Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 1, January 2018, pp. 89-100
DOI: 10.23940/ijpe.18.01.p10.89100

# An Intelligent Discovery and Error Correction Algorithm for Misunderstanding Gesture based on Probabilistic Statistics Model

Kaiyun Sun[a], Zhiquan Feng[a,*], Changsheng Ai[a], Yingjun Li[a], Jun Wei[a], Xiaohui Yang[a], Xiaopei Guo[a], Hong Liu[b], Yanbin Han[a,b], Yongguo Zhao[c]

[a]*School of Information Science and Engineering, University of Jinan, Jinan, 250022, China*
[b]*School of Information Science and Engineering, Shandong Normal University, Jinan, 250014, China*
[c]*Institute of Automation Shandong Academy of Sciences, Jinan, 250022, China*

**Abstract**

Numerous experiments have shown that there are similar gestures in visual-based gesture recognition. In order to solve the problem, this paper proposes a new algorithm based on Convolutional Neural Network. According to the model test results, the confusion matrix is established. According to the correspondence between each gesture and the predicted result, probability matrix of misjudgement is established. Based on the probability matrix of misjudgement, we correct the gestures that have been incorrectly identified by the Convolutional Neural Network Model. After this algorithm, the recognition rate of similar gestures is increased from 5% to 12%. The innovation of this paper lies in the secondary error correction of the wrong gesture of Convolutional Neural Network structure.

*Keywords*: convolutional neural network; probability matrix of misjudgement; automatic error correction algorithm

(Submitted on October 20, 2017; Revised on November 23, 2017; Accepted on December 7, 2017)

© 2018 Totem Publisher, Inc. All rights reserved.

## 1. Introduction

Human-computer interaction refers to the specific language between people and computer, carries on a series of inter-operation, and realizes the process of information exchange between man and computer [3,22]. The ways of traditional human-computer interaction include mouse, keyboard, and scanner etc. [6]. People as a passive role need to understand how to use these interactively. An important research direction in modern interactions is to interact with computers by sensing organs.

Gesture is a special language and an important means of human interaction with the outside world [51]. It transmits the gesture information mainly by gesture action and vision. In addition, in human language, gesture language and natural language contain the most and richest amount of information [28]. Different gesture shapes and various trajectories can represent a large amount of semantics [21,29] so gestures can be viewed as a natural, intuitive way of interacting. However, in the process of natural interaction, if the recognition rate of gestures is low, it will increase the user's load. Then, this interactive mode will gradually be abandoned by the user. Therefore, it is imperative to improve the natural gesture recognition rate. According to the different input mode, it can be divided into gesture recognition based on wearable equipment and gesture recognition based on machine vision.

The digital glove is a typical gesture input device, which can accurately obtain the position of the user's gesture and the direction of the finger. Because the information obtained by the digital glove is very accurate, the data glove can get a good recognition effect. Xiao Ling [47] proposed a gesture recognition method based on three-axis accelerometer. In this paper, the recognition problem was transformed into the sparse representation problem between samples. Zheng huan [53] proposed a method of identifying dynamic gesture sequences using continuous data flow. Xie Renqiang presents an

* Corresponding author.
*E-mail address*: ise_fengzq@ujn.edu.cn

accelerometer-based smart ring and a similarity matching-based extensible hand gesture recognition algorithm. Users can wear the ring to perform gestures in 2-D space. Although the wearable equipment has high accuracy, it needs people to wear certain equipment, which affects people's freedom of movement and has certain limitations. This limitation also limits its development [4].

The gesture recognition based on vision makes up for the disadvantage of wearing style. For static gestures, Haitham proposes a method for gesture recognition using neural networks. In this method, the geometric moment feature and contour feature of the sample are extracted by edge detection, and then the gesture is identified by neural network. However, the method has lower recognition for static gestures. For static gestures, Haitham [11] proposes a method for gesture recognition using neural networks. In this method, the geometric moment feature and contour feature of the sample are extracted by edge detection, and then the gesture is identified by neural network. However, the method has lower recognition for static gestures. Li proposed a gesture recognition method using Hierarchical Elastic Map Matching (HEGM). The Boosting algorithm was used to determine the hierarchical structure of a given graphic. The HOG was used to extract the visual features [18] Padam proposed a static gesture recognition algorithm based on geometric normalization and Krawtchouk moments. The hand was extracted from the hand and forearm regions according to the gesture measurement method. Then, the hand contour was normalized by Krawtchouk moments feature and minimum distance classifier. The method can recognize the small training sample sets well [29]. The paper [50] proposes a recognition method based on Hidden Markov Model (HMM). HMM has a strong ability to describe gesture station-temporal signal changes. However, HMM model needs to calculate a large number of state probability density and needs to estimate multiple parameters, making the recognition slower [46]. The document [54] proposes a method of extracting the gesture edge feature pixels, and using the Hausdorff distance template matching to achieve the Chinese sign language recognition. The advantage of this method is that the calculation is small and the adaptability is strong. The disadvantage is that the effect of gesture rotation, scaling, and skin tone on gesture recognition is not considered [54]. The paper [46] proposes a gesture recognition method combining skin color model and Convolutional Neural Network, which effectively avoid the subjectivity and limitation of artificial feature extraction and improves the recognition rate, but it does not apply to similar gestures.

By consulting the reference literature, we found that algorithms for gesture recognition are already quite mature. However, there are few algorithms for automatically identifying errors and for implicit error correction. In addition, experiments show that some gestures through the convolutional model will be confused. To resolve this problem, we propose a new algorithm. According to the result of gesture recognition of Convolutional Neural Network, the probability statistical model is established. The error correction algorithm is proposed to optimize the Convolutional Neural Network. Experiments show that this algorithm does improve the gesture recognition rate compared with the Convolutional Neural Network algorithm.

## 2. Convolutional Neural Network model for gesture

### *2.1. Introduction to Convolutional Neural Network model*

Convolutional Neural Network is a neural network that is improved and optimized based on BP neural network [5,16,40]. Compared with BP neural network, its particularity is reflected in two aspects [1]. On one hand, its neuron's connection is not all connected. On the other hand, the weight in the same layer is shared. These two aspects can reduce the complexity of the network model and greatly reduce the number of network training parameters [2,15,34]. Therefore Convolutional Neural Network is more similar to biological neural networks.

In a convolutional layer, the feature map of the upper layer is convoluted by an observable convolutional kernel [13,42]. Through an activation function, we can get this feature map. The formula (1) is expressed as follows:

$$Y_j^l = f\left(\sum_{i\in M_j} Y_i^{l-1} + b_j^l\right) \tag{1}$$

Where $M_j$ represents the upper map set, $f(x)$ represents the activation function, $K^l_{ij}$ represents the learned convolutional kernel, b represents the bias, and l represents the current convolutional layer.

Sub-sampling based on the image local correlation principle image can reduce the amount of data processing while preserving useful information [14]. The neurons of each feature map in the sampling layer can be calculated according to equation (2) [43,44].

$$Y_j^l = f\left(\beta_j^l p\left(Y_j^{l-1}\right) + B_j^l\right) \tag{2}$$

In the formula, $\rho$ is a descending function, and the down sampling function is generally a weighted sum of the regions of the input image $n \times n$. $f(x)$ is the activation function. And each output feature has a bias $B$ and a weighting factor $\beta$ [23,25].

The weight training method uses the back propagation algorithm. We assume that the output of the j time neuron in the n times iteration is $y_j(n)$, then the error signal of the neuron is given as:

$$e_j(n) = d_j(n) - y_j(n) \tag{3}$$

We define the square error of the unit $j$ as $1/2e_j^2(n)$. Then the instantaneous value of the total squared error at the output can be expressed by the formula (4).

$$E = \frac{1}{2}\sum_{j=c} e_j^2(n) \tag{4}$$

And $c$ contains all the output cells.

Assuming that the total number of samples in the training set is $N$, the mean of the squared errors can be expressed as :

$$E_{AV} = \sum_{n=1}^{N} E(n) \tag{5}$$

In the formula, $E_{AV}$ is the function of all weights and thresholds as well as the input signal, the objective function. The training network aims is to minimize the $AV$.

*2.2. Convolutional Neural Network model reconstruction for gestures*

The frame of the network consists of the Convolutional layer and the sampling layer alternately [26]. The Convolutional layer has a plurality of different feature maps. The number of feature maps is determined by the Convolutional kernel [17,33]. The network model uses 5 layers of Convolution, 3 layers of pooling, and 2 layers of full-connection. The Convolutional kernel sizes from top to bottom are 11 *11, 5 * 5, 3 * 3, 3 * 3, and 3 * 3. The training model parameters are set as follows: training batch_size is 256; display is 50; max_iter is 200000; momentum is 0.9; weight_decay is 0.0005; snapshot is 1000. Figure 1 shows the Convolutional Neural Network model[16,32, 39]. Training flowchart is shown in Figure 2 (a).

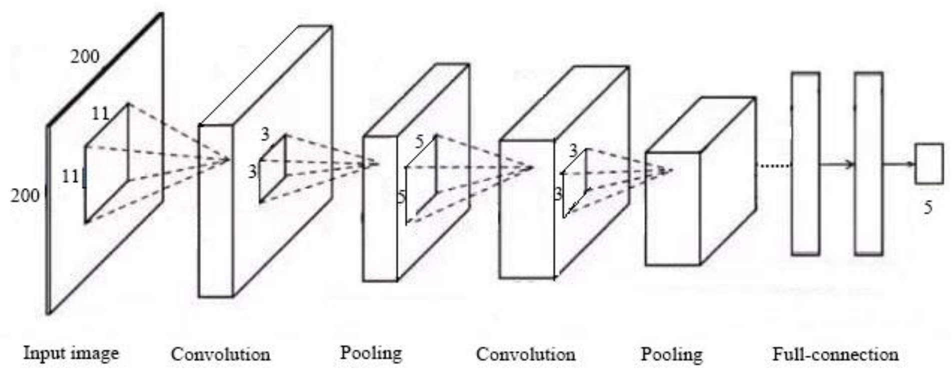


Figure 1. The Convolutional Neural Network model in this paper

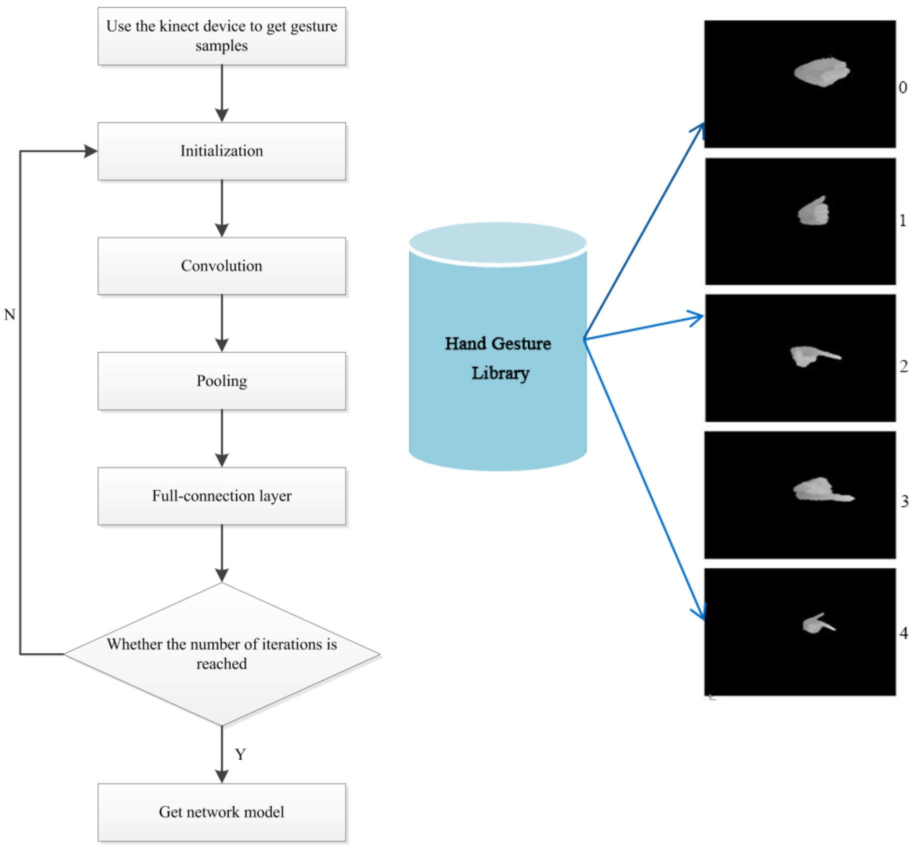
*2.3. The process of model testing*

The database used in this article is the gesture collected in the indoor scene. Five different gestures were collected by Kinect 2.0 [19,27,36,37,41]. The number of each sample is about 50,000. These gestures are taken from 600 students. In order to increase the sample's inclusiveness, we also require experimenters to make different rotation angles when collecting the same type of gesture. The gesture category is defined as shown Figure 2 (b).

In visual studio 2013, we call the trained Convolutional Neural Network model to test the new data, which includes 3000 samples per class. During the test process, we need the following files [25,35]:

- Tag file [48]
- New database
- Network model
- Network architecture documentation, corresponding to the training network structure
- Classifier

The information and format of the output are set in the classifier [9]. The last output information used in this paper is the probability of each image belonging to each category. The sum of the total probabilities is 1. Further, the convolutional process is observed by using Python visualization features [45].




(a) (b)

Figure 2. (a) The Training flowchart (b) Figure Gesture category definition

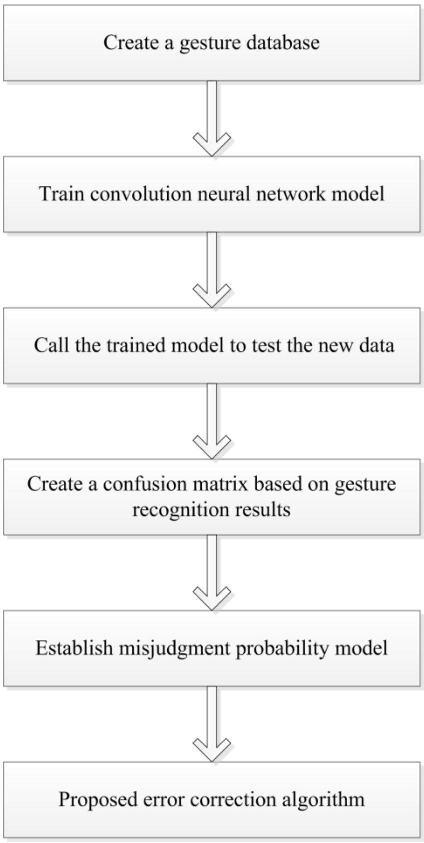
## 3. Establishing the probability model of misjudgement and proposing an error correction algorithm

For similar gestures, the Convolutional Neural Network model cannot distinguish them. Therefore, this paper proposes to reconstruct the recognition result of CNN by constructing the probability statistical model. The basic idea of algorithm is shown in Figure 3 (a).

### *3.1. Create a confusing matrix*

In machine learning (in the field of artificial intelligence), confusion matrices are visual tools, especially for supervised learning. Confusion matrices are generally called matching matrices in unsupervised learning. The columns of the matrix represent examples of prediction classes, and the rows represent examples of actual classes. So, the accuracy of the algorithm can be gauged by some of the metrics in the confusing matrix. The confusion matrix is also known as the possibility table or the error matrix. Each row represents the predicted value, and each row represents the actual category. It can be very easy to show whether categories are confused (that is, a class is predicted to be another class). Each column of the confusion matrix represents a prediction category, with the total number of each column representing the number of data predicted for that category; each row represents the true ownership of the data. The total number of each row indicates the true number of data for that category. The values in each column indicate that the real data is predicted as the number of that class. In short, the confusion matrix is an N × N table that shows the prediction of an N-ary classifier.

For Convolutional Neural Network model test results, we use the simple classification algorithm processing data to achieve the same test results. Then, we can conveniently count the number of each category. Finally, we establish the following confusion matrix shown in Figure 3(b). The sum of each line is 3000, representing 3000 samples. The first line shows that 2958 of the 3000 samples of class 0 are correctly classified. 11 of them are misclassified into 1, 16 of them are misclassified into 2, and 15 are misclassified into class 3. By analyzing the figure, we can get Category 2 easily confused with Category 1. Category 1 and Category 3 is easily confused into Category 0. We call them as similar gesture.




(a)

| | | Predicted class | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 |
| Actual class | 0 | 2958 | 11 | 16 | 15 | 0 |
| | 1 | 1458 | 1419 | 12 | 111 | 0 |
| | 2 | 329 | 1255 | 1049 | 323 | 44 |
| | 3 | 1016 | 687 | 40 | 1257 | 0 |
| | 4 | 58 | 381 | 652 | 582 | 1328 |

(b)

Figure 3. (a) The algorithm flow of this paper (b) Confusion matrix

### *3.2. Establishing Misjudgement Probability Model*

According to the confusion matrix, there are some gestures that can easily be misjudged as other types of gestures [10]. In order to correct wrong gesture, we propose a misjudgment probability model. The probabilistic relationship between the predicted result and the actual class are obtained by analyzing the above recognition results.

Use $i$ to indicate the predicted label number. Use $j$ to indicate the actual label number. $P_{ij}$ represents the probability that class $j$ is misjudged $i$ class. $a_{ij}$ represents that quantity that class $j$ is misjudged $i$ class. Equation (6) represents $P_{ij}$:

$$P_{ij} = \frac{a_{ij}}{\sum_{j=0}^{4} a_{ij}} \tag{6}$$

The value range of j is 0 ~4.

After the above formula, the misjudgment probability matrix is established. According to Table 1, if the predicted number of a picture is 0, the probability of its source from 0, 1, 2, 3, and 4 is 0.51, 0.25, 0.06, 0.17, and 0.001. Based on the above matrix, we can calculate the probability that convolutional neural network test results come from each class. Table 1 describes in detail the probability of each category. In addition, we also analyze the confusion of gestures from a probabilistic perspective. The establishment of misjudgment probability matrix lays a foundation for the error correction algorithm.

Table 1. Misjudgement probability matrix

| category | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0.51 | 0.00 | 0.01 | 0.01 | 0.00 |
| 1 | 0.25 | 0.39 | 0.01 | 0.04 | 0.00 |
| 2 | 0.06 | 0.33 | 0.59 | 0.14 | 0.03 |
| 3 | 0.17 | 0.18 | 0.02 | 0.56 | 0.00 |
| 4 | 0.001 | 0.1 | 0.37 | 0.25 | 0.97 |

Table 2. Category 0 corresponding relationship

| random number | category |
|---|---|
| 1~59 | 1 |
| 60~100 | 3 |

Table 3. Category 1 corresponding relationship

| random number | category |
|---|---|
| 1~42 | 1 |
| 43~79 | 2 |
| 80~100 | 3 |

Table 4. Category 2 corresponding relationship

| random number | category |
|---|---|
| 1~62 | 2 |
| 63~100 | 4 |

Table 5. Category 3 corresponding relationship

| random number | category |
|---|---|
| 1~68 | 3 |
| 69~100 | 4 |

Table 6. Category 4 corresponding relationship

| random number | category |
|---|---|
| 1~97 | 4 |
| 98~100 | 2 |

### *3.3. Error correction algorithm*

According to the above analysis, we know which gestures are easy to confuse gestures. This paper wants to improve the recognition rate of confusing gestures, in which case, there is only CNN model's recognition results. This paper proposes a method of probability statistics to reduce the error rate and improve the recognition rate. The core algorithm is as follows:

Predictive results are simply divided into 5 categories. According to the established miscarriage probability model, these five categories have corresponding probability matrices with the actual class. 100 random positive numbers (1 to 100) are divided on the basis of the probability matrix. The corresponding random number distribution for each class is shown as table 2-table 6.

Suppose an image is identified by a Convolutional Neural Network model as an m class and assume a space of x= {0, 1, 2, 3, and 4}. Generate random numbers of $x_m$, where $x_m$= {0, 1, 2…, 99}. Assuming the adjusted category is $y_m$, the range of $y_m$ is {0, 1, 2, 3, and 4}. Then based on table 2-table 6, establishing $y_m$*'s* section function $f(x_m)$ .The algorithm is shown in Algorithm 1.

---

**Algorithm 1** Error correction algorithm

---

**Input:** Convolutional Neural Network model identification results *m*
**Output:** New category number $y_m$
1. Generate a random integer $x_m$;
2. Select the appropriate function $f(x_m)$ based on the $x_m$ domain;
3. Output the results to the document;
4. **end**

Under the known CNN result, a random number is randomly generated to adjust its new class according to its falling range. Perform a lot of statistical experiments so that each type of gestures is as close as possible to the actual category. The error correction algorithm proposed in this paper is aimed at confusing gestures. The probability model of misjudgement is established by the probability statistic experiment of each kind of gesture. We designed a probabilistic generator to control new categories with random numbers. The emphasis of the algorithm is on the control of random numbers. We allow random numbers to be generated depending on a certain probability. Since the establishment of the confusion matrix and the misjudgement probability model is based on the model test results, the algorithm also relies on the model used. This algorithm is also an optimization of Convolutional Neural Network model.

## 4. Experimental results and analysis

### *4.1. error correction algorithm experiment results*

First, enter a picture to the Convolutional Neural Network model. In order to reveal the convolutional process of pictures in the network, visualize the feature map for each layer of convolution [10]. In CNN settings, the Feature map is the result of convolution using convolutional cores, and different feature extractions (nuclei) extract different Feature [7]. The model achieves optimization by finding the best set of convolutional cores that can explain the phenomenon. In the case of gesture No. 2nd, the visualization result of the convolutional layer is shown in the following Figure 4.

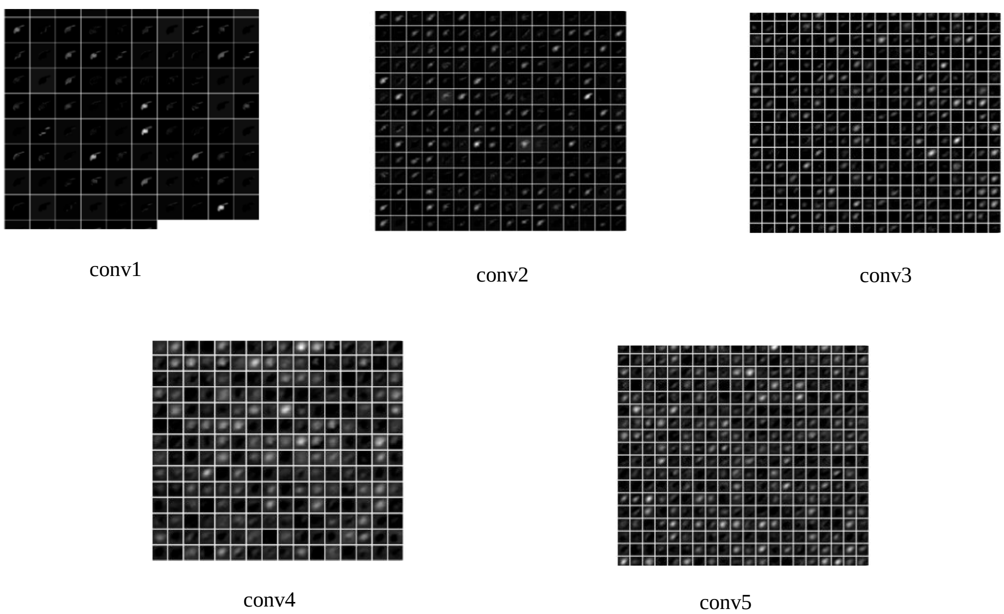



Figure 4. The visualization result of the convolutional layer

Then, save the output and use it as an input to the error correction algorithm. If the output is 1, then 1 is the input of the error correction algorithm. After the error correction algorithm, new categories are generated. An example of the experimental flowchart is shown in Figure 5.

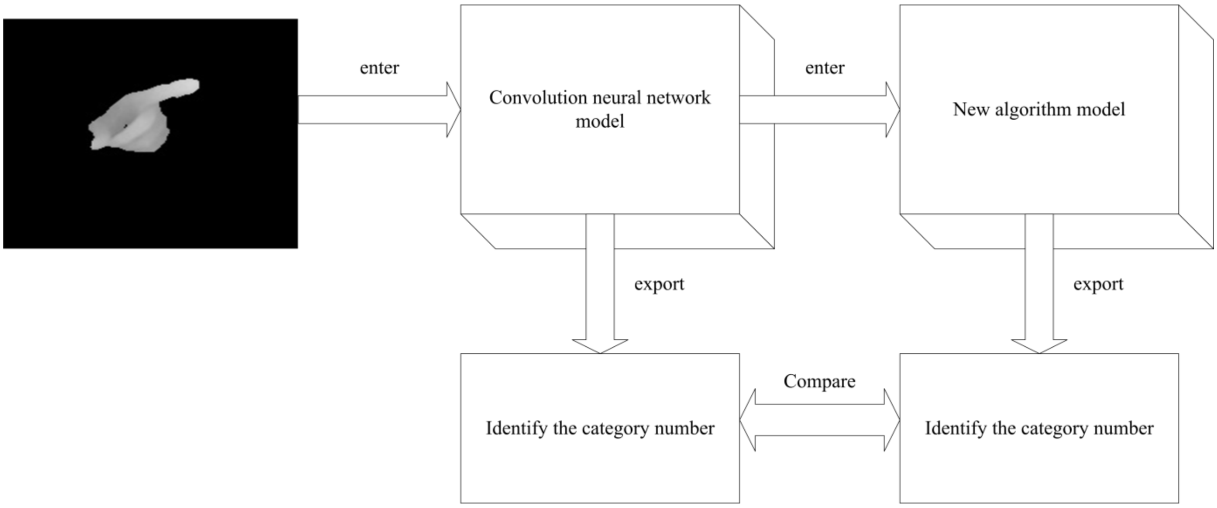



Figure 5. The visualization result of the each layer

Pick 3000 pictures for each type of gesture, and perform a lot of statistical experiments. Then, statistically recognize the rate of confusion gestures according to the experimental results. Create a confusing matrix for confusing gestures. The confusion matrix is shown in the Figure 6. The recognition result of this algorithm is shown in the Table 7.

| | | Predicted class | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| Actual class | 1 | 1432 | 526 | 952 | 40 |
| | 2 | 702 | 1118 | 614 | 516 |
| | 3 | 851 | 284 | 1410 | 405 |
| | 4 | 203 | 564 | 479 | 1704 |

Figure 6. Confusion matrix

Table 7. The accuracy of the algorithm

| category | accuracy |
|---|---|
| 1 | 52% |
| 2 | 41% |
| 3 | 47% |
| 4 | 57% |

### 4.2. Comparative experiment

#### 4.2.1. Comparison experiment with Convolutional neural network model

The paper [24] presents an algorithm for gesture recognition based on Convolutional Neural Network. A comparative experiment is conducted based on our database. In order to ensure the diversity of samples, the number of training samples is up to 83880 and the number of verification sets is 9000. Before the training model, the data are pre-treated and unified into 200*200 pictures. Output test accuracy when the number of times each training up to 500 times. The maximum number of iterations is 100,000. Finally, according to the loss value and the accuracy rate, the iteration number 83000 model is considered the best model. The test data set is 15000, and the convolutional neural network model used in this paper is invoked for comparison experiments. The experimental results output the accuracy of each type of gesture. The accuracy rate of the Fist gesture is 98%, the accuracy rate of the thumb gesture is 47%, the accuracy rate of the index finger gesture is 34%, the accuracy rate of the little finger gesture is 42%, and the accuracy rate of the finger gesture is 45%. Experimental results show that the thumb, index finger, and little finger gestures easily are confused as other gestures.

The algorithm in this paper is added after the convolutional neural network model is called. In this way, each picture is tested with 2 results, which are the results of the model test and the results of this algorithm. Compared with the recognition rate of CNN model and the recognition rate of the algorithm, it is found that the recognition rate of thumb gesture is increased by 5% and so on. Table 8 shows the results.

Table 8. The growth rate of each gesture

| category | grow rate |
|---|---|
| 1 | 5% |
| 2 | 7% |
| 3 | 5% |
| 4 | 12% |

#### 4.2.2. Verification experiment based on Thomas Moslund database

The database used in this paper is a database established by 600 students in our lab. We also validated our algorithm on the Thomas Moeslund database. The Thomas Moslund static gesture database contains 24 alphabetic gestures, J and Z were removed from 26 English letters. Figure 7 shows the data set.

Figure 7. Thomas Moeslund static data set

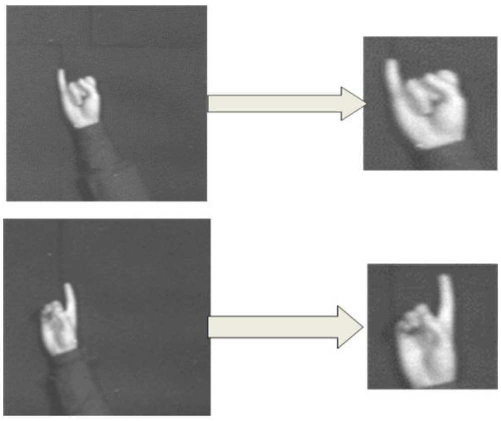


Figure 8. The Pre-treatment process

Because the original gesture picture in the database is 248*256, and the picture contains the arm, the gesture in the picture occupies a small proportion. So, I did some preprocessing on the gesture before I imported it into the model. I chose three kinds of gestures to do the test verification. The segmentation method is used to remove the useless background and preserve the gesture area. The processed picture is normalized to a 200*200. Pre-treatment process is shown Figure 8. After the preprocessing is completed, we first call the convolutional neural network model and then call our algorithm. Compared with error correction, the recognition rate was increased by about 5%.

## 5. Conclusions

This paper studies the gesture recognition based on Convolutional Neural Network. Its weight sharing and drop-sampling technology improve the recognition rate of the image to some extent. This article has done a validation experiment on five kinds of gesture databases. Studies have shown that there are four kinds of gestures that are easily confusing gestures. In response to this situation, we re-adjust the results of the Convolutional Neural Network model recognition gesture by the probability statistics model.

By analyzing the CNN model recognition results, the confusion matrix is established. Through the matrix, gestures are easily confused gestures and the relationship between them can be determined. We made a correction to the easily confused gestures. Eventually, this algorithm improves the recognition rate of these gestures. The algorithm is suitable for similar gestures, and the recognition rate is improved by a large number of statistical results. But the rate of improvement is not great. So, I think that the characteristics of each layer of the network structure need to be analysed in order to distinguish these gestures. This is my next step in the work plan.

## Acknowledgements

This work was supported in part by the National Key R&D Program of China (No. 2016YFB1001403), the National Natural Science Foundation of China (Nos. 61472163, 61603151,61472232), and the Shandong Provincial Key R&D Program (No. 2017GGX10146).

## References

1. M. K. Bhuyan, D. A. Kumar and K. F. Macdorman, “A Novel Set of Features for Continuous Hand Gesture Recognition,” *Journal on Multimodal User Interfaces*, vol. 8, no. 4, pp. 333-343, 2014
2. P. Barros, S. Magg and C. Weber, “A Multichannel Convolutional Neural Network for Hand Posture Recognition,” *Artificial Neural Networks and Machine Learning – ICANN 2014*, pp. 403-410 2014
3. C. C. Chang, J. J. Chen, and W. K. Tai, “New Approach for Static Gesture Recognition,” *Journal of information science and engineering*, vol. 22, no. 5, pp. 1047-1057, 2006
4. J. Cao, X. L. Zhao, J.H. Wang, “Animated Gesture Recognition Method Based on RGB-D,” *Journal of Computer Applications*, no. 6, pp. 1-7, 2018
5. C. Dong, C. C. Loy, and K. He, “Learning a Deep Convolutional Network for Image Super-resolution.” *European Conference on Computer Vision, Springer, Cham*, pp. 184-199, 2014
6. C. S. Fahn, and K. Y. Chu, “Hidden-Markov-Model-Based Hand Gesture Recognition Techniques Used for a Human-Robot Interaction System,” *Human-Computer Interaction, Interaction Techniques and Environments, Springer Berlin Heidelberg*, pp. 1043-1058, 2010
7. S. Fong, Z. Yan, I. Fister “A Biometric Authentication Model using Hand Gesture Images,” *BioMedical Engineering OnLine*, vol. 12, no. 1, pp. 1-18 2013
8. W. T. Freeman, and M. Roth, “Orientation Histograms for Hand Gesture Recognition,” *International workshop on automatic face and gesture recognition*, vol. 12, pp. 296-301, 1995
9. R. Girshick, “Fast R-cnn,” *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448, 2015
10. A. R. Hafiz, M. F. Amin, and K. Murase, “Real-time Hand Gesture Recognition using Complex-valued Neural Network,” *International conference on neural information processing*, pp. 541-549, 2011
11. H. Haitham, S. Abdul-Kareem, “Static Hand Gesture Recognition using Neural Networks,” *Artificial Intelligence Review*, pp. 1-35, 2012
12. H, S. Hasan, S. B. A. Kareem, “Gesture Feature Extraction for Static Gesture Recognition,” *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, vol. 38, no. 12, 2013
13. S. Ji, W. Xu, and M.Yang, “3D Convolutional Neural Networks for Human Action Recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221-231, 2013
14. Y. Jia, E. Shelhamer, J. Donahue “Caffe: Convolutional Architecture for Fast Feature Embedding” *Proceedings of the 22nd ACM international conference on Multimedia*, PP. 675-678, 2014
15. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks” *Advances in neural information processing systems*, pp. 1097-1105,2012
16. S. Lawrence, C. L. Giles, and A. C. Tsoi, “Face Recognition: A Convolutional Neural-Network Approach,” *IEEE transactions*

*on neural networks*, vol. 8, no. 1, pp. 98-113, 1997
17. S. Li, and Z. Q. Liu, and A. B.Chan, "Hetero Geneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 482-489, 2014
18. Y. T. Li, J.P. Wachs, "HEGM: A Hierarchical Elastic Graph Matching for Hand Gesture Recognition," *Pattern Recognition*, vol. 47, no. 1, pp. 80-88, 2014
19. G. Marin, F. Dominio, and P. Zanuttigh, "Hand Gesture Recognition with Leap Motion and Kinect Devices," *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 1565-1569, 2014
20. P. Molchanov, S. Gupta, and K. Kim, "Hand Gesture Recognition with 3D Convolutional Neural Networks" *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1-7, 2003
21. S. Mitra, T. and Acharya Wu, "Gesture Recognition: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311-324, 2007
22. Z. Mo and U. Neumann, "Lexical Gesture Interface," *IEEE International Conference on. IEEE*, pp. 7-7, 2006
23. J. Nagi, F. Ducatelle, and G. A. Di Caro, "Max-pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition," *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pp. 342-347, 2011
24. T. N. Nguyen, M. M. Huynh, and J. Meunier, "Static Hand Gesture Recognition using Artificial Neural Network," *Journal of Image and Graphics*, vol. 1, no. 1, pp. 34-38, 2013
25. M. Oquab, L. Bottou and I. Laptev, "Learning and Transferring Mid-level Image Representations using Convolutional Neural Networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717-1724 ,2014
26. L. Pigou, S. Dieleman, and P. J. Kindermans, "Sign Language Recognition using Convolutional Neural Networks," *Workshop at the European Conference on Computer Vision, Springer, Cham*, pp. 572-578, 2014
27. O. Patsadu, C. Nukoolkit, and B. Watanapa, "Human Gesture Recognition using Kinect Camera," *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, pp. 28-32, 2012
28. Q. Pu, S. Gupta, and S. Gollakota, "Whole-home Gesture Recognition using Wireless Signals," *Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 27-38, 2013
29. S. P. Priyal, and P. K. Bora, "A Robust Static Hand Gesture Recognition System using Geometry Based Normalizations and Krawtchouk Moments," *Pattern Recognition*, vol. 46, no. 8, pp. 2202-2219, 2013
30. T. Pfister, J. Charles, and A. Zisserman, "Flowing Convnets for Human Pose Estimation in Videos," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1913-1921, 2015
31. V. Pitsikalis, A. Katsamanis, and S. Theodorakis, "Multimodal Gesture Recognition Via Multiple Hypotheses Rescoring," *Gesture Recognition. Springer, Cham*, pp. 467-496, 2017
32. S. Qin, X. Zhu, and H. Yu, "Real-time Markerless Hand Gesture Recognition with Depth Camera," *Advances in Multimedia Information Processing – PCM 2012*, pp. 186-197, 2012
33. S. S. Rautaray,and A. Agrawal, "Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1-54, 2015
34. S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks," *Advances in neural information processing systems*, pp. 91-99, 2015
35. T. Roska, L. O. Chua, and G. Wen, "The CNN Universal Machine: An Analogic Array Computer," *J IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 3, pp. 163-173, 1993
36. Z. Ren, J. Yuan, and J. Meng, "Robust Part-based Hand Gesture Recognition using Kinect Sensor," *IEEE transactions on multimedia*, vol. 15, no. 15, pp. 1110-1120, 2013
37. Z. Ren, J. Yuan, and Z. Zhang, "Robust Hand Gesture Recognition Based on Finger-earth Mover's Distance with a Commodity Depth Camera," *Proceedings of the 19th ACM international conference on Multimedia*, pp. 1093-1096, 2011
38. A. Sharif Razavian, H. Azizpour, and H. Sullivan, "CNN Features Off-the-shelf: An Astounding Baseline for Recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806-813, 2014
39. C. Szegedy, W. Liu, and Y. Jia, "Going Deeper with Convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1-9, 2015
40. 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
41. J. Suarez, and R. R. Murphy, "Hand Gesture Recognition with Depth Images: A Review," *Ro-man, 2012 IEEE*, pp. 411-417, 2012
42. k. Simonyan, and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," *arXiv preprint arXiv*, pp.1049, 1556, 2014
43. P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," *ICDAR*, vol. 3, pp. 958-962, 2003
44. M. Turk, "Gesture Recognition," *Computer Vision: A Reference Guide*, pp. 346-349, 2014
45. M. F. Tolba, A. Samir, and M. Aboul-Ela, "Arabic Sign Language Continuous Sentences Recognition using PCNN and Graph Matching," *Neural Computing and Applications*, vol. 23, no. 3, pp. 999-1010, 2013
46. L. Wang, H. Liu, B. Wang, and P. Li, "Gesture Recognition Method Combined with Skin Color Model and Convolution Neural Network," *Computer Engineering and Applications*, no. 6, pp. 209-214, 2017
47. L. Xiao, R. Li, F.Z. Zeng, "Animated Gesture Recognition method Based on Self-learning Sparse Representation," *Journals of Communications*, vol. 34, no. 6, pp. 128-135, 2013
48. L. Xu, J. S. J. Ren, and C. Liu, "Deep Convolutional Neural Network for Image Deconvolution," *Advances in Neural Information Processing Systems*, pp. 1790-1798, 2014

49. R. Xie, X. Sun, X. Xia, "Similarity Matching-Based Extensible Hand Gesture Recognition," *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3475-3483, 2015
50. X. Xu, "Study on Gesture Recognition Based on Hidden Markov Model," *South China University of Technology*, 2012
51. B. Yang, X. N. Song, Z. Q. Feng, and S. Yan, "A Gesture Recognition Algorithm Based on Spatial Distribution Feature in Complex Background," *Journal of Computer Aided design and graphic Science*, no. 10, pp. 1841-1848, 2010
52. H. S. Yeo, B. G. Lee, and H. Lim, "Hand Tracking and Gesture Recognition System for Human-computer Interaction using Low-cost Hardware," *Multimedia Tools and Applications*, vol. 74, no. 8, pp. 2687-2715, 2015
53. H. Zheng, X.K. Shen, "Animated Gesture Recognition Method Based on Self-learning Sparse Representation," *Beijing University of Aeronautics and Astronautics*, vol. 38, no. 2, pp. 273-279, 2012
54. L. G. Zhang, J. Q. Wu, and G. Wen, "Hand Gesture Recognition Based on Hausdorff Distance," *Journal of Image and Graphics: A*, vol. 7, no. 11, pp. 1144-1150, 2002