Static Hand Gesture Recognition Based on Gaussian Mixture Model and Partial Differential Equation

Qinghe Zheng, Xinyu Tian, Shilei Liu, Mingqiang Yang, Hongjun Wang and Jiajie Yang

***Abstract*—In the hand gesture recognition process, manually designed features are difficult to achieve good results under the condition of changeable gestures and complex backgrounds. In this paper, we propose a hand gesture recognition method based on Gaussian skin color model and deep convolutional neural network (DCNN). For gesture images in different backgrounds, we first use the Gaussian skin color model to segment the gesture area, then we use the DCNN to establish gesture classification model. Finally, we use the back propagation algorithm based on partial differential equation to train the neural network on the pure gesture data samples to converge to the global optimum, and obtain the classification results. The model combines the process of feature extraction and classification, simulates the biological visual transmission and cognition, and effectively avoids the subjectivity and limitations of artificial features. And model reduces the size and the complexity of network by using weights sharing and pooling technology. Experimental results show that the method is efficient for gesture representation and classification. The average classification accuracies under two datasets (indoor and outdoor environments) are both more than 99%. Compared with the traditional methods, the proposed method has higher classification accuracy and speed.**

***Index Terms*—hand gesture recognition, Gaussian mixture color model, deep convolutional neural network, partial differential equation**

I. Introduction

With the rapid development of human computer interaction technology in recent decades, hand gesture recognition technology has been widely used in smart office, hospital monitoring, intelligent education and other fields. It bridges the gap of communication between equipment and people. Recently, gesture-based human-computer interaction technology [1] [56] has become an important direction for the future development, providing intelligent and natural way for human-computer interaction. From medical rehabilitation to electronic control, gesture interaction has been widely used. Gesture recognition technology [2, 59] is the basis of gesture interaction and is becoming a hotspot research in the field of human-computer interaction. As the mainstream of gesture recognition, there are two main problems in computer vision based gesture recognition technique [3, 4]: hand segmentation and gesture classification.

Hand segmentation is the premise of gesture recognition, and the quality of gesture area will directly affect the accuracy of hand gesture recognition. Skin color [5, 6] is one of the most discriminative features of the hand. However, the varied gestures, complex background environments, different light sources and color shift in the practical application will lead to changes in skin color. And distortion in the shape of the hand, including bending and reversing, can also make the shadow and the angle of light source change, which makes the skin color of the entire hand area may not be consistent, or even a great deal of difference. In the aspect of hand segmentation algorithm [7] based on contour, there are two following difficulties. One is that the initial contour is difficult to get because of the rotation or bending of the hand. Another difficulty is that it is difficult for algorithm to converge due to the presence of deep concave regions in the gesture. In some improved models, the increase of the number of iterations and the amount of computation results in real-time performance degradation. Motion-based hand segmentation methods are divided into frame difference method [8] and background subtraction method [9]. The frame difference method uses the differences between adjacent image frames to determine whether moving objects (*i.e.*, hands) are generated in the foreground. Background subtraction method first constructs the background image model, and then the hand is segmented by comparing the background image and the gesture. It can be seen from many experiments that the changes of light and shadow produced in the movement and the dynamic change of the background affect the segmentation results. At present, there are no methods that can achieve good segmentation results under all applications and background conditions.

In terms of hand classification, Grimes [10] first uses data glove to accomplish this task. The gesture recognition based on data glove requires the operator to wear data glove, and the sensor information, including finger movement track and time information, can be obtained by the glove. Then, the computer analyzes and identifies the acquired data to complete the human-computer interaction process. Although recognition method based on data glove has fast speed and high accuracy,

Manuscript received August 13, 2017; revised October 19, 2017. This work was supported by National Natural Science Foundation of China (Grant 61571275) and Shandong Provincial Natural Science Foundation (Grant ZR2014FM030, ZR2014FM010).

Qinghe Zheng, Shilei Liu, Mingqiang Yang (Corresponding author) and Hongjun Wang are with the School of Information Science and Engineering, Shandong University, Qingdao 266237, China (e-mail: 15005414319-@163.com; 364781424@qq.com; yangmq@sdu.edu.cn; imageinstitute-@outlook.com;).

Xinyu Tian is with College of Mechanical and Electrical Engineering, Shandong Management University, Changqing, Jinan, Shandong, China (e-mail: 18769796159@163.com).

Jiajie Yang is with Department of Science, University of British Columbia, Vancouver V6T1Z4, British Columbia, Canada (e-mail: 1040051920a@gmail.com).

the equipment is expensive and uncomfortable to use, which is not a natural way for human-computer interaction. Vision based hand gesture recognition is a method of classifying a sequence of images containing hand gestures. The gesture classification methods based on computer vision are mainly divided into the following categories: 1. Template matching method. 2. Classification method based on geometric features. 3. Classification method based on deep convolutional neural network. 4. Classification method based on Hidden Markov Model (HMM). Template matching method first converts the image sequence into a set of static shape patterns and then compares them with pre-stored behavior templates during the classification process. Finally, the closest known template is selected as the recognition result of the measured behavior. Bobick [11] converts the motion information of the target into two two-dimensional templates, namely MEI (Motion Energy Image) and MHI (Motion History Image), and then uses Mahalanobis distance to test the similarity between samples and templates. The problem with this method is that it cannot remove the influence of movement time. The classification method based on geometric features [12] uses the edge of the contour and the regional structure features of hand gesture as recognition features, and has good adaptability and stability in hand gesture recognition. However, the shortcomings of this method are that the learning ability is not strong, and the recognition rate will not be significantly improved when the size of samples becomes larger. The statistical method based on deep convolutional neural network [13] [14] can realize complex nonlinear mapping, and has the characteristics of anti-interference, so it is widely used in static hand gesture classification. But the learning ability of the shallow neural network is not strong, and it is easy to fall into the bad local optimal situation and lead to overfitting problem. HMM [15] has a strong ability to describe gesture spatial-temporal signal variations, but it has a large number of state probability density values and the number of parameters needs to be estimated, which makes recognition slower. At present, the existing methods need further research in gesture robust feature extraction and representation.

In this paper, we propose a gesture recognition method combining skin color model and deep convolution neural network. For the hand gesture images collected in different backgrounds, the Gaussian skin model is firstly used to segment the gesture area. Then we use the DCNN to establish the hand gesture classification model, which combines the gesture feature extraction and classification process. It can effectively avoid the subjectivity and limitation of artificially designed features by simulating visual transmission and cognition. Finally, we use the backpropagation algorithm based on partial differential equations to train the DCNN, and then obtain the classification results of the test set. The structure of this paper is organized as follows. In Section II, we introduce the related works of our method. In Section III, we introduce the image preprocessing process, that is, using Gaussian color model to extract the key area of hand in the image. In Section IV, we describe the construction process of deep CNN for hand gesture classification. The experiments that used to validate the effectiveness and rationality of the method are described in Section V. Finally, we discuss the conclusion and shortcomings of the algorithm and the future works in Section VI.

II. Related Work

According to the different data input devices, the existing gesture recognition methods are divided into two categories: gesture recognition based on data gloves and that based on computer vision. The gesture recognition algorithm based on computer vision is cheap and easy to operate, which is useful for the development of natural human-computer interaction. Therefore, gesture recognition based on computer vision is the main emphasis of our research. Therefore, we focus our research on the active perception of hand gesture recognition algorithms based on computer vision. In the first part, we introduce some of the relevant research results on gesture segmentation. And in the second part, we introduce the progress of gesture classification techniques that based on computer vision.

A. Related Work of Hand Segmentation

In computer vision, color space mainly consists of RGB, HSV and HIS, and the most commonly used color space is RGB color space. Researchers can segment the image by using image with deep information based on the Kinect sensor or to fuse the RGB and deep information. The former focuses on the speed of the algorithm, while the latter focuses on the classification accuracy. In the paper [16], hand segmentation task is considered as a depth clustering problem, and the pixels are grouped at different depth levels. By analyzing the human gesture dimension, a threshold is determined, which corresponds to the depth of hand. Lee [17] uses the K-means clustering algorithm and the predefined threshold to perform hand detection, and the opponent pattern is analyzed by the convex hull to locate the finger. Both of these two methods assume that the hand is closest to the sensor, and the effect of algorithm is greatly affected by the accuracy of Kinect depth data. Manuel Caputo [18] uses Kinect generated skeletal data to determine hand positions, and determines the size of the human hand at different depths by looking up tables storing standard hand information.

If there is no depth information, the difficulty of hand segmentation will increase. Wang [19] uses RGB and YCbCr space for threshold segmentation, and then uses a variety of morphological operations for image pre-processing. This method reduces the computational complexity and improves the computation speed, realizing the representation of contour features. Marin [20] uses the inter-frame difference method and the skin color model to calibrate the dynamic gesture area, which is small in computation, fast in speed and can overcome the influence of shape change to a certain extent. Oikonomidis [21] integrates the color information for hand detection, and converts the hand detection problem to the labeling problem of hand pixels or non-hand pixels. The skin detection operator of RGB image and the clustering operator of depth image are two conditions for confirming the hand pixel, and the hand region is the intersection of two pixel sets. Jiang [22] treats the different features into different operators, uses the continuity principle to the depth information and color information of adjacent pixels in the hand area and non-hand area, and searches from the palm of the hand as the starting point to

ensure that all the pixels form a connected effective area. This method avoids the problem that the hand must be located at the front or multiple targets in the traditional depth problem, and effectively solves the problem of uneven color and depth data of the hand. Cao [23] presents a monocular vision gesture segmentation method that combines skin color information with motion information. By analyzing the characteristics of skin color and chroma separation of color in YCbCr space and clustering characteristics of Cb-Cr plane, Gaussian mixture model is constructed and skin error decision algorithm with minimum total error criterion is proposed.

On the other hand, deep learning has developed rapidly, and a series of segmentation algorithms based on deep CNN [55] [57] [58] have been proposed, which have a high degree of guidance. Neverova [24] proposes a deep learning based hand posture estimation method for gesture segmentation, which requires little label information. It uses unlabeled data and synthetic data as training samples. The key to this work is to integrate structure information into model structure, which improves the inference speed. It is found that the addition of unlabeled samples compared with the pure supervised training method results in significant improvements in segmentation results. Zhang [25] builds a hybrid structure which includes three models: depth model (depth with morphology), skin model (skin color) and background model (codebook algorithm). The inputs are the overlap rate between any two models, and the segmentation is done by using the three-layer neural network, which reflects the consistency of the results of two models. Based on the theory of relaxation labeling technique, Akinin [26] proposed a master-slave segmentation method using Kohonen neural network. This method can effectively segment images with low SNR, and can realize the real-time processing. The experimental results show that it is better than the optimum discriminant threshold segmentation method and the matrix preserving threshold segmentation method. Long [27] proposed fully convolutional network (FCN) for image segmentation, which tries to recover the category of each pixel from the abstract features. It has achieved very good results in image semantic segmentation. But the algorithm requires a large number of samples for training and takes a lot of time to complete the semantic segmentation task.

B. Related Work of Hand Gesture Classification

In gesture classification, Bilal [28] chooses skin color features and uses particle filter to detect palm and finger, and then uses template matching method to perform hand gesture classification. Bjom [29] uses human skin colour and motion features to track human hand, and uses the nearest neighbor classifier for hand gesture classification, which has a classification accuracy of 82.2% for the 100 sign words in Irish language. Kaufmann [30] proposes gesture classification method through using intelligent evolutionary algorithm, which achieves the classification accuracy of 85.0% for the 192 sign words in American language. Weng [31] uses bias color model to segment human hands, and uses multi-feature fusion to perform hand gesture classification. Flasinski and Myslinski [32] construct a Gaussian skin color model and propose a gesture graph analytic classification method based on this model. Ren [33] integrates the information of skin color, movement and edge, extracts the characteristic lines that can reflect the structural features of human hand, and finally extracts the model parameters of translation invariant plane to complete the hand gesture classification. Through the fuzzy operations of the background, movement and color of the video stream in space and time domain, Zhu [34] segments the hand and uses the pyramid image to extract features for gesture classification. Similarly, Yang [35] proposes a gesture classification algorithm based on space distribution feature of hand gesture, which uses the Gaussian brightness model to segment the skin area and uses the search window to filter the skin area to realize the gesture location. Lahamy and Lichti [36] use hand color camera arrays to obtain depth information for hand gesture classification. Vogler and Metaxas [37] use three vertical color RGB cameras and a position tracker as gesture input devices, and recognize a gesture classification accuracy of 89.9% on 53 isolated sign words.

In the case of end-to-end gesture classification algorithms based on DCNN, a large number of models have achieved remarkable results. Among them, the three models of Alexnet [38], VGGNet [39] and GoogLeNet [40] have achieved good results in the field of image classification. AlexNet contains 5 convolutional layers, 3 pooling layers and 3 fully connected layers, which won the championship in ILSVRC 2012 and has increased the classification accuracy from 74.3% to 83.6%. VGGNet-16 consists of 13 convolutional layers, 5 pooling layers, 3 fully connected layers and 1 softmax layer, which ranked first place in the object positioning task and ranked second place in the image matching task in ILSVRC 2014. Its outstanding contribution is to demonstrate that using very small convolution kernel (3×3) and increasing network depth can effectively improve the classification effect, and it has good generalization ability for other datasets. GoogLeNet cannot only keep the sparsity of network structure, but also get high computing performance of dense matrix by using inception structure. The model won the championship in the image classification task in ILSVRC 2014 because of its excellent classification ability. Furthermore, a large number of training techniques are used in these frameworks to avoid overfitting and to improve the generalization ability of DCNN, such as regularization, dropout and data augmentation.

III. Image Preproessing

In this section, we mainly introduce the preprocessing steps of gesture images before we feed them into the DCNN. In part A, we introduce a color balance algorithm to remove the interference caused by unstable light sources and other factors. Part B introduces the gesture area extraction method based on Gaussian mixture model. Part C introduces reconstruction process of the whole gesture area based on morphological operation and single connectivity method. In fact, the part B and part C are designed to remove interference from complex backgrounds for hand gesture recognition.

A. Color Balance Algorithm

Due to the impact of light illumination, the acquired gesture images produce color shift when the illumination is not ideal, which leads to larger noise of gesture data. Therefore, we introduce the color balance algorithm which called gray world

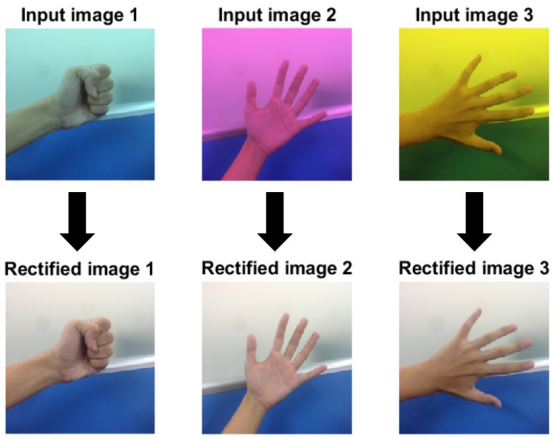


Fig. 1. Three sets of images before and after correction.

theory (GWT) [41] for color correction. The GWT algorithm is based on the gray world hypothesis, which assumed that the average value of three components R, G and B tends to be the same gray value gray for an image with a large number of color variations. The color balance algorithm applies this assumption to the image that to be processed and eliminates the impact of ambient light from the image and obtaining the original scene image. The GWT algorithm is expressed as follows.

$$R'(C) = \frac{\overline{gray}}{\overline{R}} \cdot R(C) \tag{1}$$

$$G'(C) = \frac{\overline{gray}}{\overline{G}} \cdot G(C) \tag{2}$$

$$B'(C) = \frac{\overline{gray}}{\overline{B}} \cdot B(C) \tag{3}$$

where

$$\overline{gray} = \frac{\overline{R} + \overline{G} + \overline{B}}{3}$$

where $\overline{R}$, $\overline{G}$ and $\overline{B}$ represent the mean values of the three channel colors R, G and B, respectively. $R(C)$, $G(C)$, $B(C)$ and $R'(C)$, $G'(C)$, $B'(C)$ represent the color values of pixel C before and after correction, respectively. The effect of the algorithm is shown in Fig. 1. It can be seen that the GWT algorithm achieves color correction on color biased images.

B. Hand Gesture Area Extraction

Because of the complex backgrounds of the gesture image and changeable skin color under different light sources, a reliable skin color model is needed to detect the gesture area. Some results [42] show that the color difference between different races is far less than the difference in chromaticity. YCbCr color space has the advantages of luminance and chrominance separation and has better clustering and stability. It is insensitive to the rotation of skin color and the difference of race, and approximately presents the statistical law of Gaussian distribution. Therefore, in the YCbCr space, the Gaussian distribution is used to model the skin color, and the probability values of each point in the image belong to the skin color are calculated, and the gesture region can be segmented.

Single Gaussian skin color model. The calculation of skin color model based on Gaussian distribution is shown as (4).

$$P(Cb, Cr) = \exp(-0.5(\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{x} - \boldsymbol{m})) \tag{4}$$

where

$$\boldsymbol{x} = (Cb, Cr)^{\mathrm{T}}$$
$$\boldsymbol{m} = E(\boldsymbol{x})$$
$$\boldsymbol{C} = E\{(\boldsymbol{x} - \boldsymbol{m})(\boldsymbol{x} - \boldsymbol{m})^{\mathrm{T}}\}$$

where Cb and Cr represent the blue and red concentration offset components respectively. By calculating the probability P of each pixel in the image that belongs to hand, we can establish a complete skin probability distribution matrix, and use the maximum interclass variance method with adaptive threshold to binarize the skin color probability matrix. In the binary image, the bright area with the pixel value of 1 is denoted as the skin color point, and the dark area with the pixel value of 0 is denoted as a non-skin color point.

The establishment of mixted model based on expectation maximization. As a nonparametric estimation method for mathematical modeling of unknown data by observing data, the K mixed distribution model is expressed as a random form. And the discrete variable Z used to describe the observed data follows the polynomial distribution as

$$p(x) = \sum_{i=1}^{K} \alpha_i p_i(x, \theta), \ \alpha_i > 0 \tag{5}$$

$$Z \sim Multinomial(\alpha_1, \alpha_2, \alpha_3, ..., \alpha_K) \tag{6}$$

$$x \mid Z \sim p_i(x, \theta_Z) \tag{7}$$

$$\sum_{i=1}^{K} \alpha_i = 1 \tag{8}$$

where $p(i)$ is the probability value of the variable x belonging to the target class, parameter K corresponds to the K single Gaussian distributions and Z represents the single-mode mark that corresponds to the polynomial distribution with which it belongs to, θ is the distribution parameter corresponding to each single Gaussian distribution and α_i is the corresponding weight. Above mixture model has higher flexibility relative to the single mode distribution.

Through the maximum likelihood estimation, we choose the likelihood function of the model as the objective function of the model parameter estimation. The optimal solution of the parameter corresponds to the extreme position of the likelihood function. The natural logarithm likelihood value $l(\theta)$ containing K single model and the n training samples and its derivative to the parameters of the distribution family are given by

$$l(\theta)=\sum_{i=1}^{n}\log\sum_{k=1}^{K}\alpha_k p(x_i;\theta) \tag{9}$$

$$\frac{\partial l(\theta)}{\partial\theta_j}=\sum_{i=1}^{n}\frac{\alpha_j p(x_i;\theta)}{\sum_{k=1}^{K}\alpha_k p(x_i;\theta)}\frac{\partial\log p(x_i;\theta)}{\partial\theta_j} \tag{10}$$

The expectation maximization method estimates the two factors in (9), and calculates the expectation of the weight parameter alternately (in E step), and estimates the parameters of the distribution family (in M step).

E step: calculate the weight according to the current model parameters according to

$$\omega_j^i=Q_i^t(z^{(i)}=j)=p(z^{(i)}=j\,|\,x^{(i)},\theta^{(t)}) \tag{11}$$

M step: solve the equation according to the current fixed weight parameters as

$$\frac{\partial(\sum_i\sum_{z^{(i)}}\omega_j^{(i)}\log\frac{p(x^{(i)},z^{(i)};\theta^t)}{Q_i^{(t)}z^{(i)}})}{\partial\theta^{(t)}}=0 \tag{12}$$

After obtaining the update value of the clustering parameter θ, we update the polynomial parameters as

$$\alpha_k=\sum_{i=1}^{N}l(z^{(i)}=k)/N \tag{13}$$

where l is a function of indicating species. According to the updating results of each E step and M step, it can be proved that the iterative process can increase the objective value of (8) monotonically, and the stable optimization process can be obtained. The specific process of the method is shown in the Algorithm 1.

Algorithm 1 The EM process

Model initialization:
Input: K, Z, X, θ;
for t=1…N **do**
  Calculate posterior expectation:
  $\omega^{(t)}=\arg\max\ l(X,\theta^{(t)}\ \omega^{(t-1)})$
  Update parameters of distribution family:
  $\theta^{(t+1)}=\arg\max\ l(X,\theta^{(t+1)}\ \theta^{(t)})$
until convergence
Output: θ_*

Learning method of mixture model. Assuming that the K-th Gaussian model mixture of an unknown multi-dimensional distribution is expressed as

$$f(x)=\sum\alpha_i N(x\,|\,t_i,\Sigma i) \tag{14}$$

Fig. 2. The hand gesture region detected by mixed Gaussian skin color model.

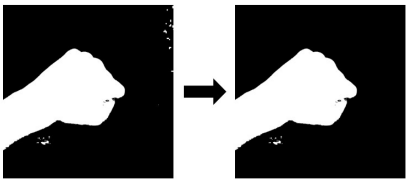


Fig. 3. The hand gesture region processed by morphological operation.

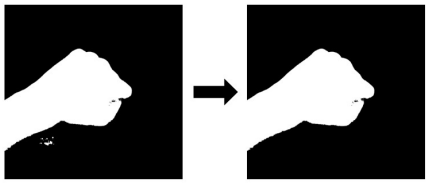


Fig. 4. The hand gesture region after single connected operation.

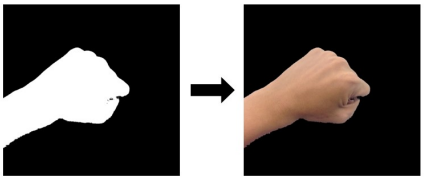


Fig. 5. Reconstructed RGB image of hand gesture region.

where N represents multidimensional Gaussian distribution, t_i is the multidimensional mean vector and Σi corresponds to the covariance matrix. The goal in the learning process of the model is to estimate all the parameters in (14) according to the training data x.

In the E step, the posterior expectation is computed by

$$\omega_j^i=P(z^{(i)}=j\,|\,x^{(i)};\alpha,t,\Sigma) \tag{15}$$

In the M step, the weights obtained above are fixed, and the likelihood values of the training data are expressed as

$$l(\alpha,t,\Sigma)=\sum_{i=1}^{n}\sum_{z^{(i)}=1}^{K}\log p(x^{(i)},z^{(i)};\alpha,t,\Sigma) \tag{16}$$

$$C(i,j)=\log\frac{\frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}}\exp(-0.5(x^i-t_j)^T\Sigma_j^{t-1}(x^i-t_j))\alpha_j^{(i)}}{\omega_j^{(i)}} \tag{17}$$

Fix parameter α_j, Σi, then calculate the partial derivative of likelihood value to t_i.

$$\nabla_{t_q}=\sum_{i=1}^{n}\omega_q^{(i)}(\Sigma_q^{-1}x^{(i)}-\Sigma_q^{-1}t_q) \tag{18}$$

Accordingly, the updated mean parameter can be obtained by

$$t_q = (\sum_{i=1}^{n} \omega_q^{(i)} x^{(i)}) / (\sum_{i=1}^{n} \omega_q^{(i)}) \tag{19}$$

Finally, we implement the parameter update of the Gaussian mixture model. Then we apply the Gaussian mixture model to gesture image segmentation, and the preliminary results are shown in Fig. 2.

C. Gesture Area Reconstruction

In fact, the mixed Gaussian skin color model usually cannot achieve satisfactory segmentation results directly because of the interference from the complex backgrounds and unstable illumination. So we need to combine some other methods to reconstruct the gesture area.

1. Morphological method. In the two binary image, the edges of the gesture area have different sizes of holes, burrs or incomplete contours. The morphological expansion algorithm can extend the bright color region in the binary image, while the corrosion algorithm can extend the dark region in the binary image. The morphological opening is operated to binary image by expansion and corrosion with the 3×3 all-one matrix, which can remove the isolated noises and bulges of edge in the binary image. The output is shown in Fig. 3.

2. Single connected method. There are still isolated blocks [43] in the gesture image after morphological processing, which are generated by the segmentation of skin color model. The existence of these areas have an adverse effect on the classification results, so we use marker connectivity method to eliminate the small areas. In the Fig. 3, each small block area of binary image is marked as $\{a_i \mid i = 1, 2, \cdots\}$. Then we calculate the area value S_i of each marked region, and get the maximum region according to (5).

$$S_{\max} = \max\{S_i \mid i \in N\} \tag{20}$$

where $S_{\max}$ represent the maximum region area, i.e., the area of gesture region. N is the number of all regions. Then we traverse the entire binary image to determine whether the mark of each pixel belongs to the largest region. The gray value of the pixel belonging to the maximum region is 1, and others are 0. The resulting gesture area is shown in Fig. 4.

3. Reconstruction of gesture area. After a single connected processing of the gesture binary image, the gesture area is complete and the background is flat. We then use the stroke method [44] to extract the edge of gesture and rebuild the gesture color image. Finally, we get the gesture segmentation results, as shown in Fig. 5.

IV. Construction of DCNN

In this section, we introduce the deep convolutional neural network model for hand gesture classification, including the specific structure, loss function, activation function, training method and processes.

A. DCNN Model

DCNN is a multilayer neural network with multiple two-dimensional planes at each layer, consisting of multiple independent neurons. The network consists of convolutional layers, pooling layers, fully connected layers and softmax layer, which ensures that both feature extraction and image classification are performed simultaneously. The network realizes the displacement, scaling and distortion invariance of image information in three ways, that is, local receptive field, weights sharing and down sampling. The local receptive field means that neurons of each layer are connected to a local region of the upper layer. Through the local receptive field, network can extract the primary architectural features of the image, such as corner, color, direction, edge, etc. Weights sharing can reduce the parameters that need to be trained and greatly enhance the generalization ability of the network. There is usually down sampling layers behind convolutional layers, which can reduce the resolution of the feature map and increase the displacement, scaling and distortion invariance of the network. It can make the training of the weight is more conducive to image classification.

In the convolutional layer, the feature map of the previous layer is convoluted with the kernels, and then the results are output through an activation function to get the feature map of this layer. Each output of the feature graph can be correlated with the convolution results of multiple input feature graphs through shared weights. The neurons for each feature map in the convolution layer can be calculated by

$$\boldsymbol{X}_j^l = f(\sum_{i \in M_j} \boldsymbol{X}_i^{l-1} * \boldsymbol{k}_{ij}^l + \boldsymbol{B}_j^l) \tag{21}$$

where l represents the position of the convolution layer in the structure. $\boldsymbol{k}_{ij}$ represent the convolution kernels from j^{th} layer to i^{th} layer. f and $\boldsymbol{B}$ represent activation functions and bias respectively. $\boldsymbol{X}_j$ is the set of input images.

The pooling layer is usually behind the convolutional layer, which performs down-sampling operation on the input feature map. The number of input images is the same as the number of output images after down-sampling, and the dimension of the output image is scaled down. The values of neurons for each feature map in the sampling layer can be calculated by

$$\boldsymbol{X}_j^l = f(\boldsymbol{\beta}_j^l * p(\boldsymbol{x}_j^{l-1}) + \boldsymbol{B}_j^l) \tag{22}$$

where p and $\boldsymbol{\beta}$ represent the down-sampling function and weights, respectively.

The last layer of DCNN is the softmax layer, that is, the network outputs the classification results through the softmax function, which was defined as

$$p(i) = \frac{\exp(\boldsymbol{\theta}_i^T \boldsymbol{x})}{\sum_{k=1}^{K} \boldsymbol{\theta}_k^T \boldsymbol{x}} \tag{23}$$

where $p(i)$ represents the probability that classification result is class i. θ and $\boldsymbol{x}$ are input feature vectors and parameters of fully connected layers, respectively. K is the number of image species.

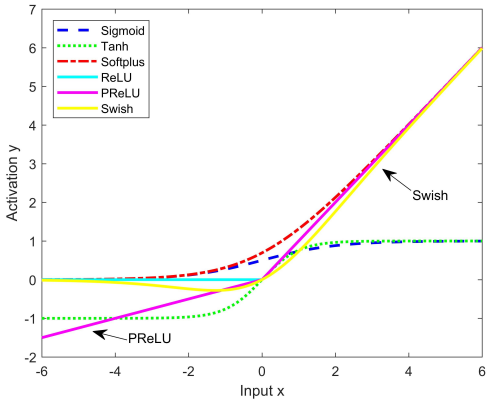


Fig. 6. Curves of various activation functions.

Input (224×224 RGB image)
Conv5-128 Conv5-128 Conv5-128
Maxpooling layer
Conv3-256 Conv3-256 Conv3-256
Maxpooling layer
Conv3-512 Conv3-512 Conv3-512
Maxpooling layer
Fully connected layer-1024 Fully connected layer-512 Fully connected layer-10
Softmax layer

Fig. 7. DCNN structure for hand gesture classification.

B. Loss Function and Network Training Method

The weights training method of deep CNN adopts the batch gradient descent method based on back propagation. Since the last layer of network is the softmax layer, we use the log-likelihood function as the loss function, as shown in (9).

$$J = -\sum_{k} y_k \log(a_k) \tag{24}$$

where a_k and y_k represent the output value and true value of the k neuron respectively. It can be seen from (9) that the higher the output probability of the neuron, the smaller the loss function. The goal of network training is to find the minimum value of the loss function J by continually updating the parameters ω and b. The calculation methods of updating the network weights and bias are given by

$$\omega_{ij}^{l} = \omega_{ij}^{l} - \alpha \frac{\partial J}{\partial \omega_{ij}^{l}} \tag{25}$$

$$b_{ij}^{l} = b_{ij}^{l} - \alpha \frac{\partial J}{\partial b_{ij}^{l}} \tag{26}$$

where

$$\frac{\partial J}{\partial \omega_{ij}^{l}} = \frac{1}{N}\sum_{j=1}^{N} a_j^{L-1}(a_i^{L} - y_i)$$

$$\frac{\partial J}{\partial b_{ij}^{l}} = \frac{1}{N}\sum_{j=1}^{N} a_j - y_j$$

where α is learning rate and N is the number of entered samples per batch. From the above gradient transfer method based on partial differential equation, it can be seen that the softmax function cooperates with log-likelihood function can train the deep CNN very well, and there is no problem that the learning speed is getting slower in the training process.

C. Network Structure used for Gesture Classification

Considering the complex change of gesture with scale transformation, rotation and translation, we need to construct the appropriate deep CNN for gesture image classification. The frame of network consists of convolutional layers and max-pooling layers, and the last layer is the output layer. There are many different feature maps in convolutional layer, and different feature maps are obtained by different convolution kernels, in which each feature map represents a kind of gesture feature. Moreover, compared with the traditional CNN, we increase the number of convolution kernels dramatically in each convolutional layer to improve the utilization ratio of the image information. And the size of kernels in the convolutional layers changes according to the depth of the network. The max-pooling layer usually uses a scaling factor of 2, which is scaled at 2×2. The roughness of features depends on the scaling degree at down-sampling, and the select of convolution kernels of size 2×2 is a good choice. The activation function of output in each layer is Swish function, which has good nonlinear mapping properties, as given in (12).

$$\text{Swish}(x) = x \cdot \text{sigmoid}(x) \tag{27}$$

where

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

where x represents the output of last layer. At present, there are various activation functions in neural network structure, such as ReLU, Tanh, Sigmoid, PReLU, Softplus and so on. As shown in Fig. 6, the output gradients of Swish function increase gradually and are stable near 1 when the inputs are greater than zero. Therefore, the activation function does not have gradient vanishing problems. On the other hand, even if the inputs are less than 0, the outputs are correspondingly small but do not equal to 0. Therefore, there is no neuronal silence phenomenon induced by ReLU activation function in the training process. And the whole network structure is

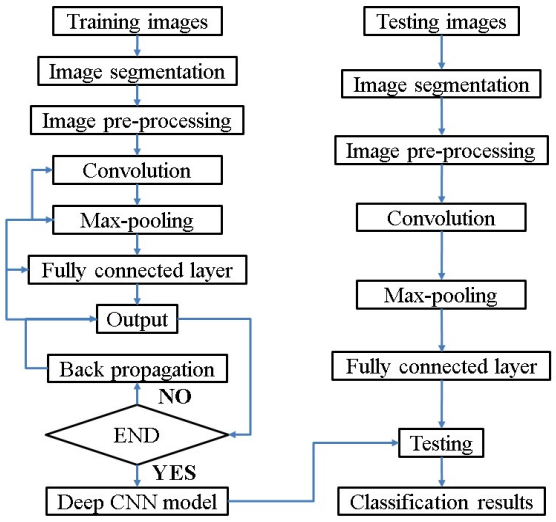
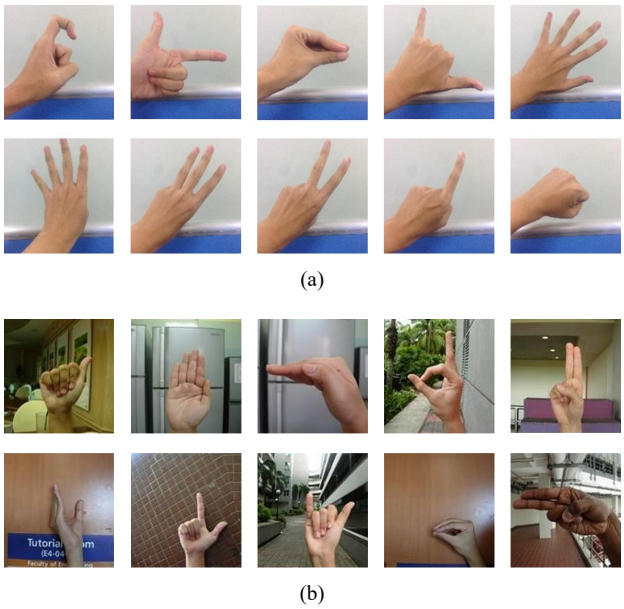


Fig. 8. Training and testing process of Deep CNN.



(a)

(b)

Fig. 9. Different kinds of hand gesture image samples in two databases. (a) shows the samples of the indoor database, and (b) shows the samples of NUS database under the outdoor condition.

sparse and the overfitting problem of deep CNN is greatly improved. The output of neural network is a floating-point vector whose component value of corresponding correct class is 1 and the rest of the values are 0. In the experimental section, we verify and compare the classification results of DCNN under various activation functions to show the effectiveness of the Swish function.

In order to get the optimal number of convolutional layers and feature maps, various DCNNs with different structures are tested in experiments. The results show that when the network is consisting of nine convolutional layers, it has the highest classification accuracy. So we finally use the deep CNN structure as shown in Fig. 7. It can be seen that in the first three convolutional layers of the structure, each layer consists of 128 convolution kernels with size 5×5. In the middle three convolutional layers, each layer consists of 256 convolution kernels of size 3×3. The last three convolutional layers contain 512 kernels of size 3×3 per layer. There is a max-pooling layer behind every three convolutional layers. Network finally connects three fully connected layers and outputs the classification results from the softmax layer.

D. Training Details and Process

In order to retain as much details of image as possible, we pre-process the sample images in the experimental dataset. By interpolating and normalizing data, the network inputs a RGB image with the fixed size of 224×224×3. Another pre-processing we do is subtracting the mean RGB value and computing on the training set from each pixel. Due to the deep layer and too many parameters of deep CNN model, the direct use of small dataset easily leads to overfitting problem. So we first use *ImageNet* dataset [45] to pre-train the network, and then fine-tune it on the target dataset to get the final classification results.

The batch size of the network is determined by the number of classes and samples of the training set, and the detailed parameters are given in the experimental section. During the training process, the three fully connected layers randomly emit neurons with a probability of 40%, which can greatly improve the generalization ability of the network. But in the test phase, the output value of fully connected layers needs to become 2.5 times of the original value. At the same time, in order to avoid the shock effect at the end of training, the learning rate decrease gradually with the increase of training epochs. When the validation set meets the requirements or the loss function is less than the manually set threshold, the training process of the network ends. The training and testing process of the algorithm is shown in Fig. 8.

V. Experiments

The experiments consist of three parts. In the first part, we introduce the two hand gesture datasets used in experiments. In part B, we demonstrate the effectiveness of segmentation algorithm based on mixed Gaussian skin color model. In the part C, we show the classification results and testing speed of DCNN in two datasets to show the effectiveness and analyze the learning process of DCNN through visualized convolution kernels and feature maps.

A. Two Datasets

The datasets in the experiments consist of two parts. The first dataset is the set of hand gesture images in the indoor background, which is used to validate the effectiveness of DCNN model. In this dataset, we collect ten classes of hand gestures through an ordinary camera, as shown in Fig. 9(a). The ten categories represent numbers 0 to 9, respectively. The dataset consists of 3000 images with different rotation angles under the single background. The background of images comes from a light stabilized indoor laboratory.

The second dataset used in the experiment is the standard hand gesture dataset of National University of Singapore (NUS), which has a total number of 3000 images in 10 categories, as shown in Fig. 9(b). All images come from various outdoor background, and the background interference (*e.g.*, similar edges, textures and colors) are deliberately intensified.

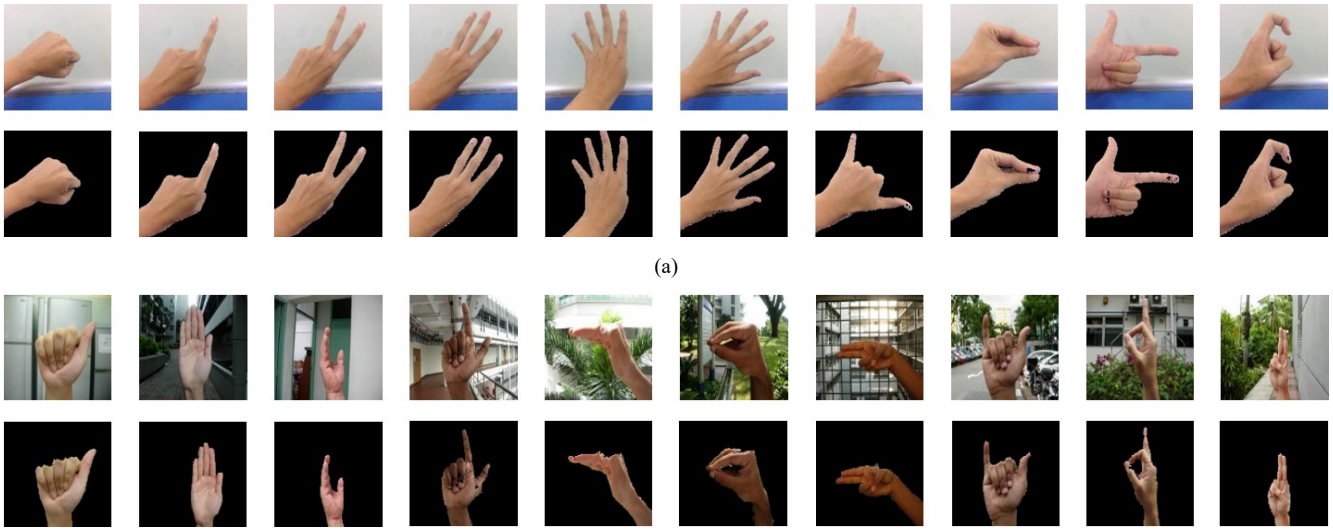


Fig. 10. Segmentation results of gesture samples in two databases. (a) shows the segmentation results of the hand gesture samples under indoor conditions. (b) shows the segmentation results of the hand gesture samples under outdoor conditions.

TABLE I
THE CLASSIFICATION RESULTS OF SAMPLES BEFORE AND AFTER SEGMENTATION IN TWO DATABASES

Indoor database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Original image	92.1	91.3	93.3	88.0	88.2	87.5	90.0	87.8	94.4	94.0	90.7
Segmented image	**100.0**	**98.7**	**98.0**	**98.4**	**100.0**	**97.4**	**99.3**	**98.3**	**100.0**	**99.9**	**99.0**
NUS database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Original image	89.2	88.7	86.3	90.1	88.0	87.6	88.1	89.0	86.9	87.1	88.1
Segmented image	**98.1**	**96.9**	**96.5**	**98.4**	**98.4**	**97.7**	**97.6**	**98.2**	**96.4**	**96.3**	**97.5**

B. Hand Segmentation Results

In this part, we show the segmentation results of hand gesture images in two datasets, as shown in Fig. 10.

It can be seen from Fig. 10(a) that the algorithm can be used to segment the entire hand area in the simple background effectively, which is favorable for subsequent classification. It can avoid the adverse effects caused by complex backgrounds and noises, and use neural networks to extract the most stable and effective features for hand gesture classification.

Fig. 10 (b) shows the segmentation results of hand gesture samples in NUS dataset. Although there are still some pixels that are wrongly segmented in the edge region, the overall segmentation effect is satisfactory. The segmented gesture image has greatly reduced the complex background area in the original image sample.

Since segmentation is not our final goal, we do not compare the experimental results with other segmentation algorithms. We will illustrate the effectiveness of image segmentation algorithm by showing the image classification effect in next section.

C. Hand Gesture Classification Results

In the image input phase of training process, since we use the batch gradient descent method to train the whole network, images of each batch are randomly selected from the dataset. In the training process of DCNN, the training time is relatively long due to the huge size of parameters in the neural network. So the performance of machine has more obvious impact on training time. In the experiments, we accelerated the training process on the GPU. We build the workstation with a NVIDIA GTX 1080, which has a speed of 400 times of CPU i5 6600K.

The classification results compared with baseline. We first compare the classification accuracy of the two datasets before and after segmentation, as shown in Table I. We compared the classification accuracy of ten categories and average classification accuracy (mAP) in two datasets. The highest classification accuracy of each column is bold to highlight. We put the image samples before and after the segmentation into the network respectively. In the training process, the training method and the strategy to prevent overfitting problem are the same. It can be clearly seen that the classification accuracy of the segmented images in the ten categories of two datasets is higher than that of the original images. The average classification accuracy of the segmented hand gesture images in two datasets are 99.0% and 97.5% respectively, while the average classification accuracy of the original hand gesture images in two datasets are only 90.7% and 88.1%respectively. The effective segmentation of hand gesture improves the classification accuracy by about 10%. The complicated background and noise have some adverse effects on the classification of gestures. Then we draw the convergence curves of the network training process on the

TABLE II
THE CLASSIFICATION RESULTS OF DCNN WITH DIFFERENT ACTIVATION FUNCTIONS IN TWO DATABASES

Indoor database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Sigmoid	89.1	86.4	86.0	86.6	90.1	85.9	87.4	86.8	87.9	88.2	87.4
Tanh	90.9	88.7	87.2	89.1	90.2	87.9	90.0	89.9	90.7	91.1	89.6
Softplus	98.2	97.6	96.0	97.1	98.0	96.9	97.4	97.2	98.0	97.8	97.4
ReLU	98.8	97.4	97.1	97.1	98.3	96.9	97.5	97.5	98.5	98.1	97.7
PReLU	98.7	**98.6**	97.8	98.0	99.2	96.6	98.2	**99.0**	99.0	99.2	98.4
Swish	**100.0**	98.5	**98.0**	**98.5**	**100.0**	**97.5**	**99.0**	98.5	**100.0**	**100.0**	**99.0**
NUS database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Sigmoid	88.2	87.3	87.3	88.1	89.0	88.3	87.8	89.2	86.7	85.6	87.8
Tanh	89.1	88.3	87.7	89.3	89.0	88.4	88.7	89.1	87.5	87.0	88.4
Softplus	96.8	95.9	95.7	95.8	96.3	95.6	96.0	97.2	96.3	96.8	96.2
ReLU	95.8	95.6	94.4	95.8	97.0	95.6	96.4	96.6	96.7	96.1	96.0
PReLU	96.3	95.7	**96.7**	96.7	**98.5**	96.4	97.0	97.5	95.2	94.4	96.4
Swish	**98.1**	**96.9**	96.5	**98.4**	98.4	**97.7**	**97.6**	**98.2**	**96.4**	**96.3**	**97.5**

TABLE III
THE CLASSIFICATION RESULTS OF DIFFERENT NEURAL NETWORK MODELS IN TWO DATABASES

Indoor database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Alexnet	96.7	97.5	96.0	95.4	96.6	95.9	95.3	96.1	97.0	97.5	96.4
VGG-16	99.0	98.1	96.5	97.2	98.8	**97.9**	98.6	**98.6**	98.8	98.1	98.2
VGG-19	98.7	97.4	97.0	97.5	99.0	96.2	96.2	97.8	98.0	98.8	97.7
GoogLeNet	98.8	97.1	**98.9**	97.4	98.5	96.5	96.8	97.5	98.1	98.4	97.8
Our model	**100.0**	**98.5**	98.0	**98.5**	**100.0**	97.5	**99.0**	98.5	**100.0**	**100.0**	**99.0**
NUS database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Alexnet	95.2	93.3	92.9	95.1	95.0	94.8	94.1	94.5	91.9	92.2	93.9
VGG-16	94.7	94.4	**96.6**	95.4	95.3	95.0	95.3	**98.4**	94.1	94.0	95.3
VGG-19	96.7	94.2	95.9	93.0	93.1	95.2	95.7	96.0	93.7	**96.6**	95.1
GoogLeNet	**98.5**	96.0	94.7	96.9	93.8	**97.8**	93.0	94.2	93.9	95.4	95.1
Our model	98.1	**96.9**	96.4	**98.4**	**98.4**	**97.8**	**97.6**	98.2	**96.4**	96.3	**97.5**

NUS dataset, as shown in Fig. 11. The red curve represents the training iteration of original images, while the blue curve represents the training iteration of segmented gesture images. It can be seen from the figure that the training process of segmented gesture image converges faster than that of the original images and leads to about 50 epochs advance. The hand gesture image after segmentation greatly reduces the useless information and makes the network more globally convergent at the right location. Moreover, the iteration curve of original images has a large range of oscillations at the end of the training, which is not what we expect to see.

On the basis of segmented hand gesture images, we verify the effect of various activation functions in our DCNN structure, as shown in Table II. In the table, we compared the respective classification accuracy of 10 classes and average classification accuracies under two datasets. The highest classification accuracy of each column is bold to highlight.

It can be seen from the classification results that network with Switsh function has the highest classification accuracy on the hand gestures in the two datasets. The network with PReLU function ranked second in the classification accuracy due to the stable learning property. The problem of deep training of deep neural networks caused by neuronal necrosis in ReLU makes it less accurate than the PReLU. Because of the lack of effective training of the deep neural network, the classification accuracy of network with ReLU function is lower than that of network with PReLU function. The network with Tanh function and Sigmoid function cannot effectively train the deep layer neural network because of the gradient vanishing problem, so the classification accuracy is the lowest.

After determining the activation function, we compare the multiple structure of DCNN and determine the optimal structure for hand gesture classification. In Table III, we compare the classification results of four structures of Alexnet, VGG-16, VGG-19, and GoogLeNet. Based on the Alexnet,

TABLE IV
THE COMPARISON OF CLASSIFICATION RESULTS OF DIFFERENT ALGORITHMS IN TWO DATABASES

Indoor database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Stergiopoulou [46]	96.6	94.0	94.3	94.3	97.2	94.7	95.0	94.4	97.1	96.9	95.5
Jiang [47]	94.5	92.3	93.0	93.5	95.4	92.1	94.3	93.2	93.6	95.2	93.7
Cai [48]	97.0	95.8	96.9	94.6	96.2	94.8	95.5	95.9	96.1	96.7	95.9
Caffe-DAG [49]	98.2	96.5	95.9	95.8	98.0	95.4	96.6	96.2	97.8	98.8	96.9
Triesch [50]	90.4	91.2	89.8	87.9	91.0	90.1	90.8	90.2	91.0	91.6	90.4
DeCAF [51]	98.0	95.5	94.9	**98.8**	96.8	96.7	**99.2**	95.6	97.1	97.4	97.0
Pisharady [52]	97.0	95.7	93.7	**98.8**	95.8	97.7	99.1	96.6	96.2	95.4	96.6
PHOG+SVM	92.2	89.4	91.4	92.1	91.1	91.0	89.5	89.9	90.0	91.2	90.8
SIFT+SVM	92.1	92.7	91.9	91.6	94.0	90.8	92.4	92.2	93.2	92.7	92.4
Hu+GLCM+SVM	95.1	96.4	96.1	95.7	97.8	96.9	96.6	95.9	97.0	97.5	96.5
Hu+GLCM+GRNN	97.2	**98.6**	95.5	95.9	97.4	**97.7**	96.5	96.1	98.0	98.3	97.1
Our model	**100.0**	98.5	**98.0**	98.5	**100.0**	97.5	99.0	**98.5**	**100.0**	**100.0**	**99.0**
NUS database	Class 1 (%)	Class 2 (%)	Class 3 (%)	Class 4 (%)	Class 5 (%)	Class 6 (%)	Class 7 (%)	Class 8 (%)	Class 9 (%)	Class 10 (%)	mAP (%)
Stergiopoulou [46]	95.7	93.8	93.0	95.5	95.1	94.4	94.4	95.6	92.5	92.2	94.2
Jiang [47]	93.1	90.2	90.7	94.1	94.5	92.8	93.0	94.1	91.2	91.7	92.5
Cai [48]	94.6	93.2	93.4	96.0	95.1	94.6	94.3	96.2	92.9	92.5	94.3
Caffe-DAG [49]	94.6	94.9	**97.2**	**98.5**	95.5	96.2	95.0	96.1	95.7	93.5	95.7
Triesch [50]	88.0	87.4	87.2	88.5	89.1	88.4	88.2	89.7	88.0	87.6	88.2
DeCAF [51]	95.2	92.3	92.0	93.7	94.4	93.6	92.2	93.0	91.2	92.3	93.0
Pisharady [52]	93.3	93.9	94.1	96.0	95.4	94.9	94.4	96.5	93.7	91.8	94.4
PHOG+SVM	91.7	91.9	92.2	93.3	94.0	92.6	92.6	94.1	91.7	92.1	92.6
SIFT+SVM	90.1	89.7	89.6	91.2	93.0	90.9	90.1	91.1	88.9	89.6	90.4
Hu+GLCM+SVM	96.0	95.6	94.9	95.5	96.2	95.9	95.7	96.7	**96.6**	94.1	95.7
Hu+GLCM+GRNN	97.7	96.5	94.4	96.7	96.1	96.6	94.5	96.0	94.7	**96.4**	96.0
Our model	**98.1**	**96.9**	96.5	98.4	**98.4**	**97.7**	**97.6**	**98.2**	96.4	96.3	**97.5**

VGGNet increased the depth of the network, making the network has a more outstanding ability to express features. And GoogLeNet increased the width of the network by adding the Inception structure to the network, and the convolution kernel of different sizes at the same convolutional layer enhances the feature extraction ability of the network at different scales. Based on the advantages of both of them, we build the final DCNN framework through a large number of experiments, as shown in Fig. 7.

The classification results compared with some other algorithms. Then we compare the classification accuracy of various algorithms, as shown in Table IV. The recognition time of each algorithm for a single image is shown in Table V.

In terms of classification performance, it can be seen that our algorithm has the highest average classification accuracy. The accuracy of two hand gesture datasets is improved from 97.1% to 99.0% and from 96.0% to 97.5% respectively. The reason is that the weights of multilayer CNN have weight sharing ability, and the network structure has limited weights. Therefore, the network has good generalization ability, and the specific performance is that the accuracy of the test set classification is very close to classification accuracy of the training set. The training process takes full advantage of the self-organizing ability of the gradient descent learning method based on partial differential equation, which can force the network to extract the most effective features of images in different feature spaces. When the traditional single hidden layer neural network is used in gesture classification, the network does not have the feature abstraction capability for the input data. It is necessary to manually select the feature of the target area as the input of the network, and the artificial selection of features is subjective. So it is difficult to establish a complete description for the hand gesture. Stergiopoulou [46] constructs a CNN model to recognize gestures. The inputs of the network are binary gesture images, which can improve the classification efficiency of the network and have better real-time performance. However, the binary images lack the local information of gesture, so that the network does not have the ability of texture description and can only describe features of hand shape. Therefore, its classification accuracy is lower than the method proposed in this paper.

TABLE V
COMPARISON OF TESTING TIME FOR SINGLE IMAGE OF VARIOUS METHODS

Method	Testing time/frame (s)
Stergiopoulou [46]	0.005
Jiang [47]	0.710
Cai [48]	0.120
Caffe-DAG [49]	0.057
Triesch [50]	0.026
DeCAF [51]	0.044
Pisharady [52]	0.026
PHOG+SVM	0.116
SIFT+SVM	0.492
Hu+GLCM+SVM	**0.015**
Hu+GLCM+GRNN	0.027
Our model	**0.034**

TABLE VI
CLASSIFICATION ACCURACIES OF VARIOUS ACTIVATION FUNCTION

Activation fuctions	Classification accuracy of indoor dataset	Classification accuracy of outdoor dataset
sigmoid	80.4%	74.9%
tanh	90.2%	84.2%
softplus	97.1%	95.6%
ReLU	98.4%	96.1%
Leaky-ReLU	98.3%	96.2%
P-ReLU	98.6%	96.6%
maxout	98.8%	97.2%
Swish	**99.0%**	**97.5%**

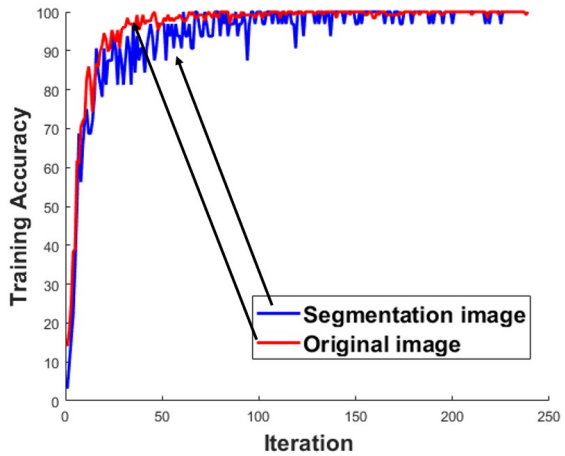


Fig. 11. Convergence curves in the training process of DCNN.

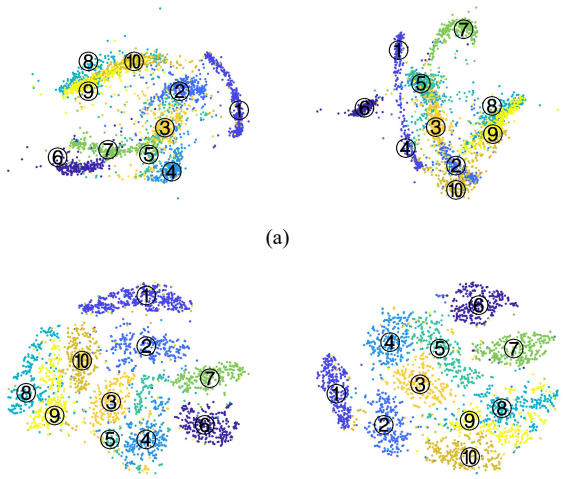
Generalized regression neural network (GRNN) and support vector machine (SVM) are two commonly used methods in image classification, and these two methods have the advantages of fast training and strong nonlinear mapping ability. In the compare experiment, when the main body of gesture is detected, we select the first moment and the second moment in the Hu invariant moment parameter as the global feature to identify the gesture, select the energy, contrast and relevance parameter of gray level co-occurrence matrix as local features to identify gestures. Then we combine the five dimensional features into the GRNN and SVM for training and classification. In traditional feature descriptors, invariant moments can represent the overall shape of objects, and have good invariance for the translation, rotation and scale transformation of objects. Gray level co-occurrence matrix has better texture description ability and can describe detail information of image. However, when the change of gesture is large, the single shape feature cannot be better described, and the texture feature based on the pixel gray value is sensitive to the deformation of the object and the change of the view angle. Jiang [47] used Hu invariant moments to extract the seven parameters of gestures as features and classify them through BP neural network. Cai [48] uses the four custom geometric invariants as a feature to classify gestures. However, these features are not as good as DCNN for the gestures, and the accuracy of these algorithms under the ten categories of gestures is both lower than the accuracy of our DCNN-based algorithms.

In the aspect of recognition efficiency, although the DCNN based approach does not achieve the fastest recognition speed, it meets the real-time requirements in practical application and can process nearly 30 images per second. In the training phase, the property of sparse expression of the activation function makes the scale of the network smaller. In the testing phase, the network automatically extracts the features of gesture image through forward propagation, and the recognition process takes shorter time. The method of Hu+GLCM+SVM method is the fastest, because the feature extraction process of Hu invariant moment is very simple, but the classification accuracy is significantly lower than the DCNN based method.

The comparison with various activation functions. We compared different activation functions on DCNN under two datasets to observe the of Swish function. The classification results are shown in Table VI. Comparing the experimental results of activation functions on two datasets, we can clearly see that Swish achieved the highest classification results. It increased the classification results of the maxout by 0.2% and 0.3%, respectively.

Visualization analysis and understanding of the learning process of DCNN. Confusion matrix is a kind of visualization method of classification results commonly used in supervised learning, which can intuitively express the precision rate and recall rate of classification model. Element $M_{i,j}$ in confusion matrix represents the number of samples in class i that are assigned to class j. The value of the matrix can be normalized between 0 and 1. Based on this ratio, we give each element of the matrix a hue from blue to red. The elements on the main diagonal represent the proportion of samples that are correctly classified. The closer the color of the main diagonal of confusion matrix is to red, the higher the classification rate. We draw confusion matrices for the classification results of two datasets, as shown in Fig. 12 (a) and (b). It can be seen from the figure that only a few samples are misclassified into other categories.

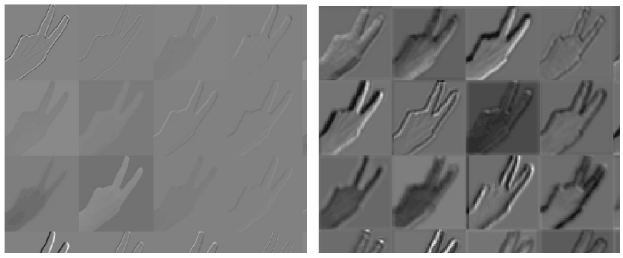
In order to observe the effect of image segmentation operation on the DCNN training process and the convergence results, we use the t-SNE [53] and largevis [54] method to reduce the dimension of the feature vector of the last fully connected layer in DCNN, and show the distribution of the two-dimensional feature in the plane, as shown in Fig. 13. The left image represents the visualization result of t-SNE and the right image represents the visualization result of largevis. In the figure, we show the feature distribution of testing samples in 10 categories of the NUS dataset. It can be clearly seen from the graph that the features of the original image are messy and loose due to the interference of the background.



(a)

(b)

Fig. 13. Distribution visualization of deep convolutional activation features of last fully connected layer. (a) represents the feature distribution of the original gesture images, and (b) represents the feature distribution of the segmented gesture images.



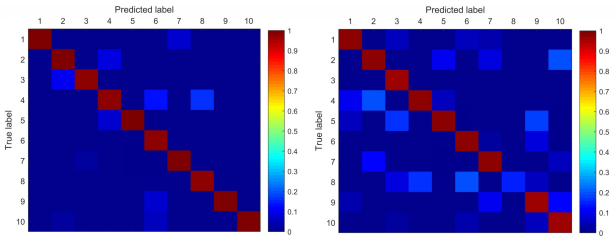
(a) (b)

Fig. 14. The output feature maps of the first and second convolutional layers in deep CNN.

The feature of the segmented image is more compact, which is more conducive to the establishment of a good classification boundary.

In order to analyze and understand the learning process of DCNN, we input the pre-processed gesture images into network, and then observe the feature maps of convolutional layers in network, as shown in Fig.14. In Fig. 14(a), we show some of the feature maps that are strongly activated in the first convolutional layer, while Fig .14(b) shows the feature maps that are strongly activated in the second convolutional layer. The feature map 3, 4, 6, and 12 in Fig. 14(b) extract structural information about the joint or edge of the gesture, which can be understood as the local feature of the gesture. The rest of the feature maps show the contour information of the gesture, which can be understood as global features. In Fig. 14(a), the feature map 6 and the feature map 12 are complementary images, and the same hand gesture regions are represented by different gray levels. By comparing Figs. 14(a) and (b), it can be seen that the feature plane in the upper layer is mapped to different gray levels by convolution operation, and the extracted features of DCNN are light robustness.

In summary, a variety of convolution kernels in the fully trained deep CNN endow it with the ability to extract different features. Since all processing of the subsequent feature maps in the network is based on the output of the previous layer, the network can gradually extract and assemble more distinct and



(a) (b)

Fig. 12. Confusion matrix of classification results in two databases. (a) is the confusion matrix of the classification results of gesture samples in indoor environment, and (b) is the confusion matrix of the classification results of gesture samples in outdoor environment.

stable features. Therefore, it has a comprehensive description of the images and therefore the hand gesture classification accuracy can be improved.

VI. Conclusion

In this paper, we construct a DCNN model for hand gesture classification. We first use the mixed Gaussian skin color model to segment the complete hand gesture area, and then use DCNN to classify hand gestures in different scales, shapes and background environments under two datasets. The back propagation algorithm based on partial differential equation enables DCNN to fully learn the discriminative representation of images. Finally, the model achieves classification accuracy of 99.0% and 97.5% on the two datasets respectively, which shows superiority to other methods. Moreover, experimental results illustrate that the proposed method has the following advantages.

1. The effective segmentation of hand areas in complicated backgrounds can guarantee the effective feature extraction, reducing redundant information and noises, improving the classification accuracy of the DCNN, and accelerating the convergence speed of the training process.

2. In dealing with two-dimensional images, the DCNN can extract the spatial features of images through convolution and down-sampling. It avoids the subjectivity of artificial features, and has translation, scaling and rotation invariance.

3. The DCNN can extract the global and local information of hand gestures comprehensively, so that the description of hand gesture is more complete and accurate.

4. By using the local receptive field and weights sharing technique, the scale of network parameters is greatly reduced and the computational efficiency is improved significantly.

In the future, we plan to build an end-to-end hand gesture recognition system that will synchronize segmentation and classification. And we will consider how to integrate the prior information such as skin color and shape into the training process of the system.

Acknowledgment

The authors appreciate the indoor hand gesture dataset produced by the computer vision and machine learning lab of Shandong University. And the authors wish to gratefully

acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] K. Niyazi, V. Kumar, S. Mahe and S. Vyawahare, "Mouse Simulation Using Two Coloured Tapes," International Journal of Information Sciences and Techniques, vol. 2, no. 2, pp. 57-63, 2012.

[2] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 3, pp. 311-324, 2007.

[3] J. Davis and M. Shah, "Visual Gesture Recognition," *in IEE Proceedings of Vision, Image and Signal Processing*, vol. 141, no. 2, pp. 101-106, 1994.

[4] Z. Lu *et al.*, "A Hand Gesture Recognition Framework and Wearable Gesture-based Interaction Prototype for Mobile Devices," IEEE Transactions on Human-Machine Systems, vol. 44, no. 2, pp. 293-299, 2017.

[5] L. Bretzner, I. Laptev and T. Lindeberg, "Hand Gesture Recognition Using Multi-scale Colour Features, Hierarchical Models and Particle Filtering," *in IEEE 5th International Conference on Automatic Face and Gesture Recognition*, pp. 423-428, 2002.

[6] B. Stenger, "Template-based Hand Pose Recognition Using Multiple Cues," *in Aisian Conference on Computer Vision*, pp. 551-560, 2006.

[7] M. C. Ornellas, "A Deformable Contour Based Approach to Hand Image Segmentation," in Proceedings of First International Conference on Cyber Crime Investigation, pp. 10-18, 2004.

[8] D. Wu *et al.*, "Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition", IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 38, no. 8, pp. 1583-1597, 2016.

[9] A. Ogihara, H. Matsumoto and A. Shiozaki, "Hand Region Extraction by Background Subtraction with Renewable Background for Hand Gesture Recognition," in IEEE International Symposium on Intelligent Signal Processing and Communications, pp. 227-230, 2007.

[10] H. Kenn, F. V. Megen and R. Sugar, "A Glove-based Gesture Interface for Wearable Computing Applications," in VDE International Forum on Applied Wearable Computing, pp. 1-10, 2007.

[11] J. W. Davis and A. F. Bobick, "The Representation and Recognition of Human Movement Using Temporal Templates," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 928-934, 1997.

[12] Y. Liu, Y. Yin and S. Zhang, "Hand Gesture Recognition Based on HU Moments in Interaction of Virtual Reality," in IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 145-148, 2012.

[13] S. Murthy and R. S. Jadon, "Hand Gesture Recognition Using Neural Networks," in IEEE Advance Computing Conference, pp. 134-138, 2010.

[14] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," Engineering Applications of Artificial Intelligence, vol. 22, no. 8, pp. 1141-1158, 2009.

[15] N. Liu, B. C. Lovell and P. J. Kootsookos, "Evaluation of HMM Training Algorithms for Letter Hand Gesture Recognition," in IEEE International Symposium on Signal Processing and Information Technology, pp. 648-651, 2004.

[16] R. Y. Tara, P. I. Santosa and T. B. Adji, "Hand Segmentation From Depth Image Using Anthropometric Approach in Natural Interface Development," International Journal of Scientific and Engineering Research, vol. 3, no. 5, pp. 1-4, 2012.

[17] U. Lee and J. Tanaka, "Hand Controller: Image Manipulation Interface Using Fingertips and Palm Tracking with Kinect Depth Data," in Proceeding-Asia Pacific Conference Computation and Human Interact, pp. 705-706, 2012.

[18] M. Caputo, K. Denker, B. Dums and G. Umlauf, "3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware," Mensch and Computer, vol. 12, no. 1, pp. 293-302, 2012.

[19] X. Wang, "Hand Gesture Recognition Based on BP Neural Network in Complex Background," Computer Applications and Software, vol. 30, no. 3, pp. 247-95, 2013.

[20] G. Marin, F. Dominio and P. Zanuttigh, "Hand Gesture Recognition with Jointly Calibrated Leap Motion and Depth Sensor," Multimedia Tools and Applications, vol. 75, no. 22, pp. 1-25, 2016.

[21] I. Oikonomidis, N. Kyriazis and A. Argyros, "Efficient Model-Based 3D Tracking of Hand Articulations Using Kinect," In British Machine Vision Conference, pp. 1-11, 2011.

[22] P. Gupta, "An Efficient Slap Fingerprint Segmentation and Hand Classification Algorithm," Neurocomputing, vol. 142, no. 1, pp. 464-477, 2014.

[23] X. Cao, J. Zhao and M. Li, "Monocular Vision Gesture Segmentation Based on Skin Color and Motion Detection," Journal of Hunan University, vol. 38, no. 1, pp. 78-83, 2011.

[24] N. Neverova and C. Wolf *et al.*, "Hand Segmentation with Structured Convolutional Learning," in Asian Conference Computer Vision, pp. 687-702, 2015.

[25] X. Zhang, Z. Ye, L. Jin, S. Xu, "A New Writing Experience: Finger Writing in the Air Using a Kinect Sensor," IEEE Transactions on Multimedia, vol. 20, no. 4, pp. 85-93, 2013.

[26] M. V. Akinin, A. I. Taganov, M. B. Nikiforov and A. V. Sokolova, "Image Segmentation Algorithm Based on Self-organized Kohonen's Neural Maps and Tree Pyramidal Segmenter," in IEEE Mediterranean Conference on Embedded Computing, pp. 168-170, 2015.

[27] J. Long，E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440, 2015.

[28] S. Bilal, R. Akmeliawati, M. Salami, A. Shafie and E. E. Bouhabba, "A Hybrid Method Using Haar-like and Skin-color Algorithm for Hand Posture Detection," in IEEE International Conference on Mechatronics and Automation, pp. 934-939, 2010.

[29] Y. She and Q. Wang *et al.*, "A Real-Time Hand Gesture Recognition Approach Based on Motion Features of Feature Points," in IEEE International Conference on Computational Science and Engineering, pp. 1096-1102, 2014.

[30] B. Kaufmann, J. Louchet and E. Lutton, "Hand posture recognition using real-time artificial evolution," in Applications of Evolutionary Computation, pp. 251-260, 2013.

[31] C. Weng, Y. Li, M. Zhang, K. Guo, X. Tang, et al, "Robust Hand Posture Recognition Integrating Multi-cue Hand Tracking," in International Conference on E-Learning and Games, Changchun, China, pp. 497-508, 2010.

[32] M. Flasiński and S. Myśliński, "On the Use of Graph Parsing for Recognition of Isolated Hand Postures of Polish Sign Language," Pattern Recognition, vol. 43, no. 6, pp. 2249-2264, 2010.

[33] H. Ren and G. Xu, "Hand Gesture Recognition Based on Characteristic Curves," Journal of Software, vol. 12, no. 5, pp. 987-993, 2002.

[34] J. Y. Zhu, "Hand Gesture Recognition Based on Structure Analysis," Chinese Journal of Computers, vol. 29, no. 12, pp. 27-32, 2006.

[35] B. Yang *et al.*, "Gesture Recognition in Complex Background Based on Distribution Features of Hand," Journal of Computer-Aided Design and Computer Graphics, vol. 22, no. 10, pp. 1841-1848, 2010.

[36] Z. Li and R. Jarvis, "Real Time Hand Gesture Recognition Using A Range Camera," in Australasian Conference on Robotics and Automation, pp. 21-27, 2009.

[37] C. Vogler and D. Metaxas, "Adapting Hidden Markov Models for ASL Recognition by Using Three-dimensional Computer Vision Methods," in IEEE International Conference on Systems, Man, and Cybernetics, pp. 156-161, 1997.

[38] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[39] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," arXiv preprint, 2014. Arxiv: 1409. 1556.

[40] C. Szegedy, W. Liu and Y. Jia, "Going Deeper with Convolutions," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.

[41] EY. Lam, "Combining Gray World and Retinex Theory for Automatic White Balance in Digital Photography," in IEEE Proceedings of the 9th International Symposium on Consumer Electronics, pp. 134-139, 2005.

[42] A. Amjad, A. Griffiths, M. N. Patwary, "Multiple Face Detection Algorithm Using Colour Skin Modelling," IET Image Processing, vol. 6, no. 8, pp. 1093-1101, 2012.

[43] H. Liu, Y. Zhang, "Curvature Computing of BOF Flame Boundary Based on Differential Chain Code," Computer Engineering and Applications, vol. 49, no. 7, pp. 171-170, 2013.

[44] H. Liu and Zhang. Y, "State Recognition of BOF Based on Flame Image Features and GRNN," Computer Engineering and Applications, vol. 47, no. 6, pp. 7-10, 2011.

[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh *et al.*, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015.

[46] E. Stergiopoulou and N. Papamarkos, "Hand Gesture Recognition Using A Neural Network Shape Fitting Technique," Engineering Applications of Artificial Intelligence, vol. 22, no. 8, pp. 1141-1158, 2009.

[47] J. Li and Q. Ruan, "Research of Gesture Recognition Based on Neuron Networks," Journal of Beijing Jiaotong University, vol. 30, no. 5, pp. 32-36, 2006.
[48] J. Cai, J. Cai, X. Liao, H. Huang and Q. Ding, "Preliminary Study on Hand Gesture Recognition Based on Convolutional Neural Network," Computer system applications, vol. 24, no. 4, pp. 113-117, 2015.
[49] S. Yang, D. Ramanan, "Multi-scale Recognition with DAG-CNNs," in IEEE International Conference on Computer Vision, pp. 1215-1223, 2015.
[50] J. Triesch and D. Malsburg, "A System for Person-Independent Hand Posture Recognition against Complex Backgrounds," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 12, pp. 1449-1453, 2001.
[51] J. Donahue *et al.*, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," in International Conference on International Conference on Machine Learning, pp. 647-655, 2013.
[52] PK. Pisharady, P. Vadakkepat, and PL. Ai, "Attention Based Detection and Recognition of Hand Postures Against Complex Backgrounds," International Journal of Computer Vision, vol. 101, no. 3, pp. 403-419, 2013.
[53] L. Maaten and G. Hinton, "Visualizing Data Using t-SNE," Journal of Machine Learning Research, vol. 9, no .11, pp. 2579-2605, 2008.
[54] J. Tang *et al.*, "Visualizing Large-scale and High-dimensional Data," in Procedings of International Conference on World Wide Web, Montreal, Canada, pp. 287-297, 2016.
[55] Q. Zheng *et al.*, "A Bilinear Multi-Scale Convolutional Neural Network for Fine-grained Object Classification," IAENG International Journal of Computer Science, vol. 45, no. 2, pp. 340-352, 2018.
[56] H. Song, M. Yi, J. Huang, and Y. Pan, "Bernstein Polynomials Method for a Class of Generalized Variable Order Fractional Differential Equations," IAENG International Journal of Applied Mathematics, vol. 46, no. 4, pp. 437-444, 2016.
[57] A. Hambarde, MF. Hashmi, A. Keskar, "Robust Image Authentication Based on HMM and SVM Classifiers", Engineering Letters, vol. 22, no. 4, pp. 183-193, 2014.
[58] Q. Feng, "Jacobi Elliptic Function Solutions For Fractional Partial Differential Equations," IAENG International Journal of Applied Mathematics, vol. 46, no. 1, pp. 121-129, 2016.
[59] Q. Zheng *et al.*, "Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process," IEEE Access, vol. 6, pp. 15844-15869, 2018.

Qinghe Zheng was born in Jining, Shandong, China in 1993. He received his B.S. degree from Xi'an University of Posts and Telecommunications in 2014 and M.S. degree from Shandong University in 2018. Now, he is pursuing his PhD degree in Shandong University. His research direction is computer vision and machine learning.