Electronic Research Archive

ERA, 31(6): 3187–3199.
DOI: 10.3934/era.2023161
Received: 19 December 2022
Revised: 02 March 2023
Accepted: 12 March 2023
Published: 27 March 2023

http://www.aimspress.com/journal/ERA

Research article

Optimized pointwise convolution operation by Ghost blocks

Xinzheng Xu*, Yanyan Ding, Zhenhu Lv, Zhongnian Li and Renke Sun

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

* **Correspondence:** Email: xxzheng@cumt.edu.cn; Tel: +8615952151616; Fax: +8651683591726.

Abstract: In the lightweight convolutional neural network model, the pointwise convolutional structure occupies most of the parameters and computation amount of the model. Therefore, improving the pointwise convolution structure is the best choice to optimize the lightweight model. Aiming at the problem that the pointwise convolution in MobileNetV1 and MobileNetV2 consumes too many computation resources, we designed the novel Ghost-PE and Ghost-PC blocks. First, in order to optimize the channel expanded pointwise convolution with the number of input channels less than the output, Ghost-PE makes full use of the feature maps generated by main convolution of the Ghost module, and adds global average pooling and depth convolution operation to enhance the information of feature maps generated through cheap convolution. Second, in order to optimize the channel compressed pointwise convolution with the number of input channels more than the output, Ghost-PC adjusts the Ghost-PE block to make full use of the features generated by cheap convolution to enhance the feature channel information. Finally, we optimized MobileNetV1 and MobileNetV2 models by Ghost-PC and Ghost-PE blocks, and then tested on Food-101, CIFAR and Mini-ImageNet datasets. Compared with other methods, the experimental results show that Ghost-PE and Ghost-PC still maintain a relatively high accuracy in the case of a small number of parameters.

Keywords: pointwise convolution; Ghost blocks; Ghost-PE; Ghost-PC; MobileNet

1. Introduction

Pointwise convolution is widely used in convolutional neural networks because the parameters required to process feature maps are far less than the conventional convolution [1–3]. However, with the development of research, it is found that pointwise convolution consumes too much computation

resources in lightweight CNN models.

In recent researches, Jia et al. [4] proposed a brand new pointwise convolutional block, called "Improved Pointwise Convolution" (IPC) block. The IPC block weighs the size and performance of the model with two hyperparameters. Sachin et al. [5] designed the Efficient Channel Fusion (EFuse) block to realize the function of fusing channels information with pointwise convolution. EFuse first extracts global vector using global average pooling operation, and then using a fully connected layer to fuse global vectors. In parallel, the EFuse encodes spatial representation by depth convolution. Yu et al. [6] applied efficient shuffle block to HRNet (high-resolution network), introduced conditional channel weighting module to replace pointwise convolution in shuffle blocks. Yang et al. [7] proposed to prune the first pointwise convolution of the inverted residual unit in the MobileNetV2, and then migrate saved computations to the second pointwise convolution. Li et al. [8] used group-adaptive convolution to factorize pointwise convolution to reduce the connection between the input feature maps and filters. Liang et al. [9] designed four types of linear-phase pointwise convolution to reduce the computational complexities of conventional means. Thus, Joao et al. [10] ensured that, in order to reduce the number of parameters of pointwise convolution in EfficientNet [11], the pointwise convolution was changed to group convolution, and each branch of group convolution processed part of the input channel, and the feature maps of the middle layer was mixed. Schwarz et al. [12] proposed a scheme of grouped pointwise convolution to reduce the complexity of deep convolutional neural networks.

Although there have been some methods to improve pointwise convolution, few of them are optimized from the perspective of channel number variation. But the effect of optimizing pointwise convolution depends largely on the change of channel number. At the same time of information fusion of channels, the pointwise convolution is also responsible for channel expansion and compression. The channel expansion means that the number of channels is more than the output, and the channel compression is the opposite.

MobileNetV1 [13] and MobileNetV2 [14] contain the above two types of structures. Therefore, based on Ghost module [15], this paper carries out lightweight optimization on the pointwise convolution of MobileNetV1 and MobileNetV2, and the experiment results on Food-101 [16], CIFAR [17] and Mini-ImageNet [18] datasets show that the proposed method is feasible.

Our main contributions can be summarized as follows:

First of all, in order to optimize the number of input channels less than the number of output channels of the extension point convolution, we propose a Ghost-PE module suitable for optimizing the pointwise convolution structure on the basis of Ghost module. To make full use of the Ghost main convolution generated feature map, Ghost-PE module further use global average pooling and deep convolution operation for information extraction.

Second, in the case of compression point convolution where the number of input channels is more than the number of output channels, each feature graph channel contains too much feature information. If directly use Ghost-PE to lightweight it, it is difficult to obtain all the information of fusion, which will have a great impact on the performance of the model. Therefore, we optimize the Ghost-PE module, put forward the Ghost-PC module.

Finally, the Ghost-PE and Ghost-PC modules are applied to the MobileNetV1 and MobileNetV2 models. The test results on the data sets Food-101, CIFAR and Mini-ImageNet show that our method shows better performance than other methods.

2. Methodology

In this section, we first introduce the proposed Ghost-PE block to reduce parameters of channel expanded pointwise convolution. We then adjust the Ghost-PE block to improve channel compressed pointwise convolution, got Ghost-PC block.

2.1. Channel expansion

We improved the Ghost module to obtain the Ghost-PE block, which optimized the channel expanded pointwise convolution in MobileNetV1 and MobileNetV2. The Ghost module has plug-and-play characteristics, which can be directly used to replace pointwise convolution to reduce computation resources without considering the model performance damage. Therefore, we first just replace pointwise convolution in MobileNetV1 by the Ghost module, the result on Food-101 is shown in Table 1. The main convolution of the Ghost module is pointwise convolution operation, and the cheap convolution is 1 × 1 depth convolution. The number of feature maps generated by the above two operations is the same, and each is half of the original module output.

Table 1. The results of MobileNetV1 on Food-101.

Model	Accuracy (%)	Parameters (MB)	FLOPs (MB)
MobileNetV1	82.22	3.3	568
MobileNetV1+Ghost	80.29	1.7	310

Compared with pointwise convolution operation in MobileNetV1, the Ghost module yields a better compression, over 45% of FLOPs are reduced and 48% of parameters are removed, but the accuracy is greatly damage. All in all, the direct replacement method is not feasible. We think the reason is that the main convolution of Ghost module only uses half of the number of original filters, so the information contained in the fused feature maps is not as rich as before. Although the cheap depth convolution operation also makes up the features, it is far from enough.

In order to make full use of the feature maps generated by the main convolution of Ghost module, we introduced the Ghost-PE block, the specific structure is shown in Figure 1.

The input $X \in R^{C\times H\times W}$ to Ghost-PE block is a three dimensional tensor, defined by depth C, width W, height H, to produce an output $Y \in R^{C\times H\times W}$. The Ghost-PE block applies pointwise convolutional kernels $W_{main} \in R^{1/2\times C\times C\times 1\times 1}$ along depth to produce $X_{main} \in R^{1/2C\times H\times W}$ that encode information from input X, and then utilizes 1 × 1 depth convolution kernels $W_{min} \in R^{1/2C\times 1\times 1\times 1}$ to generate $Y_{min} \in R^{1/2C\times H\times W}$, as shown in Eqs (1) and (2), respectively. Furthermore, we squeeze spatial information of X_{main} using global average pooling (GAP) to extract global vector descriptor $V_{avg} \in R^{1/2C}$, and then fuse spatial information by utilizing depth convolutional kernels $W_{1\times 1} \in R^{1/2C\times 1\times 1\times 1}$ to V_{avg} to produce an output V_{dw}, as formulated in Eq (3). Next, according to Eq (4), we multiply V_{dw} and Y_{min}, and then add X_{main} to produce Y_{Add}. Finally, X_{main} and Y_{Add} are connected in the channel dimension, as shown in Eq (5).

$$X_{main} = X * W_{main} \tag{1}$$

$$Y_{min} = X_{main} * W_{min} \tag{2}$$

$$V_{dw} = \left(AvgPool(X_{main})\right) * W_{1\times 1} \tag{3}$$

$$Y_{Add} = V_{dw} \times Y_{min} + X_{main} \tag{4}$$

$$Y = concatnate([X_{main}, Y_{Add}], axis = 1) \tag{5}$$

In the Ghost-PE block, 1 × 1 depth convolution is used for V_{avg} in order to get the result by learning harmonic averaging pooling. Because the function of V_{avg} is only to supplement the feature information, it is necessary to prevent the existence of information in V_{avg} that will greatly interfere with the Y_{min} feature map. Finally, X_{main} is added to Y_{min} to make the most of the features in it.

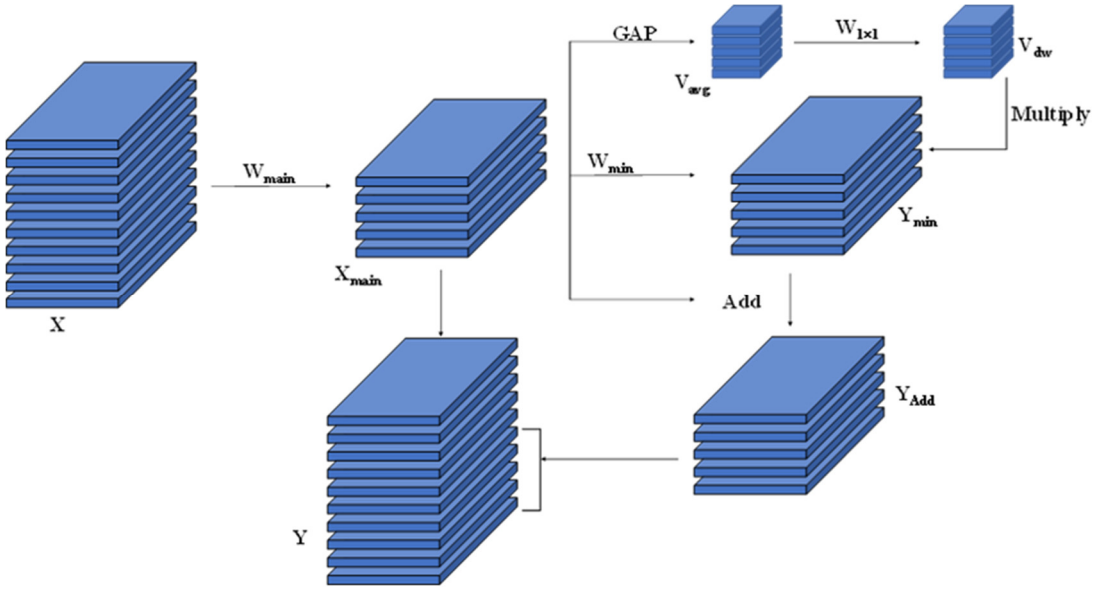

Figure 1. An illustration of the proposed Ghost-PE block. W_{main} and W_{min} represent the main and cheap operation respectively.

We can use the proposed Ghost-PE block to optimize the channel expanded pointwise convolution, and reduce the computational cost. Here, we analyze the profit on parameters and FLOPs usage by employing the Ghost-PE block. To produce an output of size C × H × W from an input of size C × H × W, the conventional pointwise convolution performs HWC^2 operations and parameters are C^2. In Ghost-PE block, the parameters and operations of main convolution are $1/2C^2$ and $HWC^2/2$ respectively, in cheap convolution are C/2 and HWC/2, the $W_{1\times1}$ performs C/2 operations and parameters are C/2. Therefore, the theoretical speed-up and compression ratio of pointwise convolution with the Ghost-PE block is shown in Eqs (6) and (7), respectively.

$$r_p = \frac{1/2\times1\times1\times C^2+1\times1\times C/2+C/2}{1\times1\times C^2} \approx \frac{1}{2} \tag{6}$$

$$r_m = \frac{H\times W\times C/2\times C+H\times W\times C/2+C/2}{H\times W\times C\times C} \approx \frac{1}{2} \tag{7}$$

2.2. Channel compression

In the above article, we introduced an improved Ghost module called the Ghost-PE block, and used it to optimize the MobileNetV1 and MobileNetV2 channel extension point convolution. The channel expanded

pointwise convolution operation output features information evenly distributed in each channel, because the number of channels unchanged or increased, so each feature maps after fusion contains less information than the compressed pointwise convolution. Thus, we can use Ghost-PE block to get all the knowledge before optimization as far as possible. However, after compressed pointwise convolution operation, each feature channel contains more information, it is difficult to obtain all the fused information compared with expanded operation, if we perform lightweight optimization, which will inevitably cause a great loss on the performance. Therefore, it is necessary to further improve and optimize the Ghost-PE module to lightweight the compressed pointwise convolution structure.

In the inverted bottleneck of MobileNetV2, there is a channel compressed pointwise convolution structure. In order to carry out lightweight optimization and maximize the acquisition of channel features information, we reuse the feature map Y_{min} generated by Ghost-PE block, designed Ghost-PC module, as shown in Eq (8).

$$Y = concatnate([X_{main} + Y_{min}, Y_{Add} + Y_{min}], axis = 1) \tag{8}$$

We used the Y_{min} many times, supplemented the feature channel information in feature maps X_{main} and Y_{Add}, and obtained all the features before optimization as much as possible. By using Ghost-PC blocks, Y_{min} can be reused to get more pre-optimized image information.

3. Experimental results and analysis

In this section, we first replace the channel expanded pointwise convolution in MobileNetV1 and MobileNetV2 by the proposed Ghost-PE block to verify its effectiveness. Then, the Ghost-PC block utilized to substitute for the channel compressed pointwise convolution in MobileNetV2 to prove its significance on the image classification task.

3.1. Datasets and settings

To verify the effectiveness of the proposed Ghost-PE and Ghost-PC blocks, we use the deep learning framework PyTorch [19] to conduct experiments on three popular visual datasets, Food-101, CIFAR and Mini-ImageNet.

Food-101 dataset consists of 75,750 training and 25,250 testing images from 101 different food classes. A common data augmentation scheme including random crop [20] and horizontal training on Food-101, we use an SGD optimizer with a momentum equal to 0.9 and a weight decay of 1e-4. We train MobileNetV1 and MobileNetV2 models for 90 epochs, the initial learning rate is set to 0.1 and then decays at epoch 30 and 60 at a rate of 0.1. CIFAR dataset consist of 60,000 color images with 32 × 32 pixels in 10 or 100 classes, including 50,000 training images and 10,000 test images. Mini-ImageNet contains a total of 60,000 color images in 100 categories, of which each class has 600 samples, and each image has a specification of 84 × 84. Since MobileNetV1 and MobileNetV2 are originally designed for ImageNet, we use their variants [21], which is widely used in literatures for conducting the following experiments. We train MobileNetV1 model for 100 epochs, MobileNetV2 for 160 epochs. The remaining training hyperparameters on CIFAR dataset are identical to the Food-101.

3.2. Experimental results of Ghost-PE block

Based on the above experimental settings, the experiment results of replacing the channel expanded pointwise convolution in MobileNetV1 with Ghost-PE block are shown in Tables 2–5, respectively.

Table 2. Comparison of other methods for compressing MobileNetV1 on Food-101 dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	3.3	568	82.22
MobileNetV1-Ghost	1.7	310	80.29
MobileNetV1-L1Norm [22]	2.0	349	80.89
MobileNetV1-Micro [8]	1.7	367	79.99
MobileNetV1-Ghost-PE	1.7	310	81.85

Table 3. Comparison of other methods for compressing MobileNetV1 on CIFAR-10 dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	3.22	46.34	91.24
MobileNetV1-Ghost	1.65	24.43	89.58
MobileNetV1-L1Norm	2.02	31.40	90.37
MobileNetV1-Micro	1.50	29.04	90.41
MobileNetV1-Ghost-PE	1.65	24.44	90.70

Table 4. Comparison of other methods for compressing MobileNetV1 on CIFAR-100 dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	3.31	46.34	67.89
MobileNetV1-Ghost	1.74	24.43	63.58
MobileNetV1-L1Norm	2.08	31.37	66.01
MobileNetV1-Micro	1.69	29.04	64.19
MobileNetV1-Ghost-PE	1.74	24.44	65.77

Table 5. Comparison of other methods for compressing MobileNetV1 on Mini-ImageNet dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	3.31	583	83.51
MobileNetV1-Micro	1.69	372	76.21
MobileNetV1-Ghost-PE	1.74	314	80.79

Although the traditional Ghost module in parameter optimization has achieved a better effect, its accuracy loss is unacceptable. When compared with the Ghost module, L1-Norm-pruning(L1Norm) and Micro-Factorized Pointwise Convolution (Micro), the Ghost-PE block delivers better performance on Food-101, CIFAR-100 and Mini-ImageNet datasets. It can be seen from the test process curve that the ghost-point module is associated with Ghost and L1Norm has certain advantages over Micro methods.

The testing process of MobileNetV1 model on Food-101, CIFAR-10 and CIFAR-100 datasets is shown in Figure 2–4.

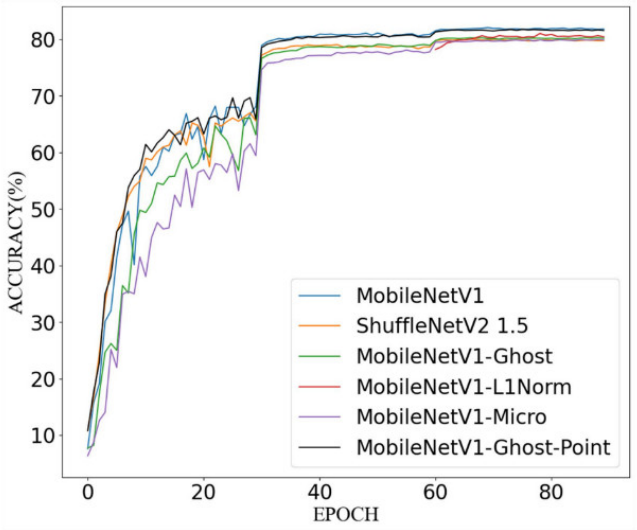

Figure 2. Comparison of the testing process of several optimization methods of MobileNetV1 on the Food-101 dataset.

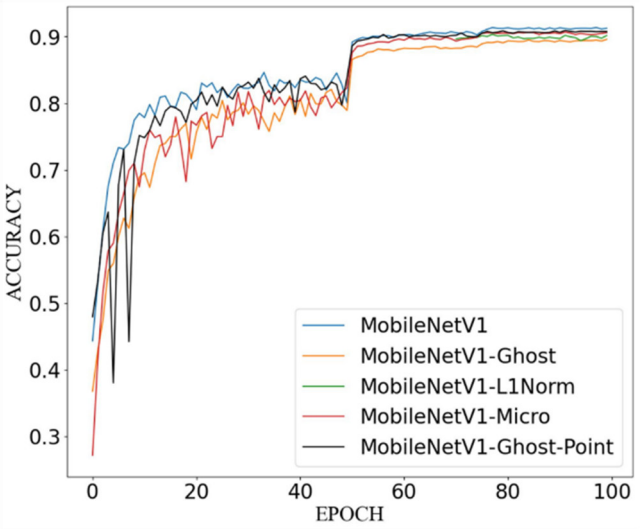

Figure 3. Comparison of the testing process of several optimization methods of MobileNetV1 on the CIFAR-10 dataset.

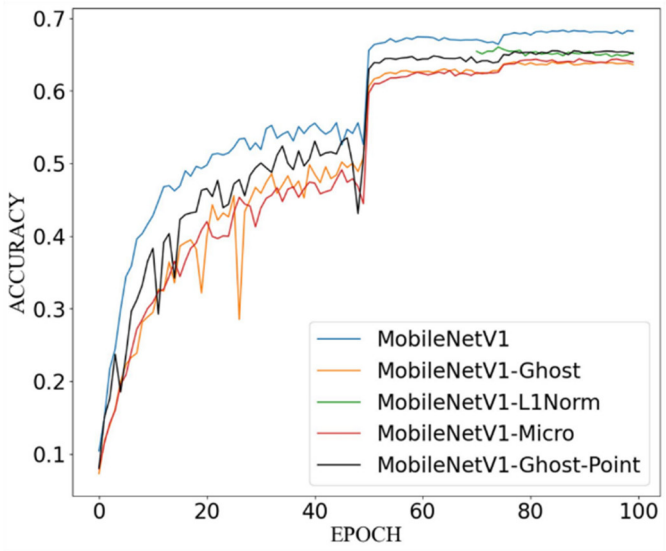

Figure 4. Comparison of the testing process of several optimization methods of MobileNetV1 on the CIFAR-100 dataset.

In order to further verify the performance of Ghost-PE block, the channel expanded pointwise convolution in inverted bottleneck of MobileNetV2 model is optimized by Ghost-PE block, the experimental results on Food-101, CIFAR-10 and CIFAR-100 datasets are shown in Tables 6–8, respectively.

Table 6. Experimental results of MobileNetV2 using Ghost-PE block on Food-101 dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	2.4	299	81.38
MobileNetV1-Ghost	1.8	226	77.82
MobileNetV1-L1Norm	2.2	257	79.51
MobileNetV1-Micro	1.8	247	79.50
MobileNetV1-Ghost-PE	1.8	226	81.37

Table 7. Experimental results of MobileNetV2 using Ghost-PE block on CIFAR-10 dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	2.30	91.14	93.33
MobileNetV1-Ghost	1.69	67.90	92.77
MobileNetV1-L1Norm	1.77	72.33	92.57
MobileNetV1-Micro	1.78	71.71	92.46
MobileNetV1-Ghost-PE	1.70	67.91	92.87

Table 8. Experimental results of MobileNetV2 using Ghost-PE block on CIFAR-100 dataset.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV1	2.41	91.14	74.83
MobileNetV1-Ghost	1.81	67.90	73.02
MobileNetV1-L1Norm	1.85	72.33	72.71
MobileNetV1-Micro	1.89	71.71	73.64
MobileNetV1-Ghost-PE	1.81	67.91	73.55

As can be seen from Tables 6 and 7, compared with other methods, the Ghost-PE block achieves the best precision while reducing the number of parameters and the amount of calculation. In Table 8, although the Ghost-PE module is lower than the Micro method experimental results, the optimized model has fewer parameters and less computation.

From Tables 6 and 7, it can be seen that, in the MobileNetV2 network through training on Food-101 and CIFAR-10 data sets, Ghost-PE module achieves the best accuracy while significantly reducing the number of parameters compared with Ghost, L1Norm, Micro and other methods. In Table 8, in the training of the CIFAR-100 data set, although the accuracy of the Ghost-PE module is not as good as that of the Micro method, it has achieved better parameter optimization effect and is superior in reducing the number of parameters and the amount of calculation.

The testing process of MobileNetV2 model on Food-101, CIFAR-10 and CIFAR-100 data sets is shown in Figures 5–7.

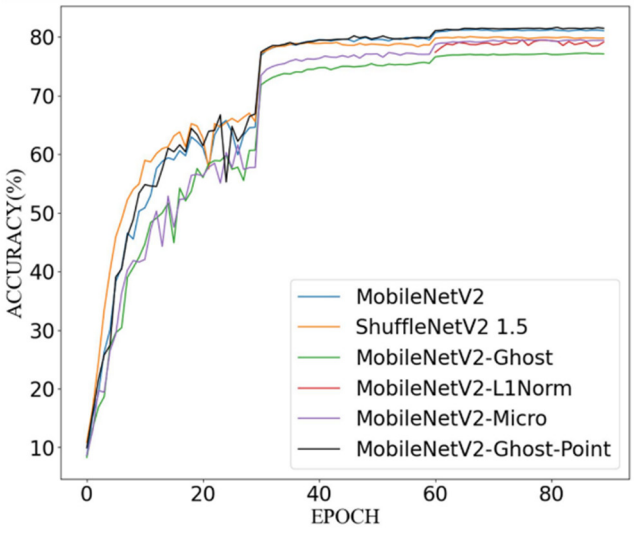

Figure 5. Comparison of the testing process of several optimization methods of MobileNetV2 on the Food-101 dataset.

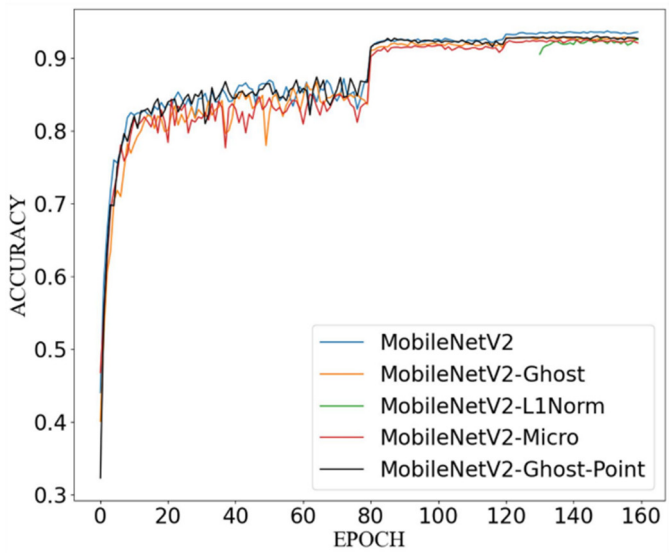

Figure 6. Comparison of the testing process of several optimization methods of MobileNetV2 on the CIFAR-10 dataset.

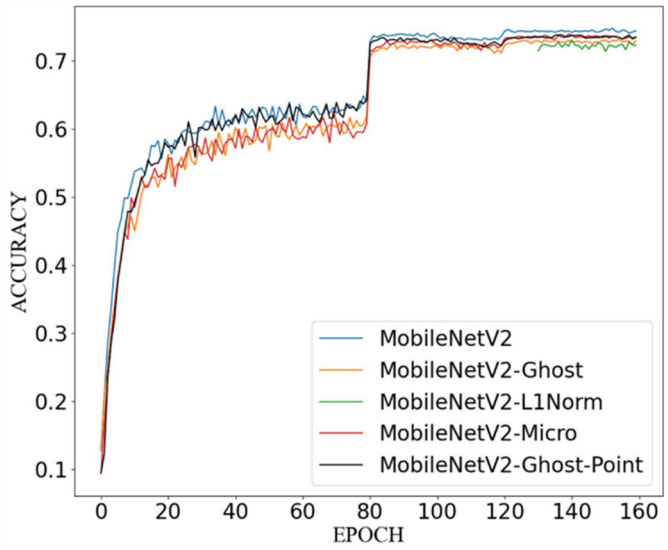

Figure 7. Comparison of the testing process of several optimization methods of MobileNetV2 on the CIFAR-100 dataset.

3.3. Experimental results of Ghost-PC block

We used Ghost-PC block to improve the channel compressed pointwise convolution in inverted bottleneck of MobileNetV2, and compared it with L1Norm, Ghost module and Ghost-PE. The experimental results on Food-101 dataset and Mini-ImageNet dataset are shown in Tables 9 and 10.

Table 9. Experimental results of the MobileNetV2 model on the Food-101 dataset lightweight optimization of the channel compressed pointwise convolution with Ghost-PC.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV2	2.4	299	81.38
MobileNetV2-L1Norm	1.8	232	77.65
MobileNetV2-Ghost	1.9	244	77.45
MobileNetV2-Ghost-PE	1.9	244	77.92
MobileNetV2-Ghost-PC	1.9	244	79.48

Table 10. Experimental results of the MobileNetV2 model on the Mini-ImageNet dataset lightweight optimization of the channel compressed pointwise convolution with Ghost-PC.

Methods	Parameters (MB)	FLOPs (MB)	Accuracy (%)
MobileNetV2	2.3	319	77.91
MobileNetV2-Ghost-PC	1.7	245	78.62

It can be seen from Tables 9 and 10 that optimize channel compressed pointwise convolution has caused a great loss to the performance. In Table 9, no matter L1Norm, Ghost module or Ghost-PE, they all decrease by nearly 4 percentage points. Although the Ghost-PC block fails to achieve the original accuracy, it has better result compared with other methods.

Combined with the above experimental results, the lightweight optimization of MobileNetV1 and MobileNetV2 by Ghost-PE and Ghost-PC modules achieves better performance compared with other methods. At the same time, it can be seen that the channel expanded pointwise convolution structure is more suitable for lightweight optimization than the compressed structure.

4. Conclusions

The pointwise convolution structure plays an important role in the compact model, but it contains a large number of parameters and computation resources. Therefore, Ghost-PE and Ghost-PC modules are proposed in this paper to improve the channel expanded and compressed pointwise convolution structure, respectively, and their effectiveness is verified by experiments. To the channel compressed pointwise convolution, the number of output features maps is far less than the input, which would cause each channel of output contains a lot of characteristic information. Thus, we use the Ghost-PC to optimize it. In the future, we will further analyze the channel compressed pointwise convolution structure and try to optimize it from a different point of view to improve its performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61976217), the Fundamental Research Funds of Central Universities (No. 2019XKQYMS87) and the Science and Technology Planning Project of Xuzhou (No. KC21193).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, (2018), 6848–6856. https://doi.org/10.1109/CVPR.2018.00716
2. N. Ma, X. Zhang, H. T. Zheng, J. Sun, ShuffleNet V2: Practical guidelines for efficient CNN architecture design, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, (2018), 116–131. Available from: https://openaccess.thecvf.com/content_ECCV_2018/papers/Ningning_Light-weight_CNN_Architecture_ECCV_2018_paper.pdf.
3. D. Zhou, Q. Hou, Y. Chen, J. Feng, S. Yan, Rethinking bottleneck structure for efficient mobile network design, in *Computer Vision – ECCV 2020*, Springer, Cham, (2020), 680–697. https://doi.org/10.1007/978-3-030-58580-8_40
4. Y. Jia, W. Miao, C. Jiang, W. Ye, An improved pointwise convolutional block for efficient model compression, in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, (2019), 24–28. https://doi.org/10.1109/ICSESS47205.2019.9040771
5. S. Mehta, H. Hajishirzi, M. Rastegari, Dicenet: Dimension-wise convolutions for efficient networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2020), 2416–2425. https://doi.org/10.1109/TPAMI.2020.3041871
6. C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, et al., Lite-HRNet: A lightweight high-resolution network, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 10440–10450. Available from: https://openaccess.thecvf.com/content/CVPR2021/papers/Yu_Lite-HRNet_A_Lightweight_High-Resolution_Network_CVPR_2021_paper.pdf.
7. H. Yang, Z. Shen, Y. Zhao, AsymmNet: Towards ultralight convolution neural networks using asymmetrical bottlenecks, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (2021), 2339–2348. https://doi.org/10.1109/CVPRW53098.2021.00266
8. Y. Li, Y. Chen, X. Dai, D. Chen, M. Liu, L. Yuan, et al., Micronet: Improving image recognition with extremely low flops, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 468–477. Available from: https://openaccess.thecvf.com/content/ICCV2021/papers/Li_MicroNet_Improving_Image_Recognition_With_Extremely_Low_FLOPs_ICCV_2021_paper.pdf.
9. F. Liang, Z. Tian, M. Dong, S. Cheng, L. Sun, H. Li, et al., Efficient neural network using pointwise convolution kernels with linear phase constraint, *Neurocomputing*, **423** (2021), 572–579. https://doi.org/10.1016/j.neucom.2020.10.067
10. M. Villaret, Grouped pointwise convolutions significantly reduces parameters in efficientnet, in *Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence*, IOS Press, **339** (2021), 383.

11. M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in *Proceedings of the 36th International Conference on Machine Learning*, **97** (2019), 6105–6114. Available from: http://proceedings.mlr.press/v97/tan19a.html.
12. J. P. S. Schuler, S. R. Also, D. Puig, H. Rashwan, M. Abdel-Nasser, An enhanced scheme for reducing the complexity of pointwise convolutions in CNNs for image classification based on interleaved grouped filters without divisibility constraints, *Entropy*, **24** (2022), 1264. https://doi.org/10.3390/e24091264
13. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., MobileNets: Efficient convolutional neural networks for mobile vision applications, preprint, arXiv:1704.04861.
14. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
15. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: More features from cheap operations, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165
16. L. Bossard, M. Guillaumin, L. V. Gool, Food-101-mining discriminative components with random forests, in *Computer Vision – ECCV 2014*, (2014), 446–461. https://doi.org/10.1007/978-3-319-10599-4_29
17. A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, 2009.
18. O. Vinyals, C. Blundell, T. Lillicrap, K. kavukcuoglu, D. Wierstra, Matching networks for one shot leaming, in *Advances in Neural Information Processing Systems*, **29** (2016). Available from: https://proceedings.neurips.cc/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.
19. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, et al., Automatic differentiation in pytorch, 2017. Available from: https://openreview.net/forum?id=BJJsrmfCZ.
20. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
21. Y. Wei, P. W. Yang, F. Ducau, K. Liu, Pytorch-cifar. Available from: https://github.com/kuangliu/pytorch-cifar.
22. H. Li, A. Kadav, I. Durdanovic, H. Samet, H. P. Graf, Pruning filters for efficient convnets, preprint, arXiv:1608.08710.

AIMS Press