IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DATA MINING FOR INTERNET OF THINGS

Received January 26, 2019, accepted February 17, 2019, date of publication February 27, 2019, date of current version March 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901930

Gesture Recognition Based on CNN and DCGAN for Calculation and Text Output

WEI FANG[1,2], YEWEN DING[1], FEIHONG ZHANG[1], AND JACK SHENG[3]

[1]Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
[3]Department of Economics, Finance, Insurance, and Risk Management, University of Central Arkansas, Conway, AR 72035, USA

Corresponding author: Yewen Ding (20171211475@nuist.edu.cn)

This work was supported in part by the Open Research Project of the State Key Laboratory of Novel Software Technology under Grant KFKT2018B23, in part by the Priority Academic Program Development of the Jiangsu Higher Education Institutions, and in part by the Open Project Program of the State Key Laboratory of CAD&CG, Zhejiang University, under Grant A1916.

ABSTRACT In the past few years, with the continuous improvement of hardware conditions, deep learning had performed well in solving many problems, such as visual recognition, speech recognition, and natural language processing. In recent years, human–computer interaction behavior has appeared more and more in daily life. Especially with the rapid development of computer vision technology, the human-centered human–computer interaction technology is bound to replace computer-centered human–computer interaction technology. The study of gesture recognition is in line with this trend, and gesture recognition provides a way for many devices to interact with humans. The traditional gesture recognition method requires manual extraction of feature values, which is a time-consuming and laborious method. In order to break through the bottleneck, we propose a new gesture recognition algorithm based on the convolutional neural network and deep convolution generative adversarial networks. We apply this method to expression recognition, calculation, and text output, and achieve good results. The experiments show that the proposed method can train the model to identify with fewer samples and achieve better gesture classification and detection effects. Moreover, this gesture recognition method is less susceptible to illumination and background interference. It also can achieve an efficient real-time recognition effect.

INDEX TERMS Calculation, CNN, DCGAN, gesture recognition, text output.

I. INTRODUCTION

In recent years, with the rapid development of science and technology, the way of human-computer interaction has also been greatly changed. Various new types of human-computer interaction methods have also appeared in the public's field of vision. The interactive mode of the mouse and keyboard has become a touch screen and voice. The form of interaction has become diversified and humanized. However, the more efficient form of interaction is to allow the machine to understand the human body language. Gestures are the most common in all kinds of body language, so it can be used as a simple and free means of human-computer interaction. It has a very broad application prospect. An important process for gesture-based human-computer interaction is to recognize gestures. When performing gesture recognition, the features of the gesture are first extracted, and then gesture recognition is performed according to the extracted features. There are many common gesture recognition methods. For example, the neural network-based recognition method has strong ability to classify and identify. However, if the number of neural network layers is generally shallow, it is easy to overfitting; the recognition method based on geometric features performs gesture recognition by extracting gesture structure, edge, contour and other features, it has good stability, but can't improve the recognition rate by increasing the sample size. The recognition method based on Hidden Markov Model has the ability to describe the time and space changes of gestures, but the recognition speed of the method is not satisfactory. With the rapid development of machine learning and deep learning in computer vision, methods based on machine learning and deep learning have attracted more and more researchers' attention. Among them, the deep neural network has the characteristics of local connection, weight

The associate editor coordinating the review of this manuscript and approving it for publication was Tie Qiu.

sharing, automatic feature extraction, etc., which brings new ideas to the task of gesture recognition. Therefore, based on the complexity of gesture changes, we propose a gesture recognition method based on deep Convolutional Neural Network (CNN) and Deep Convolution Generative Adversarial Networks (DCGAN).

We use the camera of the computer to collect the data directly. But the quality of the sampled data is obviously affected by the illumination. So we perform the light detection firstly. The purpose of this step is to obtain high-quality samples. We only need to adjust the camera angle, light intensity or other methods to achieve this step. Next, we use DCGAN to generate new images to solve the problem of overfitting. Finally, 5/6 of the data is used for training, and the remaining 1/6 is used for testing. We design two network structures to realize the expression recognition function, calculation and text output. It mainly adjusts the depth of the network and the number of parameters according to the complexity of the task.

The main contributions of this paper are as follows:

1) We propose a new gesture recognition method based on CNN and DCGAN;
2) And, we evaluate our model in some real data sets. The experimental results show that our model can achieve good results. First, for a specific gesture, by using the recognition model, it can effectively recognize the actual meaning of the gesture. The new model can achieve full automation, and its accuracy can reach a very high level;
3) In addition, in the case of a small number of samples, the problem of overfitting can be solved only by DCGAN. In the state where the illumination conditions are not particularly good, the accuracy of recognition without treatment can be effectively improved by our pre-processing.

II. RELATED WORK

Below we will introduce some related work on gesture recognition and neural network.

A. GESTURE RECOGNITION

In recent years, virtual reality has gradually appeared in people's daily life, and it is undoubtedly the mainstream of human-computer interaction in the future. However, at the input of human-computer interaction, there is no unified way. With the unique advantages of gestures, it will become the mainstream of future interactions. At present, gesture recognition is mainly divided into two types: contact and non-contact. The contact interaction method mainly acquires three-dimensional information of gestures by means of equipment such as gloves, but the manner of using peripherals largely limits the flexibility of human-computer interaction and brings inconvenience to the operator. The non-contact type of interaction is mainly a visual-based method, which eliminates the need for the operator to wear any peripherals, and the interaction is more natural and comfortable.

Early gesture recognition was based on data gloves. In 1983, Grime *et al.* first used gloves with node markers. They used the palm skeleton to recognize gestures and complete simple gesture recognition. In the 1990s, with the advantage of accurate positioning of peripherals, many excellent systems appeared at home and abroad. Takahashi *et al.* [17] used data gloves to achieve the recognition of 46 specific gestures; The finger marking method replaced the data gloves and completed the recognition of several specific gestures, it achieved good results. In many human-computer interactions, dynamic gestures were often required, thereby promoting the development of dynamic gestures. Lee *et al.* [1] used the information entropy algorithm to segment the hand from the background image, and successfully applied it to the video data stream through the parallel computing algorithm, and identified the extracted target image with an accuracy rate of 95%, but there were fewer gesture categories that could be recognized.

During this period, gesture recognition mostly needed to be performed by means of peripherals. Therefore, the application of gesture interaction was greatly limited. In 2010, Microsoft released a depth sensor ''Kinect'' for somatosensory games, which could measure the distance between the human body and the device, and could track the movements of the human body. Since then, many gesture recognition algorithms and systems have been based on Kinect.

At the same time, many electronic information companies had also joined the topic of gesture interaction and achieved good results. Wachs *et al.* [2] used face recognition, speech recognition and gesture recognition to apply it to ES8000 series TVs for browsing web pages, TV remote control and other functions. In the same year, Microsoft used the Doppler effect, built-in speakers and microphones to achieve target positioning and gesture recognition, and developed the gesture interaction tool ''SoundWave''; Newcombe *et al.* [3] introduced the gesture recognition tool ''Handpose'' based on depth information to track the movement of the hand in real time. Shin and Sung [4] also tried to recognize dynamic gestures.

At this stage, some gesture algorithms and devices had reached the requirements of practical applications. However, such products and algorithms still had great problems, and there were many restrictions in the application process. There was still a gap between the identification and application of bare hands.

B. NEURAL NETWORK

Convolutional Neural Network is a common deep learning architecture inspired by biological natural visual recognition mechanisms. In 1959, Hubel and Wiesel [18] found that animal visual cortical cells were responsible for detecting optical signals. Inspired by this, Kunihiko and Sei [5] proposed CNN's predecessor, neocognitron.

In the 1990s, Lecun *et al.* [6] published a paper that established the modern structure of CNN and later improved it. They designed a multi-layer artificial neural network called

LeNet-5 to classify handwritten numbers. Like other neural networks, LeNet-5 could also be trained by using backpropagation algorithms.

LeNet-5 had achieved gratifying results. However, due to the lack of ability to process large-scale training data, LeNet-5 did not perform well on complex issues. Therefore, the convolutional neural network once fell into a low tide.

With the development of GPU accelerators and big data, the number of CNN layers has been deepened, and the recognition accuracy has been greatly improved, so it has received a lot of attention and research. Since 2006, researchers have designed many ways to overcome the difficulty of deep convolutional neural network training. Among them, AlexNet [7] was one of the most famous. AlexNet used a classic CNN structure to achieve breakthrough performance in image recognition. The overall structure of AlexNet was similar to that of LeNet-5, but with more layers.

After the success of AlexNet, researchers further designed a lot of better classification models, including the four most famous ones: ZFNet [8], VGGNet [9], GoogleNet [10] and ResNet [11]. They achieved a higher classification accuracy. In terms of structure, the number of layers of CNN increased. The number of layers of the ILSVRC 2015 champion ResNet was 20 times deeper than AlexNet and 8 times deeper than VGGNet. By increasing the depth, the network can use additional nonlinearity to derive the approximate structure of the objective function, thereby further better characterizing the features and achieving better classification results.

GAN was inspired by the two-player game in game theory, pioneered by Goodfellow *et al.* [12]. Based on actual results, they appear to produce better samples (more sharp and clear images) than others. DCGAN [13] was an extension of GAN that introduced a convolutional neural network into a generative model for unsupervised training, using the powerful feature extraction capabilities of the convolutional network to improve the learning of the generated model.

Nowadays, various neural networks emerge in an endless stream and are applied to a wide range of fields. Fang *et al.* [14] applied it to image recognition, Meng *et al.* [15] applied it to information hiding, Xiong *et al.* [16] applied it to natural language processing. We believe that it will continue to develop and make people's lives better.

III. EXPRESSION RECOGNITION

In order to test the effect of our method, we output the corresponding expression by recognizing the gesture. Here we collect 10 gestures, corresponding to 10 expressions.

A. DATA

We use the camera of the computer to collect the training data directly. The amount of sampled data for each gesture is 1200 images, and the size of the image is 50×50. Adjusting the position of the hand to ensure that no large batches of the same image appear in a training set. We collect 10 gestures, and the corresponding expressions are also displayed above the gestures, as shown in Figure 1. The meanings of each gesture are as follows, from 1 to 10 are: "yeah," "high-five," "good," "bad," "ok," "smile," "cry," "fist," "dog" and "love." We will use these numbers to describe 10 gestures later.

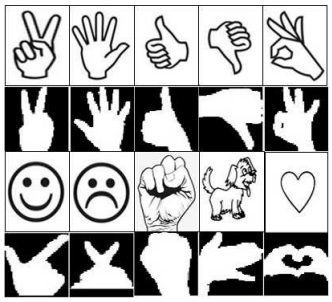


FIGURE 1. Contrast map of gestures and expressions.

B. IDENTIFICATION MODEL

We use a convolutional neural network to train our recognition model. We want to use more economical convolutional neural network models, such as shallow networks like LeNet-5 and AlexNet, rather than models with more powerful classification capabilities, such as VGG and GoogLeNet. Although VGG and GoogLeNet have more powerful classification performance, but they have more parameters. For example, although the VGG network uses a 3×3 convolution filter, but the number of parameters is still large compared to AlexNet. Taking its model VGG16 as an example, the total number of parameters is about 130 million, and the parameters increase the training time. If the model of the VGG19 is only deployed on a single CPU for training, it will take more than 8 hours to train an epoch. Obviously, using this model is impractical.

Since we only classify images of 10 gestures here, we don't need to use convolutional neural network models such as VGG16, GoogLeNet, which are applied to the classification of thousands of images. We will make adjustments to AlexNet to train our gesture recognition model. Here we make the following changes:

The model contains a total of 6 layers. The front 4 layers are a convolution layer plus a pooling layer, and the last 2 layers are fully connection layers.

The input to the first convolutional layer is the original image, which is $50 \times 50 \times 1$. The convolution filter has a size of 5×5 and a depth of 32. Instead of using full 0 complement. The activation function used is relu. The filter used in the second convolutional layer has a size of 5×5 and a depth of 64. It also does not use full 0 complement. The activation function used is sigmoid.

We use max-pooling at the pooling layer. The first pooling layer uses a 2×2 filter size with a step size of 2, and full 0 complement. The second pooling layer uses a 5×5 filter size with a step size of 5 and full 0 complement.

The Flatten layer is used to "flatten" the input. It makes a multidimensional input one-dimensional for transitioning to a fully connection layer. The number of output nodes of the first

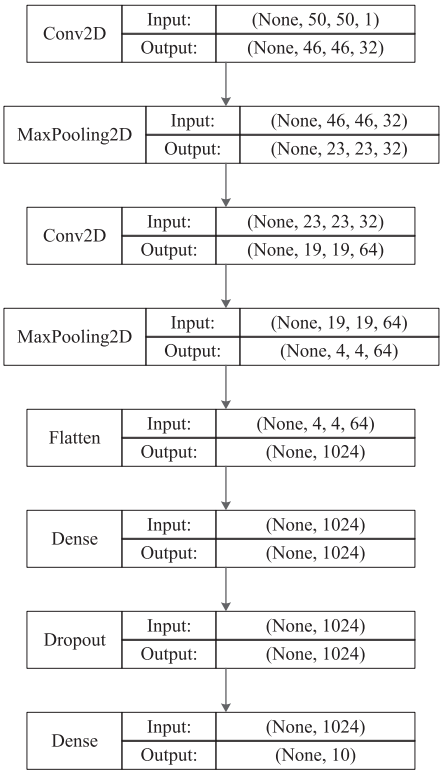


FIGURE 2. Gesture recognition network structure.

fully connection layer is 1024. At the same time, we introduce the Dropout mechanism after the first fully connection layer to suppress overfitting. The size of the Dropout parameter is 0.6. The second fully connection layer has 1024 input nodes and 10 output nodes. Finally, we use the softmax function to get the final prediction. The network structure is shown in Figure 2.

C. TRAINING

We use ''adam'' as optimizer, which uses the default settings in keras, lr $=$ 0.001, beta_1 $=$ 0.9, beta_2 $=$ 0.999, epsilon $=$ 1e $-$ 08, decay $=$ 0.0. In our experiment, there are 10 gestures, a total of $1200 \times 10 = 12000$ images, we use 10000 images in the data set for training, and the remaining 2000 images are used for testing. We train all images for 10 epochs with a batch size of 64 (epochs $=$ 10, batch size $=$ 64), which allow us to save the training model on a regular basis.

D. RESULTS

After getting the model of gesture recognition, we actually test the model. Because the test results are affected by lighting conditions, in order to ensure the validity of the test, we strive to keep the test environment and the conditions of the sampling environment basically the same. To visualize the test results, we predict each frame of the image and compare it to the actual gesture. When the start of each test, you may get a misjudgment because the gesture is not fully prepared, so we use the results after 11 frames as the test results. Here we record the total number of test frames as ''Total_frames,'' and the correct number of frames is recorded as ''Correct_frames.'' Then our accuracy can be recorded as:

$$Accuracy = \frac{Correct\_frames}{Total\_frames} \times 100\% \quad (1)$$

We test all 10 gestures and achieve good recognition results, with an average recognition accuracy of over 90%. We also find that the accuracy of recognition depends on the different categories in the data set. Some categories are easier to identify, such as gestures ''2'' and ''8,'' because they are more discriminating in the data set. The recognition model can achieve almost 100% recognition rate on these categories. However, the gestures ''1'' and ''7'' have a high degree of similarity and they are susceptible to finger movement, so the accuracy is relatively low. But overall, the accuracy of gesture recognition is still very high. Table 1 shows the accuracy of gesture recognition.

TABLE 1. Accuracy of gestures corresponding to expressions.

Number	1	2	3	4	5
50 frames accuracy	88%	94%	92%	86%	94%
100 frames accuracy	86%	96%	94%	87%	95%
Average accuracy	87%	95%	93%	86.5%	94.5%
Number	6	7	8	9	10
50 frames accuracy	88%	86%	96%	92%	86%
100 frames accuracy	86%	87%	97%	90%	89%
Average accuracy	87%	86.5%	96.5%	91%	87.5%
Global accuracy	90.45%				

IV. CALCULATION AND TEXT OUTPUT

Based on our good results, so we decide to use the same model for more complex tasks. Finally, we decide to use the model to complete the calculation and text output tasks. The reason why I want to accomplish this task here is to increase the complexity and test the reliability and practicability of the model. At the same time, my mother has been affected by presbyopia, often typo because she cannot see the letters on the keyboard, and her typing speed is very slow. I really want to help her solve this problem. We combine these two tasks into one, and you can easily use different functions by selection buttons.

A. DATA

We also use the camera of the computer to collect the training data directly. The amount of sampled data for each gesture is 1200 images, and the size of the image is 50×50. This time, because the training data is collected under the condition of poor lighting conditions, we obviously find that the image contour in the training set is not particularly obvious, which may affect the training model and the final recognition effect. Therefore, we add a step of light detection before collecting the training set. By adjusting the angle of the camera and the position of the hand, a better sampling effect is obtained. This is a very important step, which guarantees that we will obtain

good sample data in the same lighting environment, as shown in Figure 3.

FIGURE 3. (left) Image outline is blurred; (right) image meets our requirements.

At the same time, in the section "Expression recognition," we find that most of the images sampled are highly similar, which leads to overfitting when training the model. To solve this problem, we use DCGAN to generate training sample. The gestures generated by DCGAN are more diverse, which helps the trained models to be more reliable. Some of the generated sample images are shown in Figure 4.

FIGURE 4. Some of the sample images generated by DCGAN.

We design all the corresponding gestures according to the American sign language alphabet. For the calculation function, we collect 10 gestures. The gestures we collected are at the top, the corresponding numbers are in the middle, and the bottom is the American sign language alphabet, as shown in Figure 5.

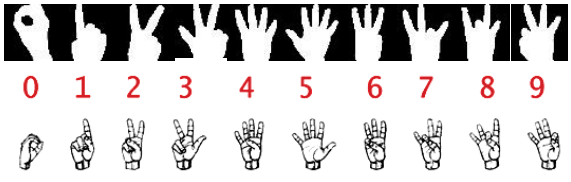


FIGURE 5. Comparison of gestures and numbers.

When we use calculation function, we use gestures to tell the computer that a number has been entered, then it will require you to enter the arithmetic symbol, and we use keyboard here. "0" represents "+"; "1" represents "-"; "2" represents "×"; "3" represents "÷"; "4" represents "$\surd$," stands for square root; "5" represents "x^2," which is the square of the number x; "6" represents "x^3," which is the cube of the number x; "7" represents "%," which is 1 ÷ 100 of the number x; "8" represents "Mod," that is, a divided by b, the remainder is returned; "9" represents "=." By repeating the above operations, we can solve most of the calculation problems in daily life. Finally, after accepting "=," the computer will automatically calculate the final result of the entire expression and display it on the screen.

For the text output function, we collect 27 gestures. Among them, 26 gestures represent the letter "A" to "Z," and the last gesture represents a space, and it plays a role in determining the completion of the input in the calculation function. As shown in Figure 6, the American sign language alphabet is on the left and the gestures we collected is on the right.

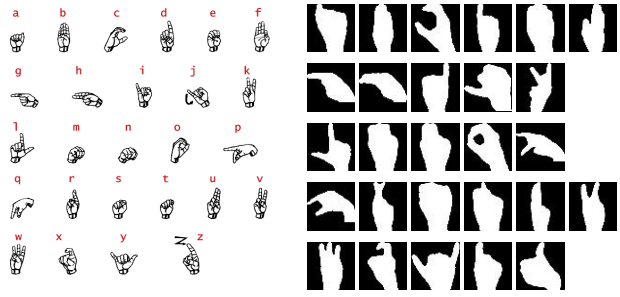


FIGURE 6. Comparison of gestures and letters.

B. IDENTIFICATION MODEL

We also train our gesture recognition model based on the convolutional neural network. This time, we put all 37 gestures for training, including 10 numbers, 26 letters and a space/confirmer. As the gestures increase nearly 3 times, the difficulty of recognition has also increased. Here, we further adjust the network structure.

The specific adjustments are as follows:

The model consists a total of 7 layers. The front 4 layers are the convolution layer plus the pooling layer, the 5th layer is the convolution layer, and the last 2 layers are the fully connection layers.

In the first convolutional layer, the convolution filter has a size of 2 × 2 and a depth of 16. Instead of using full 0 complement. The activation function used is relu. The filter used in the second convolutional layer has a size of 5 × 5 and a depth of 32. It does not use full 0 complement. The activation function used is relu. The filter used in the third convolutional layer has a size of 5 × 5 and a depth of 64. It also does not use full 0 complement. The activation function used is relu.

The first pooling layer uses a 2 × 2 filter size with a step size of 2, and full 0 complement. The second pooling layer uses a 5 × 5 filter size with a step size of 5, and full 0 complement.

The number of output nodes of the first fully connected layer is 128. At the same time, we introduce the Dropout

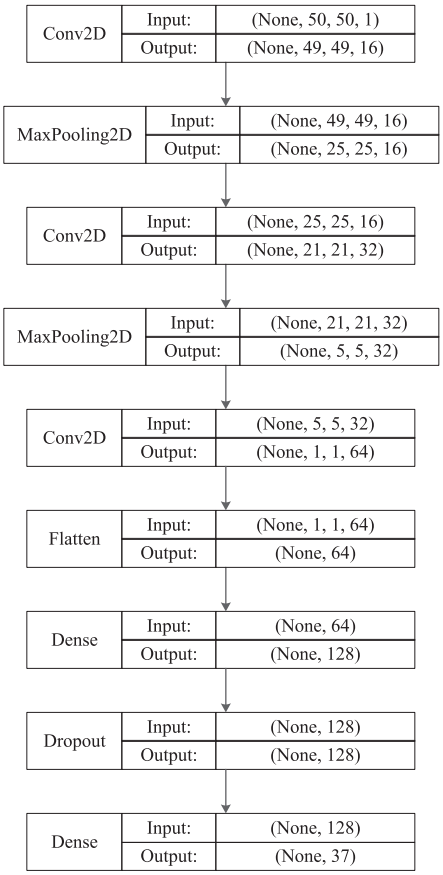


FIGURE 7. Gesture recognition network structure.

mechanism after the first fully connection layer to suppress overfitting. The Dropout parameter is 0.2. The second fully connection layer has 128 input nodes and 37 output nodes. Finally, we use the softmax function to get the final prediction. The network structure is shown in Figure 7.

C. TRAINING

We use "SGD" as optimizer, which uses the default settings in keras, lr = 0.01, momentum = 0.0, decay = 0.0, nesterov = False. In this experiment, including 37 gestures, a total of $37 \times 1200 = 44400$ images, all of them are generated by DCGAN. We use 37000 images in the data set for training, and the remaining 7400 images are used for testing. Due to a large number of training samples, we train all images for 20 epochs with a batch size of 500 (epochs = 20, batch size = 500), which allow us to save the training model on a regular basis.

V. EXPERIMENTS EVALUATION

We decide to actually test the training model in a real environment. Obviously, illumination affects the accuracy of our gesture recognition, so it is very important that the model is robust to illumination. We choose to test our model under different lighting conditions, first of all, experimenting with only natural light. We choose the location in the dark room at 8 o'clock in the morning; then in the environment with the artificial light source, we choose the same time and place, the artificial light source is an incandescent lamp with a power of 15w. We test two different functions, the calculation function is abbreviated as "Cal" and the text output function is abbreviated as "Text." We have obtained a large number of junior high school students' calculation questions online, and the first chapter in "Artificial Intelligence: a Modern Approach, Third Edition" is our test content. When the prediction result is the same for 30 consecutive frames, then a number or letter is output on the screen. The formulas for defining the accuracy of these two functions are:

$$Cal\_Accuracy = \frac{Correct\_num}{Total\_num} \times 100\% \quad (2)$$

$$Text\_Accuracy = \frac{Correct\_num}{Total\_num} \times 100\% \quad (3)$$

where "Correct_num" represents the correct number in the test and "Total_num" represents the total number of the test. The test results are shown in Table 2.

TABLE 2. Identification accuracy.

Accuracy/Quantity	500	1000	1500	2000
Cal^-	85.6%	82.3%	83.4%	84.2%
Cal^+	91.1%	92.7%	89.6%	90.8%
$Text^-$	77.3%	74.1%	75.1%	75.8%
$Text^+$	82.0%	83.4%	81.1%	79.8%

Among them, the result with "−" is the test without the artificial light source, and the result with "+" is the test with the artificial light source. Through experiments, it can intuitively find that our model can still have a relatively high recognition accuracy even without artificial light sources.

In order to improve the efficiency of the input, we decide to reduce the number of frames required to obtain the predicted results. At the same time, we choose to carry out it in the environment with an artificial light source, the test results are shown in Figure 8.

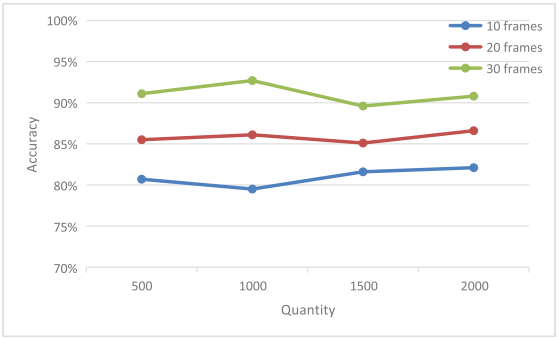


FIGURE 8. Accuracy under different frame counts.

From 30 frames to 20 frames, our recognition speed has increased by about 1 time, but the accuracy rate has dropped by only 6 %. From 30 frames to 10 frames, our recognition speed has increased by about 2 times, but the accuracy rate

was only reduced by about 10 %. This show that our model can be almost recognized and output in real time, which can meet the needs of people in daily life.

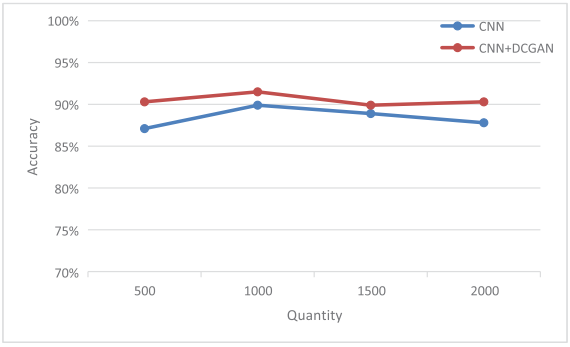


FIGURE 9. Average accuracy of participants.

In order to further ensure the validity of the test results, we test the people who did not participate in the sampling, and the test results are shown in Figure 9. We use CNN and CNN+DCGAN models to test. We select 10 participants and test the calculation function to obtain the average accuracy of the recognition, which is almost consistent with our previous test results. It also shows that our model has wide adaptability. It can also be seen from the experiment that the model of CNN+DCGAN is better than the CNN model. Because the images generated by DCGAN are more diverse, it is possible to avoid overfitting caused by images being too similar in the date set.

VI. DISCUSSION AND CONCLUSION

We propose a method of gesture recognition based on CNN and DCGAN, and we evaluate our model in a real-world environment. The experimental results show that our model can achieve good results. First, for a specific gesture, by using the recognition model, it can effectively recognize the actual meaning of the gesture. The model can also achieve full automation, and its accuracy can reach a high level; In addition, in the case of a large number of similar images in the sample, we use DCGAN to generate training data, which effectively solve the overfitting problem; Moreover, in the state where the illumination conditions are not particularly good, the recognition accuracy can be effectively improved by our pre-processing. Next, we will further test and improve our model. We have some preliminary thoughts on how to improve the results.

At present, our network supports only calculation and text output. We can increase functions by adding more gestures. In the future, we can even use gestures to play games, chat and email with others.

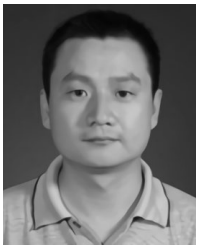
Although the accuracy obtained by the experiment has been very high, we feel that it is necessary to further improve for the application to real life. We plan to further optimize our model by adding training data and changing the network structure.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviewers for their constructive comments and suggestions, which improve the quality of this paper.

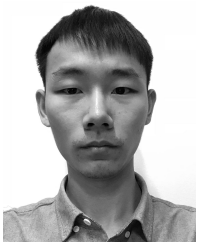
REFERENCES

[1] J. Lee, Y. Lee, E. Lee, and S. Hong, "Hand region extraction and gesture recognition from video stream with complex background through entropy analysis," in *Proc. Conf. IEEE Eng. Med. Biol. Soc.*, Jan. 2004, vol. 2, no. 2, pp. 1513–1516.
[2] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Commun. Acm*, vol. 54, no. 2, pp. 60–71, Feb. 2011.
[3] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," presented at the 10th Int. Conf. Symp. Mixed Augmented Reality, Basel, Switzerland, Oct. 2011, pp. 127–136.
[4] S. Shin and W. Sung, "Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks," in *Proc. ISCAS,* Montréal, QC, Canada, May 2016, pp. 2274–2277.
[5] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 826–834, Sep. 1983.
[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," presented at the NIPS, Lake Tahoe, NV, USA, Dec. 2012.
[8] M. D. Zeiler and R. Fergus. *Visualizing and Understanding Convolutional Networks*. Berlin, Germany: Springer, 2014, pp. 818–833.
[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," presented at the ICLR, San Diego, CA, USA, May 2015.
[10] C. Szegedy *et al.*, "Going deeper with convolutions," presented at the CVPR, Boston, MA, USA, Jun. 2015.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR* Las Vegas, NV, USA, 2016, pp. 770–778.
[12] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, Jun. 2014, pp. 2672–2680.
[13] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," presented at the ICLR, San Juan, PR, USA, May 2016.
[14] W. Fang, F. Zhang, V. S. Sheng, and Y. Ding, "A method for improving CNN-based image recognition using DCGAN," *CMC, Comput., Mater. Continua*, vol. 57, no. 1, pp. 167–178, 2018.
[15] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *CMC, Comput., Mater. Continua*, vol. 55, no. 1, pp. 1–16, 2018.
[16] Z. Xiong, Q. Shen, Y. Wang, and C. Zhu, "Paragraph vector representation based on word to vector and CNN learning," *CMC, Comput., Mater. Continua*, vol. 55, no. 2, pp. 213–227, 2018.
[17] T. Takahashi and F. Kishino, "Hand gesture coding based on experiments using a hand gesture interface device," *ACM Sigchi Bull.*, vol. 23, no. 2, pp. 67–74, 1991.
[18] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, 1968.



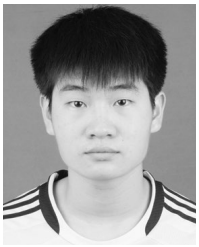
WEI FANG was born in Anhui, China, in 1975. He received the M.S. and Ph.D. degrees in computer application technology from Soochow University, Jiangsu, China, in 2006 and 2009, respectively.

He is currently an Associate Professor with the Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, China, and the State Key Laboratory for Novel Software Technology, Nanjing University. He has authored two books, seven inventions, and more than 30 articles. His research interests are in the areas of cloud computing, big data, deep learning, and artificial intelligence. He is a PC Member for a number of international conferences and a reviewer for several international journals.



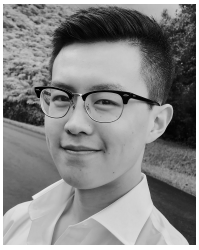
YEWEN DING was born in Changzhou, Jiangsu, China, in 1994. He received the B.S. degree in information and computing science from Yancheng Teachers University, Yancheng, Jiangsu, in 2017. He is currently pursuing the M.S. degree in computer science and technology with the Nanjing University of Information Science and Technology, Nanjing, Jiangsu.

He has authored three articles. His research interests include machine learning, image processing, and weather information processing. His recent research content includes using convolutional neural networks and deep convolution generative adversarial networks for radar image recognition, gesture recognition, and image background segmentation.



FEIHONG ZHANG received the B.S. degree in computer networking engineering from Hechi University, Yizhou, China, in 2017. He is currently pursuing the M.S. degree in computer science and technology with the Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China.

He has participated in the research and development of many meteorological projects, such as hybrid CNN-based satellite big data cloud map classification method research, deep machine learning-based thunderstorm gale classification, and recognition technology development. His research interests include deep learning and image processing applications in meteorology systems.



JACK SHENG received the degree *(magna cum laude)* from the Department of Economics, Finance, and Insurance and Risk Management, School of Business, University of Central Arkansas.

His research interests include data mining, data analytics, and business intelligence.

Mr. Sheng was a recipient of the Arkansas Distinguished Governor's Scholarship.