The Influence of Parameter Optimization of VGGNet on Model Performance in Terms of Classification Layers

Yizhen He

Joint Institute, Shanghai Jiao Tong University, Shanghai, 200240, China
h1zn666@sjtu.edu.cn

Abstract. This paper aims to explore the effect of parameter adjustment in classification layers of VGGNet. It provides suitable amounts of parameters for VGGNet with FC and FCN layers, which are available for reference. In the research, FER13 dataset, which contains gray-scaled images with shape of 48 x 48 with one channel is divided into training, validation and testing dataset. During the procedure of data processing, resizing, color jitter and flipping is adopted to the images to enhance training, and then the images are put into data loaders. As for the architecture of the model, for the reason of the limitation of time and computing capacity, the VGGNet is simplified. In detail, its width and height of convolutional layers are reduced. Also, for the classification layers, FC layers as well as FCN layers with different kernels are adopted. With Adam as optimizer and cross entropy as loss function, the accuracy of each model is tested and compared after training of 20 epochs. Experimental results show the suitable amounts of parameters with which the model has best performance. Also, the results indicate that FCN layers with smaller kernels have better performance than those with larger kernels.

Keywords: Machine learning, VGGNet, parameter adjustment.

1 Introduction

Nowadays, with the development of deep learning, facial expression analysis attracts much attention in recent years. Technology achieves this by detecting and recognizing the details on the face, ranging from the wrinkles to the angle of the corners of eyes as well as mouth. Not only can it be used to analyze human's psychological activities, but also can be applied in human-computer interaction. Therefore, technology is of great value and worth studying. However, due to the varieties of ages, races and genders, the task is faced with great challenges. Also, even at the same age, race and gender, facial expressions still differ from person to person for the different degree of expression exaggeration and facial muscle composition. As a result, facial expression analysis is never a simple problem. Therefore, the challenge of achieving high accuracy in recognition results emerges.

Currently, several models have been adopted to achieve facial expression recognition. For the reason that it is a problem about computer vision, Convolutional neural network (CNN) gains outstanding performance [1] due to their excellent performance

Y. Wang (ed.), *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Advances in Computer Science Research 115,
https://doi.org/10.2991/978-94-6463-540-9_89

in many tasks [2-5]. Furthermore, because of the complexity of facial expression, deep neural networks are preferred. Visual Geometry Group Network (VGGNet) [6] is a kind of CNN model with 3x3 kernel mainly. Its advantage is its relatively low parameter and also higher accuracy compared with single large kernel. Moreover, VGGNet tries to adopt fully-convolutional layers (FCN) (1 convolution layer with 7x7 kernel and 2 convolution layers with 1x1 kernel) to replace fully-connected (FC) layers. As is mentioned by the author, fully-convolutional layers can have receptive field on the whole feature map output by convolutional layers before.

In order to get higher accuracy or reduce the amounts of parameters, several networks are born, like GoogleNet, ResNet and DenseNet. These models adopt new strategies in the convolution layers, for example, ResNet avoids the problem of gradient vanishing by adding Residual block [7]. In this paper, in contrary, the purpose is to test the effect of adjusting the parameters in the model. The reason is that most of the past research is focused on discovering new architectures of models, but few are focused on the effect of parameters. Hence, this paper is focused on the adjustment of the parameter in fully-connected layers as well as fully-convolutional layers.

The contributions of this study can be summarized as comparing the performance between models with different parameters. In detail, two kinds of models are under research: first VGGNet with fully-connected layers and second VGGNet with fully-convolutional layers. For the model with fully-connected layers, the parameters in the fully-connected layers adjusted. Then, this paper compares the accuracy as well as the loss between the models with different parameters. Similarly for the model with fully-convolutional layers.

2 Methodology

2.1 Data Preprocessing

The dataset used in this research is facial expression recognition (FER-2013) dataset [8]. The images in it are gray scaled with shape of 48x48, and there are 7 classes, which represent 7 different emotions, from “angry” to “surprise”. To help split data set, data are put in “train” and “test” files, and the data in “train” file are split into 2 subsets for training and validation, and the validation part accounts for 20%.

The images are first transformed into “tensor” type for convenience for training. Also, color jitter and horizontal flipping are also performed to improve training effect. The shape of images is resized to 56x56, in order to better imitates VGGNet, because the input shape of VGGNet is 224, four times of 56. Then data is shuffled randomly, divided into batches with size of 32 and put into data loaders, of which the loading rate is about 30 times faster than original data loading implementation [9]. After the data processing, 16 sample images are provided (Fig. 1).

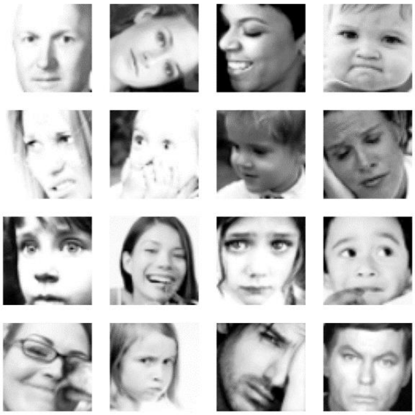

Fig. 1. Sample Images.

2.2 VGGNet Model

Facial expression recognition is a problem with computer vision. Thus, CNN stands out for its strength in feature extracting. A RGB image can be considered as a 3 channels x width x height matrix, and CNN is input with the matrix. First, convolutional layers apply filters to extract features from the input data through convolution operations. For example, 3x3 kernel multiply with the corresponding 3x3 spatial values and add up all the products (like Fig. 2).

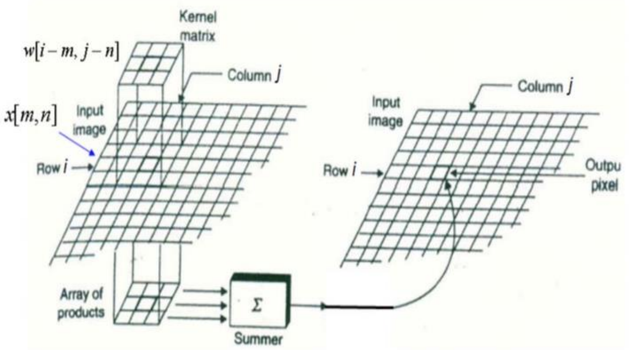

Fig. 2. Convolution Operation.

One convolution layer has several filters, so the layer actually output kernel maps with several channels. Second, Pooling layers like max-pooling or average-pooling are adopted between the convolution layers to reduce the spatial dimensions of the feature maps. For example, max-pooling layers "summarize" the feature map into smaller size by replacing the feature values in 3x3 spaces (if kernel is 3x3) with the largest value of them. At last, Fully-connected layers combine the extracted features to make predictions.

VGGNet is one type of CNN, which is proposed by Visual Geometry Group of University of Oxford in the 2014 ILSVRC competition and wins the 2nd place [6]. Small convolution kernel (3x3 most), small max-pooling kernel (2x2 all) and its wider and deeper convolutional layers are its characteristics. Its strength is that, when faced with complex problems, the wider and deeper layer can have good performance compared with other models. However, with wide and deep layers, the parameter in VGGNet is too large, which will bring about long training time. Compared with the packaged VGG16 model, the VGGNet model adopted in this research is simplified and there are several differences. First of all, the shape of images used is (1, 56, 56) instead of (3, 224, 224). What's more, the width and depth of VGGNet are reduced to a certain extent to reduce the number of parameters. The change is necessary for the limited time and GPU computational capacity.

In detail, the model contains 3 VGG blocks. Each VGG block contains convolutional layers (all with RELU activation) with kernel of 3x3 and padding of same. In addition, one VGG block has a max-pooling layer with kernel of 2x2 and stride of 2. The amounts of convolutional layers and the input as well as the output amounts of channels are set by a variable named convolutional architecture. For example, if the variable is (2, 1, 32), (2, 32, 64), (3, 64, 128), it means that the first block has two convolution layers with input channel 1 and output channel 32. Similarly, for the second and third blocks.

After the VGG blocks, the model (with fully-connected layers) flattens the (128, 7, 7) matrix and has fully-connected layers. The activation of each fully-connected layer follows the order of "Linear-RELU-Linear-RELU-Linear". What this paper adjusts is the unit of each fully-connected layer. To regularize the model and reduce overfitting, dropout layers are added after each FC layer with RELU activation, and the dropping rate is 0.5.

As for the model with fully-convolutional layers, as is mentioned in the introduction part, one convolutional layer with 7x7 kernel and two convolutional layers with 1x1 kernel are included in the fully-convolutional layers. What this paper adjusts is also the unit of each fully-convolutional layer. The reason why such full-convolutional layers can replace fully-connected layers is worth mentioning. The convolutional layers output feature maps with shape of 7x7 if the shape of images is 56x56. Then with a convolutional layer with kernel of 7x7 and no padding, the feature maps output is 1x1. Hence, the output of the model is numbers of 7x1x1 (if the output channel is 7), which can be squeezed to output the scores for 7 emotions. That's why the images are resized to 56x56. Otherwise, the output feature map becomes 6x6. To deeper explore the performance of FCN layers, architecture of three 3x3 kernels and two 1x1 kernels is also tested.

For the amounts of parameters in both fully-connected layers and fully-convolutional layers, the powers of 2, from 128 to 9192 are selected. VGGNets with different types of classification layers (fully-connected layers and fully-convolutional layers) and different amounts of parameters are the test targets of this paper.

2.3 Implementation Details

Adam optimizer [11] is used to compile the model, when the learning rate and the weight decay are both 0.0001. As for the loss function, cross entropy is adopted, which is widely used in classification problems. When training, images and labels are first put into CUDA as GPU has a much faster training rate. To reduce interference from initial weights, the seed of weights is fixed. In addition, each model is trained for 20 epochs, and accuracy is adopted as the evaluation metric. After training and validation of each epoch, the models work on the test dataset, and the accuracy is the criteria for evaluation.

3 Results and Discussion

To better show the performance of each model with different parameters, related results are provided in Table 1, Fig. 3, Fig. 4 and Fig. 5.

Table 1. The accuracy of different models

Parameter amount	Simplified VGGNet with FC layers	Simplified VGGNet with FCN layers* (kernel size: 711)	Simplified VGGNet with FCN layers (kernel size: 33311)
128	50.3%	48.7%	49.2%
256	50.2%	48.6%	53.1%
512	50.2%	49.5%	52.8%
1024	49.7%	50.6%	48.4%
2048	50.7%	49.4%	24.7%
4096	50.4%	49.3%	24.7%
9192	51.1%	49.6%	24.7%

* Simplified VGGNet with FCN layers (kernel size: 711)" means the FCN layers contain one convolutional layer with 7x7 kernel and two with 1x1 kernel. Similar for "kernel size: 33311"

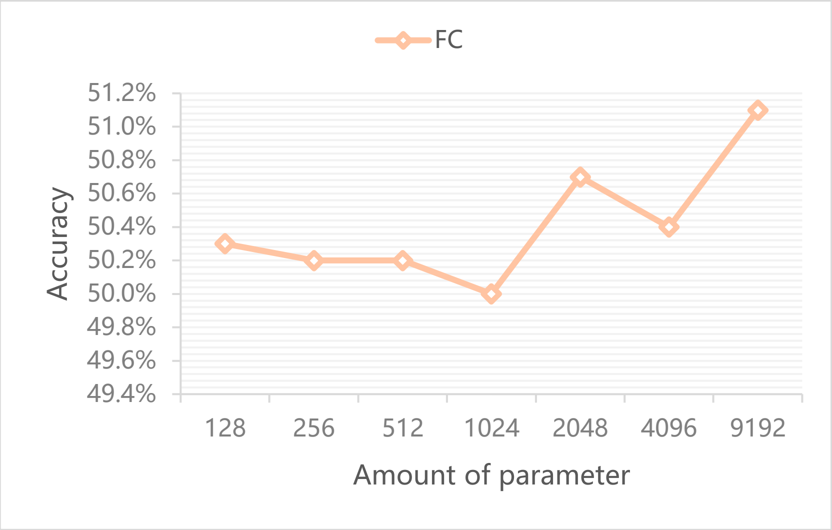

Fig. 3. The accuracy of VGGNet with FC layers.

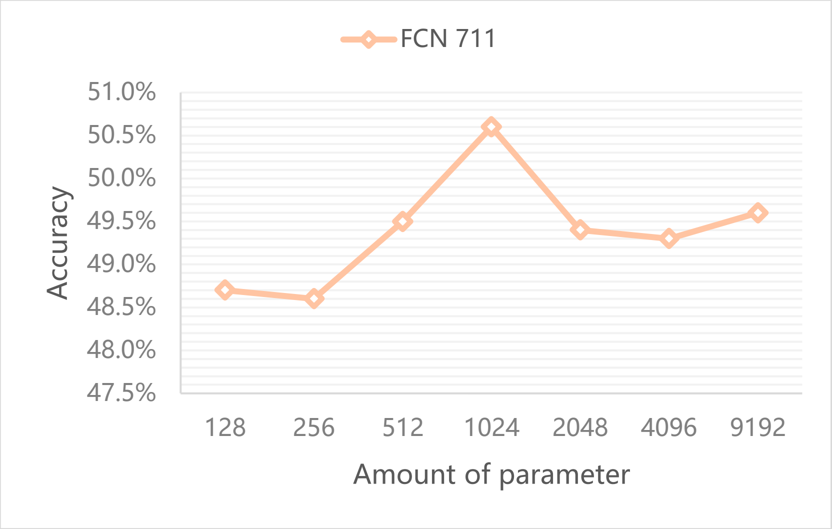

Fig. 4. The accuracy of VGGNet with FCN layers (kernel size: 711).

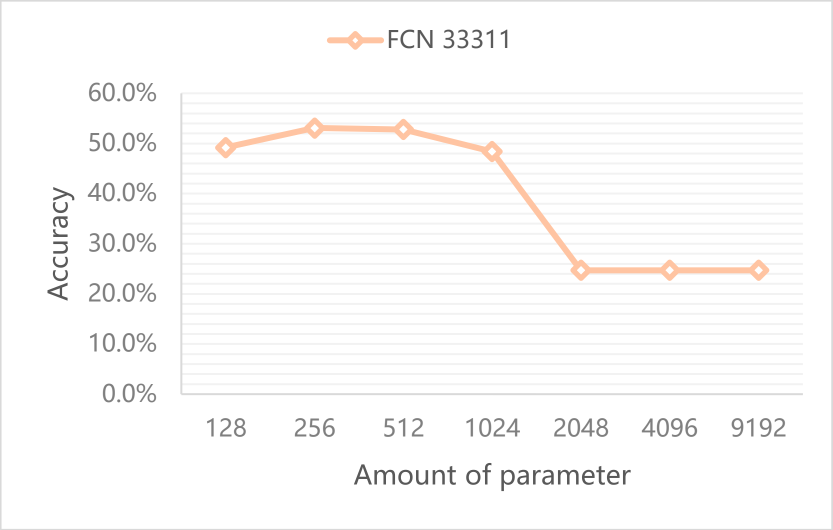

Fig. 5. The accuracy of VGGNet with FCN layers (kernel size: 33311).

As is shown in the results mentioned before, accuracy differs according to the number of parameters. In detail, for the VGGNet with FC layers, accuracy totally rises to a small extent when the parameter grows larger, though the accuracy drops in some ranges, which is within acceptable error range. To deeper explore the relationship between the accuracy and the amount of parameter, the performance of model with larger amount of parameter is tested, and the accuracy is numerically similar. For the VGGNet with FCN layers (kernel size: 711), the largest accuracy occurs when the amount of parameter is 1024. For that with kernel size 33311, the accuracy gets largest when the amount of parameter is 256. It is worth paying attention to that the accuracy stays small when the amount of parameter is too large.

The results indicate that, for the VGGNet with FC layers, when the amount of parameter grows, the accuracy rises to a certain extent. However, the improvement in accuracy is not significant (<1%). The reason is that when the amount of parameter grows to a certain extent, it is able to contain all the features needed to recognize emotions. After this, when the number of parameters keeps growing, there is no qualitative change happening. Thus, when building models with FC layers, the amount of parameter can be determined based on computing capacity, because larger amount bring a bit better performance but also longer training time.

For the VGGNet with FCN layers (kernel size: 711), larger amount of parameter can better deal with the features, but it is also more likely to cause overfitting. Under the influence of the two factors, the accuracy gets the largest when the amount is 1024. Hence, the model has the best performance when the amount of parameter is between 512 and 2048. When deciding the parameter in FCN layers with large kernel, 512, 1024 or 2048 may be the best choice.

For the VGGNet with FCN layers (kernel size: 33311), the model has good performance when the amount of parameter is small (get the best at 256, thus the best ranging from 128 to 512), but has bad performance when the amount grows too large, which may result from gradient vanishing. As a result, 128, 256 or 512 can be suitable for the parameter in FCN layers with small kernel.

FCN layers with kernel size of 33311 have better performance than those with size of 711. Though the former model suffers from gradient vanishing when the amount of parameter is too large, its largest accuracy is 2.5% larger than the latter model. With smaller kernels, to get good performance, smaller number of parameters are needed, which is preferred because it can reduce training time. Thus, when building FCN layers to classify, layers with small kernel may have better performance.

However, there are some limitations of this study: 1) the VGGNet used in this paper is simplified, so the accuracy cannot reach high value. Also, the amounts of parameters may be unable to be promoted to VGGNet for the difference of architecture. 2) The procedure of testing each model with a certain number of parameters should be repeated several times to reduce errors. In addition, the interval between the amounts of parameters should be reduced to more accurately find the relationship between the accuracy and the amount. 3) The results may only apply to facial expression recognition, because different problems have different images, and the amounts of features also differ. In the future, some more advanced methods will be considered to further optimize parameters and improve the performance of models [12-15].

4 Conclusion

This paper explores the relationship between the amounts of parameters and the performance of simplified VGGNet with FC and FCN layers. It shows the accuracy of the models with FC layers, FCN layers (kernel size: 711) and FCN layers with different amounts of parameters, from 128 to 9192, with exponential growth as interval. From the results, the ranges of the amounts can be inferred, with which models can have better performance. Also, the results demonstrate the demand for different models for parameter quantities. For example, large number of parameters can be adopted in the VGGNet with FC layers, but for that with FCN layers, with kernel size of 33311, large amount of parameter will lead to overfitting as well as gradient vanishing. In the future, with adequate time and computing capacity, adjustment of parameter for original VGGNet and other problems will be carried out. Also, each test will be repeated several times to reduce errors.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (2012).
2. Qiu, Y., Hui, Y., Zhao, P., Cai, C. H., Dai, B., Dou, J., Bhattacharya, S., Yu, J.: A novel image expression-driven modeling strategy for coke quality prediction in the smart cokemaking process. Energy 294, 130866 (2024).

3. Ye, X., Wu, P., Liu, A., Zhan, X., Wang, Z., Zhao, Y.: A deep learning-based method for automatic abnormal data detection: Case study for bridge structural health monitoring. International Journal of Structural Stability and Dynamics 23(11), 2350131 (2023).
4. Liu, Y., Bao, Y.: Intelligent monitoring of spatially-distributed cracks using distributed fiber optic sensors assisted by deep learning. Measurement 220, 113418 (2023).
5. Qiu, Y., Wang, J., Jin, Z., Chen, H., Zhang, M., Guo, L.: Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training. Biomedical Signal Processing and Control 72, 103323 (2022).
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778 (2016).
8. Kaggle FER2013: Available at https://www.kaggle.com/datasets/msambare/fer2013?select=test (2013).
9. Yang, C. C., Cong, G.: Accelerating data loading in deep neural network training. In 2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC), pp. 235-245 (2019).
10. Albawi, S., Mohammed, T. A., Al-Zawi, S.: Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), pp. 1-6 (2017).
11. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
12. Liu, Y., Liu, L., Yang, L., Hao, L., Bao, Y.: Measuring distance using ultra-wideband radio technology enhanced by extreme gradient boosting decision tree (XGBoost). Automation in Construction 126, 103678 (2021).
13. Qiu, Y., Wang, J.: A Machine Learning Approach to Credit Card Customer Segmentation for Economic Stability. In Proceedings of the 4th International Conference on Economic Management and Big Data Applications, ICEMBDA 2023, October 27–29, 2023, Tianjin, China, (2024).
14. Ye, X., Luo, K., Wang, H., Zhao, Y., Zhang, J., Liu, A.: An advanced AI-based lightweight two-stage underwater structural damage detection model. Advanced Engineering Informatics 62, 102553 (2024).
15. Hao, Y., Chen, Z., Jin, J., Sun, X.: Joint operation planning of drivers and trucks for semi-autonomous truck platooning. Transportmetrica A: Transport Science, 1-37 (2023).