**Title: Static hand recognition based on CNN model**

***DengSuming****, HuangLantao, LiangZhenxuan, GaoKunwei, JiaChangjing*

*[m730026014@mail.uic.edu.cn](mailto:m730026014@mail.uic.edu.cn)*

*Unite International College, HuiTong Village, TangJiaWan Town, China*

**Abstracts:** With the popularization of computers in society, the development of Human-Computer Interaction (HCI) technology will have a positive impact on the use of computers. Based on efficient image preprocessing and deep convolutional neural network (CNN) architecture, this proposal proposes some gesture recognition methods which want to perfectly complete gesture recognition even in complex backgrounds. The first stage is that we collected our own gestures pictures and do the data preprocessing as the data set. These gestures represent the numbers from 0 to 9 and a total of 18,447 data sets were collected. The image preprocessing includes skin detection, morphological processing, and contour extraction. The methods are all from OpenCV-python. At last stage, we use different kind of CNN(AlexNet, VGG-16) models from TensorFlow to training the data to get different performances. And also we change the parameters and the structure of VGG-16 and Alexnet. The accuracy of the VGG-16 and Alexnet models changed by us in our own data set are 99.56% and 99.43%, respectively. VGG-16 prediction time is nearly 25 times that of AlexNet

**MOTIVATION/BACKGROUND/RELATED WORK:**

**Motivation**: The motivation is to recognize human gestures through mathematical algorithms. Gestures can originate from any body movement or state, but usually originate from the face or hands. Users can use simple gestures to control or interact with devices without touching them. Hand gesture recognition can provide convenience for people with disabilities in life. It also can be applied to many fields or professions, such as sign language recognition, virtual reality, or robot control.[12]

**Background**: In gesture recognition, gestures can be static or dynamic.[1][2] Static gestures are also called gestures in other aspects. They are formed by hands of various shapes and directions and do not represent any movement information. Dynamic gestures are composed of a series of gestures and related motion information.

Gestures can represent letters, names, place names, ages, numbers, dates, years, etc. This fragmented information can form a complete piece of information through a series of expressions.

In order to reduce our workload, we will use the Convolutional neural network in this project without developing complex algorithms to extract image features and learn them.

Convolutional neural network (CNNs) is mainly used in figure classification and target detection. Author use three main types of layers to build Convolutional Net architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer, as shown in Fig. 1.

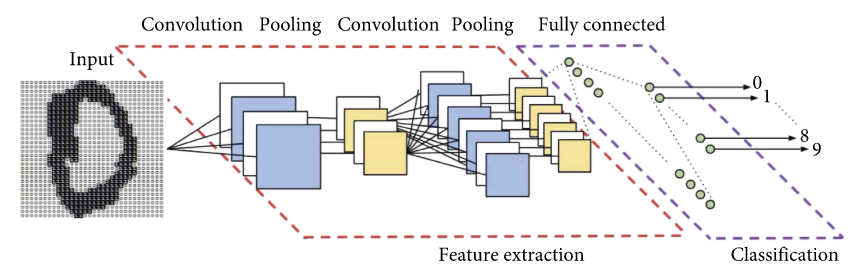


Fig. 1: Example of CNN and its layers. Image adapted from Vargas et al.[2]

The purpose of the convolution Layer is to extract the characteristics of the images. In CNN model, the image convolution is actually to multiply the different local matrices of the input image and the elements at each position of the convolution kernel matrix, and then add them. For example, for two-dimensional convolution, it is defined as the following formula.

Among the formula, W is convolution kernel, and X is the input.

Usually, after the convolution layer, features with large dimensions will be obtained. In the pooling layer, the features will be cut into several regions and the maximum value will be obtained to obtain new features with small dimensions. Fully connected layers play the role of "classifier" in the entire convolutional neural network.

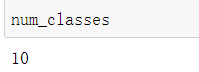
**Related work:**

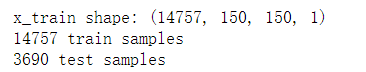
Convolutional neural networks have the characteristics of self-adaptation and self-organization, and can automatically discover features and regularities, so they are often used for classification and recognition. With the development of deep learning methods, more and more Convolutional neural networks have been used in various fields to solve computer vision problems in recent years. Kaiming He et al.[4] used a 152-layer residual network for training on the ImageNet dataset, which was 8 times deeper than the VGG network, but the complexity was lower. At the same time, ResNet has a single-model Top-5 validation error of 4.49% which had a nice performance. In order to find the most suitable model for gesture recognition, different models be compared and analyzed. Gjorgji Strezoski et al.[1] used Marcel dataset as a data set, used models such as VGG-16 , AlexNet, Googlenet and Letnet to train on this data set and compared the results. In the VGG-16 model, the fixed input image size is 224 x 224px and minus the meanRGB value as preprocessing. The final performance of VGG-16 and AlexNet was average. Norah A. Al-johania et al. used transfer learning with CNN (AlexNet, VGG16 and VGG19) models for both features’ extraction and classification. The recognition accuracy give best result when epoch number is 50 where Dr. Badawi dataset in VGG16 and AlexNet reaches to 100% recognition accuracy and BOSPPHORUS dataset reaches to 99.25 % accuracy in VGG19.

**OBJECTIVE:**

The research question of our project: How to extract and process the hand gestures efficiently? What kind of CNN model is the best? Whether the models can predict the new testing data precisely.

The data used: The data will be collected by the laptop’s camera. All the data have been preprocessed by our preprocessing code. We have 10 classes. Each class represents a number. The data is separated two parts. Train samples and test samples. The input size is as followed:





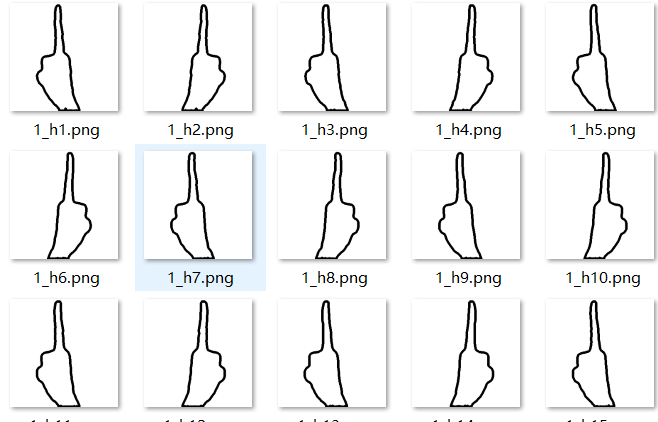


Fig. These are the sample of our dataset used preprocessing method

**METHODS:**

Skin detection: This method helps to extract the skin color. In the preprocessing, we are going to use method of YCrCb color space and method of Otsu Threshold segment from OpenCV-Python[5]. In the RGB color space, the color of skin will be affected by the luminance. Therefore, it is far from easy to extract skin’s color. If change the RGB into YCrCb space color, it can ignore the influence of Y(luminance). By using the YCrCb space color, the skin color can cluster better. For the processing of Cr component in YCrCb, the Cr channel is separately processed by Otsu. Otsu algorithm is used to cluster the gray level of the image. By these ways, we can extract the skin color of our hand successfully.

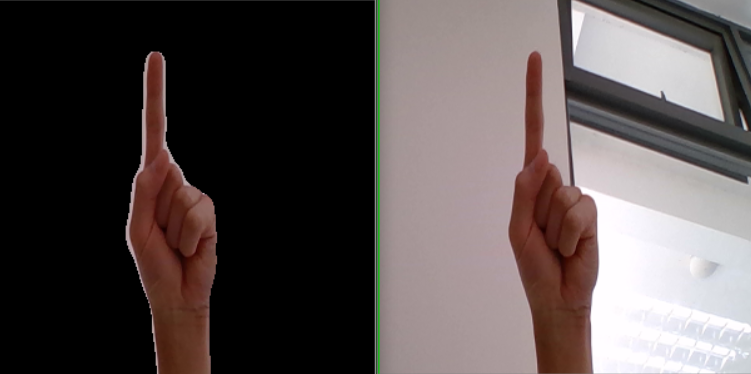


Fig.3 Generated by our laptop’s camera. Base on YCrCb color space + Otsu Threshold segment.

Morphological processing: This method is to further process the segmented gesture images. Since after the process of skin detection, there still has the problems that many black noisy points both in the hand and background. We are going to use cv2.erode and cv2.dilate from OpenCV-Python[6] to figure out this problem. We can choose the open operation or close operation. The former is that erosion followed by dilation, remove isolated small points, burrs. The latter is that dilation followed by erosion, fill in small holes and close small cracks. Therefore, through this method, it can further process the segmented hand gesture images.

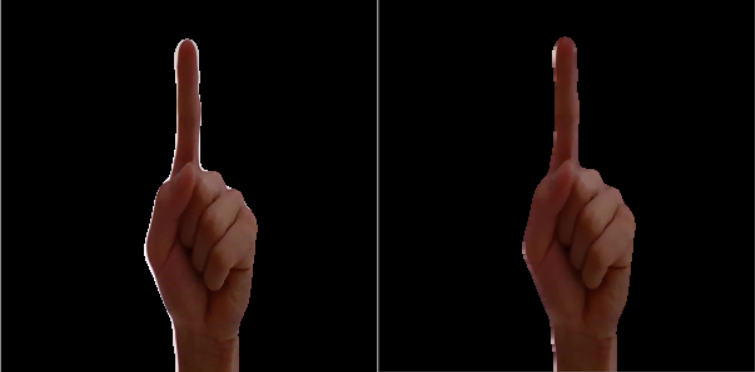


Fig.4 Generated by our laptop’s camera. After erosion and dilation. (right)

Contour processing: The method’s goal is to find the contour of the hand gesture. Contour processing will mainly use two function: cv2.findContours and cv2.drawContours from OpenCV-Python. [7]These two functions can help to draw the contour of the hand gestures. To be noticed, the contours in one image may be numerous but we only need hand’s contour. Therefore, we use sorted function to find the biggest contour. Finally, we get the counter of the hand gesture.

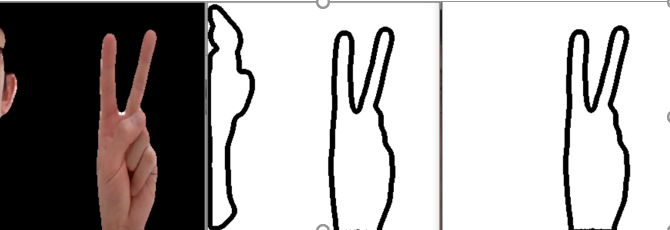


Fig.5 Generated by our laptop’s camera. After contour and sorting

Model selection and training:

**AlexNet**: AlexNet was proposed by Alex Krizhevsky in 2012 and won the champion of ILSVRC 2012.  is the 2012 ImageNet Champion. This model can train and inference fast. It is because three characteristics of this models: activation function, Max pooling layer and Dropout method. Firstly, activation function can convergence better (compared with sigmoid). When doing the back propagation, the model will calculate the derivative of the activation function. Sigmoid activation function has some limitation, the derivative will become zero, which means the gradient will disappear (see the figure). While the activation function will not like that (see the figure). Secondly, Max pooling layer can extract max feature, which can help to decrease the pressure of the model and avoid overfitting problems. Thirdly, Dropout method that decrease the complexity of neural. It saves time and prevent overfitting. There still have some disadvantages: the structure of Alex Net is very simple and the accuracy score still not very high. .[7]

In our models, input size is (224,224,1). It has 6-layer neural network, 3 convolutional layers and 3 full connection layers, the maximum pooling layer is added after the 3 convolutional layers, which contains about 25 million parameters. We changed the number of convolutional layers from 5 to 3. The first dense layer changed from 4096 to 512 and dropout layer changed from 0.5 to 0.3. The second dense layer changed from 4096 to 128 and dropout layer changed from 0.5 to 0.3. The last dense layer changed from 1000 to 10. See the modified version and the original version. The reasons why we modified the model like mainly are that: decrease the time complexity and space complexity. Our datasets are single channel. We don’t need too many parameters and channels, otherwise will cause the overfitting problem. Here are the two figure about our modified model’s FLOPs and structure. (Fig.6 & Fig.7)

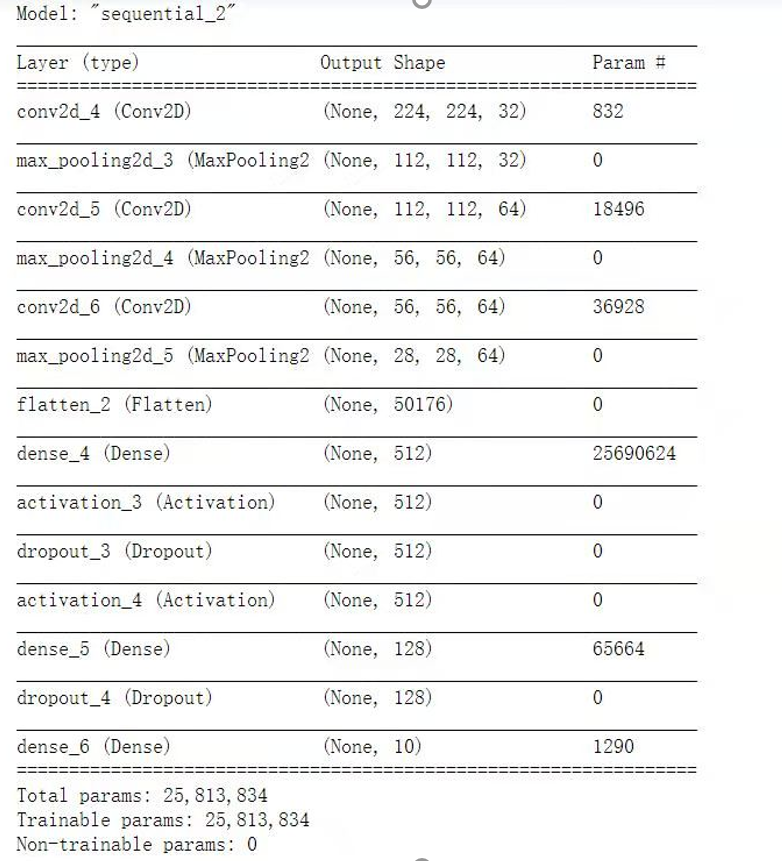


Fig.6 Modified Alexnet structured

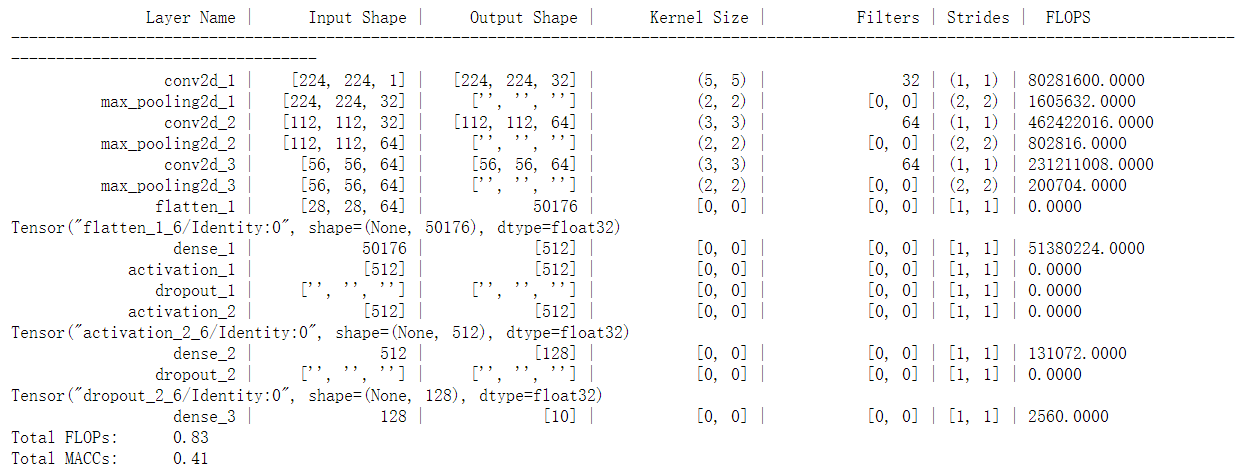


Fig.7 FLOP Estimator (GB)

**VGG-16**: is come up with and in the paper regarding as CNN models in 2015. Compare with the models,  is deeper. It has activation function, Max pooling layer and Dropout method too. In model, the kernels become smaller, which can let the layers more and let the parameters become smaller. The strides become smaller, which let the features can be extracted more. In our models, it adopts 10-layer neural network, 7 convolutional layers and 3 full connection layers. It contains about 68 million parameters. We deleted all the convolutional layer which have 512 filters. The first dense layer changed from 4096 to 256 and dropout layer changed from 0.5 to 0.1. The second dense layer changed from 4096 to 512 and dropout layer changed from 0.5 to 0.1. The last dense layer changed from 1000 to 10. The main reason why we modify the VGG model is the same as Alexnet. Just want to decrease the space complexity and time complexity. The figure14 shows the result of the structure of the original version and our version of VGG models.[6] Here are the two figures about our modified model’s FLOPs and structure. (Fig.8 & Fig.9)

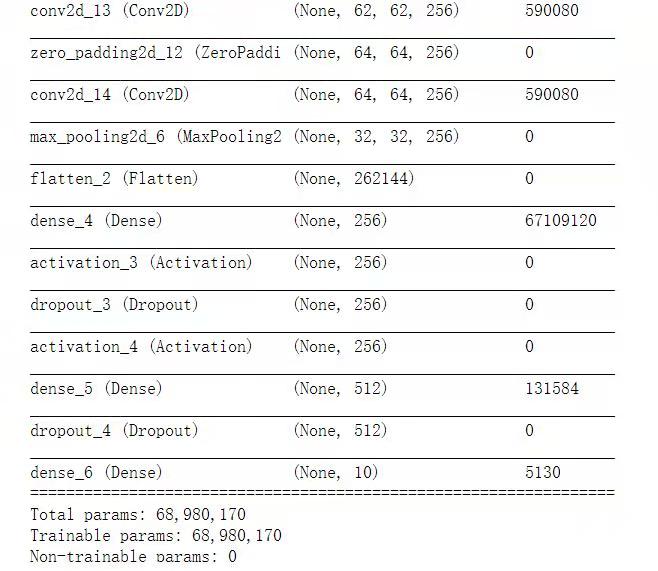


Fig.8 modified Vgg-16 structure.

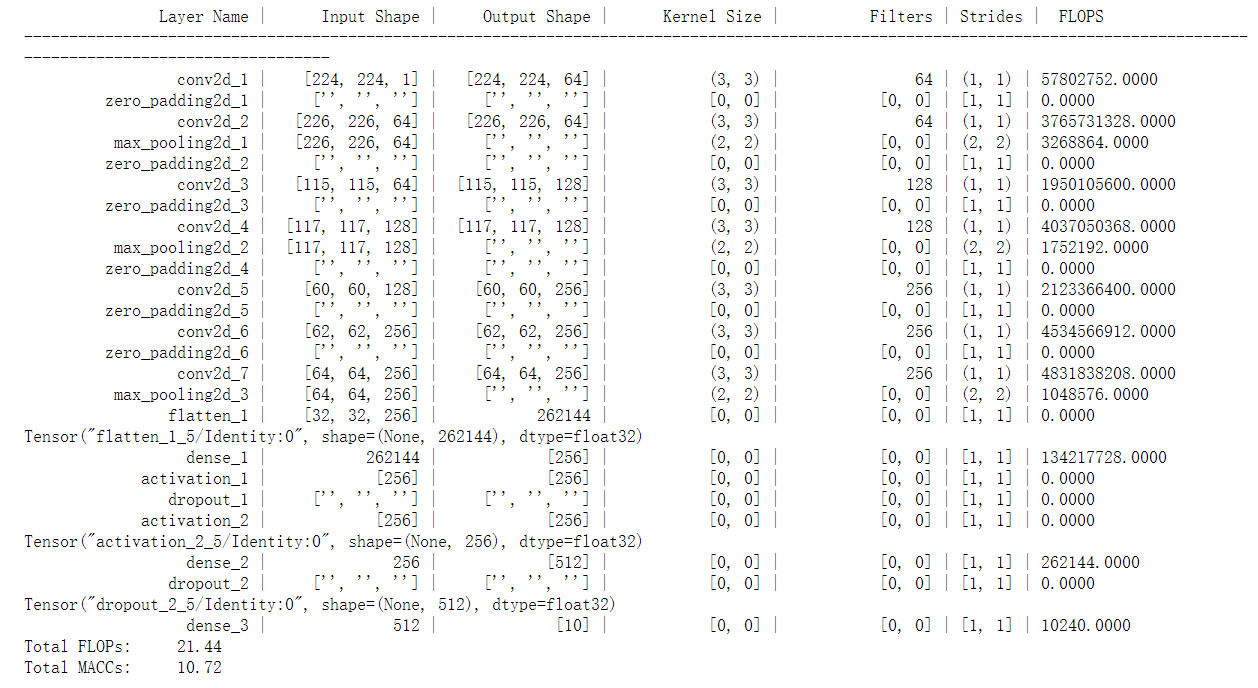


Fig9. FLOP Estimator

When we modified our models, firstly, we cut the channels number. Then we decrease the kernel size and dense layers number. We don’t want to cut too many convolution layers because in a great extent, the convolution layers can increase the accuracy, which have a good ability to express the feature. After modified the model. The FLOPs is decreased. Full name of FLOPs is floating point operations, which deternmine the training time and prediction time. Here’s the folumar of FLOPs in convolution layers. (Fig.10). M is feature map, K is kernel size, Cin is input Channle, Cout is output channel.

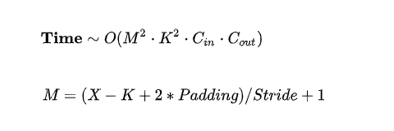


Fig10. FLOPs formular.

## Accuracy and loss analysis

To evaluate a machine learning model, the most important thing is accuracy and loss. Fig. 11 shows the accuracy and loss of AlexNet while the epoch number getting bigger. “+” follow the model name means the version we mortified. The red solid line, which represent the accuracy of the mortified model, show a sharp rising at the beginning of the training process. After the second epoch, it remains stable around 99%. But the dash one takes longer time for the same process, and only reach a lower accuracy around 96% stable level when it stops rising. It means that our modification and data preprocessing work.

Another line chart shows the same thing, the original AlexNet take more time to converge into a stale level and the final loss is higher than the mortified AlexNet. In a nutshell, no matter accuracy or loss, the mortified AlexNet perform better than the original AlexNet.

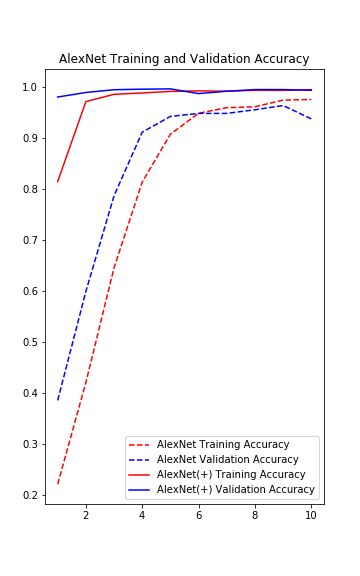
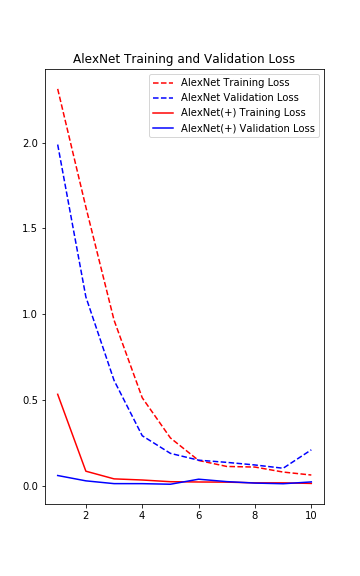


Fig. 11: AlexNet result

(+) means that the modified model

Figure 12. shows almost the same result when training VGG-16. The trend of the line is very similar to AlexNet. From the line chart we can easily find the first thing: no matter AlexNet or VGG-16, the solid one reaches the accuracy around 99%, perform better than the dash one that is the original model.

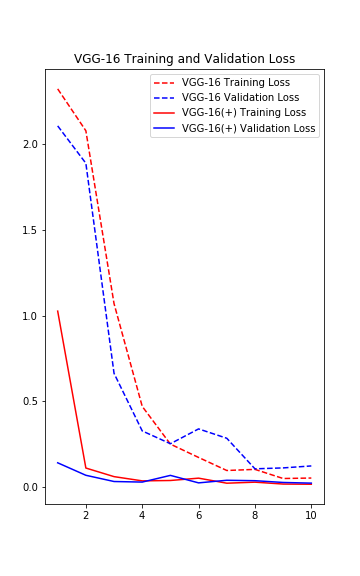
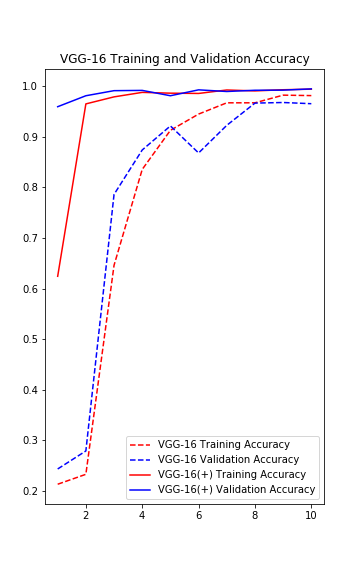


Fig. 18 VGG-12 result

(+) means that the modified model

## Time Consume Analysis (Train consuming)

After compare the accuracy and loss, we abandon the original model without data preprocessing. We only compare two modify model in the time consume analysis part.

Firstly, compare the train consuming. As Fig. 13 and Fig. 14 shows, we have 10 epochs for training, it takes AlexNet 40s for training while VGG-16 take 230s, that is almost 6 times than AlexNet.

The possible reason is the different number of parameters. In the previous chapter we mention that the number of parameters for AlexNet and VGG-16 are 25,813,834 and 68,980,170. The number of parameters of AlexNet is one-third of VGG-16 may be the cause that AlexNet faster than VGG-16.

文本

描述已自动生成

Fig. 13: Train consuming of AlexNet

文本

描述已自动生成

Fig. 14: Train consuming of VGG-16

## Time Consume Analysis (Recognize consuming)

Recognition time consumption is also a criterion for measuring a model. Especially when we need to build a real-time gesture recognition system, we need to minimize this part of the calculation time. As Fig .15 shows, AlexNet costs 0.03s for each image, while VGG costs 0.83s. VGG-16 takes 25 times time for prediction higher than AlexNet.

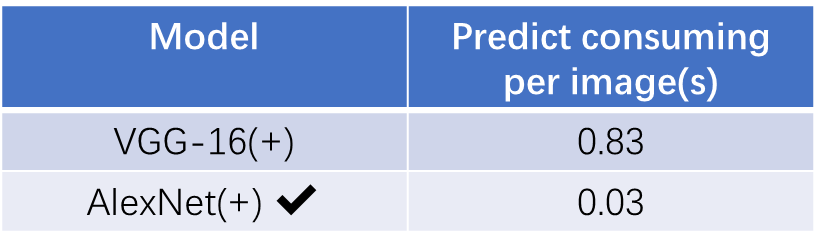


Fig. 15: Recognize consuming (up is the AlexNet+, down is the VGG+)

## Time complexity analysis

Although AlexNet has been selected as its time consuming in recognize is much lower than VGG-16. We briefly consider the reasons for this difference. Time complexity determines the training or prediction time of the model. If the complexity is too high, it will cause model training and prediction to consume a lot of time. Neither can it quickly verify ideas and improve models, nor can it make rapid predictions. FLOPs (floating point operations): Can be used to measure the complexity of the algorithm/model. We calculated the FLOPs of two model to explain why mortified VGG-16 need 7.5 times time than mortified AlexNet to recognize hand gesture.

In[13], the time complexity for one convolutional layer is:

Here is the spatial size of the output feature map, and is the width of each convolution kernel. is the number of input channels. is the number of output channels.

According to the formula, the time complexity for all convolutional layers in mortified AlexNet is . The time complexity for all convolutional layers in mortified VGG-16 is .

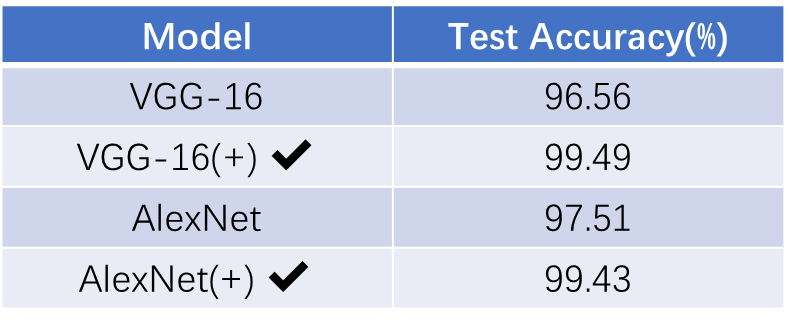
In[13], the time complexity for one dense layer is:

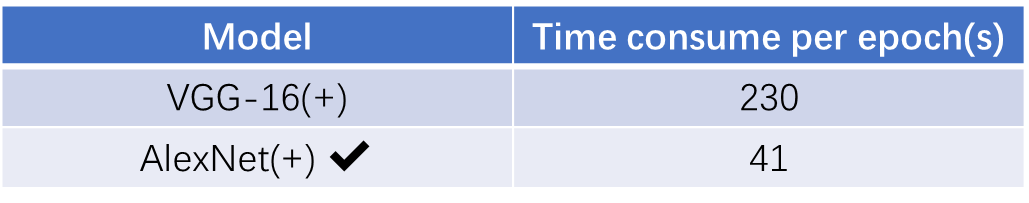
Here is input neuron numbers, is output neuron numbers.

According to the formula, the time complexity for all three dense layers in mortified AlexNet is . The time complexity for all three dense layers in mortified VGG-16 is .

Now, we have already had the time complexity of both models. After we sum the complexity time of dense and convolution layers. We found that the VGG-16’s time complexity is about 25 times as the AlexNet’s counterpart. Therefore, it close to the processing (prediction) time’s multiple (25).

## The best performing combination





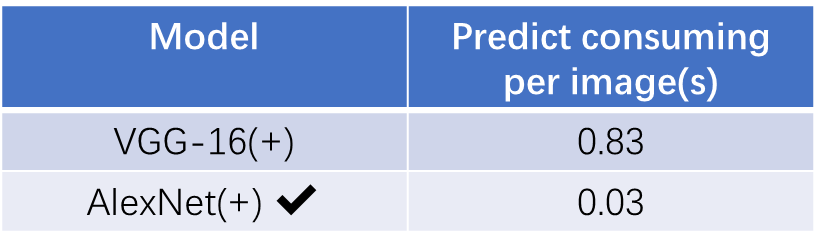


Table 1 Comparison of the three criteria for each model

(+) means that the modified model

According to the data in the Table 1 after aggregation, we finally choose one best combination that is modified AlexNet with data preprocessing as our hand recognize system method.

**Future work**

Now we have finished a static 2D hand gesture recognize system that can be used in the real time recognize. If we want more gesture, we can just simply add the new gesture in training set. But what if we want to recognize dynamic gesture? That mean a series of continuous hand gestures. Or even recognize hand gesture based on not only the pix, but also the depth that can recognize hand gesture in 3D. [14][15]

We also have some idea for complete these tasks. For example, combine hand gesture and its following movements, or use the feature that inside the contour. We may also do the data augmentation in the training process. Moreover, we can use more popular model likes Resnet or even non-CNN models.

**CONCLUSION:**

In this paper, we developed a human hand gesture recognition program based on the CNN model. We proposed a useful preprocessing method to process the hand gestures pictures and made them as the dataset. We changed the parameters and the structure of VGG and AlexNet and use them for training. Compared with the data set without preprocessing and the model without change, the accuracy after our preprocessing and model changes has been improved. By comparing the time consumption and accuracy in the experimental results, we came to the conclusion that Alextnet performed better in our data set. We use FLOPs to calculate the time complexity of the two models we changed and found that time complexity is the main reason that affects the training time and prediction time. We found that the accuracy of the original VGG-16 with single-channel in our data set was very low, so we trim the structure and change the parameters. This processing improved the accuracy, and we concluded that the complex space complexity and the data set with smaller features may be matched lead to over-simulation. And the more parameters of the model, the larger the amount of data required for training. What’s more, the deeper the model, although the representation ability becomes stronger, it may not necessarily improve accuracy.

**References**

1. Strezoski, G., Stojanovski, D., Dimitrovski, I., & Madjarov, G. (2016, September). Hand gesture recognition using deep convolutional neural networks. In International conference on ICT innovations (pp. 49-58). Springer, Cham.
2. Y. Liu, Z. Gan, Y Sun. Static hand gesture recognition and its application based on support vector machines. The Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel／Distributed Computing Phukgt, Thailand, 2008: 517-521
3. Y. Fang, K. Wang, J. Cheng, et. A real-time hand gesture recognition method. IEEE International Conference on Multimedia and Expo, Beijing, China, 2007: 995-998
4. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
5. Sushmita Mitra, Tinku Acharya (2007), “Gesture Recognition: A Survey.” IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and Reviews 37(3).
6. Siddharth S. Rautaray and Anupam Agrawal (2015), “Vision based Hand Gesture Recognition for Human Computer Interaction: A Survey.” Artificial Intelligence Review, Springer 43: 1–54.
7. Kaushik Deb and Ashraful Huq Sunny, *Shadow Detection and Removal Based on YCbCr Color Space (page3, p1-p4),* February, 2014.
8. Ravi Srisha and Am Khan, *Morphological Operations for Image Processing: Understanding and its Applications (page1, p2-p4)*, December, 2013.
9. Mei Wang, Jzau-Sheng Lin, and Guo Qing Meng, *Fingertip Detection and Gesture Recognition Based on Contour Approximation (page1, p1-p4)*, July, 2015.
10. Saeed Khaki, *HIGH-THROUGHPUT IMAGE-BASED PLANT STAND COUNT ESTIMATION USING CONVOLUTIONAL NEURAL NETWORKS (page2, p1-p5),* November 5, 2020
11. J Pyo, S Ji, S You, T Kuc, [*Depth-based hand gesture recognition using convolutional neural networks*](https://ieeexplore.ieee.org/abstract/document/7625742/), 2016 13th International Conference …, 2016 - ieeexplore.ieee.org*(page3, p2-p6),*
12. H. I. Lin, C. H. Cheng and W. K. Chen, "Learning a pick-and-place robot task from human demonstration", Proc. Int. Conf. Automat. Control. IEEE, pp. 312-317, 2013.
13. He, K. , & Sun, J. . (2015). Convolutional neural networks at constrained time cost.
14. Wang, Z. , Li, G. , & Yang, L. . (2020). Dynamic hand gesture recognition based on micro-doppler radar signatures using hidden gauss-markov models. IEEE Geoence and Remote Sensing Letters, PP(99), 1-1.
15. Camgoz, N. C. . (2017). Using Convolutional 3D Neural Networks for User-Independent Continuous Gesture Recognition. International Conference on Pattern Recognition. IEEE.