Visual Hand Gesture Recognition with Convolution Neural Network

Mengmeng Han, Jiajun Chen, Ling Li, Yuchun Chang*
State Key Laboratory on Integrated Optoelectronics
Jilin University
Chang Chun, China
*changyc@jlu.edu.cn

***Abstract*—Hand gestures are a type of communication that is multifaceted in a number of ways and they provide an attractive alternative to the cumbersome interface devices used for human–computer interaction (HCI). However, there are still limitations regarding its usage in unfavorable live situations where hand gestures variation, illumination change or background complexity are an issue. Therefore, this paper propose a convolution neural network (CNN) method to reduce the difficulty of gestures recognition from a camera image. To achieve the robustness performance, the skin model and background subtraction are applied to obtain the training and testing data for the CNN. Since the light condition seriously affects the skin color, we adopt a simple Gaussian skin color model to robustly filter out non-skin colors of an image. In addition, it also employs elastic distortions to obtain lager database for more effective training and reduce potential overfitting. Experimental evaluation achieves an average correct classification rate of 93.8%, which shows the feasibility andreliability of the method.**

Keywords: hand gesture recognition; convolution neural network; background subtraction; human computer interaction

I. INTRODUCTION

In recent years, a number of vision-based applications have been proposed for gesture recognition. The gesture information can provide a rich signal or message for home entertainment and other applications.

Several studies developed various methods or solutions to meet the requirements of various applications.In [1][2], a gesture recognition system was proposed using AdaBoost and Support Vector Machine (SVM) to detect hand location and to recognize hand type, respectively. In object detection[3], the color processing is based on skin color information. In recognition, the k-cosine two procedures of predefined and pre-training in [4] was used. However, the pre-training process and excessive mathematical operation are required which increase the complexity of the system.

Convolutional neural network (CNN) can extract feature from 2D images directly, so that it has been applied to image classification gradually. LeCun et al. [5] first designed CNN and trained it based on the error gradient in 1989. They used CNN to classify the handwritten digits and achieved the best result at that time. Lawrence et al. [6] presented a hybrid neural network solution for face recognition, which combined SOM and CNN. Christophe et al.[7] realized face detection using CNN with three layers, including a convolutional layer, a sampling layer and a MLP layer. In 2012, Hinton et al. [8] trained a large and deep CNN and achieved unprecedented success in the ImageNet contest. So far, it has already achieved great success in facial point detection [9], pedestrian detection [10], human attribute inference [11], image quality assessment [12], image classification [13], and video classification [14]. However, there are few literature about gesture recognition using convolution neural network.

Therefore, this paper proposed a novel framework of gesture classification using a convolutional neural network. The convolutional neural network is a multi-layer feed-forward neural network which is biologically inspired. Unlike traditional methods by using hand-crafted features, the convolutional neural network is able to automatically learn multiple stages of invariant features for the specific task. Hand gesture recognition in natural gray-scale image is a extremely difficult task. Hence, we divide the recognition task into two easier ones as shown in Fig.1. Considering light and backgrounds may seriously affect the recognition accuracies, especially the light condition, Gaussian skin color model and background subtraction are used to performe better accuracy. Background subtraction is utilized to remove the interference close to skin color in an image. And then binary image is obtained by filtering out non-skin colors of an image using Gaussian skin model.

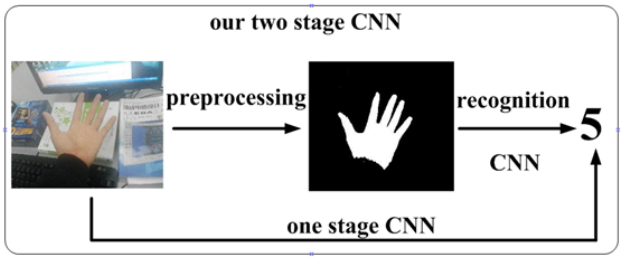

Fig.1 System diagram of our method.

The remainder of this paper is organized as follows. Section II explains the architecture of proposed framework. Parameters learning is described in Section III. Experiment results and its analysis are presented in Section IV.

This work was supported by the National Natural Science Foundation of China (No.61274023), the New Century Excellent Talents Support Program of the Ministry of Education (No. NCET-12-0236) and the Development and Reform Commission of Jilin Province (No. 2015Y041).

978-1-5090-2239-7/16/$31.00 ©2016 IEEE

II. Architecture of Proposed Framework

The convolutional neural network in our method takes an original gesture image as the input and outputs the probability of each gesture type to which the gesture belongs. Framework is shown in Fig.2. The network contains two stages, the first stage is used to preprocess the input gesture images and the second stage with CNN is used to recognize which gesture type the input image belongs to. The second stage consists of the convolutional layer, the average pooling layer, and the subsampling layer. In which, the convolutional layer computes the convolutions between the input and a set of filters (filter bank), and provides a nonlinear representation of the input signal by using a point-wise nonlinear function. The average pooling and subsampling layer reduces the spatial resolution of the representation to achieve the robustness to both geometric distortions and small shifts.

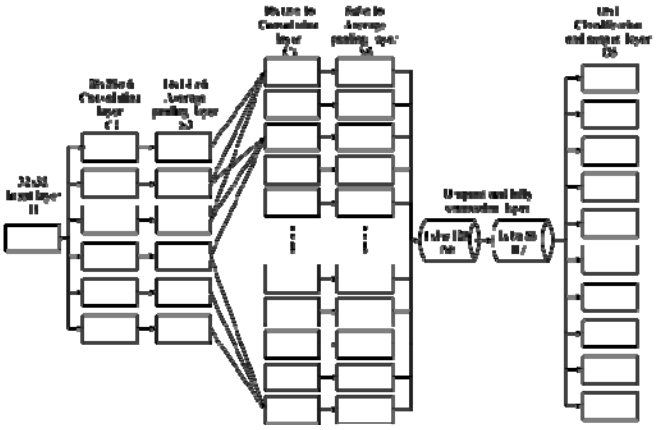

Fig. 2. The architecture of our convolutional neural network.

A.Convolutional Layer

In the convolutional layer, convolutions between the input and a series of filters are first computed. A nonlinear activation function is then executed on the convolutions. The layer provides a non-linear mapping from the low level image representation to high level semantic understanding, which simulates the "simple cells" in the standard models of the visual cortex. A convolutional layer is parametrized by: the number of maps, the size of the maps and kernel sizes. The input x_i is the i^{th} 2D feature map with the size of $s1 \times s2$. The output y is a 2D array whose size is $t1 \times t2$, and y_j is defined as the j^{th} 2D feature map of the output. The softplus function s $softplus(\cdot)$ is chosen as the nonlinear activation function, which is detailed in the Section III-A. Hence, y_j is computed by

$$y_j = softplus\,(W_i^T \otimes x_i + b_i) \tag{1}$$

where $\otimes$ denotes the convolution operation, b_i is the bias and W_i^T is a 2D filter learned by the error back propagation described in Section III-B.

B.Average Pooling and Subsampling Layers

The pooling and subsampling layers aim to make the representation robust to both geometric distortions and small shifts. The pooling layer performs average pooling on the results of the convolution layer. Average pooling takes regions of an image and reduces them to a single value by taking the mean of all values in this region. In this implementation, non-overlapping regions of 2x2 pixels were used for pooling. The subsampling procedure is performed on the output of the average pooling layer to reduce the feature size.

C.Dropout and Fully Connection Layer

Dropout [15] regularizes neural network training by preventing co-adaptation of model parameters, thus reducing overfitting with limited training data. To be more specific, let hl be the activation vector in the l-th hidden layer of the neural network. A binary mask ml is generated for each training case, and the forward pass for the dropout training becomes

$$y_j = m_i * softplus\,(W_i^T \otimes x_i + b_i) \tag{2}$$

The $*$ denotes element-wise multiplication. The elements of m_i obey the distribution $Bernoulli(1-r)$, where r is often referred to as the dropout rate. No neuron is dropped out for the recognition phase, but the activations are scaled by $1-r$ to compensate for the dropout training.

D.Softmax Classification Layer

The results of the fully connected layer get passed into a softmax classifier. As shown in Fig. 2, the input of the softmax classification layer is the feature vector learned by previous layers, and the output is the type probability vector. A linear function is applied to model the relationship between the feature and the probability distribution of the hand gestures.

$$v = W^T \cdot x + b \tag{3}$$

where $x \in R^{D\times 1}$ represents the input feature, $v \in R^{C\times 1}$ is a intermediate variable for describing the distribution, and C is the number of hand gesture types. Because the probability has the properties of nonnegativity and unitarity, y is normalized as

$$y_i = \frac{1}{V} e^{v_i}, i = 1,2,\cdots,C \tag{4}$$

$$V = \sum_{i=1}^{C} e^{v_i} \tag{5}$$

where v_i is the i^{th} element of v, and $y = [y_1, y_2, \cdots, y_C]^T$ is the output of the softmax classifier laye

III. Network Parameters Learning

A.Activation Function

All along, nonlinear sigmoid is commonly used in traditional neural networks for activation function.Sigmoid

function has a good effect in the signal feature space mapping due to a feature of monotonous increment both itself and its inverse function. 2001, neuroscientist Dayan and Abbott simulated a more precise neuron activation model when receiving signals from a biological point of view[16], and the model is shown in Fig.3. Attention is needed in the figure, state in the front of red box did not activate.

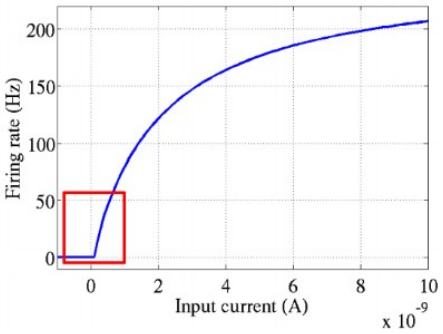

Fig.3 Precise neuron activation model.

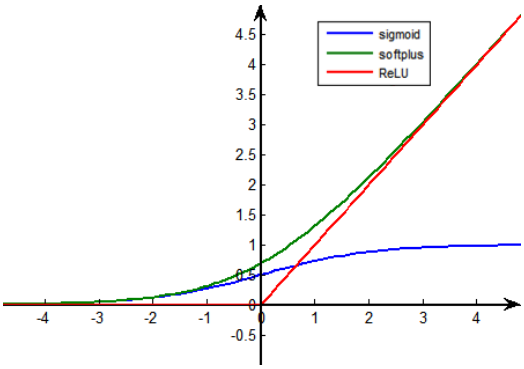

Fig.4 Graphs of the softplus, ReLU and sigmoid.

When doing positive regression prediction, Charles Dugas et al.[17] occasionally utilizesd softplus function for activation function. It is found that softplus function can greatly shorten the learning period of network while improving the prediction accuracy of the data. Except softplus function，ReLU (a forced non-negative correction function) is closer to neuronal frequency activation function. Graphs of the softplus, ReLU and sigmoid is shown in Fig.4.

Compared to the Sigmoid, Softplus and ReLu are closer to neuron frequency activation function. ReLu is the most similar function to neuron frequency activation model, however, it can not be used for back-propagation of errors for it is not derivative. To compromise, this paper choose smoother softplus as the activation function.

B.Parameters Learning

All parameters W such as the kernel elements, the weights between the units and the bias, are randomly initialized. We use backpropagation to update the parameters W_{t+1} in iteration $t+1$ as defined in

$$W_{t+1} = W_t - \alpha_{t+1} \frac{\partial L}{\partial W} \tag{6}$$

The update efficiency will decrease depending on the update time t as defined in

$$\alpha_t = \frac{\alpha_0 \tau}{\max(t, \tau)} \tag{7}$$

The α_0 is initially update efficient and τ is the defined parameter. We use the $L2$ norm for the loss function L. However, softmax is used for the loss function L in the classification layer learning.

IV. Experiments and Analysis

A.Datasets

A complex and challenging hand gesture dataset including 10 gestures was involved in to test the recognition accuracy of our method. Original hand gesture images were collected from 30 participants and 10 gestures in 4 directions under 4 backgrounds with illumination changes. We use the 4800 images as seed and obtain 12000 images after elastic distortions. 10 gestures are shown in Fig.5, and gesture in 4 directions as well as 4 backgrounds in Fig.6.

Fig.5 10 gestures in evaluation dataset.

Fig.6 Gesture in 4 directions and 4 backgrounds with illumination changes.

B. First Stage Processing

The framework of proposed system is divided into two stages. The first stage is utilized to preprocess the original input gesture images. Two steps is needed in the first stage: background subtraction and binarization based on skin color.

There is inevitable interference close to skin color in the collected gesture images. Hence, background subtraction is of vital importance for Gaussian mixture model binarization. A relatively simple method is used in this paper. The average of 6 images containing background is regarded as the real background. Image that contains only a gesture can be obtained by subtraction between image to be processed and real background to eliminate interference.

Considering the time-consuming and efficiency of recognition algorithm, a simple Gaussian model based on skin color is adopted in this paper for binarization process. By comparing clustering analysis in different color spaces, skin color clustering in the YCbCr space is more compact and easier to be implemented. Hence YCbCr color model is used to separate “skin area” from “non-skin area”. Fig.7 shows the preprocessing results of original input gestures.

Fig.7 Preprocessing of original input gestures.

C.Secong Stage Recognition

Compared to previous TECHNOLOGIES using complicated image feature extraction, the CNN provides a robust and systematic methodology to classify the type of hand gestures. Fig.2 shows the architecture of the CNN. Dataset consists of 12000 pictures, in which 10000 pictures are used for training the neural network and 2000 image for testing recognition accuracy of the CNN. The total iteration of updating parameters in convolutional neural network is 12000 with 10 mini-batches.

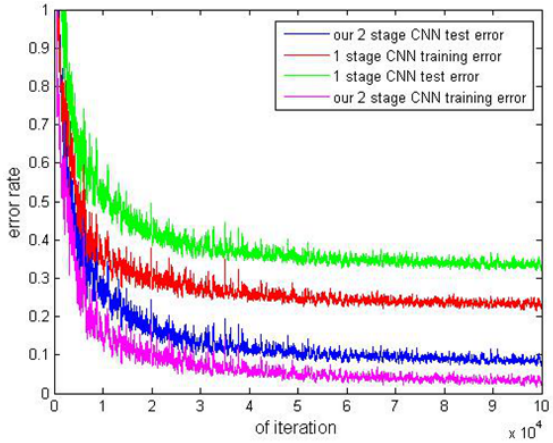

Fig. 8. Error rate for the iterations of each initial update coefficient.

We compare our two stage CNN with the one stage CNN that has not preprocessing layer. The performance results of our two stage CNN with the one stage CNN are shown in Fig.8. The curves show the network does not overfit even after a very large number of epochs, which proves that we have adopted robust structure with stable performance.

Fig. 9 show the accuracy confusion matrices uses ten-fold cross validation based on our two stage CNN. Each cell in the matrix is computed as the average of the corresponding values of all confusion matrices. The low standard deviations are invisible here, as most values are either zero or quite low. In the best case we achieved an accuracy of 99.6% for gesture 8 due to no similar gestures. While in the worst case we only achieved an accuracy of 88.3% for gesture 5. There may be finger blockings resulting in misjudging of finger numbers which leads to a lower recognition rate for gestures with more fingers.

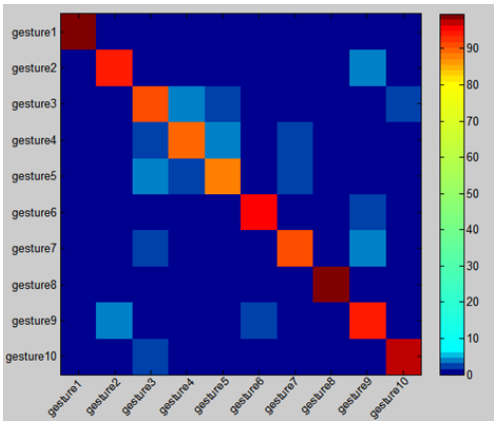

Fig.9 Confusion matrices with ten-fold cross validation based on our two stage CNN.

V. CONCLUSION

In this paper, we presented a two stage convolutional neural network with perprocessing layer to reduce the complexity of the recognition task. The salient feature of the system is that there is no need to build a model for every gesture using hand features such as fingertips and contours. Background subtraction is used in preprocessing stage to exclude interference close to skin color and simple Gaussian skin color model is used to detect skin color and input image is inverted to binary one. After preprocessing, redundant information in gesture images is reduced and process of background information is omitted, which greatly helps to reduce training time of CNN. Then, a CNN was trained to learn 10 gestures in this paper. In the experiments, we conducted ten-fold cross-validation on the system where 10000 and 2000 images were used to train and test respectively. The results showed that the average recognition rates of the 10 gestures were around 93.8% which demonstrates the effectiveness of the proposed method.

REFERENCES

[1] Y. Liu, P. Zhang, “Vision-based Human-Computer System using Hand Gestures”, In Proceeding(s) of the International Conference on Computational Intelligence and Security, pp. 529-532, 2009.

[2] Jane J. Stephan, Sana'a Khudayer, “Gesture Recognition for Human-Computer Interaction (HCI)”, International Journal of Advancements in Computing Technology(IJACT), vol. 2, no. 4, pp. 30-35, 2010.

[3] Tao Ji, Wencheng Wang, “A Fast Face Detection Method Based on Skin Color”, Journal of Convergence Information Technology(JCIT), vol. 6, no. 9, pp.50-58, 2011.

[4] T. Sun, “K-Cosine Corner Detection”, Journal of Comp., vol. 3, no. 7, pp. 16-22, 2008.

[5] LeCun Y, Boser B, Denker JS et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541 – 551

[6] Nebauer C (1998) Evaluation of convolutional neural networks for visual recognition. IEEE Trans Neural Netw 9(4):685 – 696

[7] Garcia C, DelakisM(2002) A neural architecture for fast and robust face detection 2002. In: Proceedings 16th international conference on pattern recognition, IEEE, vol 2, pp 44 – 47

[8] Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp 1097 – 1105

[9] Y. Sun, X. Wang, and X. Tang, “Deep convolutional network cascade for facial point detection,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog., 2013, pp. 3476–3483.
[10] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning,” in Proc. IEEE CVPR, 2013, pp. 3626–3633.
[11] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, “PANDA: Pose aligned networks for deep attribute modeling,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog., 2014, pp. 1637–1644.
[12] L. Kang, P. Ye, Y. Li, and D. Doermann, “Convolutional neural networks for no-reference image quality assessment,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog., 2014, pp. 1733–1740.
[13] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1106–1114.
[14] A. Karpathy et al., “Large-scale video classification with convolutional neural networks,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog., 2014, pp. 1725–1732.
[15] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing coadaptation of feature detectors,” arXiv:1207.0580, 2012.
[16] Dayan, Peter, and Laurence F. Abbott. Theoretical neuroscience. Vol. 806. Cambridge, MA: MIT Press, 2001.
[17] Dugas C, Bengio Y, Bélisle F, et al. Incorporating second-order functional knowledge for better option pricing[J]. Advances in Neural Information Processing Systems, 2001: 472-478.