A lightweight convolutional neural network model for target recognition

Chunqian He*, Dongsheng Li2 and Siqi Wang3

1College of Electronic Engineering, National University of Defence Technology, Hefei, Anhui, China
2College of Electronic Engineering, National University of Defence Technology, Hefei, Anhui, China
3Graduate school of engineering, The University of Tokyo, Bunkyo, Tokyo, Japan

*Corresponding author’s e-mail: chunqianhe@nudt.edu.cn

Abstract. Convolutional neural networks have achieved excellent performance in a wide range of applications, but the huge resource consumption makes a great challenge to their application on mobile terminals and embedded devices. In order to solve such problems, it is necessary to balance the size, speed and accuracy of the network model. This study proposed a new shallow neural network on the bases of ResNet and DenseNet. We use different size convolution kernels to obtain feature maps and then concat them. Afterwards we build two convolution layers to reduce the size of the feature maps and increase the depth of the network. By stacking this structure, we get our net model. Experiments show that our nine-layers network recognition performance is better than 18-layers ResNet and 19-layers DenseNet, and its training time is shorter. The final recognition rate of our network is 97.37%, ResNet recognition rate is 96.93%, and DenseNet is 96.31%.

1. Introduction
In recent years, Convolutional Neural Networks (CNN) have been widely used in computer vision, natural language processing, data mining, and have achieved excellent performance. These technological breakthroughs are closely related to the huge amount of data and powerful computing resources. For example, AlexNet [1] has made great progress in the field of natural image recognition. It uses about 1.2 million images for training on multiple computing devices. Since then, people have realized that the performance of CNN is better than other methods, and the demand for its applications has been continuously improved. CNN's computational complexity and storage requirements have also increased dramatically. Networks such as VGG [2] and GoogleNet [3] require more than 100MB of storage space and hundreds of millions of calculation operations. The emergence of GPU and CPU clusters makes it possible to train a powerful CNN in a reasonable time. In recent years, people have made tremendous progress in mobile devices such as unmanned driving, unmanned aerial vehicles, smart watches, smart glasses. The need to use CNN models on these devices has become more and more strongly. However, these devices are very scarce in resources such as storage capacity, computing unit, and battery power, so it is important to design a lightweight network with less computational complexity and parameter volume.

The current research on lightweight network models can mainly divided into network structure design and model compression. Network structure design focuses on optimizing the spatial...
The convolution operation method, designing special structured operation core or lightweight computing unit to reduce model parameters and complexity. Model compression is to decrease the calculation of the model through compressing the model parameters. In recent years, some excellent network models have been proposed. SqueezeNet [4] uses $1 \times 1$ convolution and group convolution methods; MobileNetV1 [5] proposed a deep separable convolution method; MobileNetV2 [6] added residual structure and linear bottleneck structure on the basis of MobileNetV1; ShuffleNetV1 [7] proposed an approach of group convolution and Channel Shuffle. These new net models reduce the size of the network and improve its operational efficiency. To meet the demands of using CNN on mobile devices, this study proposed a new shallow neural network on the bases of ResNet [8] and DenseNet [9]. Our network not only has fewer layers and fewer parameters than DenseNet and ResNet, but also has a better recognition performance.

This article proceeds as follows. Section 2 introduces the dataset and the main networks of our experiment. Section 3 presents our experimental results, comparing different networks performance and characteristics. Finally, Section 4 sets forth our conclusions.

2. Materials and Methods
In this part we will clarify the dataset, and then have a quick preview of ResNet, DenseNet and our proposed lightweight network sequentially.

2.1. Pokémon dataset
The dataset used in this experiment is collected from the network, which called Pokémon. We collected five types of Pokémon pictures. These Pokémon have different physical characteristics, colors, sizes, backgrounds. As shown in figure1, from left to right is Bulbasaur, Chamander, Mewtwo, Pikachu and Squirtle.

![Figure 1: Sample pictures of five Pokémon](image)

| Dataset   | Mewtwo | Squirtle | Bulbasaur | Pikachu | Charmander |
|-----------|--------|----------|-----------|---------|------------|
| Training  | 143    | 128      | 140       | 136     | 137        |
| Validation| 48     | 43       | 47        | 45      | 45         |
| Test      | 48     | 43       | 47        | 45      | 45         |
| Total     | 239    | 214      | 234       | 226     | 227        |

Table 1 gives the details of the Pokémon dataset. Each type of Pokémon has about two hundred pictures and we divide them into three parts. The first part is the training set, which includes 60% images and is used to train the neural network. The second part is the verification set, which possesses 20% pictures and is used to evaluate the performance of the network. The last 20% of the pictures are the test set which is used to test the accuracy of network classification.

2.2. ResNet
ResNet[8] is a convolutional neural network proposed by He Kaiming in 2016. This network introduces shortcut structure, which speeds up the network training process while reducing complexity. It makes the network lightweight. In this experiment, we use ResNet18 (18 refers to the number of convolution layers, which represents the depth of the network) as a comparison network. Its structure...
is shown in figure 2 (a). Basic block (figure 2 (b)) is the basic module of the ResNet, which includes a shortcut structure. The network depth is obtained through stacking basic blocks.

2.3. DenseNet

DenseNet [9] was proposed by Huang on 2017. Compared to ResNet, DenseNet has a more aggressive dense connection mechanism: connecting all layers to each other. Each layer will accept all the layers before it as its additional input. Figure 3(a) shows dense connection mechanism. ResNet is a short-circuit connection between one layer and a previous layer (usually 2~3 layers), its connection method is element-level addition. In DenseNet, each layer output will be concatenated with the previous layers’ outputs in the channel dimension (the feature map size of each layer is the same), and then used as the input of the next layer. The feature maps of DenseNet come from different layers is directly concatenate, which can achieve feature reuse and improve efficiency. The DenseNet network structure used in this experiment is shown in figure 3(b). We simplified the network structure and only stacked 3 dense blocks. The three dense blocks’ layers are 4, 8, and 6.
2.4. MyNet

We propose a new lightweight network that combines the advantages of DenseNet and ResNet. As shown in the figure 4, the first step is to convolve the input image with convolution kernels of different sizes (7, 5, 3). It is to obtain different feature maps. Then we concat these feature maps in the channel dimension. Next, we use two convolutional layers to control the number of channels and the size of the feature maps. This is the module of our network. The complete structure of Mynet stack the module 3 times. As the feature map size decreases, the convolution kernel used in the module in the first step need to be adjusted. In stack 2, the kernel size is (5, 3), and in stack 3 is 3. The advantage of this network is that it uses different convolution kernels to obtain feature maps, which improved the quality of feature maps, and at the same time, the feature maps come from previous module direct concatenate makes feature reuse efficient.

3. Experiment and discussion
We train three networks and compared their training process, performance and complexity.
As shown in figure 5, the abscissa indicates the epoch of training set, and the ordinate indicates the network recognition rate. The three network training processes are roughly similar. They have a relatively good effect in the twelfth epoch and their final recognition rates are around 93%. We set the learning rate decay with the epoch in training progress, which makes the training slow down but can find the optimal network parameters.

Table 2: Recognition rate of three networks on each dataset

| Dataset  | DenseNet | ResNet | Mynet |
|----------|----------|--------|-------|
| Test     | 93.99    | 93.86  | 94.20 |
| Validation | 93.42    | 93.58  | 94.73 |
| Overall  | 96.31    | 96.93  | 97.37 |

Table 2 shows the recognition capabilities of the three networks. The high recognition rate obtains on the overall dataset is that the training set accounts for 60% of the entire data, and the network
training is performed on it. Figure 6 intuitively compares the performance of the three networks on each dataset, Mynet is slightly better than ResNet and DenseNet.

Table 3: Recognition results using Mynet

| Pokémon     | Bulbasaur | Charmander | Mewtwo | Pikachu | Squirtle | Total |
|-------------|-----------|------------|--------|---------|---------|-------|
| Bulbasaur   | 225       | 0          | 0      | 0       | 9       | 234   |
| Charmander  | 0         | 217        | 0      | 6       | 4       | 227   |
| Mewtwo      | 1         | 0          | 237    | 0       | 1       | 239   |
| Pikachu     | 2         | 0          | 0      | 220     | 2       | 226   |
| Squirtle    | 1         | 1          | 2      | 0       | 210     | 214   |
| Recognition rate (%): | 96.15 | 95.59 | 99.16 | 97.35 | 98.13 | \ |

Average target recognition accuracy: 97.37%

Figure 7: False recognition of the three networks

Table 3 lists the recognition performance of Mynet on each type of target in detail. The numbers on the diagonal are the number of correctly recognized pictures, and the rest are false recognition. The final recognition rate of Mynet is 97.37%, ResNet recognition rate is 96.93%, and DenseNet is 96.31%. Figure 7 compares the number of misrecognitions of the three networks under each type of target, Mynet performs better obviously.

Finally, we compare the total parameter numbers of the three networks, the network depth and the training time. The network depth refers to the number of convolutional layers. The comparison results are shown in Table 4. It can be seen that the number of Mynet parameters is only 37% of ResNet, and its depth is only half of the other two networks. In summary, in the Pokémon dataset recognition task, Mynet achieved better performance than ResNet and DenseNet with a shallower depth and a smaller number of parameters.

Table 4: Comparison of network characteristics

| Network model | DenseNet | ResNet | Mynet |
|---------------|----------|--------|-------|
| Parameter number | 9360     | 38592  | 14500 |
| Training time  | 10min27s | 11min42s | 9min29s |
| Network depth  | 19       | 18     | 9     |

4. Conclusion

In this paper, we propose a new lightweight convolutional neural network with a depth of only 9, which can achieve good recognition results. One reason is that we use different size convolution
kernels to extract feature maps. The other reason is that concat is used to improve the feature maps utilization. The comparison with ResNet and DenseNet also proves the superiority of ours network. In future work we will increase the depth of the network and look forward to have better performance.

Acknowledgments
The authors would like to thank the editors and anonymous reviewers for their insightful comments and suggestions, which could significantly help in improving the quality of this paper.

References
[1] Morgado P, Vasconcelos N. Semantically consistent regularization for zero-shot recognition. 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, July 21-26, 2017. Los Alamitos, 2017: 2037-2046
[2] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556,2014.
[3] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 7-12, 2015. Piscataway, 2015:1-9
[4] Iandola F N, Han S, Moskewicz, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5Mb model size [J]. arXiv preprint arXiv:1602.07360. pdf.
[5] Howard A G, Zhu M L, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017
[6] Sandler M, Haward A, Zhu M L, et al. MobileNetV2: inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 18-23, 2018. Los Alamitos, 2018:4510-4520
[7] Zhang Xiangyu, Zhou Xinyu, Lin Mengxiao, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 18-23, 2018. Los Alamitos, 2018:6848-6856.
[8] He K, Zhang X, Ren S.et al. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition. Seattle, WA, June 27-30, 2016: 770-778.
[9] Huang G, Liu Z, et al. Densely Connected Convolutional Networks. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition. Honolulu, HI, July 21-26, 2017:2261-2269.