A new lightweight network based on MobileNetV3

Liquan Zhao[*] and Leilei Wang
Key Laboratory of Modern Power System Simulation and Control and Renewable Energy Technology, Ministry of Education, Northeast Electric Power University
Jilin, 132012 China
[e-mail: zhaoliquan@neepu.edu.cn]
[*]Corresponding author: Liquan Zhao

Received April 7, 2021; revised September 23, 2021; revised November 18, 2021; accepted December 5, 2021; published January 31, 2022

Abstract

The MobileNetV3 is specially designed for mobile devices with limited memory and computing power. To reduce the network parameters and improve the network inference speed, a new lightweight network is proposed based on MobileNetV3. Firstly, to reduce the computation of residual blocks, a partial residual structure is designed by dividing the input feature maps into two parts. The designed partial residual structure is used to replace the residual block in MobileNetV3. Secondly, a dual-path feature extraction structure is designed to further reduce the computation of MobileNetV3. Different convolution kernel sizes are used in the two paths to extract feature maps with different sizes. Besides, a transition layer is also designed for fusing features to reduce the influence of the new structure on accuracy. The CIFAR-100 dataset and Image Net dataset are used to test the performance of the proposed partial residual structure. The ResNet based on the proposed partial residual structure has smaller parameters and FLOPs than the original ResNet. The performance of improved MobileNetV3 is tested on CIFAR-10, CIFAR-100 and ImageNet image classification task dataset. Comparing MobileNetV3, GhostNet and MobileNetV2, the improved MobileNetV3 has smaller parameters and FLOPs. Besides, the improved MobileNetV3 is also tested on CPU and Raspberry Pi. It is faster than other networks

Keywords: MobileNetV3, Real-time image classification, Lightweight network, Deep convolutional neural network, residual structure

1. Introduction

Deep convolutional neural networks are widely used in computer vision tasks, natural language processing and machine fault diagnosis. To obtain higher accuracy, large-scale data sets such as ImageNet and the application of data processing methods can be used to improve the performance of deep learning networks [1-3]. Increasing the depth and width of the convolutional neural network also can improve the network performance. Increasing the depth and width will increase the complexity of the network, which requires more computing power. In many practical applications (such as augmented reality, autonomous driving and smart robot), the deep convolutional neural networks are deployed on embedded devices or mobile devices which have limited computing power and memory. Therefore, it is difficult to realize the real-time object detection on these devices. To solve the problem, lightweight neural networks are proposed to reduce the complexity of network. Although they have lower accuracy than a normal neural network, they have better real-time performance in image recognition. The accuracy can also be accepted in real application. The lightweight network has been successfully applied in object detection [4], solar irradiance prediction [5] and network reconfiguration [6].

The ShuffleNet series (ShuffleNet[7], ShuffleNetV2[8]), MobileNet series (MobileNetV2[9], MobileNetV3[10]) and GhostNet[11] are typical lightweight convolutional neural networks. The original residual block is composed of three traditional convolutional layers. The sizes of the convolution kernel of the three layers are 1×1, 3×3, 1×1, respectively. To reduce the amount of convolution calculation, ShuffleNetV1 uses point wise group convolution and 3×3 depthwise convolution to replace 1×1 convolution and 3×3 convolution in the residual block, respectively. The ShuffleNetV2 proposes four effective network design guidelines and uses these guidelines to improve the residual block in shuffleNet. The MobileNetV2 proposes the inverted residuals block that increases the number of channels in the first convolutional layer in the residual block. MobileNetV3 uses the inverted residual block, the SE module and the non-linear function h-swish to improve the network. The GhostNet proposed the Ghost module and used it to construct the residual structure, which can greatly reduce the amount of calculation compared with the original residual block. They use different residual blocks to construct the network. The performance of residual blocks directly affects the performance of lightweight convolutional neural networks.

The previous works mainly focused on the increase or decrease of the number of convolutional layers and the number of channels in the residual block without considering the network structure. In this paper, we improve the network structure of partialresidual block. In the original network structure, it consists of shortcut connection, residual learning part and element-wise addition operation. The output of upper layer is directly used as the input of network. For our network structure, it has two branches and the output of the upper layer is divvied into two equal parts to be used as the input of two branches, respectively. The two branches have different sizes of convolution kernels and same number of channels. The outputs of two branches are connected by concatenate operation without using element-wise addition operation. Besides, the MobileNetV3 network directly uses many partial residual blocks to construct network. We design new MobileNetV3 network based on our proposed partial residual block and transition layers to reduce the parameters and FLOPs of the network.

The contributions of the method we proposed are summarized as follows:

1. In partial residual block, the feature maps of the base layer are divided into two parts, one

part is used as the input of the residual learning block, and the other part is directly connected to the output of the residual learning block. This structure can greatly reduce computation.

2. Based on our designed partial residual block, we designed a new module named DB-block A that is used as one part of improved MobileNetV3 network to reduce parameter. Compared with original module, it has two branches. The two branches are composed of two independent partial residual structure modules with different convolution kernel sizes, and channel dimension connect operation is used to connect the outputs of two branches. In the end, we design transition layer to fuse feature to ensure the accuracy of the model

3. The MonileNetV3 network contains modules that do not use residual structure. These modules consist of a single branch. To reduce parameters, we designed an improved module named DB-block B. It also has two branches with different convolution kernel sizes, and channel connect operation is also used at the end of the two branches. To ensure the accuracy of the module, the designed transition layer is also used to realize feature fusion.

The rest of our works as follows: In section 2, we introduced the main research methods of the lightweight neural networks. In section 3, we introduced our proposed partial residual blocks and the partial residual network. Then we introduced the proposed lightweight neural network DP-Net witch network structure is based on the proposed partial residual blocks and one of the state-of-the-art lightweight architectures. In section 4, we verified the effectiveness of the proposed partial residual network and DP-Net on several basic classification datasets and compared the proposed DP-Net with several popular lightweight neural networks. In section 5, we summarized the network proposed in this paper.

2. Related work

Research of lightweight neural networks can be divided into two directions, one is the model compression based on the pre-training model, and the other one is the model design that proposes the lightweight model. The model compression method can be realized by network pruning, quantization or knowledge distillation. In the aspect of network pruning, Han et al. proposed deep compression lightweight network by using pruning, trained quantization and Huffman coding [12]. It reduces the weight storage without loss of accuracy. To prune the deep neural network, He et al. proposed an iterative two-step channel pruning algorithm, and used it to prune the VGG16. It can achieve 5× speed-up, with only an error rate increase of 0.3% [13]. In the aspect of network quantization, Rastegari et al. proposed two binary networks that binary the filters and the input of convolution layer to save memory [14]. Wang et al. proposed a hardware-aware automatic quantification framework that uses reinforcement learning to automatically determine quantification strategies [15]. In the aspect of knowledge distillation, Chen et al. used GANs and teacher-student network to enable student network to learn the ability of teacher network without training data [16]. The RKD method utilized distance-wise and angle-wise distillation to improve educated student models [17]. All the above methods are based on the basic deep neural network.

Model design is the second research direction of lightweight neural networks. It includes ShuffleNet series method (ShuffleNet[7], ShuffleNetV2 [8]), MobileNet series method (MobileNets[18], MobileNetV2 [9], MobileNetV3 [10]). To reduce the computation of the network, the ShuffleNet method replaces traditional convolution with pointwise group convolutions and uses channel shuffle to solve the problem that information can’t be well communicated in different feature channels. According to the four design principles of the lightweight network, ShuffleNetV2 is proposed by redesigning the network architecture of ShuffleNetV1. In ShuffleNetV2, the pointwise group convolutions are abandoned and the element-wise addition is replaced by concatenate operation to improve the network speed. The MobileNets is proposed by Howard et al. [18]. it uses depth-wise separable convolutions to build neural networks and introduces two global hyper-parameters to trade off latency and accuracy. The MobileNetV2 [9] is based on linear bottlenecks and inverse residual to solve the problem of information loss in low dimensional feature maps and improve the network accuracy. The network structure of MobileNetV3 [10] is based on Auto ML technology [19, 20]. This structure not only includes depth-wise separable convolutions, inverted residuals and linear bottlenecks, but also SE [21] models and the new activation functions h-swish. Therefore, this network can achieve better performance with fewer FLOPs.

To balance the accuracy and speed, the EfficientNet[22] changes the size of the network from the network depth, width and image resolution aspects. Tan et al. proposed the MnasNet[23] method. They take the model delay as the objective function and use the network search technology to design resource-efficient convolutional neural network mobile. The SqueezeNet[24] designs a new network module by using the 1×1 convolution to replace 3×3 convolution. It can achieve AlexNet-level accuracy on ImageNet with 50x fewer parameters. The Xception[25] is based on inception network [26]. It replaces convolutions operation in the inception network with the depthwise separable convolutions to improve accuracy. Wu et al. point that the computational complexity of spatial convolution will grow quadratically with respect to filter size, so they use shifts and point-wise convolution to build end-to-end shift-base modules instead of spatial convolution to reduce computational complexity [27]. GhostNet[11] proposed by Han et al. is based on original feature maps by using a series of linear transformations to generate some effective feature maps. On the ImageNet dataset, this network can achieve a better classification effect with lower computational cost. LegoNet[28] designs the new filter modules that are assembled by a set of shared Lego filters, which are usually much smaller in size. Using the LegoNet as part of the network structure can reduce parameters and speed up the network.

In the research of lightweight networks, the model compression methods use pruning, quantization or knowledge distillation to reduce the size of the pre-training model. The model design methods are to design networks with smaller sizes. The final goal of lightweight networks is to reduce the parameters and computation complexity with acceptable accuracy.In this paper, we propose a new lightweight network based on MobileNetV3.Adouble branches module that combines the channel splitting operation with the residual structure to reduce the amount of parameters and the calculation cost is designed. In each module, input is divided into two low-dimensional branches. In order to maintain accuracy, the two branches adopt convolution kernels with different sizes to extract the multi-scale image features and enrich the information of each layer. Transition layer is also designed to realize the information exchange between the two branches. Using the proposed double branches module, we redesigned the network structure of MobileNetV3. It occupies less memory and has lower computational complexity than MobileNetV3 with almost similar classification accuracy.

3. Proposed method

3.1 Proposed partial residual network

Residual structure [29] has better performance in convergence and prevention of network degradation. Therefore, it is widely used in MobileNetV3 [10], ShufflenetV2 [8], GhostNet[11] and other lightweight models. The output of residual structure network with k layers can be expressed as follows:

$$\begin{aligned} x_k &= R_k\left(x_{k-1}\right) + x_{k-1} \\ &= R_k\left(x_{k-1}\right) + R_{k-1}\left(x_{k-2}\right) + \ldots + R_1\left(x_0\right) + x_0 \end{aligned} \tag{1}$$

where R represents the calculation operator of the residual layer. x_0 represents the initial input, $R_1\left(x_0\right)$ represents the output of the first residual module, and so on, x_{k-1} represents the output of the $k-1$ residual modules. It is usually composed of two or three convolutional layers. It is also called residual block in the network. The input of each residual block is the sum of previous residual block outputs. It can slow down network degradation and improve network performance. However, this will make the input of the next layer contain much similar or repeated information in residual blocks. There will be more repeated or similar feature maps in the middle layer of the network, as shown in **Fig. 1**. With the number of layers increases, it will make the network repeatedly learn redundant information and increase the computation cost of the network.

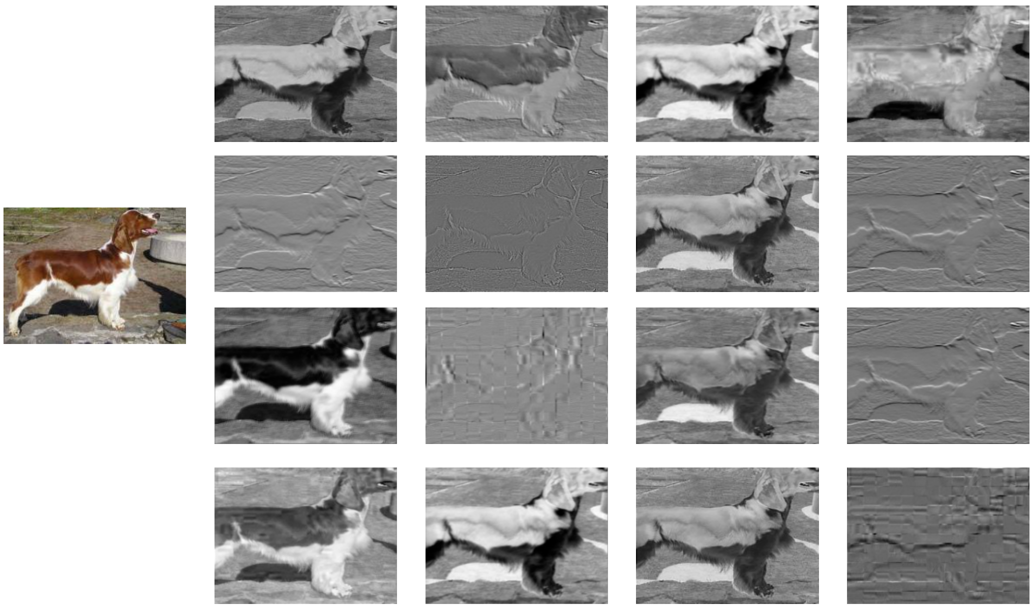


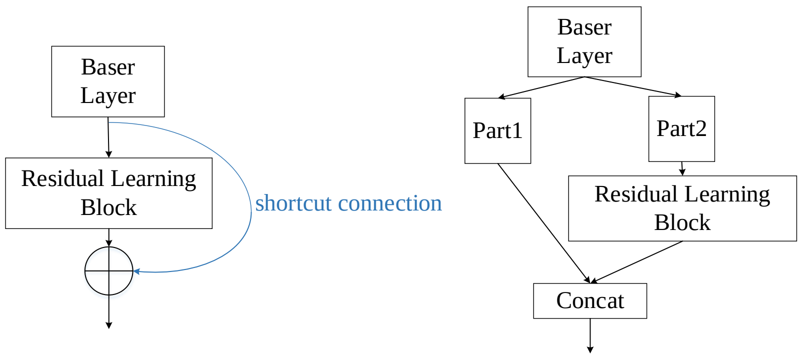
Fig. 1. Visualization of some feature maps composed of the first residual of ResNet-50.

To avoid repeatedly learning redundant information, we divide the input feature maps into two different parts and they pass different branches, respectively. The redundant information will be reduced in dividing input feature maps. Based on this idea, we designed new residual structure. The proposed structure is called partial residual structure (It is also called partial residual block in the network.), and the neural network stacked by partial residual blocks is called partial residual network. The original residual structure and proposed residual structure are shown in **Fig. 2**. As shown in **Fig. 2(b)**, we evenly divide the input feature maps into two parts in the channel dimension. The two parts have the same number of channels. Concatenate operations are used to connect the output of two parts also in the channel dimension, and the final output channel number is the same as the original structure. The main difference between the original residual structure and proposed residual structure is that the residual learning block and shortcut connection requires processing all input feature maps, whiling the residual learning block and shortcut connection only requires to process half input feature maps in proposed residual structure. Therefore, proposed residual structure reduces the complexity of original residual structure.

The theoretical speedup ratio of partial residual structure and residual structure is expressed as follows:

$$r = \frac{residual\ structure}{partial\ residual\ structure} = \frac{W \times H \times C_{in} \times C_{out} \times K^2}{W \times H \times C_{in} / 2 \times C_{out} / 2 \times K^2} = 4 \tag{2}$$

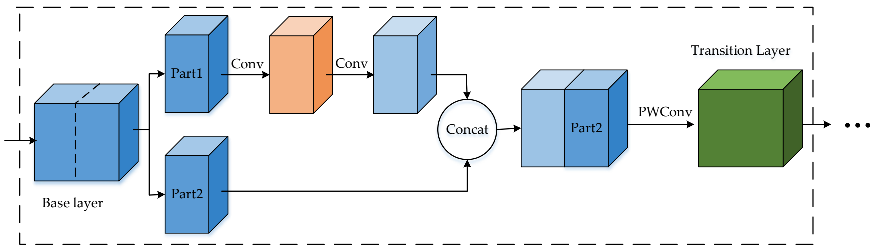
where H and W respectively represent the length and width of the output feature map, K represents the size of the convolution kernel, C_{in} and C_{out} represent the number of input feature channels and the number of output feature channels, respectively.



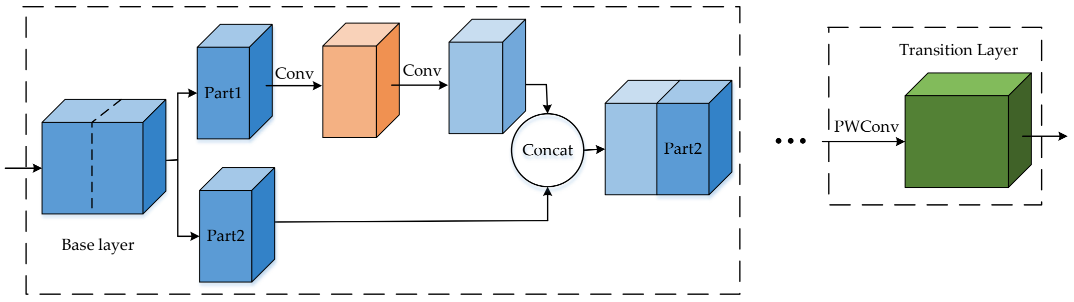
(a) (b)

Fig. 2. Residual structure. (a) Original residual structure; (b) proposed residual structure.

The concatenate operation is just to connect the outputs of the residual learning block and shortcut connection. If we directly use the output of concatenate operation as the input of the next network and divide feature maps into two parts. One part may be still as the input of the residual learning block, and the other part is still as the input of shortcut connection. The information of two parts has not any fusion. This affects feature extraction. To solve the problem, we design the transition layer to realize the feature fusion. The transition layer is made up of a point wise convolution. The proposed transition layer is shown in **Fig. 3**. In **Fig. 3(a)**, the transition layer and partial residual block compose a whole block. We can stack the new block to construct a network. In **Fig. 3(b)**, the transition layer is an independent block which is added to the network. If the number of partial residual block channels changes, the transition layer will be added between the two partial residual blocks. Taking ResNet50 as an example, the network is divided into 4 stages according to the number of channels. The numbers of channels are 256, 512, 1024, 2048 for the 4 stages, respectively. The numbers of residual blocks are 3, 4, 6, 3 for the 4 stages, respectively. Therefore, the transition layers are added behind the third, seventh, thirteenth, and sixteenth residual blocks. Due to that each partial residual block refers to feature fusion, the parameters and calculations of the transition layer will sharply increase, with the number of channels increasing. Using this strategy can reduce the parameters which caused by transition layers.



(a)



(b)

Fig. 3. Residual block with transition layer. (a) A residual block with a transition layer. (b) Many residual blocks with a transition layer.

3.2 Improved MobileNetV3

In this section, we use our proposed partial residual structure to improve the MobileNetV3. The architecture of MobileNetV3 is composed of a series of bottleneck blocks. Some of the bottleneck blocks include residual structure. To reduce the computation cost of the residual structure, we use the proposed partial residual structure to replace the original residual structure in the bottleneck blocks. In the original bottleneck blocks, there is only one feature extraction path. In order to further reduce the parameters and ensure the accuracy of the network, we change the single feature extraction path in the original bottleneck blocks into dual feature extraction paths. The designed dual-path feature extraction bottleneck with partial residual structure is shown in **Fig. 4**.

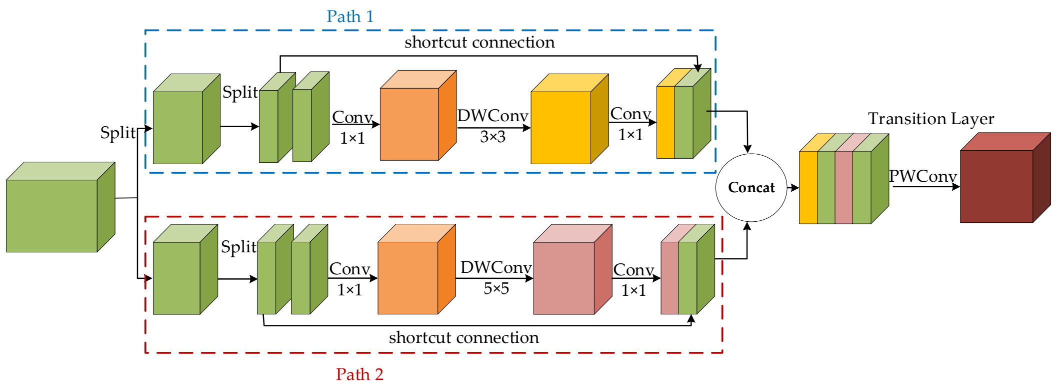


Fig. 4. Dual-path feature extraction bottleneck with partial residual structure.

The input feature map of the bottleneck block is evenly divided into two parts. One part is used as input feature map of path1, and the other part is used as input feature map of path2. In path1, the input feature map is evenly divided into two parts again. One part passes through 1×1convolution layer, 3×3 depthwise convolution layer, and 1×1 convolution layer. The other is directly connected with the output of the residual learning block through the shortcut connection. used as one part of the output feature. In path2, the input feature map also is evenly divided into two parts. One part passes through 1×1 convolution layer, 5×5 depthwise convolution layer, and 1×1convolution layer. The other directly also is used as one part of the output feature. The concatenation operation is used to connect the output feature maps of two paths. We also use the pointwise convolution to construct the transition layer to fusion the concatenated feature maps. When the size of the recognition target in the picture is too small or too large, if we use the same convolution kernel to extract features in CNN, the extracted information will be incomplete. Therefore, we use different convolution kernels in path1 and path2 to improve accuracy and reduce the affection of branch structure on accuracy. Due to that the depthwise convolutions are different in two paths; we use the padding method to make the output feature maps have the same dimensions. The padding is 1 in path1 and 2 in path2.

For simplicity, we only discuss the computational complexity of the deep convolutional layer in the bottleneck block using the original residual structure and improved residual structure. In the case of considering bias, FLOPs of this layer can be expressed as follows:

$$\begin{aligned} FLOPs_{3\times 3} &= 2\times C_{in}\times W\times H\times 3^2 \\ FLOPs_{5\times 5} &= 2\times C_{in}\times\times W\times H\times 5^2 \\ FLOPs_{our} &= 2\times C_{in}/4\times W\times H\times(3^2+5^2) \end{aligned} \tag{3}$$

where $FLOPs_{3\times 3}$ and $FLOPs_{5\times 5}$ represent the computational complexity when the convolution kernel size of the deep convolution layer in the original bottleneck is 3 and 5, respectively. $FLOPs_{our}$ represents the computational complexity of the improved depthwise convolution layer of the bottleneck block. H and W respectively represent the length and width of the output feature map, K represents the size of the convolution kernel, C_{in} and C_{out} represent the number of input feature channels and the number of output feature channels, respectively. The complexity comparison is as follows:

$$\begin{aligned} \frac{FLOPs_{3\times 3}}{FLOPs_{our}} &= \frac{2\times C_{in}\times W\times H\times 3^2}{2\times C_{in}/4\times W\times H\times(3^2+5^2)} = \frac{3^2\times 4}{3^2+5^2}\approx 1.06 \\ \frac{FLOPs_{5\times 5}}{FLOPs_{our}} &= \frac{2\times C_{in}\times W\times H\times 5^2}{2\times C_{in}/4\times W\times H\times(3^2+5^2)} = \frac{5^2\times 4}{3^2+5^2}\approx 2.94 \end{aligned} \tag{4}$$

It can be seen from (4) that no matter whether the size of the convolution kernel used in the original bottleneck block is 3×3 or 5×5, the computational complexity of the improved bottleneck block is always lower than the original block.

The architecture of MobileNetV3 also has some bottleneck blocks without residual blocks. For these bottleneck blocks, we also design a dual-path feature extraction structure. The designed dual-path feature extraction structure is shown in **Fig. 5**. The input feature maps of the bottleneck block are also evenly divided into two parts. Different from the partial residual structure in **Fig. 4**, each path of the bottleneck block has no shortcut connection. Therefore, the input feature maps of each path are no longer divided into two parts. The rest is the same as the partial residual structure in **Fig. 4**.

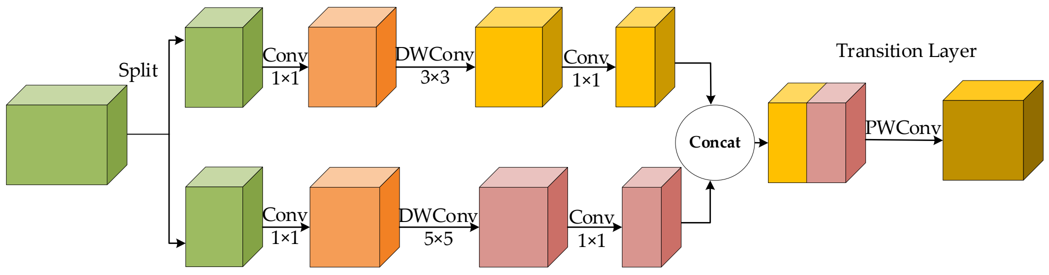


Fig. 5. The dual path feature extraction blocks.

For simplicity, we named the block that is shown in **Fig. 4** DP-block A, and the block that is shown in **Fig. 5** DP-block B. Based on DP-block A and DP-block B, the detailed network structure of the proposed method based on MobileNetV3 is shown in **Table 1**. The first layer of the network is a standard convolutional layer with 16 filters. The middle part of the network is composed of a series of DP-block A and DP-block B, and the number of network channels is gradually increasing. According to the input feature map size in the DP-block A and DP-block B, the network of our proposed method can be divided into several stages. The stride of the last DP-block is 2 in each stage, and the strides of other DP-blocks are 1. This makes the feature map size of our proposed method equal to the feature map size of MobileNetV3 in the same stage. The last part of the network structure consists of an average pooling layer and a standard convolutional layer. Squeeze-And-Excite (SE) [21] is also used in some DP-blocks. In **Table 1**, the "#exp" means expansion size. "#out" expresses the number of output channels. SE indicates whether to use the SE module.

Table 1. Network structure of our proposed method

Input	Operator	#exp size	#Out	SE	NL	s
224^2×3	Conv2d	-	16	-	HS	2
112^2×16	DP-block A	16	16	-	RS	1
112^2×16	DP-block B	64	24	-	RS	2
56^2×24	DP-block A	72	24	-	RS	1
56^2×24	DP-block B	72	40	√	RS	2
28^2×40	DP-block A	120	40	√	RS	1
28^2×40	DP-block A	120	40	√	RS	1
28^2×40	DP-block B	240	80	-	HS	2
14^2×80	DP-block A	200	80	-	HS	1
14^2×80	DP-block A	184	80	-	HS	1
14^2×80	DP-block A	184	80	-	HS	1
14^2×80	DP-block B	480	112	√	HS	1
14^2×112	DP-block A	672	112	√	HS	1
14^2×112	DP-block B	672	160	√	HS	2

7^2×160	DP-block A	960	160	√	HS	1
7^2×160	DP-block A	960	160	√	HS	1
7^2×160	Conv2d,1×1	-	960	-	HS	1
7^2×960	Pool, 7×7	-	-	-	-	1
1^2×960	Conv2d 1×1, NBN	-	1280	-	HS	1
1^2×1280	Conv2d 1×1, NBN	-	k	-	-	1

4. Experiments

Large-scale data sets play a key role in providing predictive analysis and solutions for deep learning models[30].In this section, we test the efficiency of the proposed partial residual structure and our proposed MobileNetV3(named DP-Net) on the CIFAR-100 dataset, CIFAR-10 dataset, and Image Net ILSVRC 2012 dataset, respectively. The CIFAR-100 dataset has 100 categories, each category contains 600 images. Each category contains 500 training images and 100 test images. The CIFAR-10 dataset contains 10 categories. There are 50000 training images and 10000 test images in the dataset. The ImageNet ILSVRC 2012 is a large data set, which contains 1.2 million training images and 50K verification images in 1000 categories. The size of the dataset is 160.7G. Three datasets adopt a general data enhancement scheme including random cropping and mirroring. For a fair comparison, the training settings for all experiments in this paper start with a learning rate of 0.1 and momentum of 0.9, and dropout of 0.2 and weight decay of 0.00001. The mini-batch size is 64 for ResNet-based experiments, and 128 for other experiments. The operating system is Ubuntu 18.04. The CPU is Intel Xeon E5-2678 v3 with 2.5 GHZ main frequency. The GPU is NVIDIA GeForce GTX 1080Ti. The deep learning framework is PyTorch.

4.1 Efficiency of proposed partial residual structure

Many lightweight networks are designed based on ResNet. Our proposed partial residual structure is designed to reduce parameters and computation of ResNet and other networks that are based on the ResNet network. On the CIFAR-100 dataset, we compare the proposed partial residual network with ResNet in accuracy and parameters when the network layers are 18 and 34, respectively. For simplicity, we name the network P-ResNet A in **Fig. 3(a)** and the network P-ResNet B in **Fig. 3(b)**. The accuracy and parameters of the ResNet method and the improved ResNet methods based on our proposed P-ResNet A and P-ResNet B are shown in **Table 2**. Compared with ResNet, when the number of layers is 18, the accuracies of P-ResNet18A and P-ResNet18B are respectively reduced by 1.1% and 1.7%, the parameters of P-ResNet18A and P-ResNet18B are respectively reduced by 65.6% and 68.6%. When the number of layers is 34, the accuracies of P-ResNet34A and P-ResNet34B are respectively reduced by 5.0% and 5.8%, the parameters of P-ResNet34A and P-ResNet34B are respectively reduced by 67.4% and 71.6%. Although the ResNet has the highest accuracy than our proposed networks, the parameters of ResNet are more than our proposed networks. Besides, the difference of accuracy between the ResNet and our proposed networks is much smaller, the difference of parameters between the ResNet and our two proposed ResNet is significant. This shows that our proposed ResNets have lower memory than the original ResNet.

Table 2. Performance of the ResNets and the improved ResNets on CIFAR100 dataset

Network	TOP-5 Acc. (%)	Parameters (M)
ResNet18	91.54	11.69
P-ResNet18A	90.56(-1.1%)	4.02(-65.6%)
P-ResNet18B	89.96(-1.7%)	3.67(-68.6%)
ResNet34	92.20	21.80
P-ResNet34A	87.55(-5.0%)	7.11(-67.4%)
P-ResNet34B	86.83(-5.8%)	6.20(-71.6%)

On the ImageNet dataset, we compare our proposed ResNets and original ResNet. The accuracies, parameters and FLPOs (the number of multiply-accumulates) of different networks are shown in **Table 3**. Compared with ResNet, when the number of layers is 18, the accuracies of P-ResNet18A and P-ResNet18B are respectively reduced by 4.5% and 5.9%, the FLOPs of P-ResNet18A and P-ResNet18B are respectively reduced by 64.3% and 67.0%. When the number of layers is 34, the accuracies of P-ResNet34A and P-ResNet34B are respectively reduced by 4.5% and 9.8%, the FLOPs of P-ResNet34A and P-ResNet34B are respectively reduced by 66.8% and 71.1%. Besides, the difference of accuracy between the ResNet and our proposed networks is much smaller, the difference of FLOPs between the ResNet and our two proposed ResNet is significant. This shows that our proposed ResNets have lower computational complexity than the original ResNet.

Based on **Table 2** and **Table 3**, although the accuracy of P-ResNetA is higher than that of P-ResNetB, the parameters and FLOPs of P-ResNetA are more than that of P-ResNetB. To balance the accuracy, memory and real-time, P-ResNet A method is used in our improved MobileNetV3.

Table 3. Performance of the ResNet and the improved ResNets on ImageNet dataset

Network	TOP-5Acc. (%)	Parameters (M)	FLOPs(G)
ResNet18	70.05	11.69	1.82
P-ResNet18A	66.87(-4.5%)	4.02(-65.6%)	0.65(-64.3%)
P-ResNet18B	65.91(-5.9%)	3.67(-68.6%)	0.60(-67.0%)
ResNet34	71.70	21.80	3.67
P-ResNet34A	68.44(-4.5%)	7.11(-67.4%)	1.22(-66.8%)
P-ResNet34B	64.70(-9.8%)	6.20(-71.6%)	1.06(-71.1%)

4.2 Efficiency of proposed partial residual structure

In this section, we test the performance of our improved MobileNetV3 and other methods on the CIFAR10 dataset, CIFAR100 dataset and ImageNet dataset.

The accuracy and parameters of MobileNetV3, improved MobileNetV3 based on the partial residual block (P-MobileNetV3), improved MobileNetV3 based on dual-core structure (D-MobileNetV3) and complete improved MobileNetV3 based on partial residual block and dual-core structure (DP-Net), are shown in **Table 4**. Comparing with MobileNetV3, the TOP-1 accuracy of D-MobileNetV3, P-MobileNetV3 and DP-Net are reduced by 1.8%, 5.1% and 2.2%, respectively. The parameters of D-MobileNetV3, P-MobileNetV3 and DP-Net are reduced by 20.4%, 21.4% and 33.8%, respectively. The DP-net with D-MobileNetV3 and P-MobileNetV3 structure has the smallest parameters. Comparing with MobileNetV3, the

TOP-1 accuracy of DP-Net reduces by 2.2%, but the parameters of DP-Net are reduced by 33.8%. The reduction of accuracy is much smaller than that of parameters.

Table 4. Performances of our proposed methods and other methods on CIFAR100 dataset

Network	TOP-1 Acc. (%)	Parameters (M)
MobileNetV3	70.15	5.15
D-MobileNetV3	68.92(-1.8%)	4.10(20.4%)
P-MobileNetV3	66.58(-5.1%)	4.05(21.4%)
DP-Net	**68.63(-2.2%)**	**3.41(33.8%)**

The accuracies and parameters of MobileNetV3, MobileNetV2, GhostNet and improved MobileNetV3 (DP-Net) on the CIFAR10 dataset are shown in **Table 5**. Comparing with MobileNetV3, GhostNet and MobileNetV2, the TOP-1 accuracy of DP-Net is reduced by 1.0%, 1.4% and 1.5%, respectively. The TOP-5 accuracy of DP-Net is reduced by 0.2%, 0.1% and 0.1%, respectively. The parameters of DP-Net are reduced by 33.8%, 34.2% and 2.6%, respectively. Although the accuracy of DP-Net is the lowest, the accuracy differences between these methods are very small. Besides, the differences between DP-Net and other networks on parameters are much larger than on accuracy. This shows that the DP-Net has lower memory than others with an acceptable accuracy.

Table 5. Performances of improved MobileNetV3 and other methods on CIFAR10 dataset

Network	TOP-1 Acc. (%)	TOP-5 Acc. (%)	Parameters (M)
MobileNetV3	91.92(1.0%)	99.81(0.2%)	5.15(33.8%)
GhostNet	92.22(1.4%)	99.75(0.1%)	5.18(34.2%)
MobileNetV2	92.34(1.5%)	99.72(0.1%)	3.50(2.6%)
DP-Net	**90.97**	**99.61**	**3.41**

The accuracies, parameters and FLOPs of different methods on the ImageNet dataset are shown in **Table 6**. Comparing with MobileNetV3, GhostNet and MobileNetV2, the TOP-1 accuracy of DP-Net is reduced by 1.8%, 0.5% and 0.2%, respectively. The TOP-5 accuracy of DP-Net is reduced by 1.0%, 0.5% and 0.3%, respectively. The FLOPs of DP-Net are reduced by 55.1%, 21.9% and 63.1%, respectively. Although MobileNetV3, GhostNet and MobileNetV2 methods can achieve high TOP-1 accuracy, they also have more parameters and computational complexity. Besides, the differences in accuracy between the DP-Net and other methods are very small. The reduction of accuracy is much smaller than that of FLOPs. The proposed DP-Net can achieve acceptable accuracy with lower computational complexity.

Table 6. Performances of improved MobileNetV3 and other methods on ImageNet dataset

Network	TOP-1 Acc. (%)	TOP-5 Acc. (%)	FLOPs(M)
MobileNetV3	71.32(1.8%)	89.83(1.0%)	263(55.1%)
Ghost Net	70.36(0.5%)	89.44(0.5%)	151(21.9%)
MobileNetV2	70.13(0.2%)	89.21(0.3%)	320(63.1%)
DP-Net	**70.01**	**88.97**	**118**

To test the performance of the proposed method on CPU and embedded devices that have limited computation, we run the different methods on CPU and Raspberry Pi on the CIFAR100 dataset, respectively. The CPU is AMD Ryzen 7 PRO 4750U with Radeon Graphics 1.70 GHz. The model of Raspberry Pi is 3B with a BC219M2835 processor. The running times of different methods are shown in **Table 7**. The DP-Net has the least running time on CPU and Raspberry Pi. This means that our proposed method has better performance in real-time image recognition than others.

Table 7. Running time of different methods on CPU and Raspberry pi

Network	**CPU (ms)**	**Raspberry pi(s)**
MobileNetV3	236	7.85
Ghost Net	213	6.67
MobileNetV2	265	8.41
DP-Net	178	5.31

5. Conclusion

We proposed a new lightweight network based on MobileNetV3. There are three main contributions: 1) To reduce the computational complexity of the residual structure, we designed a partial residual structure. 2) In order to further reduce the network parameters and ensure accuracy, we designed a dual-path feature extraction structure with different size filters. 3) To ensure the performance of the network, we designed a transition layer to fuse the feature map of the two paths. Comparing with the original ResNet, the ResNet-designed partial residual structure has smaller parameters and FLOPs on the CIFAR-100 dataset and Image Net dataset for a different number of layers. The improved MobileNetV3 based on the designed partial residual structure, dual-path feature extraction structure and transition layer has lower computation and memory than other networks with acceptable accuracy. Besides, we also test the different networks on CPU and Raspberry pi that has limited memory and computation power. The improved MobileNetV3 has a faster speed in image recognition than other networks. It is more suitable for real applications on embedded devices.

In our work, we cannot ensure that all similar feature maps are completely separated. Therefore, in our feature work, we will focus on identifying similar feature maps and dividing them exactly into two parts. The two parts will be used as inputs of two network branches to further improve accuracy. Besides, we will also research how to further reduce redundant information without increasing computation complexity.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61271115), Research Foundation of Education Bureau of Jilin Province (JJKH20210095KJ) and Science and technology development plan project of Jilin City (20190104124)

References

[1] G. A. Khan, J. Hu, T. Li, B Diallo and H. Wang, “Multi-view data clustering via non-negative matrix factorization with manifold regularization,” *Int. J. Mach. Learn. & Cyber*, pp.1-13, March, 2021. Article (CrossRef Link)

[2] B. Diallo, J. Hu, T. Li, G. A. Khan, A. S. Hussein, “Multi-view document clustering based on geometrical similarity measurement,” *Int. J. Mach. Learn. & Cyber*, pp.1-13, March, 2021. Article (CrossRef Link)

[3] G.A. Khan, J. Hu, T. Li, et al., “Multi-view low rank sparse representation method for three-way clustering,” *Int. J. Mach. Learn. & Cyber*, vol. 13, pp. 233-253, 2022. Article (CrossRef Link)

[4] Y. J. Wang, P. P. Cao, X. S. Wang, X. Y. Yan, “Research on Insulator Self Explosion Detection Method Based on Deep Learning,” *Journal of Northeast Electric Power University*, vol. 40, no. 3, pp. 33-40, June, 2020. Article (CrossRef Link)

[5] Z. X. Zhong, X. B. Ma, A Wei, “A Lightweight Solar Irradiance Prediction Model Based on Ground-based Cloud Images and Meteorological Data,” *Journal of Northeast Electric Power University*, vol. 41, no. 1, pp. 24-30, June, 2021. Article (CrossRef Link)

[6] H. Yang, K. Zhan, B. Bao, Q. Yao, J. Zhang and M. Cheriet, “Automatic guarantee scheme for intent-driven network slicing and reconfiguration,” *J NETW COMPUT APPL*, vol. 190, no. 15, 103163, September, 2021. Article (CrossRef Link)

[7] X. Zhang, X. Zhou, M. Lin and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in *Proc. of CVPR 2018*, Salt Lake City, USA, pp. 6848-6856, June 2018. Article (CrossRef Link)

[8] N. Ma, X. Zhang, H. T. Zheng and J. Sun, “ShuffleNetV2: Practical guidelines for efficient cnn architecture design,” in *Proc. of ECCV 2018*, Munich, Germany, pp. 122-138, September 2018. Article (CrossRef Link)

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. of CVPR 2018*, Salt Lake City, USA, pp. 4510-4520, June 2018. Article (CrossRef Link)

[10] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen et al., “Searching for MobileNetV3,” in *Proc. of ICCV 2019*, Seoul, Korea, pp. 1314-1324, 2019. Article (CrossRef Link)

[11] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu et al., “GhostNet: More features from cheap operations,” in *Proc. of CVPR 2020*, Seattle, WA, USA, pp. 1580-1589, June 2020. Article (CrossRef Link)

[12] S. Han, H. Mao and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” *arXiv preprint arXiv:1510.00149*, 2015.

[13] Y. He, X. Zhang and J. Sun, “Channel pruning for accelerating very deep neural networks,” in *Proc. of ICCV 2017*, Italy, pp. 1389-1397, October 2017. Article (CrossRef Link)

[14] M. Rastegari, V. Ordonez, J. Redmon and A. Farhadi, “Xnor-net: Imagenet classification using binary convolutional neural networks,” in *Proc. of ECCV 2016*, Amsterdam, the Netherlands, pp. 525-542, October 2016. Article (CrossRef Link)

[15] K. Wang, Z. Liu, Y. Lin, J. Lin and S. Han, “Haq: Hardware-aware automated quantization with mixed precision,” in *Proc. of CVPR 2019*, Long Beach, CA, USA, pp. 8612-8620, June 2019. Article (CrossRef Link)

[16] H. Chen, Y. Wang, C. Xu, Z. Yang, C. Liu et al., “Data-free learning of student networks,” in *Proc. of ICCV 2019*, Seoul, Korea, pp. 3514-3522, 2019. Article (CrossRef Link)

[17] W. Park, D. Kim, Y. Lu, M. Cho, “Relational knowledge distillation,” in *Proc. of CVPR 2019*, Long Beach, CA, USA, pp. 3967-3976, June 2019. Article (CrossRef Link)

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[19] Z. Yang, Y. Wang, X. Chen, B. Shi, C. Xu et al., “Cars: Continuous evolution for efficient neural architecture search,” in *Proc. of CVPR 2020*, Seattle, WA, USA, pp. 1829-1838, June 2020. Article (CrossRef Link)

[20] X. Gong, S. Chang, Y. Jiang and Z. Wang, “Autogan: Neural architecture search for generative adversarial networks,” in *Proc. of ICCV 2019*, Seoul, Korea, pp. 3224-3234, October 2019. Article (CrossRef Link)

[21] J. Hu, L. Shen and G. Sun, “Squeeze-and-excitation networks,” in *Proc. of CVPR 2019*, Salt Lake City, USA, pp. 7132-7141, June 2018. Article (CrossRef Link)

[22] M, Tan, Q, Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. of ICML 2019*, PMLR, Long Beach, USA, pp. 6105-6114, June 2019. Article (CrossRef Link)
[23] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler et al., “MnasNet: Platform-aware neural architecture search for mobile,” in *Proc. of CVPR 2019*, Long Beach, CA, USA, pp. 2820-2828, June 2019. Article (CrossRef Link)
[24] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashrafet, W. J. Dally and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size,” *arXiv preprint arXiv: 1602.07360*, 2016.
[25] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. of CVPR 2017*, Honolulu, HI, USA, pp. 1251-1258, July 2017. Article (CrossRef Link)
[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. of CVPR 2016*, Las Vegas, NV, USA, pp. 2818-2826, June 2016. Article (CrossRef Link)
[27] B. Wu, A. Wan, X. Yue, P. Jin, S. Zhao et al., “Shift: A zero flop, zero parameter alternative to spatial convolutions,” in *Proc. of CVPR 2018*, Salt Lake City, USA, pp. 9127-9135, June 2018. Article (CrossRef Link)
[28] Z. Yang, Y. Wang, C. Liu, B. Shi, C. Xu et al., “LegoNet: Efficient convolutional neural networks with lego filters,” in *Proc. of ICML 2019*, PMLR, Long Beach, USA, pp. 7005-7014, June 2019. Article (CrossRef Link)
[29] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” in *Proc. of CVPR 2016*, Las Vegas, NV, USA, pp. 770-778, June 2016. Article (CrossRef Link)
[30] J. F. Qiu, Q. H. Wu, G. R Ding, Y. H. Xu and S. Feng, “A survey of machine learning for big data processing,” *EURASIP J Adv Signal Process*, vol. 2016, no. 67, pp.1-16, May, 2016. Article (CrossRef Link)

Liquan Zhao was born in Heilongjiang province in 1982. He received the B.S degree in Electrical & Information Engineering from Harbin University of Science and Technology, Harbin, China, in 2005 and the Ph.D. degree in Communication and Information System at Harbin Engineering University, Harbin, China, in 2009.From 2009, he was an associate professor at Northeast Electric Power University, Jilin, China. His research interests include deep learning and blind source separation.

Leilei Wang was born in Henan province in 1995. She received the bachelor’s degree in college computer and information from the Henan Normal University, Henan, China, in 2018. She is working toward the master’s degree in School of Electrical Engineering, Northeast Electric Power University, Jilin, China. Hers research interests include deep learning and Lightweight neural network.