Hindawi
Advances in Multimedia
Volume 2022, Article ID 9973814, 9 pages
https://doi.org/10.1155/2022/9973814

Research Article

Self-Attention Multilayer Feature Fusion Based on Long Connection

Chu Yuezhong, Wang Jiaqing, and Liu Heng

School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, China

Correspondence should be addressed to Chu Yuezhong; mychu@126.com

Received 7 March 2022; Accepted 22 April 2022; Published 4 May 2022

Academic Editor: Anil Kumar

Feature fusion is an important part of building high-precision convolutional neural networks. In the field of image classification, though widely used in processing multiscale features of the same layer and short connections in the same receptive field, feature fusion is rarely used in long connection operations across receptive fields. In order to fuse the high- and low-level features of image classification, a feature fusion module SCFF (selective cross-layer feature fusion) for long connections is designed in this work. The SCFF can connect the long-distance feature maps in different receptive fields in a top-down order and apply the self-attention mechanism to fuse them two by two. The final fusion result is used as the input of the classifier. In order to verify the effectiveness of the model, the image classification experiment was done on a number of typical datasets. The experimental results prove that the model can fit the existing convolutional neural network well and effectively improve the classification accuracy of the convolutional network only at the cost of a small amount of calculation.

1. Introduction

With the wide application of deep learning in the field of image classification, various convolutional neural networks emerge in an endless stream, and the classification accuracy records on public datasets are constantly refreshed. For the purpose of effectively improving the classification accuracy, current convolution network models tend to be deeper, wider, and more refined. For example, IResNet [1] has more than 3,000 layers and can be trained on Cifar10 and Cifar100, WRN [2] adopts wider convolution kernel, and the Inception series [3–6] has a detailed design for each level. Moreover, various improvement schemes for feature fusion are proposed. For example, the Inception series use the convolution kernels with different sizes for feature extraction and fusion in the same layer, increasing the complexity of the output features. ResNet [7] uses the residual connections to connect the input and output features of the backbone structure, increasing the network's effective depth. Res2Net [8] divides the feature channels into multiple groups for volume accumulation so that the output contains the characteristics of a variety of receptive fields. Besides that, SKNet [9] uses a self-attention mechanism for multiscale feature fusion, and ResNet [10] applies the group feature dynamic weighting strategy to extract feature maps.

Among the existing feature fusion methods, most of them are apt to increase the feature diversity within the scope of the same kernel. The networks, such as ResNet [7, 11] and DenseNet [12], use local short-range connections to the features of the same receptive field. From the perspective of the network model structure in the field of image classification, there are few studies on fusion of high and low-level features across receptive fields. However, similar studies appear more in the field of target detection and semantic segmentation, such as the FPN [13] model and U-Net [14] model, which merge high-level features and low-level features through long-distance connections to obtain high-resolution or strong semantic features. Naturally, there raises an issue that whether the long-connection fusion of features between different levels can improve the performance of the classification network. To find the solution, this paper makes an exploration to the fusion methods that can improve classification performance. Firstly, the feasibility of applying cross-layer long-connection feature fusion in the field of

image classification is analyzed. Then, using a structure similar to the pyramid model of FPN [13], the long-distance feature maps in different receptive fields are connected in a top-down order, and the self-attention mechanism [15] is used to fuse them two by two. The final fusion result can be input to the classifier network to get better classification results. Since this method can selectively fuse cross-layer features, we named this fusion module as SCFF (selective cross-layer feature fusion). SCFF can make the network no longer propagate in the top-down order but make the features of each receptive field directly related to the input features of the classifier, thus enhancing the gradient propagation during backpropagation and enabling the network to fuse the characteristics of different receptive fields with weight bias.

Since the deep network model structure in the field of image classification is relatively unified, SCFF can be combined with most networks. We conduct the comparative experiments on multiple datasets on different network structures, such as ResNet [10] series, Inception series [3–6], VGG [16], and EfficientNet [17], and the experiments prove that SCFF can be easily embedded in the current image classification networks with a minimal increase of calculation but an effective improvement of the classification accuracy.

The contributions of the paper mainly lie in two aspects. Firstly, an improved method that can integrate different levels of feature information is proposed, which can effectively improve the recognition performance. Secondly, this method is easy to be combined with various network models and its effectiveness has been verified on multiple datasets.

2. Research Methods

2.1. Benchmark Network. For its strong hierarchical nature, the ResNet model is used as the benchmark network for our research. As shown in Table 1, the network structure of ResNet50 can be divided into three parts. The first part is the stem layer consisting of convolutional layer, BN (Batch Normalization) operation, and activation function ReLU. The second part is composed of 4 modules from stage-1 to stage-4, and each stage is composed of several bottlenecks (the basic residual module of ResNet) to formally extract image features. The third part is a classifier composed of a downsampling layer and a full connection layer, which downsamples and classifies the final features extracted earlier. The stride of the first bottleneck in each stage is equal to 2, so every time when the feature passes through a stage, the size of the feature map will be reduced to a quarter of the original, and the receptive field will expand. The downsampling of stage-1 is done by the maxpool function with stride = 2. Therefore, when ResNet performs feature extraction, it outputs the features of 4 different receptive fields, which provides convenience for us to perform cross-layer feature fusion.

2.2. Feasibility Analysis. What is to be considered first is the feasibility of cross-layer feature fusion between modules in image classification. The network structure used by conventional image classification algorithms is similar to ResNet's top-down sequential extraction mode. Although DenseNet adopts a complex dense connection model inside the stage, there is still a single route of feature transfer between stages. We set that the input feature of the first layer stage is y_0, W is a convolution operation, and then the output feature of the i-th layer can be defined as

$$y_i = W(y_{i-1}), \quad 0 \le i \le 4. \tag{1}$$

The y_i of each layer is calculated from y_{i-1} through complex convolution. During the convolution process, the receptive field of y_i becomes larger, the number of feature channels increases, and the size of the feature map becomes 1/4, which can be described as

$$y_i \cdot s = \left(\frac{1}{4}\right)^{i-j} y_j \cdot s, 0 \le j < i \le 4. \tag{2}$$

Therefore, before the output features y_i of different stages are fused, their sizes must be unified. That is, it is necessary to downsample and channel-expand the features of the upper layer. In fact, no matter what downsampling method is adopted, the output data will lose certain effective features compared with the input data. Therefore, in each stage of the ResNet, after the feature passes through the first bottleneck with stride = 2, it also needs to go through the bottleneck with stride = 1 multiple times for feature extraction to reduce the impact of downsampling feature loss on network accuracy. However, if the feature maps with a loss of effective features are directly fused, it may have a negative impact on the classification results.

If the feature loss during downsampling has a negative effect, will the feature fusion between stages have a positive effect? There may be two answers. Firstly, the feature fusion of different stages is based on long connections across layers as a carrier, and long connections can provide additional gains to the network. For example, the concept of auxiliary networks is introduced in the Inception series [3]. The network sets optional parameters to output the features of the intermediate layer through auxiliary networks. These features will eventually participate in the error calculation with a certain weight, which is equivalent to building a long connection between the final output layer and the intermediate layer. By adding auxiliary classifiers connected to these intermediate layers, the author would expect to encourage discrimination in the upper stages of the classifier, increase the gradient signal that gets propagated back, and provide additional regularization. In addition, the ROR [18] network optimizes the operation of the network by building

Table 1: The network structure of ResNet50.

Output_size	Layer_name	ResNet50
$112 \times 112 \times 64$	Stem	conv, maxpool
$56 \times 56 \times 256$	Stage-1	Bottleneck × 3
$28 \times 28 \times 512$	Stage-2	Bottleneck × 4
$14 \times 14 \times 1024$	Stage-3	Bottleneck × 6
$7 \times 7 \times 2048$	Stage-4	Bottleneck × 3
$1 \times 1 \times 2048$	Classifier	avgpool, fc

dense long connections between different layers. Secondly, during feature extraction, it is inevitable that some valid features will be missed or erroneous features will be generated. After such errors occur, it will increase the difficulty of extracting valid features later, and eventually lead to errors in the classification results. Therefore, the features closer to the input are often closer to the original features, and the error is smaller. So, it can be considered to use the upper-layer features to fuse the lower-layer features, thereby reducing the error of the lower-layer feature output. Of course, the premise of achieving this goal is to minimize the effective feature loss when the upper-layer features are downsampling.

2.3. Module Design

2.3.1. Efficient Network Architecture. As analyzed above, the loss of accuracy of downsampling is a problem that cannot be ignored when designing the long connection fusion of the ResNet, and it is difficult to solve in a simple way. Thus, this paper is to reduce the loss as much as possible. The pyramid model of the FPN [13] fuses the features of different scales generated by the network from high to low so that the output results contain richer semantics. Accordingly, the structure as shown in Figure 1 is designed. Figure 1 is the original structure of the ResNet, and Figure 2 is the overall structure of SCFF, where D-S is the downsampling module and F-F (Feature Fusion) is the feature fusion module. The output feature after fusion of the i-th layer is described as follows:

$$Y_i = \begin{cases} X_1, & i = 1, \\ F\left(X_i, D\left(Y_{i-1}\right)\right), & 1 < i \leq 4. \end{cases} \tag{3}$$

In the formula above, X_i is the feature output by stage-i, D is the downsampling algorithm, and F is the fusion algorithm. Each X_i except X_1 will be fused with the previous layer. Thus, the advantages of this design lie in as follows: First of all, it does not interfere with the feature extraction on the backbone, which is only used as a feature extractor to generate features $\{X_1, X_2, X_3, X_4\}$ with different receptive fields for SCFF to perform feature fusion on the original network, and the intrusion to the original network is very small; secondly, the fusion features output by each stage will only undergo a minimum downsampling and then fuse with the features of the next layer so that different downsampling methods and feature fusion algorithms can be combined to minimize the loss of features; thirdly, each layer Y_i contains $\{X_1, \ldots, X_i\}$ information, which enables the final output of SCFF to integrate the feature map information of all receptive fields; last but not least, as each layer of Y_i contains the information of $\{X_1, \ldots, X_i\}$, the output features of each stage are associated with the final input features of the classifier through SCFF, which enhances the gradient transfer in this part.

2.3.2. D-S Module. The purpose of the D-S (downsampling) module is to complete the mapping of upper-level features to lower-level features, which contains two tasks: feature map downsampling and channel expansion. The tasks can be accomplished in a variety of ways, such as average pooling or maximum pooling often used in downsampling with point convolution, directly using a 2×2 convolution kernel to complete the mapping, or point convolution (stride = 2) is used for residual connection downsampling in the ResNet network. The characteristics of these methods are analyzed by comparative experiments on the Cifar10 dataset. The F-F module is unified into addition, and the results are shown as in the upper part of Table 2. The experimental results prove that the maximum pooling and average pooling have a relatively obvious improvement in classification accuracy. When ignoring F-F, the output features of each stage are associated with the final features, which enhance the gradient propagation. The advantages of max pooling and average pooling lie in no additional weights, nor increasing the difficulty of gradient transfer. On the other hand, the point convolution needs to a few weights to be trained and has less influence on the gradient. Therefore, the network accuracy can also be effectively improved after the two are combined with point convolution.

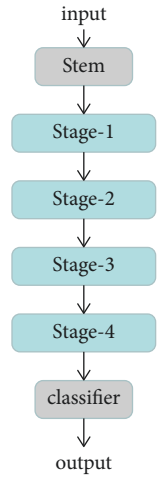

Figure 1: The original structure of the ResNet.

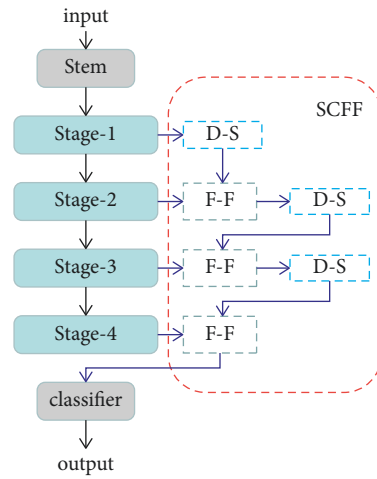

Figure 2: The structure of SCFF.

Table 2: Downsampling and gradient blocking tests.

Model	Acc. (%)-Cifar10
ResNet50 [7]	92.32
ResNet50_avg	92.45
ResNet50_max	92.54
ResNet50_2 × 2conv	92.24
ResNet50_res	92.34
ResNet50_avg_no_grad	90.84
ResNet50_max_no_grad	88.97

Next, we will analyze the effect of gradient transfer on max pooling and convolution pooling. As shown in Figure 3, we use the detach () method provided by PyTorch to cut off the gradient flow passed by SCFF from stage 1 to stage 3 and only release the gradient passed to stage 4 to train the parameters of ResNet. The gradient flow during backpropagation is the same as the original ResNet. The experimental results are shown in the lower part of Table 2. After losing the gradient gain brought by SCFF, the accuracy of both average pooling and max pooling decreased, and the accuracy of max pooling decreased more obviously. This verifies the feature loss problem mentioned in the feasibility analysis, that is, the feature loss during pooling will have a negative impact on the network, but in practice, considering the positive gain of gradient transfer, this negative impact can be accepted.

From the experimental results of blocking gradients, the accuracy of average pooling drops less, so less feature loss occurs. The reason is average pooling synthesizes all features in the mapped range, while maxpooling ignores features other than the maximum value. Our module design is to use the upper-layer features to correct the lower-layer features, hoping that the upper-layer features will minimize the loss of effective features during downsampling, so the downsampling method in this paper uses average pooling. The structure of D-S module is shown in Figure 4.

2.3.3. F-F Module. For the features from different stages, how to fuse them needs analyzing. As described above, it is hoped that the upper-level features closer to the initial features can correct the lower-level features, so it is necessary to assign different weights to the feature maps according to the requirements before their fuse. For this purpose, we use the attention mechanism. After comparing various attention mechanisms, we choose the structure shown in Figure 5 through experimental verification. Firstly, the input features are compressed by global average pooling to obtain channel-level global features. Then, two fully connected layers (fc) cooperate with the ReLU function to form a funnel structure to model the correlation between channels. Finally, the channel weights normalized between 0 and 1 are obtained through the Sigmoid function. Compared with using only one fully connected layer, the funnel structure using two fully connected layers can make the extraction process more nonlinear and better fit the relationship between channels.

The classic achievement of the SE (squeeze and excitation) mechanism in multiscale fusion is SKNet [9]. The main

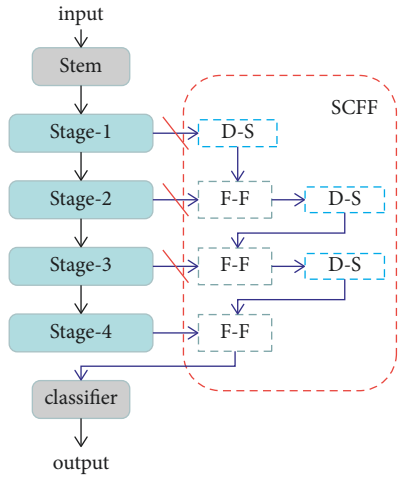

Figure 3: Gradient blocking in SCFF.

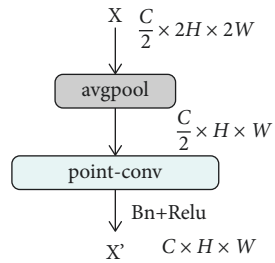

Figure 4: The structure of D-S module.

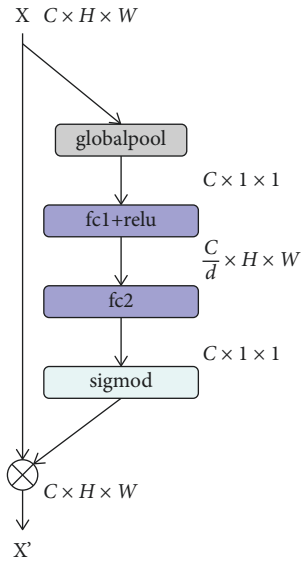

Figure 5: SE module.

purpose of SKNet is to give attention to the convolution kernel level of the network. Unlike the SE module, which targets single-channel feature maps, SKNet needs to process multichannel features, so the focus is on how to generate weights to fuse the information of feature maps of different scales. The feature fusion structure of SKNet is shown in Figure 6, where X and Y are the input feature maps of

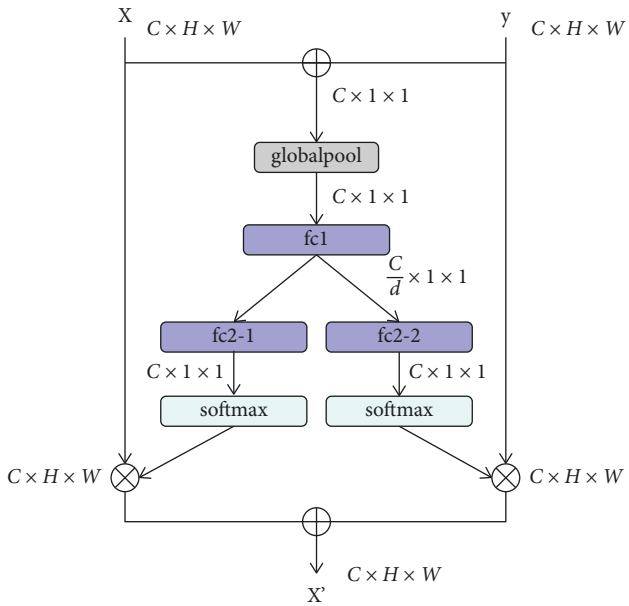

Figure 6: SKNet fusion module.

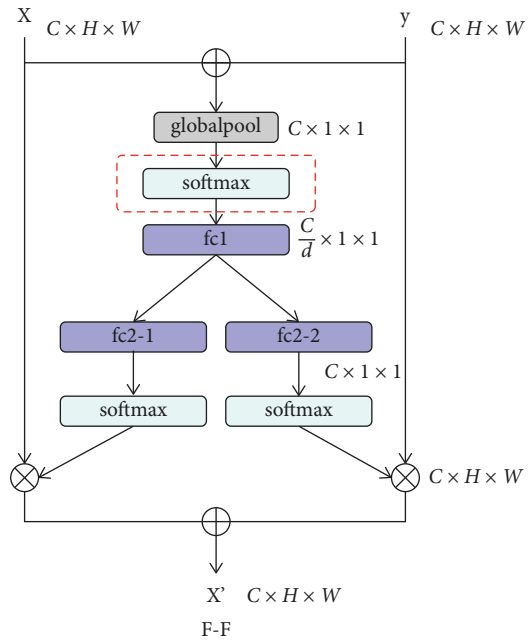

Figure 7: The improved SKNet feature fusion module.

different scales. The SKNet adopts the global average pooling after adding X and Y to generate the sum of the global features of X and Y, and then performs the first weight extraction through the first layer of full connection with BN (Batch Normalization) and ReLU; then, different fully-connected layer and the softmax activation function are used to generate the channel weights corresponding to X and Y, respectively. Finally, the weights are assigned to X and Y, and the obtained results are added to complete the feature fusion. In fact, looking at the data flow of X or Y alone, it can be found that this improved process is still the same as the SE module, but in order to improve the correlation between the weights, the global average pooling and FC1 operations are combined.

We make an adjustment to the parameters on the SCFF model and compare various models of SE with additive and concatenated combinations. Since there is no fusion model that has a better overall structure than SKNet, this paper chooses to improve it based on the SKNet model. For the features after global pooling, we perform a softmax activation, then extract features, and keep the others as they are. The original full-connection operation is aimed at the sum of the global features of X and Y at the channel level, and the improvement is activated by softmax. The sum of the global features is mapped to a weight that sums to 1, and the full connection operation is against the weight. Performing a softmax activation is equivalent to initializing the weights, instead of generating weight information until the end like the original SKNet. The experiments show that this method can improve the classification accuracy by 0.1–0.2 percentage. The improved SKNet feature fusion module is shown in Figure 7.

So far, the SCFF model design is completed. In general, the advantages of the SCFF model lie in two aspects: First, the pyramid-like layer-by-layer structure combined with average downsampling can effectively alleviate the gradient propagation. Second, the feature fusion algorithm based on the SE mechanism can ensure that different receptive field features can be fused with different weights; thus, the features finally input to the classifier that contains all the effective information of the receptive field.

3. Experiment

We conduct comparative experiments on various network structures based on multiple public datasets. The training strategy adopts simple random cropping, rotation, and normalization. The SGD optimizer is trained for 100 epochs. The initial learning rate is 0.1, and then it is reduced to 1/10 of the original value every 30 epochs. The resulting data is the average of more than three experiments.

3.1. Cifar10, Cifar100, and Tiny-Image Net Datasets. Cifar10, Cifar100, and TinyImageNet-100 are three relatively mature and widely used public datasets. The image size of Cifar10 and Cifar100 is 32×32, and the image size of Tiny-ImageNet is 64×64. We mainly conduct experimental comparisons of the ResNet series on these three datasets.

ResNet's bottleneck has high scalability. Many ResNet series networks tend to improve the intermediate 3×3 convolution kernel on the basis of maintaining their original structure, such as ResNet [18], ResNetV2 [19], Res2Net [8], and SKNet [9]. This provides great convenience for our experiments, as we only need to keep the previous design structure and embed SCFF on each stage. The experimental results are shown as in Table 3 where the calculation amount is for the Cifar10 dataset, and B is billion. The SCFF model has a very good performance, as the classification accuracy of each network has been significantly improved, and the increase in the amount of calculation is not large.

Table 3: Comparative experiment.

Model	The calculation amount (B)	Acc. (%)		
		Cifar10	Cifar100	Tiny-ImageNet
ResNet50 [7]	0.328	92.39	68.14	54.84
ResNet50-SCFF	0.362	93.20	74.27	60.78
ResNext50 [18]	0.340	92.78	70.84	54.80
ResNext50-SCFF	0.374	93.35	75.79	62.08
Res2Net50 [8]	0.342	92.58	70.85	57.06
Res2Net50-SCFF	0.375	93.37	74.31	61.29
PreActResNet50 [19]	0.328	93.35	73.08	54.74
PreActResNet50-SCFF	0.362	93.98	76.14	54.54
SKNet50 [9]	0.622	92.59	67.36	57.22
SKNet-SCFF	0.656	93.64	72.56	61.34
VGG16 [16]	0.333	92.66	—	—
VGG16-SCFF	0.339	93.30	—	—

3.2. 101_ObjectCategories Datasets. Considering that most networks are more suitable for images with relatively large sizes, 101_ObjectCategories are further used here for classification experiments. 101_ObjectCategories datasets contain 101 categories, each of which contains about 30~400 pictures, totally 8,677 pictures. We adjust the size into 224×224 and divide them into training sets and test sets according to the ratio of 8 : 2.

The overall structure of the convolutional network is not necessarily as regular as the ResNet series. When combining with the SCFF module, the position of the combination needs to be considered. The testing proves that the position before the size of the feature map decreases by an integer multiple is the most suitable. Based on this, experiments are performed on a variety of network structures against the 101_ObjectCategories dataset. The classification experiment results are shown in Table 4. Similar to ResNet, SCFF shows a good accuracy improvement whether it is InceptionV3 with a more complex structure or DenseNet121 with only 3 stages in the stage, which further demonstrates the effectiveness of SCFF.

3.3. SIRI-WHU Datasets. To better analyze the impact of SCFF on the classification task, we again conduct classification experiments on the remote sensing image dataset SIRI-WHU [20] and conduct in-depth analysis of the training and testing data. SIRI-WHU dataset is a remote sensing image dataset, which contains a total of 2,400 scene images of 12 categories, of which each category has 200 images, the pixel size of each image is 200 ∗ 200, and the spatial resolution is 2 meters. The dataset resource comes from Google Earth, mainly covering urban areas in China, and the scene image dataset is designed by the RS-IDEA Group of Wuhan University.

The optimizer for this part of the experiment uses Adam with an initial learning rate of 0.001. The experimental results are shown in Table 5. Similar to the experimental results above, various networks introduced with SCFF achieved accuracy gains. We recorded the test results for each epoch during training, as shown in Figure 7. Since SCFF establishes a long connection from the intermediate layer to the classifier on the network, which enhances the gradient transfer, it is also easier to train. As can be seen from the abscissa in Figure 7, with the increase of training batches, the accuracy of the network after the introduction of SCFF improves with a faster initial speed and finally maintains a better accuracy range than the original version. The faster initial speed of the network is due to the enhanced gradient transfer, making it easier to train. The network accuracy can be maintained in a higher range in the end, because the attention mechanism is introduced to fuse the features (Figure 8).

Next, we compared the correspondence between the features extracted by the network to the classifier after the introduction of SCFF and the original image by means of a heat map. We used Grad-CAM++ [21] to generate a heat

Table 4: The experimental results of 101_ObjectCategories datasets.

Model	The calculation amount (B)	Acc. (%)
EfficientNet-b0 [17]	0.013	79.50
EfficientNet-b0-SCFF	0.019	84.17
DenseNet121 [12]	2.865	78.20
DenseNet121-SCFF	2.906	79.50
InceptionV3 [5]	2.845	82.35
InceptionV3-SCFF	2.995	85.48
Res2Net50 [8]	4.278	78.99
Res2Net50-SCFF	4.598	80.81
ResNet50 [7]	4.109	77.68
ResNet50-SCFF	4.421	79.44
ResNext50 [18]	4.257	80.24
ResNext50-SCFF	4.577	81.20
SKNet [9]	7.636	79.50
SKNet50-SCFF	7.955	83.37

Table 5: The experimental results of SIRI-WHU dataset.

Model	Acc. (%)
ResNet50	94.67
ResNet50-SCFF	95.66
ResNext50 [18]	95.83
ResNext50-SCFF	96.00
Res2Net [5]	95.83
Res2Net-SCFF	96.33
InceptionV3 [8]	93.00
InceptionV3-SCFF	94.50

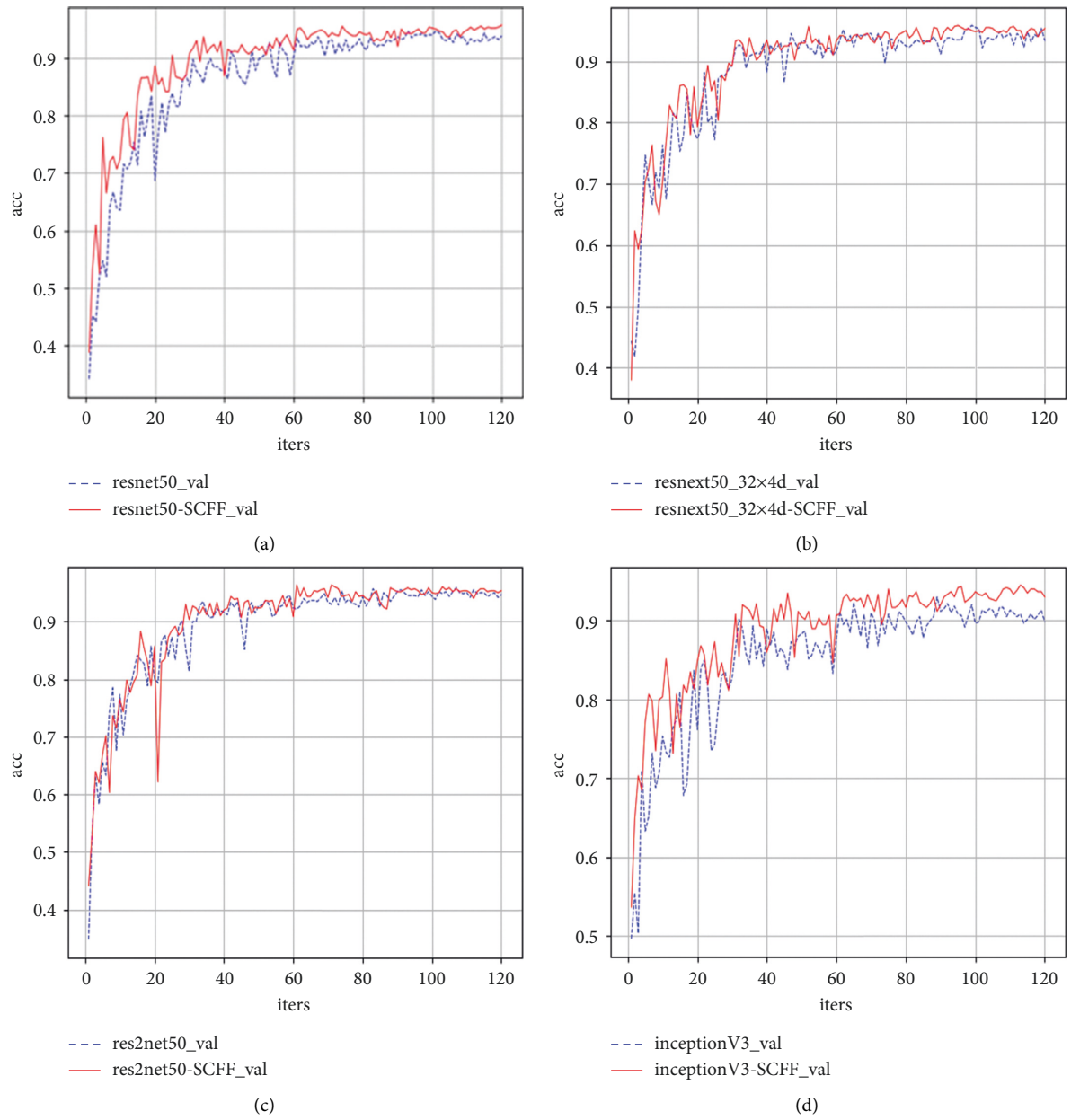

Figure 8: Comparison of the network accuracy curve before and after the introduction of SCFF model. (a) ResNet50 Model. (b) ResNext50 Model. (c) Res2Net Model. (d) InceptionV3 Model.

map of the features corresponding to the original image. The result is shown in Figure 9. On the overpass image, we compared the heatmaps of the ResNet50, ResNext, and InceptionV3 models before and after the introduction of SCFF. As can be seen from Figure 9, after the introduction of SCFF, the red part of the heat map covers more of the overpass area. It shows that SCFF based on multilayer feature fusion can well enhance the network's attention to the target object in the image, and the enhancement of accuracy is reasonable.

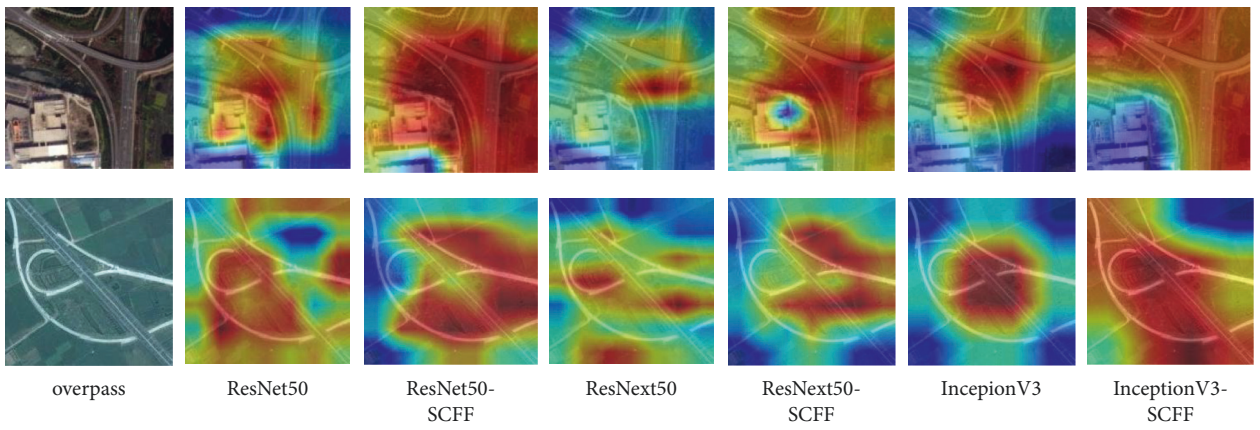

Figure 9: Heatmap comparison.

4. Conclusion

This paper analyzes the feasibility of long connections across different receptive fields to improve the performance of image classification networks and designs a feature fusion model SCFF using a similar pyramid structure combined with a self-attention mechanism. SCFF can be embedded into a convolutional neural network in a non-invasive manner and selectively fused according to the characteristics of different receptive fields. We conduct classification and comparison experiments on multiple datasets by embedding multiple convolutional neural networks. The results show that SCFF can effectively improve the accuracy of network classification at the expense of a small increase in computational complexity. The results of this paper show that image classification is not the only choice to extract features in order from large to small. It is also important to think from more dimensions and connect the features of high and low layers. However, the fusion module of SCFF is an improvement on the basis of SKNet, but SKNet was originally designed to fuse multichannel features of the same layer, not for long-connected cross-layer features. Are there other feature fusion algorithms that are more suitable for SKNet? In addition, the SCFF fuses the features extracted by the backbone. The current strategy is to train SCFF and the main network together. Will this have a certain negative impact on the features extracted by the backbone network? If the backbone network and the SCFF are trained separately, or if the SCFF is directly introduced into the pre-trained network for retraining, will the network be improved better? We will further study these ideas in the future.

Data Availability

CIFAR-10 and CIFAR-100 dataset were used to support this study and are available at http://www.cs.toronto.edu/~kriz/cifar.html. The Google image dataset of SIRI-WHU is available at http://www.lmars.whu.edu.cn/prof_web/zhongyanfei/Num/Google.html.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded in part by the National Natural Science Foundation of China (Grant no. 61971004) and the Key Project of Natural Science of Anhui Provincial Department of Education (Grant nos. KJ2019A0083, KJ2021A1289).

References

[1] I. C. Duta, L. Liu, F. Zhu, and L. Shao, "Improved residual networks for image and video recognition," in *Proceedings of the 2016 IEEE Conference on Pattern Recognition, ICPR 2020*, pp. 9415–9422, Cancun, Mexico, December 2016.

[2] S. Zagoruyko and N. Komodakis, "Wide residual network," 2017, https://arxiv.org/abs/1605.07146.

[3] C. Szegedy, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 1–9, IEEE, Boston, MA, USA, June 2015.

[4] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2017, https://arxiv.org/abs/1602.07261.

[7] K.-m. He, X.-y. Zhang, S.-q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[8] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone

architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.

[9] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Recognition, CVPR 2019*, pp. 510–519, IEEE, Long Beach, CA, USA, June 2016.

[10] H. Zhang, C. Wu, Z. Zhang et al., "Resnet: Split-attention networks," 2020, https://arxiv.org/abs/2004.08955.

[11] S. S. Verma, A. Prasad, and A. Kumar, "CovXmlc: High performance COVID-19 detection on X-ray images using Multi-Model classification," *Biomedical Signal Processing and Control*, vol. 71, no. 1, pp. 1–7, 2022.

[12] H. Gao, Z. Liu, V. D. M. Laurens, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 2261–2269, IEEE, Honolulu, HI, USA, July 2017.

[13] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 2117–2125, IEEE, Honolulu, HI, USA, July 2017.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the 2015 International Conference on Medical image computing and computer-assisted intervention, MICCAI 2015*, pp. 234–241, Springer, Munich, Germany, 2015.

[15] J. Hu, Li. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 7132–7141, IEEE, Salt Lake City, UT, USA, June 2018.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[17] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 2019 International Conference on Machine Learning, ICML 2019*, pp. 6105–6114, Long Beach, CA, USA, June 2019.

[18] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1492–1500, IEEE, Honolulu, HI, USA, July 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the 2016 European Conference on Computer Vision, ECCV 2016*, pp. 630–645, Amsterdam, Netherlands, October 2016.

[20] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 99, pp. 1–16, 2015.

[21] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proceedings of the 2018 IEEE winter conference on applications of computer vision, WACV 2018*, pp. 839–847, IEEE, Lake Tahoe, NV, USA, March 2018.