Research on Remote Sensing Image Classification Based on Lightweight Convolutional Neural Network

Zhengwu Yuan and Xinjie Liu(✉)

Chongqing University of Posts and Telecommunications, Chongqing, China
624561901@qq.com

Abstract. There is an increasing demand for remote sensing image classification in many civilian applications. Inputting the feature map into the convolutional neural network can obtain more abstract features of the input feature map, and this ability can be applied to remote sensing image classification. The popularity of portable devices makes the network model develop in the direction of light weight. Therefore, this paper uses MobileNetV3 as the basic network, applies convolutional neural network to remote sensing image classification, and uses grouped convolution to group input features. In order to reduce the parameters of the model, a lightweight attention mechanism is used, and the convolution operation of different receptive fields is used in this paper to extract image features. The main purpose of this paper is to use convolutional neural networks on portable devices, and to ensure the accuracy of the network to a certain extent. The experimental results show that the parameter amount of the model has changed from 20.92M to 5.66M after several processing, which is only a quarter of the original, but its accuracy rate better.

Keywords: lightweight convolutional neural network · remote sensing image classification · grouped convolution · receptive field · attention mechanism

1 Introduction

In remote sensing applications, remote sensing image classification is a very important part. By classifying remote sensing images, it can promote the application of remote sensing technology in various fields such as developing urban and rural construction, strengthening military and national defense, and conducting resource surveys. Remote sensing can investigate mineral resources, monitor the epidemic situation in various places and analyze natural disasters and other civilian purposes. Therefore, the accuracy of remote sensing image classification has a great influence on the development of remote sensing. In recent years, deep learning has gradually become a research hotspot and has been widely used in the field of remote sensing [1].

Researchers have developed convolutional neural networks with superior performance such as ZFNet [2], SENet [3] and DenseNet [4].

N. Radojević et al. (Eds.): ICAID 2022, AHIS 7, pp. 127–138, 2023.
https://doi.org/10.2991/978-94-6463-010-7_15

The advent of the Internet era has led to the popularization and development of smartphones and other portable devices. In order to apply convolutional neural networks on these devices, lightweight networks such as MobileNet [5–7], SqueezeNet [8], and ShuffleNet [9, 10] have been successively developed.

2 Related Work

Scene classification was first to use hand-crafted feature descriptors to extract the features of remote sensing scenes, but unsupervised feature learning methods emerged due to the inability to automatically extract image features by hand-made, but unsupervised learning can only extract shallow image features, and it is not well applied to remote sensing scene classification. In recent years, deep learning has gradually become a research hotspot. In [11], Cheng G et al. conducted experiments on the NWPU-RESISC45 dataset based on VGG [12], AlexNet [13] and GoogleNet [14] and found that the use of convolutional neural networks can greatly improve the classification accuracy. In addition, convolutional neural networks have also made great progress.

AlexNet used dual GPU mode to improve the running speed and running scale. The VGG network continuously extracted the feature map by using a small convolution kernel. The Mobilenet V1 network uses a depthwise separable convolution method to extract feature information. Mobilenet V2 used a residual structure on the basis of Mobilenet V1. EspNet [15] considers the effect of receptive field on the model.

In 2019, Mobilenet V3 advanced the average pooling layer of the V2 version, reducing the depth of the network and reducing the number of channels in the convolutional part. Secondly, the author replaced part of the Relu function with the Hard_Swish activation function, which improved the calculation accuracy. An attention mechanism is also used in the V3 network to increase the performance of the network.

3 Methodology

3.1 Group Convolution

In Fig. 1, the left picture is a common convolution operation, and the right picture is a schematic diagram of using grouped convolution to process feature maps.

In order to effectively reduce the number of parameters and reduce the computational cost, the grouped convolution divides the feature map into several groups, and then performs convolution operations on each group independently. Assuming that the information of $C \times H \times W$ is input, the output channel is N, and the size of the convolution kernel is $K \times K$, then, $C \times K \times K \times N$ is the parameter amount of the conventional operation. However, if grouped convolution is used, the feature map is first divided into G groups, then each group needs to process C/G channels, the output channels are also G groups, each group is C/G, and the convolution operation becomes $N \times K \times K \times C/G$ convolution kernel. Each group uses N/G convolution kernels, then the parameter amount of group convolution is $N \times K \times K \times C/G$.

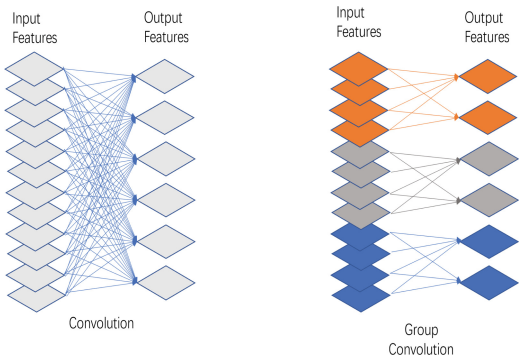


Fig. 1. Convolution comparison chart.

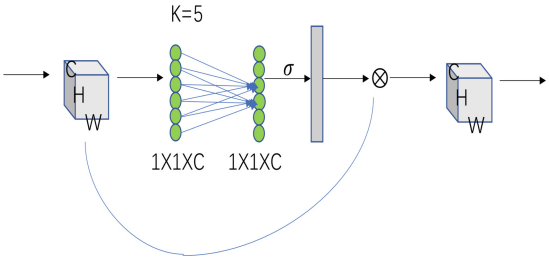


Fig. 2. The attention mechanism.

3.2 Lightweight Channel Attention Mechanism

MobilenetV3 has a significant improvement in accuracy compared to the V2 version, the number of parameters is about 2M more than that of the V2 version. Therefore, we need to make lightweight improvements to it. In the SE module, in order to realize the cross-channel interaction of each feature channel, two fully connected layers are added, but the experiment found that it is not necessary to interact with each channel.

Via the formula:

$$f_{\{W_1,W_2\}}(\mathrm{y}) = \mathrm{W}_2 \mathrm{Re}LU(W_1 y) \tag{1}$$

Among them, w represents the weight. We can find that there is an indirect correspondence between the channel and its weight. In order to reduce the amount of calculation between the fully connected layers, and to reduce the complexity of the model, in this paper, we use a method to capture only the information interaction of local channels.

The attention mechanism is shown in Fig. 2.

The difference between the two attention mechanisms is only reflected in the difference in the information captured by the fully connected layer. However, in the study, it was found that in the SE module, reducing the number of channels by reducing the number of parameters in the first fully connected layer will affect the weights obtained, so we do not perform dimensionality reduction in the first fully connected layer.

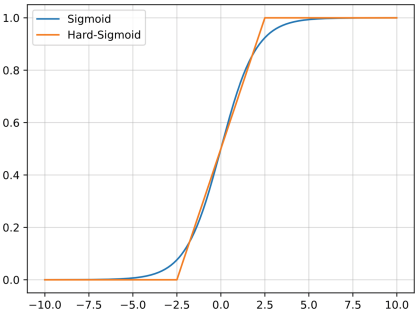


Fig. 3. Function comparison chart.

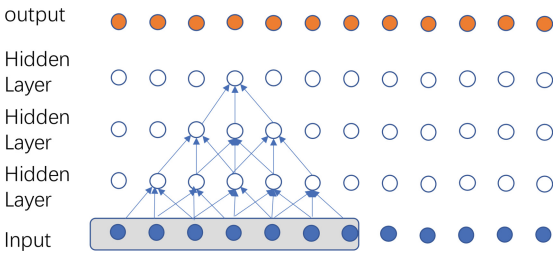


Fig. 4. Receptive field.

To speed up the training, we replace the normalized Sigmoid activation function in the SE module with the Hard_Sigmoid activation function. The formula for the Hard_Sigmoid activation function is:

$$f(x) = \begin{cases} 1, (x > 3) \\ \frac{x}{6} + 0.5, (3 \geq x \geq -3) \\ 0, (x < -3) \end{cases} \tag{2}$$

The two functions are basically the same in shape and direction, but Hard_Sigmoid only uses a simple piecewise function, the training speed has been greatly improved.

Comparison of the two images (Fig. 3).

3.3 Receptive Field

In recent years, research has gradually focused on the impact of receptive fields on the network. In convolutional neural networks, the receptive field is how many pixels in the input feature affect the output of the feature. For a more intuitive introduction, take the one-dimensional graph of the three-layer convolutional neural network below as an example. After three 3×3 convolutions, the top receptive field is 7×7 (Fig. 4).

The formula for calculating the receptive field is as follows:

$$RF_{l+1} = RF_l + (\mathrm{k}_{l+1} - 1) \times s_l \tag{3}$$

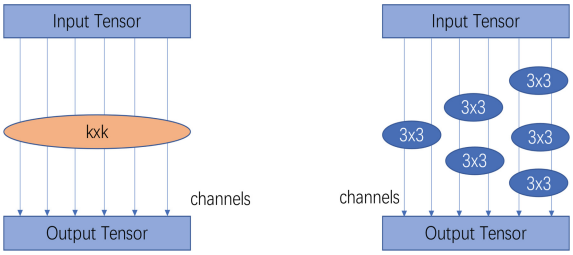


Fig. 5. Feature extraction unit.

Among them, the number of layers is represented by l, RF_l is the size of the receptive field of the previous layer, k is the size of the convolution kernel, and s is the step size of the convolution kernel. At first input, l equals 0, RF_0 equals 1, s_0 equals 1.

According to the formula, taking the convolution kernel 3×3 as an example, it can be calculated that the receptive field of the first layer is 3.

$$RF_1 = RF_0 + (k_1 - 1) \times s_0 = 1 + (3 - 1) \times 1 = 3 \tag{4}$$

In the second layer feature map, the receptive field is 5.

$$RF_2 = RF_1 + (k_2 - 1) \times s_1 = 3 + (3 - 1) \times 1 = 5 \tag{5}$$

The main reason why a small convolution kernel can be used to replace a large convolution kernel in VGG to reduce the amount of parameters is that after multiple small convolution kernels are superimposed, the size of the receptive field of the two is the same. In traditional training, the larger the receptive field, the deeper the network, and the better the classification effect can be achieved.

In this paper, the receptive field is enlarged by stacking small convolution kernels. In the research, it is found that when using a smaller receptive field, although the parameter amount of the network model will be reduced, it will affect the accuracy of classification. On the contrary, using a convolution kernel with a large receptive field to extract features will improve the accuracy of the network model, but will increase the number of parameters. Therefore, this paper designs the following feature extraction unit in combination with grouped convolution (Fig. 5).

The first image is the traditional convolution mode, where the input feature map is operated on by the same convolution kernel. The second picture is the convolution pattern we designed. In the figure on the right, firstly, the feature map is divided into equal amounts, and then the convolution operation is performed in groups. The convolution operations of each group are independent of each other and have different receptive fields. Although they all use 3×3 convolution kernels, by using the method of stacking multiple convolution operations can not only increase the range of the receptive field, improve the recognition rate of the network model, but also ensure that too many parameters are not involved in the convolution process.

4 Experiment and Result Analysis

4.1 Convolutional Neural Network Model

This paper uses MobilenetV3 as the prototype network, and the CNN model diagram is as in Fig. 6.

The large convolution kernel in Fig. 6 is formed by superimposing 3 × 3 convolution kernels.

4.2 Data Sets

In this paper, a total of three datasets are used, namely NWPU dataset, AID dataset and UC Merced Land-Use dataset. The AID dataset is a remote sensing image dataset released by Wuhan University and Huake. It includes 30 categories. NWPU is a remote sensing image dataset created by Northwestern Polytechnical University. There are a total of 31,500 remote sensing scene images. The UCM dataset mainly includes 21 categories such as buildings, forests, and airplanes. Each category has 100 remote sensing scenes of 256 × 256 image (Table 1).

4.3 The Setting of Experiments

In this paper, all experiments are carried out using the Pytorch framework. The graphics card is an NVIDIA GeForce GTX 1660 Ti with 6 GB of memory. The version number of Pytorch is 1.7.0. Python is version 3.6.0.

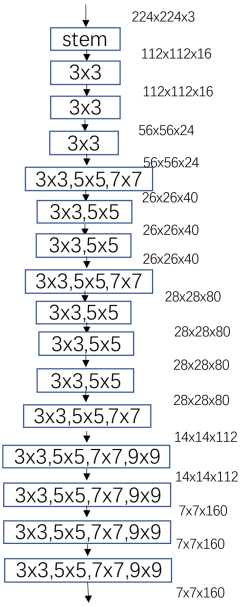
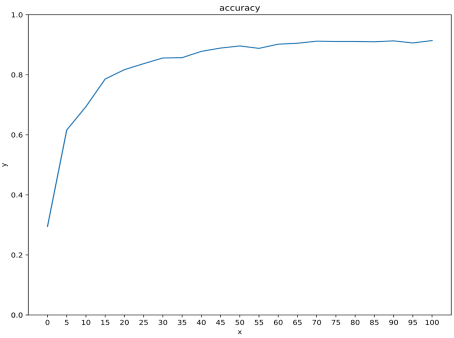


Fig. 6. Model structure.

Table 1. Sample image of the dataset

Airport	BareLand	Baseball Field	Beach



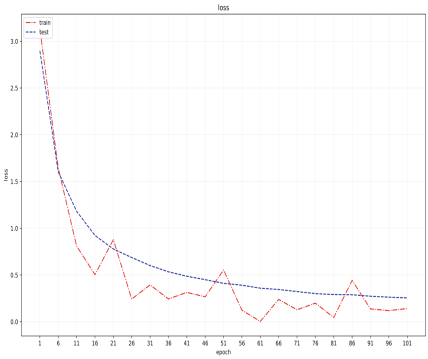


Fig. 7. Experimental results of NWPU.

4.4 Comparative Experiments and Analysis

Figures 7 and 8 are the data display figures of the network in the three remote sensing image datasets of NWPU and UCM.

In the accuracy image, the x-axis represents the training epoch, and the y-axis is its accuracy. It can be seen intuitively from the figure that the accuracy of the network improves with the increase of epoch and finally achieves good results. The data comparison chart is the comparison of training loss and test loss. It can be seen from the image that both gradually decrease with the increase of training times, and there is no

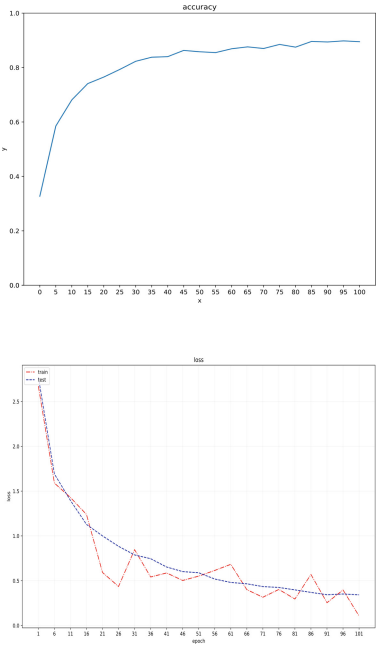


Fig. 8. Experimental results of AID.

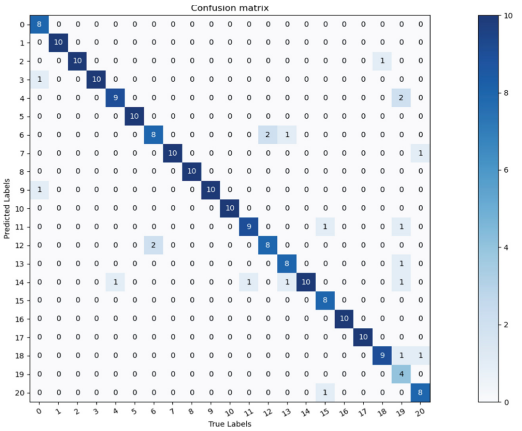


Fig. 9. Confusion Matrix for UCM.

situation that one side is large and the other is small in terms of value, which shows that when training the network, it is a normal training process, and there is no overfitting or underfitting.

This paper still uses the confusion matrix to view the effect of the network model on the validation set.

We can see from the Fig. 9 that the overall performance of the model is good when using this dataset.

This paper compares the improved model with three convolutional neural networks, shuffleNet, MobilenetV2 and ResNet. The experimental results are as in Tables 2, 3 and 4.

From the experimental results, we can see that the improved network model not only reduces the number of parameters by nearly three-quarters, but also compared with the previous MobilenetV3 network model, whether on NWPU, AID or UCM datasets, the network in this paper still has the advantage in accuracy. Compared with the lightweight convolutional neural network, the improved network not only has a smaller amount of parameters, but also has a certain improvement in accuracy. However, compared with the non-lightweight convolutional neural network, although the model in this paper has obvious advantages in the amount of parameters, there is still a certain gap with the ResNet network in terms of accuracy.

Table 2. Experimental results on NWPU

Model	Accuracy (%)	Params (M)
shuffleNet	88.3	8.69
MobilenetV2	89.9	13.37
ResNet	91.7	83.15
MobilenetV3	91.3	20.92
Ours	91.4	5.66

Table 3. Experimental results on AID

Model	Accuracy (%)	Params (M)
shuffleNet	87.8	8.69
MobilenetV2	88.2	13.37
ResNet	90.3	83.15
MobilenetV3	89.4	20.92
Ours	89.8	5.66

Table 4. Experimental results on UCM

Model	Accuracy (%)	Params (M)
shuffleNet	80.4	8.69
MobilenetV2	86.8	13.37
ResNet	91	83.15
MobilenetV3	89.8	20.92
Ours	90.2	5.66

Table 5. Experimental results on NWPU

Model	Accuracy (%)	Params (M)
MBV3	91.3	20.92
MBV3_SE	91.1	15.15
MBV3_G	90.7	7.65
MBV3_SE_G	91.4	5.66

Table 6. Experimental results on AID

Model	Accuracy (%)	Params (M)
MobilenetV3	89.4	20.92
MBV3_SE	88.2	15.15
MBV3_G	89.3	7.65
MBV3_SE_G	89.8	5.66

Table 7. Experimental results on UCM

Model	Accuracy (%)	Params (M)
MobilenetV3	89.8	20.92
MB_SE	89.9	15.15
MBV3_G	90.5	7.65
MBV3_SE_G	90.2	5.66

4.5 Ablation Experiment

In this paper, the original network, the improved attention mechanism and the network after the improved receptive field are compared.

MB_SE refers to the network after improving the attention mechanism. MB_G is the network after improving the receptive field. MB_SE_G refers to using both of the above methods to improve the network. Tables 5, 6 and 7 show that the number of parameters is reduced from the original 20.92M to 5.66M, and the accuracy rate is also slightly improved.

5 Conclusion and Discussion

In this paper, based on MobilenetV3, a more lightweight attention mechanism and feature extraction method are used to classify remote sensing images. It can be seen from the experimental results that not only the amount of parameters is reduced, but also the

accuracy is slightly improved. However, there is still a certain gap between the accuracy and traditional convolution, and future work will further improve the accuracy of the model.

References

1. Li E, Xia J, Du P, Lin C, Samat A (2017) Integrating multilayer features of convolutional neural networks for remote sensing scene classification. IEEE Trans Geosci Remote Sens 55(10)
2. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) ECCV 2014, vol 8689. LNCS. Springer, Cham, pp 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
3. Hu J, Shen L, Albanie S et al (2017) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell
4. Huang G, Liu Z, Maaten LVD, Weinberger KQ (2017) Densely connected convolutional networks. In 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, pp 2261–2269
5. Howard AG et al (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 432–445
6. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
7. Howard A, Sandler M, Chu G et al (2019) Searching for mobilenetv3. arXiv:1905.02244
8. Iandola FN et al (2017) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. In: 5th international conference on learning representations
9. Zhang X, Zhou X, Lin M, Sun J (2018) ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6848–6856
10. Ma N, Zhang X, Zheng H, Sun J (2018) ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Proceedings of the European conference on computer vision, pp 116–131
11. Cheng G, Han J, Lu X (2017) Remote sensing image scene classification: benchmark and state of the art. Proc IEEE 105(10):1865–1883
12. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd international conference on learning representations
13. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. In: NIPS. Curran Associates Inc.
14. Szegedy C et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
15. Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) ESPNet: efficient spatial pyramid of dilated convolutions for semantic segmentation. In: Proceedings of the European conference on computer vision, pp 552–568