many misjudgment areas in the segmentation map [11]. The Gaussian model uses the Gaussian probability density function to count the probability of a pixel in the skin tone. The cutting effect is better than the threshold model, but the calculation time is longer. The ellipse model uses elliptical regions to contain skin color pixels, and the segmentation effect is basically the same as the Gaussian model, and the segmentation is faster. Therefore, this paper chooses to create an elliptical model in the $YCbCr$ color space to segment the gesture image, and finally, the data collected by the method of rotation and shearing are expanded, as shown in Fig. 4.

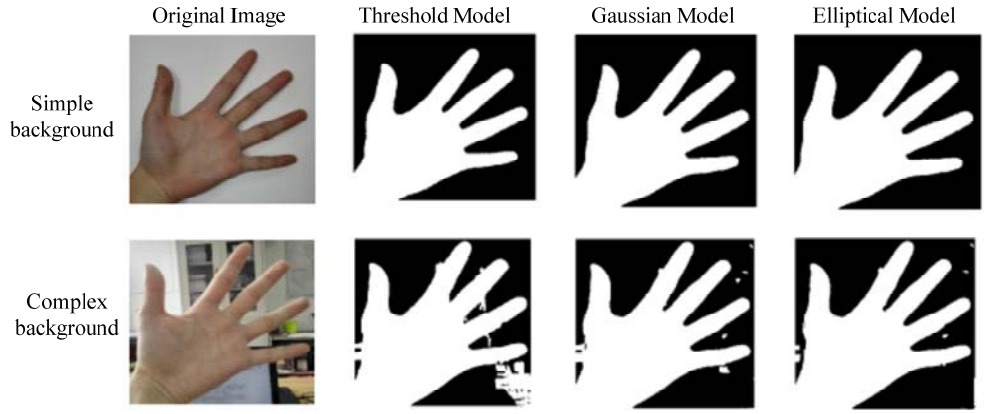


Fig. 3 Comparison of gesture segmentation of different skin color models

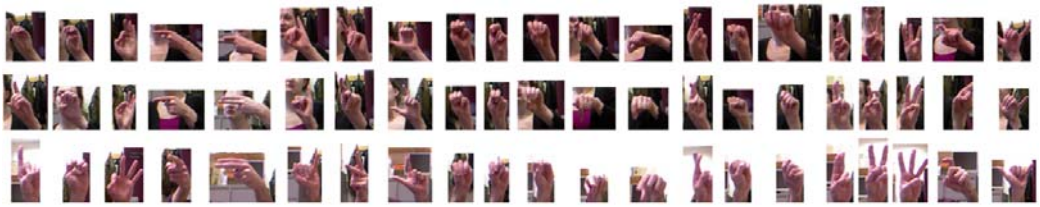

Fig. 4 Partial gesture sample data

Among them, the total sample size of ten types is 10000 under the simple background. It is a very difficult task to detect and recognize gestures in complex background, which requires a large number of positive and negative samples, and because it takes a lot of time to make hand gesture samples with complex background, so we control the specific environment. At present, the total number of samples in the complex background is 7000.

## 4 Neural Network Structure

### 4.1 Convolution Neural Network

There are many structures of convolutional neural networks, such as DeepID network structure for face recognition, LeNet-5 for identifying digital handwriting, and ImageNet-2010 network structure. Convolution neural network contains three structural features to help ensure that some of the translation, rotation, scale transforms invariance, respectively, local receptivity, weight sharing, and space or time down-sampling. Local receptive wild neurons can extract basic visual features such as edges, endpoints, angles, and then connect these features to subsequent layers to detect higher-order feature combinations [8]. And these features are connected to subsequent layers to detect higher-order feature combinations. Weight sharing makes the number of network parameters reduced a lot, the fewer the network parameters, the lower part of the objective function will be less, which helps us to find a better training in the local minimum. Down-sampling can reduce computation time, build deeper abstractions, and highlight certain feature information such as zooming in and out of a graph. Its profile information can change dramatically, and outline information disappears as you zoom in, conversely, the outline information will gradually appear, while down-sampling helps to suppress the noise. Fig. 1 shows a typical convolution neural network

for identification-LeNet-5 [9].

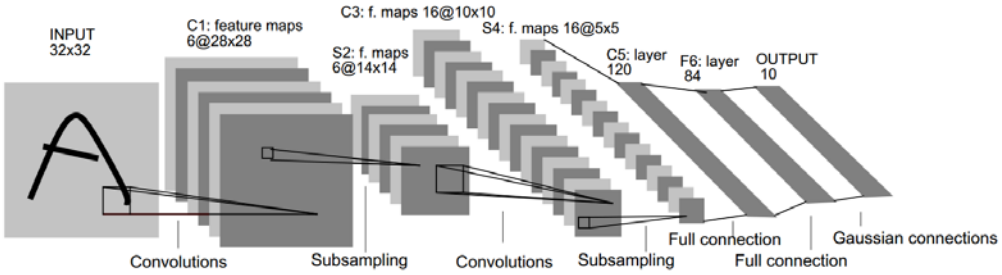


Fig. 5 LeNet-5 network structure

As shown in Fig. 5, the network structure consists of eight layers from the input layer to the output layer. In addition to the input layer and output layer, there are two Convolutions layers, two pooling layers and two full connection layers. Convolutions layers are alternately carried out with the pooling layers. Full connection layers are placed behind the final pooling layers, and the two-dimensional feature map of the pooling layer is spread into a one-dimensional feature vector.

The convolution operation of the convolutions layer like equation (1) shows:

$$x_i^n = f\left(\sum_{j \in M_i} x_j^{n-1} \times K_{ij}^n + b_i^n\right) \tag{1}$$

In the formula, $n$ represents the number of layers in the model, $K$ represents the convolution kernel, $M_i$ represents the $i$ characteristic map of the $n$-1 layer, $b$ is the bias of the output graph, and $f$ is the activation function. The activation functions of traditional convolution neural networks generally adopt saturated nonlinear functions such as *sigmoid* and *tanh*.

Subsampling is to further reduce the amount of data and reduce the spatial resolution of the input image, which is to sample the input feature map [12-13]. By extracting the features of the feature map, the down sampling can be implemented simply, so as to ensure the robustness and distortion of the model's displacement and scaling.

As shown in Fig. 5, the input layer inputs the image with a size of 32×32, and the convolution operation is carried out on the input layer to get the $C_1$ layer convolution feature. In order to fully extract the feature of the input image, the convolution kernel is used to extract features [14-15]. LeNet-5 uses 5×5×6 convolution kernels to extract features, and then 6 feature graphs was obtained. If the size of the input image is 32×32, 5×5 convolution kernel is used to calculate the size of the next feature map are 28×28.

The $S_2$ layer performs the down sampling operation on the convolution layer on the top layer, and the sampling operation can reduce the amount of data without losing the feature. The relationship between $C_3$ and $S_4$ is similar to $C_1$ and $S_2$. The difference is that $C_3$ is the convolution kernel of 3×3×16, so as to get more feature maps and better describe image content information. The $C_1$ layer convolution operation is directly related to the input image, while the $C_3$ layer convolution operation needs to act on the $S_2$ layer, while the $S_2$ layer is the characteristic map of the 14×14×6. Therefore, the way of connection between $C_3$ and $S_2$ is different from that between $C_2$ and input layer. The connections between $C_3$ and $S_2$ are shown in Table 1.

Table .1 Connections between the $C_3$ feature map and the $S_2$ feature map

| $S_2$ \ $C_3$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | √ | | | | √ | √ | √ | | | √ | √ | √ | √ | | √ | √ |
| 1 | √ | √ | | | | √ | √ | √ | | | √ | √ | √ | √ | | √ |
| 2 | √ | √ | √ | | | | √ | √ | √ | | | √ | | √ | √ | √ |
| 3 | | √ | √ | √ | | | √ | √ | √ | √ | | | √ | | √ | √ |
| 4 | | | √ | √ | √ | | | √ | √ | √ | √ | | √ | √ | | √ |
| 5 | | | | √ | √ | √ | | | √ | √ | √ | √ | | √ | √ | √ |

In Table 1, columns 0-15 are the numbers of the 16 convolution kernels of the $C_3$ convolutional layer, and rows 0-5 are the numbers of the six feature maps of the $S_2$ pooling layer. As can be seen from the above table, the convolution kernels 0-5 in the 16 convolution kernels of the $C_3$ convolutional layer are connected with the three feature maps in the $S_2$ layer, and the convolution kernels 6-14 are in the $S_2$ layer. Four feature maps are connected, and the last $C_3$ convolution kernel

is connected to the six feature maps of $S_2$. For example, as can be seen from the tenth column of Table 1, the convolution kernel numbered 9 are obtained by combining the four feature maps of the $S_2$ pooling layer numbers 0, 3, 4, and 5.

### 4.2 Restricted Boltzmann Machine

Restricted Boltzmann machine, RBM, is a probabilistic graphical model, rather a special form of log linear Markov random field. It can also be regarded as a statistical neural network that contains input layer and output layer [16]. Similarly, in order to enhance its ability to express complex distribution, the number of hidden variables can be increased. These hidden variables are regarded as variables that cannot be observed, which the number is more, the stronger the ability of simulation and distribution. Due to the enhancement of computing power and the development of machine learning algorithm, the Boltzmann machine can be applied to many related fields. There are many ways to train RBM. We mainly introduce a training method based on the energy model, which is often using Markov Monte Carlo sampling method to train [17-18]. The Boltzmann machine is simplified by the Boltzmann machine, which cancels the interconnection between the Boltzmann machine and the same layer. In the RBM graph model, $A$ represents the hidden layer nodes, and $B$ represents the visible layer nodes. Bidirectional and fully connected neurons between the explicit and hidden neurons, as shown in Fig. 6.

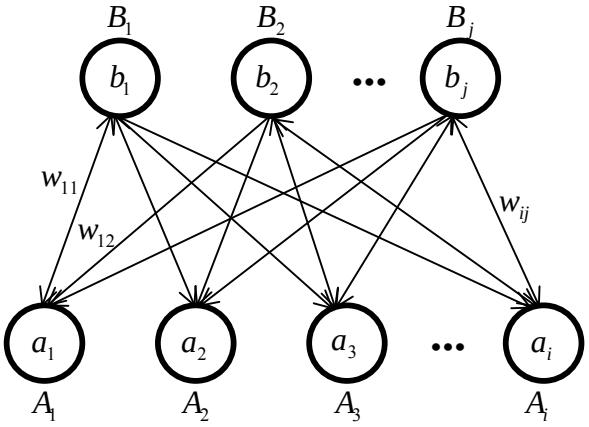


Fig. 6 RBM structure model

According to Fig. 6, for a given set of $(B,A)$, the following energy function can be defined[19]:

$$E_\theta(B,A) = -\sum_{j=1}^{B_j} x_j b_j - \sum_{i=1}^{A_i} y_i a_i - \sum_{j=1}^{B_j}\sum_{i=1}^{A_i} b_j w_{ij} a_i$$
$$E_\theta(B,A) = -X^T \times B - Y^T \times A - A^T \times W \times B \tag{2}$$

In which $X$, $Y$, $W$ are weight matrix. The standard $w_{ij}$ represents the weight of the connection between the hidden layer $A_i$ and the visible layer $B_j$. As you can see from the RBM model, In the case of known $B$, nodes $A$ of the hidden layer is conditionally independent.

$$p(A \mid B) = \prod_{n=1}^{i} P(a_i \mid B) \tag{3}$$

$$p(B \mid A) = \prod_{m=1}^{j} P(b_i \mid A) \tag{4}$$

Assuming that both $B$ and $A$ belong to the Bernoulli distribution, the conditional probability of a single variable can be expressed as a logistic function:

$$
\begin{aligned}
p(A_n = 1 \mid B) &= \frac{p(A_n = 1 \mid N)}{p(A_n = 1 \mid B) + p(A_n = 0 \mid B)} \\
&= \frac{p(A_n = 1 \mid V) \times p(B)}{p(A_n = 1 \mid B) \times p(B) + p(A_n = 0 \mid B) \times p(B)} \\
&= \frac{1}{1 + p(A_i = 0 \mid B) / p(A_i = 1 \mid B)} \\
&= \frac{1}{1 + e^{-Y_n - B \times W_{:,n}}} \\
&= \sigma(Y_n + \sum_{m=1}^{j} B_m \times W_{m,n})
\end{aligned} \tag{5}
$$

Similarly:

$$
p(B_m = 1 \mid A) = \sigma(C_m + \sum_{n=1}^{i} W_{n,m} \times A_n) \tag{6}
$$

Among them: $\sigma(x) = \frac{1}{1 + e^{-x}}$

The same level of variables is independent of each other makes Gibbs sampling easier. In the actual update of each step is often not only for a single variable but all the variables in the same layer.

### 4.2.1 RBM Release Gradient

The log likelihood function gradient of the Markov random field can be written as the formula (7) [20]:

$$
\frac{\partial Ln(L(\theta \mid x))}{\partial \theta} = -\sum_{a} p(a \mid x) \frac{\partial E(x,a)}{\partial \theta} + \sum_{a,x} p(x,a) \frac{\partial E(x,a)}{\partial \theta} \tag{7}
$$

In formula (7), $L(\theta \mid x)$ is a loss function; $-\sum_{a} p(a \mid x) \frac{\partial E(x,a)}{\partial \theta}$ is the conditional distribution expectation of the hidden variable $a$ under the training sample $x$; $\sum_{a,x} p(x,a) \frac{\partial E(x,a)}{\partial \theta}$ representative model distribution expectation.

Then we derive the partial derivatives of *W*, *X* and *Y* parameters of RBM respectively, where *x* represents *b*:

The partial derivative of *W* is the follow:

$$
\begin{aligned}
\frac{\partial Ln(L(\theta \mid x))}{\partial W_{i,j}} &= -\sum_{a} p(a \mid x) \frac{\partial E(x,a)}{\partial W_{i,j}} + \sum_{a,x} p(x \mid a) \frac{\partial E(x,a)}{\partial W_{i,j}} \\
&= \sum_{a} p(a \mid x) x_j a_j - \sum_{a,x} p(x \mid a) x_j a_i \\
&= x_j \times p(a_{i=1} \mid x) + \sum_{x} p(x) \times p(a_{i=1} \mid x) \times x_j
\end{aligned}
$$

The partial derivative of *X* is the follow:

$$
\begin{aligned}
\frac{\partial Ln(L(\theta \mid x))}{\partial X_i} &= -\sum_{h} p(a \mid x) \frac{\partial E(x,a)}{\partial X_i} + \sum_{a,x} p(x \mid a) \frac{\partial E(x,a)}{\partial X_i} \\
&= x_i - \sum_{x} p(x) \sum_{a} p(a \mid x) x_j \\
&= x_i - \sum_{x} p(x) x_i
\end{aligned}
$$

The partial derivative of *Y* is the follow:

$$\frac{\partial Ln(L(\theta \mid x))}{\partial Y_j} = -\sum_a p(a \mid x)\frac{\partial E(x,a)}{\partial Y_j} + \sum_{a,x} p(x \mid a)\frac{\partial E(x,a)}{\partial Y_j}$$
$$= p(a_j = 1 \mid x) - \sum_x p(x)\sum_{a,-j} p(a_{-j} \mid x)p(a_j \mid x)a_j$$
$$= p(a_j = 1 \mid x)$$

Written in the general form:

$$\Delta W = \varepsilon(Ep_{data}(xa^T) - Ep_{\mathrm{mod}el}(xa^T)) \tag{8}$$

$$\Delta X = \varepsilon(Ep_{data}(x) - Ep_{\mathrm{mod}el}(x)) \tag{9}$$

$$\Delta Y = \varepsilon(Ep_{data}(a) - Ep_{\mathrm{mod}el}(a)) \tag{10}$$

Among them, $\Delta W$ , $\Delta X$ and $\Delta Y$ represent the partial derivatives of logarithmic loss function to $W$ , $X$ and $Y$, respectively; $\varepsilon$ represents the learning rate, we can set these three different parameters were different learning rates; $Ep_{data}(\cdot)$ said that the visible layer to get the distribution of training samples expected expectations; $Ep_{\mathrm{mod}el}(\cdot)$ said the model expected distribution.

In the RBM model, accurate maximum likelihood learning is difficult because the time complexity of experiential distribution expectations and model distribution expectations increases exponentially with the number of implicit variables. Hinton and T. Sejnowski [21] propose an algorithm that uses Gibbs sampling to approximate these two expectations. In the each iterative training, a new Markov chain is used to approach and another Markov chain is used to approximate. The main problem is that the algorithm itself requires a certain amount of time to achieve a smooth distribution, especially when approaching the model expectation because the Gibbs-sampled object may be a multi-headed energy distribution.

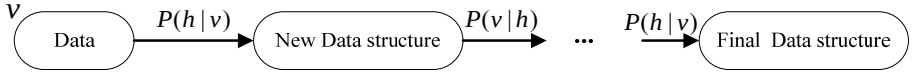


Fig. 7 Sampling process on a Markov chain

**4.2.2 RBM Training Algorithm**

The approach of approximating the RBM relief gradient is usually used for training, as obtaining an unbiased estimate of relief requires the sampling of many steps when using the MCMC sampling method [22-23]. However, in actual training, it has been found that studying the model only after a few steps of sampling is the Contrastive Divergence (CD) algorithm and has now become a standard algorithm for training RBM. More specifically, the CD_K algorithm is shown in the formula (11). $K$ represents a set of new training data coming in to perform K-step Gibbs sampling while training the RBM. In other words, every time a set of training data is to be established a Markov chain, and in this chain continuous collection of $K$ samples. Theoretically, when $K$ approaches to infinity, the distribution of learning model is more accurate, but the time it takes will be very long. The most common one is when $K$ is taken as 1, which is simple, fast and works well, especially in reconstructing errors.

$$CD_K(\theta, b^{(0)}) = -\sum_a p(a \mid b^{(0)})\frac{\partial E(b^{(0)}, a)}{\partial \theta} + \sum_a p(a \mid b^{(k)})\frac{\partial E(b^{(k)}, a)}{\partial \theta} \tag{11}$$

CD_K is a popular gradient approximation algorithm in training RBMs. CD_1 is common, however, no obvious signs indicate that it is the best. Tijmen Tielema [24] proposed a Persistent Contrastive Divergence (PCD) algorithm to train the RBM. The difference with the CD_K algorithm is that only one Markov chain needs to be established in the whole process, and all the collected samples come from this chain. Although CD_1 is fast and has a low bias, it is a suitable likelihood gradient approximation algorithm. However, when the mixing rate between a Markov chain and a Markov chain requiring approximation is low, the likelihood of CD_1 algorithm approximation Gradients is very different from the true likelihood gradient. In our simulation, we found that the reconstruction error of CD_1 when training RBM is much smaller than PCD algorithm, and PCD can extract more effective feature information.

Although the Gibbs sampling method is the main choice for RBM training, it is sometimes

difficult to approximate the distribution in real models. Therefore, many researchers began to improve their sampling techniques by other methods. Parallel tempering [25] is one of the most promising sampling techniques for RBM training. It is a supplement to Gibbs sampling in order to make the original target distribution smoother. By sampling back and forth at different temperatures, the sampled samples are even more Sample close to the real model. Suppose there are $M$ temperatures $T_1, T_2, T_3, ..., T_M$, where $1 = T_1 < T_2 < T_3 < ... < T_M$. First, define $M$ Markov chain, the jointly distribution under steady state is the follow:

$$P_r(b,a) = \frac{1}{Z_r} e^{-\frac{1}{T_r}E(b,a)} \tag{12}$$

Where $r = 1,2,...,M$, $Z_r = \sum_{b,a} e^{-\frac{1}{T_r}E(b,a)}$, and $p_r$ are the target model distributions.

In the algorithm, we use the (13) and odd even exchange rules to complete the $B$ and $A$ exchange, so that we can often get better results.

$$\min\{1, \exp(\frac{1}{T_r} - \frac{1}{T_{r-1}}) \times (E(B_r, A_r) - E(B_{r-1}, A_{r-1}))\} \tag{13}$$

After parallel tempering, the mixing speed of the Markov chain is accelerated so that the sample sampled from normal temperature is closer to the sample under real model [26]. Since the parallel tempering method in this paper is based on the PCD. It is called TPCD.

**4.3 CNN and RBM jointly network structure**

Due to the advantages of convolutional neural networks and RBM in feature extraction, we use this to form a jointly network of CNN and RBM. CNN is a powerful supervised learning network. Once the CNN for specific purposes has been trained, we can use it as a hidden feature of the middle layer of treatment, and then the CNN has become a supervised feature extractor. The RBM itself can be thought of as a feature extraction tool because one RBM is undirected. In which *B* and *A* can be restructured, and *A* is another representation of the signal *B*, which is the feature of *B*.

In feature extraction, Scott E. Fahlman et al. [27] proposed the Cascade-Correlation learning structure, as shown in Fig. 8. First, the candidate neurons are linked to all input and implicit neurons (ie, the dashed lines in the figure), and the output of the candidate neurons is not linked to the network. Then fix the solid line part of the graph and train only the weights of the candidate neurons (that is, the dotted lines in the figure). When the weights are trained, the candidate neurons are installed on the blank layer in the figure, which is the fourth area. At this time, connection rights of the options cannot be changed. The candidate neurons are then linked to the output of the network, at which point the candidate neurons are activated and begin to train all of the output connections of the network. The advantage of a Cascade-Correlation learning structure is that it learns quickly. It determines the size and topology of the network itself, and these topologies can be preserved even after training sets have changed, without the need to return the error signal to the network's connection layer.

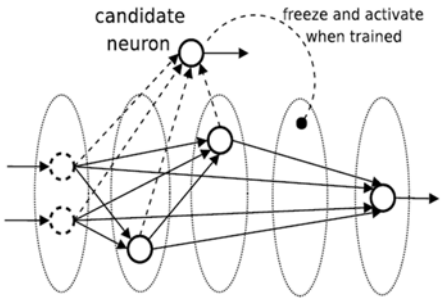


Fig. 8 Cascade-Correlation network structure

Li Deng et al. [28] proposed a Deep Stacking Networks, as shown in Fig. 9. In each module of the DSN, the output cells are linear and the hidden layers are non-linear sigmoidal cells. The training of DSN networks has the feature of making every module unit output a value similar to the value of

the tag as much as possible. DSN has a parallel learning and extensible learning ability: accumulative networks can learn complex functions, making DSN a great success in information retrieval. The advantage of DSN over other deep networks is that they are easy to learn and do not need to compute statistical gradients. The mean square error as an objective function makes it easy to train a unit of DSN, and scalability learning is useful for unlimited training data [29].

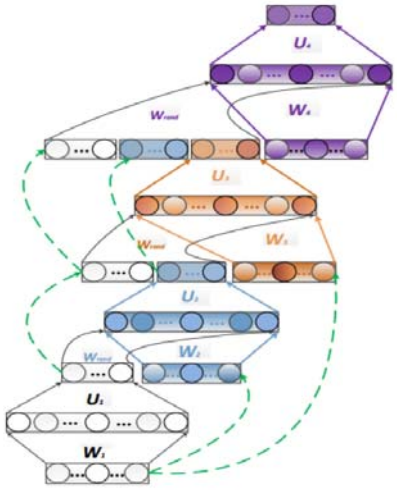

Fig. 9 Deep Stacking Networks structure

Inspired by this, this paper presents a deep feature extraction network based on RBM, as shown in Fig. 10. Similar to the structure of the Cascade-Correlation network, our network uses RBM to generate a set of features. Instead of using a function to generate a single feature like a Cascade-Correlation network, our network can produce more and more efficient features. Again, with the DSN network for comparison, we use unsupervised RBM feature extraction. In terms of computational complexity, our network is easier to implement. First, feature $A_1$ is extracted from raw data using RBM. Then, $A_1$ is merged with the original $data$. Finally, extract feature $A_2$ with another RBM until we meet our needs. The final feature $A_{RBM} = [A_n, A_{n-1}, ..., A_1, data]$ is combined with the feature $A$ of the second layers of CNN reciprocal in the graph to form a feature $A = [A_{RBM}, A_{CNN}]$. Finally, $A$ is trained in a single layer neural network, as shown in Fig. 11.

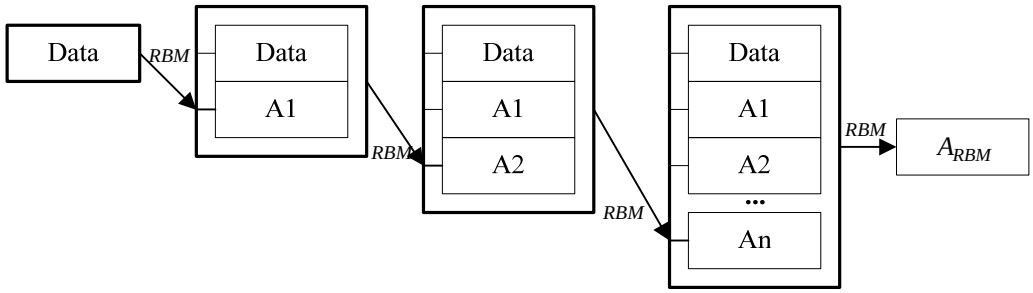


Fig. 10 RBM stacking networks feature generator

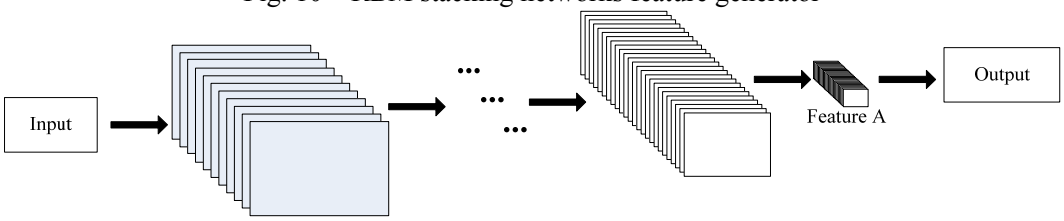


Fig. 11 CNN network feature A extraction

## 5 Simulation Comparison Experiment

The latest theoretical results show that in order to learn complex functions that can represent high-level abstract information, such as visual and linguistic abstract information, a deep structure is needed and includes multi-layer nonlinear elements such as multi-layer neural networks. We hope that the input raw data will gradually be transformed into higher-level abstract information [30-32]. In this section, we mainly use four kinds of neural networks to recognize and classify gestures under

different backgrounds, namely, deep belief nets, a deep neural network, convolutional neural network and CNN and RBM jointly networks [33-34].

### 5.1 Simple Background of the Gesture Recognition Results and Analysis

#### 5.1.1 Deep Belief Nets Recognition

Deep Belief Nets (DBNs), a type of neural network that can be used for both unsupervised learning and supervised learning [35]. When used for unsupervised learning, it is similar to a self-encoder, which is used as a classifier for supervised learning. In the case of unsupervised learning, the goal is to preserve the characteristics of the original features as much as possible while reducing the dimensions of the features. In terms of supervised learning, the purpose is to make the classification error rate as small as possible. Whether it is supervised learning or unsupervised learning, the essence of DBN is the process of feature learning, that is, how to get better feature expression [36-37]. The DBN consists of several layers of neurons, and the constituent elements are Restricted Boltzmann machines (RBMs). Therefore, different RBM training algorithms can be selected to train DBNs. Fig. 12 shows the results of four different training algorithms for DBNs training.

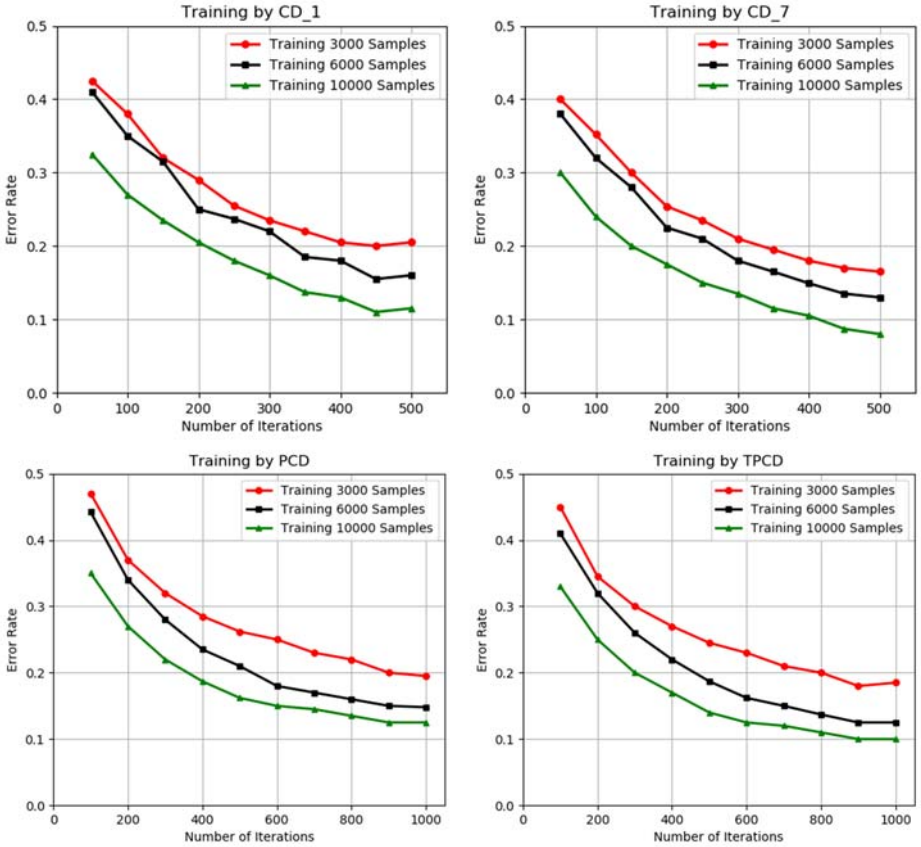


Fig. 12 Training DBNs recognition results with three hidden layers

As can be seen from the above figure, we divide the sample set. The network is trained by different numbers of samples to compare the effects of different training algorithms on gesture recognition under the same depth of DBNs. As can be seen from the figure, when the largest training sample, the recognition error of CD_1 is 11.2%, and the error of CD_K is 8.72%. Therefore, when the K value in the CD_K training algorithm is increased, the error for the recognition of the gesture is smaller, that is, the distribution of the original training samples is more likely to be approximated. Compared with the PCD and TPCD methods, the PCD gesture recognition error is 11.4%, while the TPCD gesture recognition error is 10%. Therefore, TPCD has a better effect than PCD, and the approximation of the original sample is higher. Finally, by comparing four different algorithms, CD_K has a better effect. Through experiments, it is found that when the K value is selected 7. It has the best effect for our gesture sample processing. Therefore, this paper chooses the CD_7 algorithm for training when training other networks.

At the same time, in order to analyze the influence of different depths on the identification error of DBNs network, we use different sample numbers at different depths to analyze. The result of the analysis is shown in Fig. 13, through the CD_7 training algorithm selected. CD_7_80 represents only one layer of network, and the number of nodes is 80. CD_7_80_80 represents two layers of network, and the number of nodes on the two levels is 80.

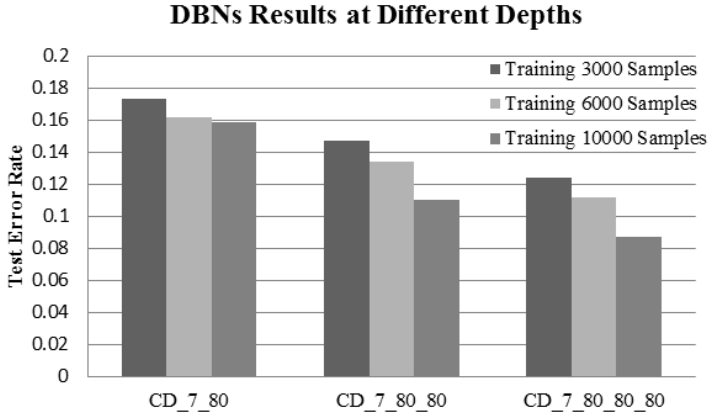


Fig. 13 DBNs simulation results at different depths

As can be seen from Fig. 13, deeper network structures can achieve smaller errors, and their performance is better than that of shallow networks. Therefore, when using RBM to construct a joint network, more layers can be used to achieve smaller errors and the accuracy of recognition is improved.

### 5.1.2 Deep Neural Network Recognition

With the development of deep learning DNN becomes possible. DNN has many advantages over shallow neural networks, the most obvious of which is its learning ability [41-42]. Therefore, in order to reduce the recognition error rate of the gesture, our recognition task turns to the DNN recognition research of the gesture. By reading other articles, we can see that the initialization of DNN's network parameters have an impact on the final effect. Therefore, this section studies different initialization conditions for DNN to determine the effect of different initialization on gesture recognition, as shown in Fig. 14-15.

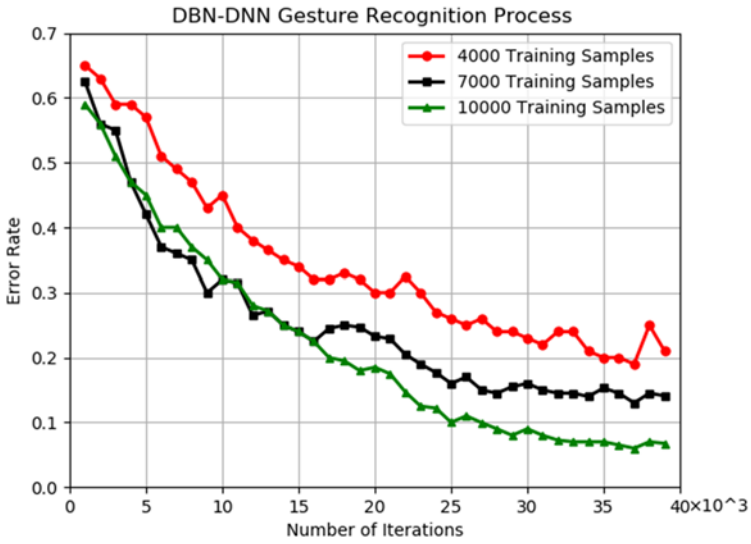


Fig. 14 DBN-DNN gesture recognition process

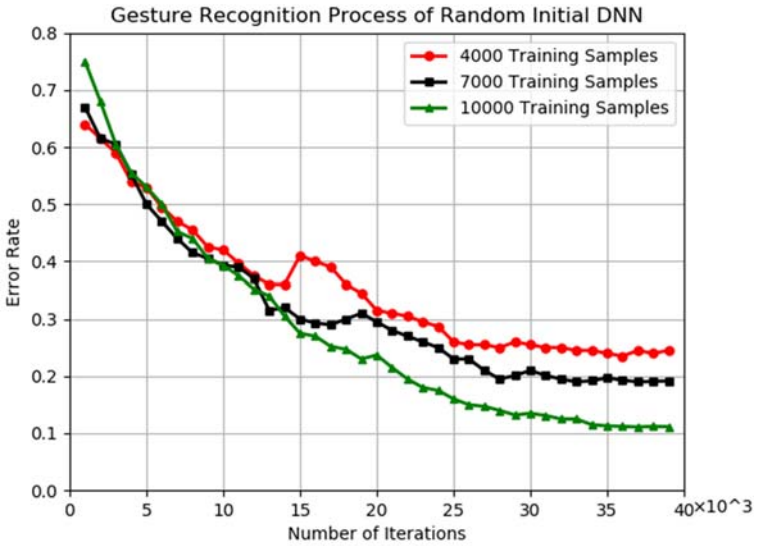


Fig. 15 Gesture recognition process of random initial DNN

As can be seen from the above figure, we choose two different ways to initialize the DNN network. One is to initialize with the DBN network, and the other is to randomly initialize the DNN network. After training the network through different samples, it is found that in the same initialization method, the larger the number of samples, the smaller the recognition error. In the DBN-initiated DNN network, the recognition error was 8.3% for 10,000 training samples. In the randomly initialized DNN network, the recognition error of 10,000 training samples is 10.23%. Therefore, it can be seen that the DBN initialization is better than the random initialization, making the performance of the DNN superior.

In order to better recognize gestures, we continue to analyze the different effects of DNN at different depths on gesture recognition. There are two initialization methods, and the final gesture recognition error is shown in Fig. 16.

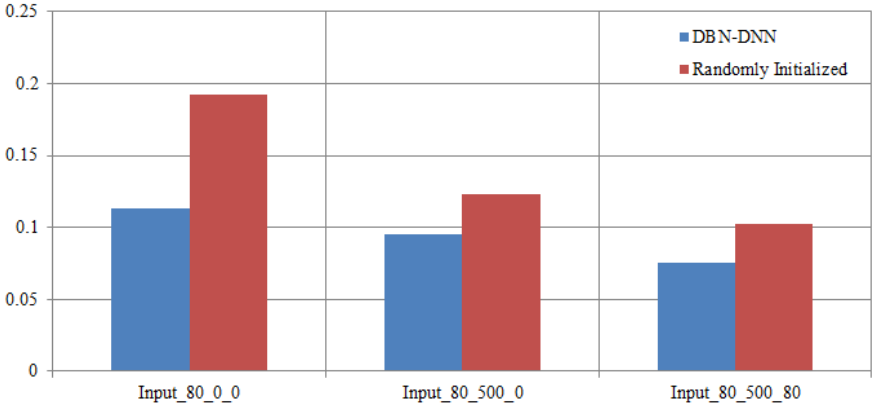


Fig.16 Comparison of DNN recognition errors at different depths

As shown in the figure above, Input_80_0_0 represents the structure of its network is one layer, and the number of nodes in this layer is 80, does not contain input and output. Input_80_500_0 represents a network structure of two layers and the number of nodes of the two layers is 80 and 500, and so on. It can be seen from the analysis that the deeper the depth of the DNN network, the smaller the error of recognition. The minimum error can reach 6.34%.

### 5.1.3 Convolutional Neural Network Recognition

When using the CNN recognition gesture, we adopt two methods to select the sample training [43-45]. One is to choose randomly the training samples from ten kinds of hand gestures through the principle of maximum entropy, and the other is to randomly select the ten samples. The simulation results show that the former is better for learning gesture samples. At the same time a 4.8% error rate was achieved in the test samples, as shown in Fig. 17.

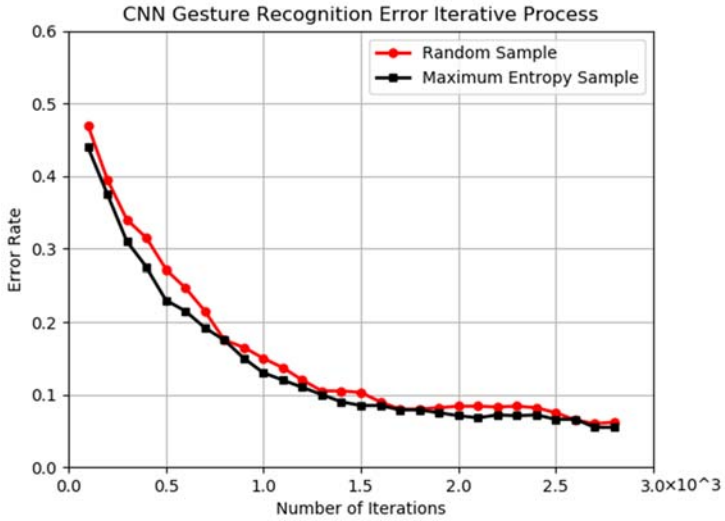


Fig. 17 CNN gesture recognition error iterative process

### 5.1.4 Jointly Network Recognition

The jointly network of CNN and RBM proposed in this paper has absorbed the advantages of CNN and RBM. Through the fusion of the two types of features, the network performance is improved on the basis of CNN and RBM. As shown in Fig. 18, the federated network achieved a 3.7% error rate in the test sample set. It can be seen from the graph that the error rate is decreasing along with the increase of iteration times. The number of iterations is $2.0 \times 10^4$ times, and the combined network is worse than the other three networks. However, as the number of iterations increases, the jointly network is superior to other networks, and has a high recognition rate.

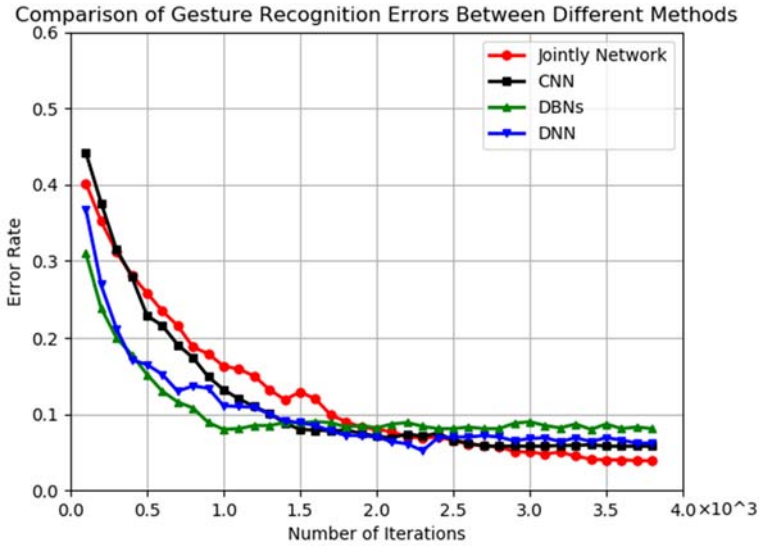


Fig. 18 Comparison of gesture recognition errors between different methods

## 5.2 Complex Background of Gesture Recognition Results and Analysis

Gesture recognition in complex contexts is much more difficult than the gesture recognition described above for simple backgrounds [46-48]. The main reason is that there are too many noise signals in the sample, which greatly increases the difficulty of network identification. A more skillful training strategy is needed to achieve a higher recognition rate [49-51]. First of all, four different training methods are used to train the DBNs network with two hidden layers, and the bottom unit of DBNs uses GB_RBM, as shown in Fig. 19. The results show that the ability of DBNs to recognize complex samples is weak. When using an RBM to simulate complex gesture samples, we found that RBM basically did not learn hand information, completely unable to reconstruct the sample information [52-54].

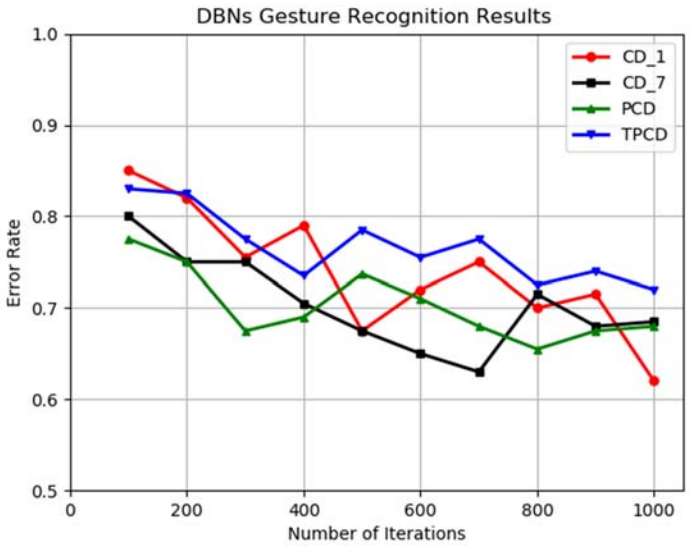


Fig. 19 DBNs gesture recognition results in complex background

For complex background gesture recognition, DNN performance [55-56] is also unsatisfactory, as shown in Fig. 20. DNN performs better than DBNs, at least in terms of convergence, while DBNs cannot always be converge.

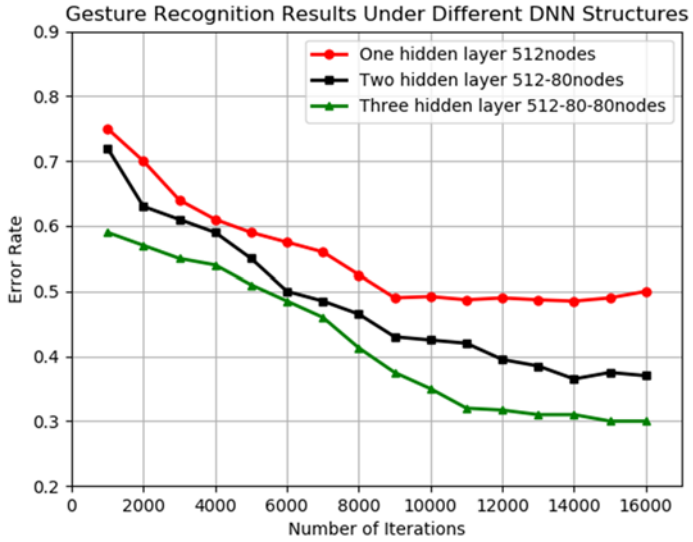


Fig. 20 Gesture recognition results under different DNN structures

It is very gratifying that the network can converge to a better local minimum due to its ability to extract invariance. However, in the simulation of DBNs, we find that the training RBM model needs a certain regular distribution, and the RBM can hardly extract useful information for complex gesture samples. Therefore, the unsupervised feature extraction based on RBM accumulation in our jointly network some cannot play a role, and even bring interference information, as shown in Fig. 21. In the latter part of the training phase, we also find that the weighted value of the unsupervised feature connected to the RBM is much smaller than the weight of the supervised feature extracted from the CNN.

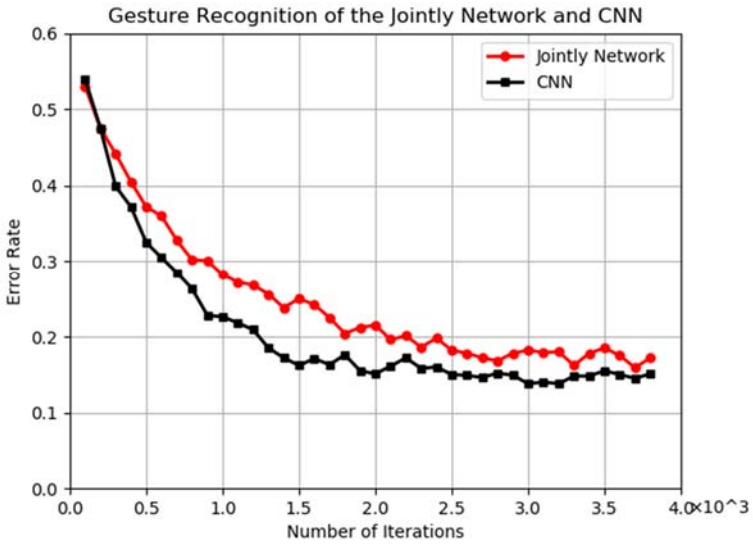


Fig. 21 The result of gesture recognition of the jointly network and CNN

## 6 Conclusions

In this paper, simulation experiments show that the recognition of gestures on RBM-based DBN networks and CNN networks has higher recognition accuracy. And often its training speed is relatively fast, and it can be well avoided for over-fitting and other issues. However, the two networks also have distinct advantages and disadvantages. The unsupervised learning network represented by RBM has poor ability to classify gestures, but it has a better effect on over-fitting than supervised learning. The supervised learning represented by CNN can greatly improve the performance of the network, but it is easy to fall into local optimum or over-fitting. Therefore, this paper proposes a joint network based on the two, combined with the supervised learning network and the unsupervised learning network, that is, the combination of CNN and RBM network. Finally, through simulation analysis, it is found that the joint network has a high recognition rate in simple sample gesture recognition, and its error is only 3.9%. Then on the complex sample, the joint network and other centralized networks do not perform well, mainly because RBM requires strict data distribution. Therefore, the future will focus on how to improve the accuracy of the joint network in a complex context.

## Acknowledgments

This work was supported by Grants of the National Natural Science Foundation of China (Grant Nos.51575407, 51575338, 51575412, 61733011) and Grants of the National Defense Pre-Research Foundation of Wuhan University of Science and Technology (GF201705). This paper is funded by Wuhan University of Science and Technology graduate students' short-term study abroad special funds.

## References

[1] Traver V.J., Latorre-Carmon Luzanin a P., Salvador-Balaguer E., Filiberto P., Bahram J., Three-dimensional integral imaging for gesture recognition under occlusions [J]. IEEE Signal Processing Letters, 2017, 24(2): 171-175.

[2] Oyedotun O.K., Khashman A., Deep learning in vision-based static hand gesture recognition [J]. Neural Computing and Applications, 2017, 28(12): 3941-3951.

[3] Nasri S., Behrad A., Razzazi F., Spatio-temporal 3D surface matching for hand gesture recognition using ICP algorithm [J]. Signal, Image and Video Processing, 2015, 9(5): 1205-1220.

[4] Li G.F., Tang H., Sun Y., Kong J.Y., Jiang G.Z., Jiang D., Tao B., Xu S., Liu H.H., Hand gesture recognition based on convolution neural network [J]. Cluster Computing, 2017: DOI:10.1007/s10586-017-1435-x.

[5] He Y., Li G.F., Liao Y.J., Sun Y., Kong J.Y., Jiang G.Z., Jiang D., Tao B., Xu S., Liu H.H., Gesture recognition based on an improved local sparse representation classification algorithm [J]. Cluster Computing, 2017: DOI: 10.1007/s10586-017-1237-1.

[6] Ding W.L., Li G.F., Jiang G.Z., Fang Y.F., Ju Z.J., Liu H.H., Intelligent computation in grasping control of dexterous robot hand [J]. Journal of Computational & Theoretical Nanoscience, 2015, 12(12):6096–6099.
[7] Li B., Sun Y., Li G.F., Kong J.Y., Jiang G.Z., Jiang D., Tao B., Xu S., Liu H.H., Gesture recognition based on modified adaptive orthogonal matching pursuit algorithm[J]. Cluster Computing, 2017: https://doi.org/10.1007/s10586-017-1231-7.
[8] Ordóñez F.J., Roggen D., Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition [J]. Sensors, 2016, 16(1): 115.
[9] Jiang D., Zheng Z.J., Li G.F., Sun Y., Kong J.Y., Jiang G.Z., Xiong H.G., Tao B., Xu S., Yu H., Liu H.H., Ju Z.J., Gesture recognition based on binocular vision [J]. ClusterComputing, 2018: https://doi.org/10.1007/s10586-018-1844-5.
[10] Xiong H.G., Fan H.L., Li G.F., Jiang G.Z., Research on steady-state simulation in dynamic job shop scheduling problem[J]. Advances in Mechanical Engineering, 2015, 7(9):1-11.
[11] Barros, P., Maciel-Junior, N.T., Fernandes, B.J., Bezerra, B.L., Fernandes, S.M., A dynamic gesture recognition and prediction system using the convexity approach [J]. Computer Vision and Image Understanding, 2017, 155: 139-149.
[12] Escalante H.J., Guyon I., Athitsos V., Jangyodsuk P., Wan J. Principal motion components for one-shot gesture recognition[J]. Pattern Analysis and Applications, 2017, 20(1): 167-182.
[13] Boughrara H., Chtourou M., Amar C.B., Chen L., Facial expression recognition based on a mlp neural network using constructive training algorithm[J]. Multimedia Tools and Applications, 2016, 75(2): 709-731.
[14] Li G.F., Gu Y.S., Kong J.Y., Jiang G.Z., Xie L.X., Wu Z.H., Li Z., He Y., Gao P., Intelligent control of air compressor production process [J]. Applied Mathematics & Information Sciences, 2013, 7(3): 1051-1058.
[15] Li G.F., Qu P.X., Kong J.Y., Jiang G.Z., Xie L.X., Gao P., Wu Z.H., He Y., Coke oven intelligent integrated control system [J]. Applied Mathematics & Information Sciences, 2013, 7(3): 1043-1050
[16] Rautaray S.S., Agrawal A., Vision based hand gesture recognition for human computer interaction: a survey [J]. Artificial Intelligence Review, 2015, 43(1): 1-54.
[17] Chakravarthi M.K., Tiwari R.K., Handa S., Accelerometer based static gesture recognition and mobile monitoring system using neural networks [J]. Procedia Computer Science, 2015, 70: 683-687.
[18] Luzanin O., Plancak M., Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network [J]. Assembly Automation, 2014, 34(1): 94-105.
[19] Pisharady P.K., Saerbeck M., Recent methods and databases in vision-based hand gesture recognition: A review[J]. Computer Vision and Image Understanding, 2015, 141: 152-165.
[20] Kılıboz N.Ç, Güdükbay U., A hand gesture recognition technique for human–computer interaction[J]. Journal of Visual Communication and Image Representation, 2015, 28: 97-104.
[21] Hinton , T. Sejnowski., Optimal perceptual inference [C]. In IEEE conference on Computer Vision and Pattern Recognition,1983.
[22] Varghese B., Buyya R. Next generation cloud computing: New trends and research directions[J]. Future Generation Computer Systems, 2018, 79: 849-861.
[23] Sun Y., Hu J.B., Li G.F., Jiang G.Z., Xiong H.G., Tao B., Zheng Z.J., Jiang D., Gear reducer optimal design based on computer multimedia simulation [J]. J Supercomput, 2018: https://doi.org/10.1007/s11227-018-2255-3.
[24] Tijmen Tieleman. Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient[C].International Conference on Machine Learning (IC-ML) 2008
[25] Li G.F., Liu Z., Jiang G.Z., Xiong H.G., Liu H.H., Numerical simulation of the influence factors for rotary kiln in temperature field and stress field and the structure optimization [J]. Advances in Mechanical Engineering, 2015, 7(6):1-15
[26] Nguyen-Dinh L.V., Calatroni A., Tröster G., Supporting One-Time Point Annotations for Gesture Recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(11): 2270-2283.
[27] Deng L., He X.D., Gao J.F., Deep stacking network for information retrieval[C], 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).
[28] Li Z., Li G.F., Jiang G.Z., Fang Y.F., Ju Z.J., Liu H.H., Intelligent Computation of grasping and manipulation for multi-fingered robotic hands [J]. Journal of Computational & Theoretical

Nanoscience, 2015, 12(12):6192-6197.
[29] Li G.F., Liu J., Jiang G.Z., Liu H.H., Numerical simulation of temperature field and thermal stress field in the new type of ladle with the nanometer adiabatic material [J]. Advances in Mechanical Engineering, 2015, 7(4):1-13
[30] Xiong H.G., Fan H.L., Jiang G.Z., Li G.F., A simulation -based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints [J]. European Journal of Operational Research, 2017, 257(1):13-24.
[31] Goh, J.E.E., Goh, M.L.I., Estrada, J.S., Lindog, N.C., Tabulog, J.C.M., Talavera, N.E.C., Presentation-Aid Armband with IMU, EMG Sensor and Bluetooth for Free-Hand Writing and Hand Gesture Recognition[J]. International Journal of Computing Sciences Research, 2017, 1(3): 54-66.
[32] Li G.F., Qu P.X., Kong J.Y., Jiang G.Z., Xie L.X., Wu Z.H., Gao P., He Y., Influence of working lining parameters on temperature and stress field of ladle [J]. Applied Mathematics & Information Sciences, 2013, 7(2): 439-448.
[33] Ohn-Bar E., Trivedi M.M., Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations[J]. IEEE transactions on intelligent transportation systems, 2014, 15(6): 2368-2377.
[34] Chen D.S., Li G.F., Sun Y., Kong J.Y., Jiang G.Z., Tang H., Ju Z.J., Yu H., Liu H.H., An interactive image segmentation method in hand gesture recognition [J]. Sensors, 2017, 17(2): 253.
[35] Liao Y.J., Sun Y., Li G.F., Kong J.Y., Jiang G.Z., Jiang D., Cai H.B., Ju Z.J., Yu H., Liu H.H., Simultaneous calibration: a jointly optimization approach for multiple kinect and external cameras [J]. Sensors, 2017, 17(7): 1491.
[36] Miao W., Li G.F., Jiang G.Z., Fang Y.F., Ju Z.J., Liu H.H., Optimal grasp planning of multi-fingered robotic hands: a review [J]. Applied and Computational Mathematics, 2015, 14(3): 238-247
[37] Chen D.S., Li G.F., Sun Y., Jiang G.Z., Kong J.Y., Liu H.H., Fusion hand gesture segmentation and extraction based on CMOS sensor and 3D sensor [J]. International Journal of Wireless and Mobile Computing, 2017, 12(3): 305-312
[38] Sun Y., Li C.Q., Li G.F., Jiang G.Z., Jiang D., Liu H.H., Zheng Z.J., Shu W.N., Gesture Recognition Based on Kinect and sEMG Signal Fusion. Mobile Networks and Applications, 2018: https://doi.org/10.1007/s11036-018-1008-0.
[39] Fang Y.F., Liu H.H., Li G.F., Zhu X.Y., A multichannel surface EMG system for hand motion recognition, International Journal of Humanoid Robotics, 2015, 12(2): 1550011.
[40] Li Z., Li G.F., Sun Y., Jiang G.Z., Kong J.Y., Liu H.H., Development of articulated robot trajectory planning [J]. International Journal of Computing Science and Mathematics, 2017,8(1):52-60.
[41] Miao W., Li G.F., Sun Y., Jiang G.Z., Kong J.Y., Liu H.H., Gesture recognition based on sparse representation [J]. International Journal of Wireless and Mobile Computing, 2016,11(4):348-356.
[42] Yin Q., Li G.F., Zhang J.G., Research on the method of step feature extraction for EOD robot based on 2d laser radar, Discrete and continuous dynamical systems-series s, 2015, 8(6): 1415-1421.
[43] Ding W.L., Li G.F., Sun Y., Jiang G.Z., Kong J.Y., Liu H.H., D-S evidential theory on semg signal recognition [J]. International Journal of Computing Science and Mathematics, 2017, 8(2): 138-145
[44] Jadooki S, Mohamad D, Saba T, et al. Fused features mining for depth-based hand gesture recognition to classify blind human communication[J]. Neural Computing and Applications, 2017, 28(11): 3285-3294.
[45] Du F., Sun Y., Li G.F., Li Z., Kong J.Y., Jiang G.Z., Jiang D., Adaptive fuzzy sliding mode control for 2-DOF articulated robot[J], Journal of Wuhan University of Science and Technology,2017,40(6):446-450.
[46] Núñez J.C., Cabido R., Pantrigo J.J., Montemayor, A.S., Vélez, J.F., Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition[J]. Pattern Recognition, 2018, 76: 80-94.
[47] Li G.F., Miao W., Jiang G.Z., Fang Y.F., Ju Z.J., Liu H.H., Intelligent control model and its simulation of flue temperature in coke oven [J]. Discrete and Continuous Dynamical Systems -

Series S (DCDS-S), 2015, 8(6): 1223-1237.
[48] Poularakis S., Katsavounidis I., Low-complexity hand gesture recognition system for continuous streams of digits and letters[J]. IEEE transactions on cybernetics, 2016, 46(9): 2094-2108.
[49] Wenjun Chang,Gongfa Li,Jianyi Kong,Ying Sun,Guozhang Jiang,Honghai Liu. Thermal Mechanical Stress Analysis of Ladle Lining with Integral Brick Joint [J]. Archives of Metallurgy and Materials 2018,63(2):659-666.
[50] Misra S., Singha J., Laskar R.H., Vision-based hand gesture recognition of alphabets, numbers, arithmetic operators and ASCII characters in order to develop a virtual text-entry interface system[J]. Neural Computing and Applications, 2017: 1-19.
[51] Li G.F., Kong J.Y., Jiang G.Z., Xie L.X., Jiang Z.G., Zhao G., Air-fuel ratio intelligent control in coke oven combustion process [J]. Information-An International Interdisciplinary Journal, 2012, 15(11): 4487-4494.
[52] Baraldi L., Paci F., Serra G., Benini L., Cucchiara R., Gesture recognition using wearable vision sensors to enhance visitors' museum experiences [J]. IEEE Sensors Journal, 2015, 15(5): 2705-2714.
[53] Gongfa Li, Leilei Zhang, Ying Sun, Jianyi Kong. Internet of Things sensors and haptic feedback for sEMG based hands [J]. Multimedia Tools and Applications,2018, https://doi.org/10.1007/s11042-018-6293-x.
[54] Oyedotun O K, Khashman A. Deep learning in vision-based static hand gesture recognition[J]. Neural Computing and Applications, 2017, 28(12): 3941-3951.
[55] Gravina R., Ma C., Pace P., Aloi G., Russo W., Li W., & Fortino G., Cloud-based Activity-aaService cyber–physical framework for human activity monitoring in mobility[J]. Future Generation Computer Systems, 2017, 75: 158-171.
[56] Singha J, Roy A, Laskar R H. Dynamic hand gesture recognition using vision-based approach for human–computer interaction[J]. Neural Computing and Applications, 2018, 29(4): 1129-1141.