Multimedia Tools and Applications (2022) 81:38513–38530
https://doi.org/10.1007/s11042-022-13157-8

# Low-res MobileNet: An efficient lightweight network for low-resolution image classification in resource-constrained scenarios

Haiying Yuan[1] · Junpeng Cheng[1] · Yanrui Wu[1] · Zhiyong Zeng[1]

Received: 30 March 2021 / Revised: 22 February 2022 / Accepted: 10 April 2022 /
Published online: 25 April 2022


**Abstract**
The convolutional neural networks (CNNs) deployed on devices for visual image processing faces the thorny problems on high system real-time requirements and resource consumption. A high-performance Low-res MobileNet model is constructed to effectively alleviate the high computing resources and storage costs in the real-time image processing. The main works are summarized as: (1) To actively match the input of low-resolution feature map, the MobileNetV2 is further optimized by clipping to simplify the network structure and improve the efficiency of image recognition. (2) To improve the classification accuracy, the Inception structure is used to fill the Dwise layer in depthwise separable convolution to extract more abundant low-resolution features; the activation function during the process of increasing the dimension is replaced to avoid the loss of useful information; Inter-layer connection structure is adopted to strengthen the fusion of feature information between layers. (3) To reduce the network scale, the gradually decreasing expansion factors are used to remove the redundant structure of the model. Subsequently, the Low-res MobileNet is validated and evaluated through data sets of different scales. The experimental results show that this model has smaller scale, less computation and higher classification accuracy compared with other CNN models. The model has 0.36 M parameters and 25.46 M floating point of operations (FLOPs), which is easy to deploy to resource-constrained mobile and embedded devices. The model runs at 35 batches per second, and it achieves an accuracy rate of 89.38%, 71.60%, and 87.08% on CIFAR-10, CIFAR-100, and CINIC-10 datasets, respectively, which is basically suitable for real-time image classification task applied in low-resolution application scenarios.

**Keywords** Low-resolution features · Image classification · Depthwise separable convolutions · Lightweight network

✉ Haiying Yuan
yhycn@126.com

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, People's Republic of China

## 1 Introduction

Powered by the progress of computer technology and the rapid increase of information, deep learning technology has made great strides. The architecture of convolutional neural network (CNN) has been an area of intense investigation in many domains, including computer vision, natural language processing and big data analysis. Since the milestone work of AlexNet [14], novel structures have emerged including VGG16 [25], GoogLeNet [27], ResNet [8] and so on. The question then becomes how to tackle such a daunting challenge that the models need high computing costs and excessive storage resources. For example, the size of the Caffe model of VGG16 exceeds 500 MB, and 16G floating point of operations (FLOPs) are consumed in the calculation cost [13]. As a result, similar network architectures can't be deployed to mobile and embedded devices with limited resources. In the information age, the application demand of real-time data processing promotes the development of network architecture to be compact and efficient, resulting in model pruning, lightweight module, binary neural network and other optimization technologies. Among them, the lightweight networks such as SqueezeNet [12], MobileNet [9], ShuffleNet [33], ESPNet [20] can reduce the scale of the model while maintaining the accuracy of reasoning. Google has developed an efficient and lightweight CNN network MobileNet V1, which doesn't perform pruning, quantization, decomposition and other operations on large networks, but uses deep separable convolution to construct network structure. The model is practiced in processing image classification, detection, positioning and other applications. Based on the structure of MobileNet V1, the inverse residual structure is proposed, and the MobileNet V2 [23] is generated by optimizing the model. MobileNet V2 has similar classification accuracy with VGG16, GoogLeNet and other networks on ImageNet dataset, but greatly reduces the amount of calculation and parameters.

Resource-constrained scenarios usually involve mobile and embedded devices. Limited by hardware resources and application scenarios, it is laborious for mobile devices or embedded systems to deploy large-scale CNN models to process real-time images. Therefore, two important insights are presented to improve network performance: (1) Devices with limited storage resources are arduous to process a large number of high-quality image data. Additionally, the trained models have hysteresis after it is deployed to the equipment. Starting from the application requirements of low-resolution image classification, low-resolution data are selected for network training and prediction, which greatly reduces the storage cost. Thereby, this allows the mobile terminal to train and update the model online. (2) Due to the lack of computing resources, it is difficult to ensure the real-time performance of visual image processing if the model involves numerous operations. In the low-resolution scene, the computational complexity of the model is reduced, which can further reduce the computational pressure of the mobile and embedded devices.

Based on the above two points, CNN model specialized in low-resolution scene is more suitable for mobile and embedded devices. The resolution size objectively reflects the image quality and the amount of information stored. Although the low-resolution image cannot reflect all details of the original image, it has the advantages of little storage resources and small computational overhead. In some embedded applications with low precision requirements, low-resolution image processing needs are still extensive.

The lightweight network MobileNet V2 developed specifically for mobile terminals uses the depthwise separable convolution structure to reduce computational effort while ensuring classification accuracy. Considering that MobileNet V2 has poor classification effect on low

resolution feature map, and its large input feature map aggravates the storage pressure of hardware, this paper proposes a more effective network architecture based on MobileNet V2.

The performance of state-of-the-art lightweight networks is firstly analyzed in Section 2, and then the application of low-resolution images is briefly described. In Section 3, this paper designs a new network structure called Tailored MobileNet V2 in order to adapt the input of low-resolution images, and four improvements are proposed to optimize the network architecture. Next, numerous experimental analysis and evaluation are given in Section 4, and the results show that Low-res MobileNet reduces computational and memory overhead on mobile and embedded devices and is expert in classifying low-resolution features for resource-constrained scenarios.

## 2 Related work

### 2.1 Lightweight models

Currently, the lightweight network architecture design is mostly guided by depthwise separable convolution, such as MobileNet V1 [9], MobileNet V2 [23], CondenseNet [10], SqueezeNet [12], ShuffleNet V1 [33], ShuffleNet V2 [19] and Xception [4]. SqueezeNet [12] is a lightweight and efficient CNN model proposed by Han et al., whose parameters are 50 times less than that of AlexNet with similar performance. ShuffleNet V1 [33] is an efficient CNN model proposed by Face++, which achieves a balance between the accuracy and the amount of calculation. The main innovation is to break the channels of feature map into new channels in an orderly manner through channel shuffle to solve the problem of poor information flow caused by group convolution. ShuffleNet V2 [19] analyzed the possible cost of non-FLOPs (such as data reading and scheduling) brought by the model in terms of memory consumption cost and GPU parallelism, which is an improvement on V1 with higher classification accuracy and computational efficiency. Compared with previous CNNs, EfficientNet [28] focus on model scaling, and it identifies that carefully balancing network depth, width, and resolution can lead to better performance.

This proposed network architecture is based on MobileNet. MobileNet V1 [9] is a lightweight network proposed by Google that can be deployed on the mobile terminal. It replaces the standard convolution with depthwise separable convolution, which ensures the accuracy and reduces weight parameters.

In the standard convolution, a single convolution kernel corresponds to the whole feature map, which means that the number of channels of the convolution kernels should be consistent with that of the input feature maps. Meanwhile, the number of convolution kernels determines the number of channels of the output feature maps. Different from the standard convolution, deep separable convolution decomposes the convolution operation into two parts: depthwise (Dwise) convolution and pointwise (Pwise) convolution. The number of channels of each convolution kernel in Dwise convolution is 1, which only corresponds to the input channel of the feature maps. This operation greatly reduces the amount of convolution computation and increases the forward propagation speed of mobile devices. Pwise convolution combines the multi-channel output of Dwise convolution, so as to learn the correlation between different channels.

Assuming that the number of channels of input feature maps is $M$, the number of channels of output feature maps is $N$, and the size of feature maps and convolution kernels are $D_F \times D_F$ and $D_k \times D_k$ respectively, the process of standard convolution and depthwise separable convolution is demonstrated in Fig. 1.

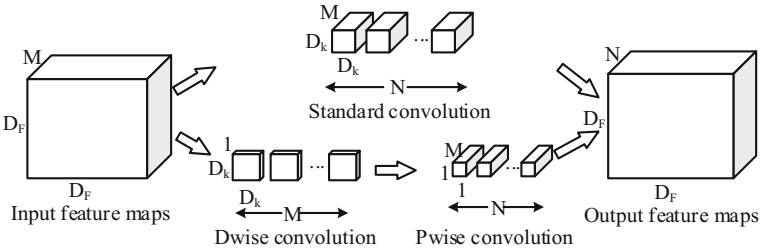


**Fig. 1** Comparison between standard convolution and depthwise separable convolution

Depthwise separable convolution can significantly reduce computational effort. In Eq. (3), the computational difference between standard convolution (denominator) and depthwise separable convolution (molecule) is compared. When 3 × 3 convolution kernel is used, the computational complexity of depthwise separable convolution is only 1/9 that of standard convolution [9].

$$\frac{D_K \times D_k \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_k \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_k^2} \tag{1}$$

On the basis of MobileNet V1, the inversed residuals structure shown in Fig. 2 is added in MobileNet V2 [23]. Pwise convolution is firstly used to expand the number of input channels by 6 times. Then, Dwise convolution is performed to accurately extract image features in a higher spatial dimension. Subsequently, Pwise convolution is used to reduce the number of channels of $C_{out}$ to the input dimension of $C_{in}$. Noteworthy, full connection layer is replaced by the convolution layer in MobileNet V2, which allows the convolution kernel slides over a larger input picture to get the output of each region. This breaks the limit of input size to meet the application requirements of computer vision processing [24].

Besides lightweight models, model compression methods such as low rank decomposition, network pruning, low quantification, and knowledge distillation are an important part of lightweight design [31], which can also reduce network size and computational effort.

## 2.2 Application scenarios of low-resolution images

Visual image processing in mobile and embedded device application shown in Fig. 3 is widely used in many aspects such as unmanned driving [1], delivery robots[18, 29, 30], wearable devices, intelligent monitoring [6, 7, 22], face recognition [2], remote sensing image [3], card swiping [34], medical imaging [15] and image security [16], which has extremely strict requirements on computing speed and storage resources.

Compared with high-resolution images, low-resolution images have smaller size and contain less information. Although some local details of the low-resolution image are lost, the overall information of the image is still complete. In the field of deep learning, small image size and complete image information mean less computational consumption and faster training speed. Meanwhile, in some embedded applications with lower accuracy requirements, such as farmland boundary mapping, low-resolution face recognition for security surveillance scenes

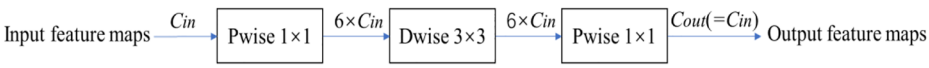


**Fig. 2** Inverted residual structure of MobileNet V2

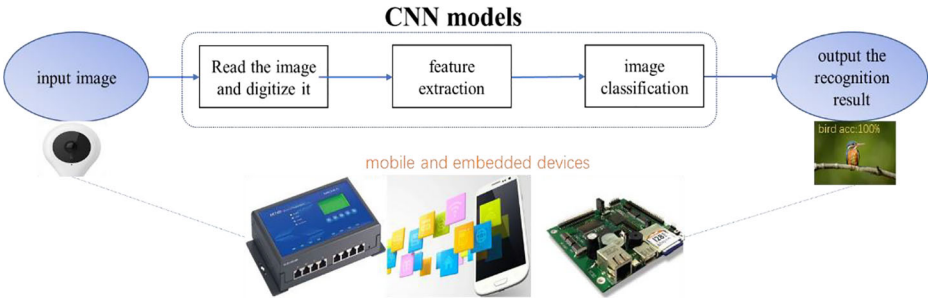


**Fig. 3** Visual image processing flow of mobile and embedded devices

and road traffic flow monitoring and so on, the low-resolution image processing can reduce the pressure of the device memory. In order to deploy the CNN with small-size input feature map to mobile and embedded devices with limited hardware resources, the models should be further optimized to classify low resolution image feature.

## 3 Low-res MobileNet: An efficient architecture

### 3.1 Modify MobileNet V2 for low-resolution features

The low computational complexity and small model size of MobileNet V2 are well adapted to the application needs of real-time data processing, but it performs poorly on low-resolution image feature classification. A 7-layers bottleneck structure is adopted to process 224 × 224 × 3 images in MobileNet V2, whose network structure need be properly tailored when it applied to 32 × 32 × 3 low-resolution images. Considering that only 32 × 32 images need to be down-sampled, the last two layers of bottleneck structure of MobileNet V2 are removed, and the stride of the first convolution layer is changed to 1. The tailored network model is called as Tailored MobileNet V2, and its network structure is described in Table 1.

In Table 1, $t$, $c$, $n$ and $s$ respectively denotes the expansion factor of the bottleneck, the channel number of the feature, the number of the bottleneck repeated, and the stride of first convolution in bottleneck (all subsequent repeat strides are 1). The diagram of Tailored MobileNet V2 is shown in Fig. 4.

**Table 1** The network structure of Tailored MobileNet V2

| Input | Operator | $t$ | $c$ | $n$ | $s$ |
|---|---|---|---|---|---|
| $32^2 \times 3$ | Conv2d 3×3 | – | 32 | 1 | 1 |
| $32^2 \times 32$ | Bottleneck1 | 1 | 16 | 1 | 1 |
| $32^2 \times 16$ | Bottleneck2 | 6 | 24 | 2 | 2 |
| $16^2 \times 24$ | Bottleneck3 | 6 | 32 | 3 | 2 |
| $8^2 \times 32$ | Bottleneck4 | 6 | 64 | 4 | 2 |
| $4^2 \times 64$ | Bottleneck5 | 6 | 96 | 3 | 1 |
| $4^2 \times 96$ | Conv2d 1×1 | – | 160 | 1 | 1 |
| $4^2 \times 160$ | Avgpool 4×4 | – | – | 1 | – |
| $1 \times 1 \times 160$ | Flatten | – | – | – | – |

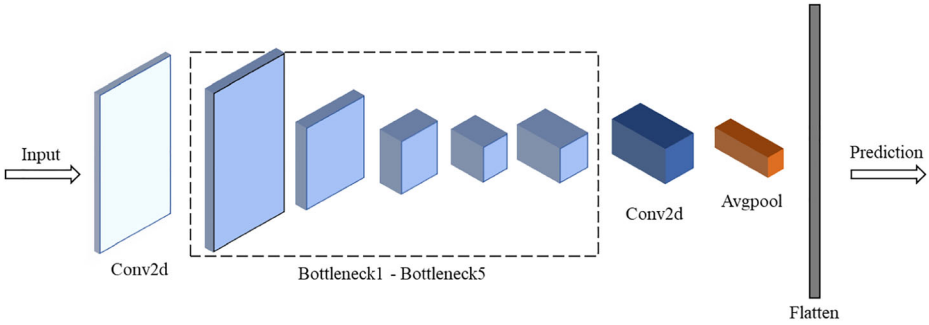


**Fig. 4** The diagram of Tailored MobileNet V2

A convolutional layer instead of a fully connected layer is adopted in MobileNet V2 to break through the limitation of the input feature size. However, in the case of low-resolution feature map input, this measure makes the model cannot fit well. Therefore, full connection operation instead of convolution operation is used in Tailored MobileNet V2 after global average pooling, which solves the model fitting problem well and reduces a large number of parameters [17]. In addition, dropout is used in the network to alleviate over-fitting problem, and dropout rate is determined by the number of training samples. Dropout is placed after BN-layer structures to avoid variance offset in Fig. 5.

### 3.2 Optimization of tailored MobileNet V2

Size matching problem on feature map is effectively solved by Tailored MobileNet V2. Usually, the methods to improve network performance include expanding the network size (increasing the depth or width of the network), upgrading hardware and expanding datasets. However, there are two thorny problems in enlarging the network size: a) When the depth and width continue to increase, the parameters that need to be learned also continue to increase. Too many parameters can easily lead to network overfitting. b) The larger the network size, the greater the amount of calculation. Therefore, the following improvements are successively performed on Tailored MobileNet V2 network structure, then this optimized network is called as Low-res MobileNet.

#### 3.2.1 Filling Dwise layer for information reuse

The block structure of depthwise separable convolution in MobileNet V2 is shown in Fig. 6, where the Dwise part makes the calculation cost much less than that of ordinary convolution operations. When the stride is 1, $1 \times 1$ convolution is firstly performed to send the input feature map into the higher-dimensional spaces. Then, depthwise convolution is performed to extract the features, and Pwise convolution is subsequently performed to reduce the dimensionality. Finally, a residual structure is formed by accumulating the input to the output (shortcut). When the stride is 2, no shortcut is adopted because of the different size between input and output, and the rest is consistent with step 1.

An inverted residual structure is applied to MobileNet V2 to increase the width of the network, and data are processed in a high-dimensional space to minimize information loss. However, the higher the data dimension, the greater the calculation amount of multiplication and addition on the convolutional layer. This inverted residual structure consumes more computation resource.

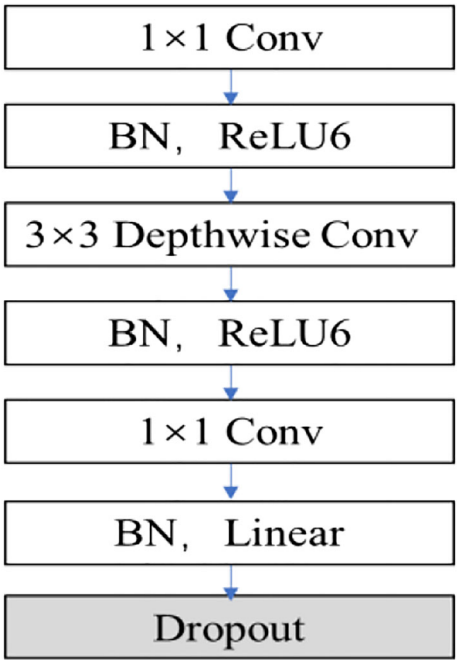


**Fig. 5** The position of Dropout in the inverted residual structure

Inception structure [27] proposed by Google is shown in Fig. 7, which stacks the convolution and pooling operations to increase the width and scale adaptability of the CNN. Inception structure not only improves the network performance but also ensures the computing efficiency without upgrading hardware.

Since network parameters are mainly concentrated on Pwise convolution rather than Dwise convolution, Inception structure is only added to Dwise layer to increase a small number of network parameters in exchange for higher classification accuracy. Therefore, the basic Inception structure is transplanted into the inverted residual structure in Fig. 8, which doubles the width of Dwise layer and improves the feature extraction capability with less computing cost.

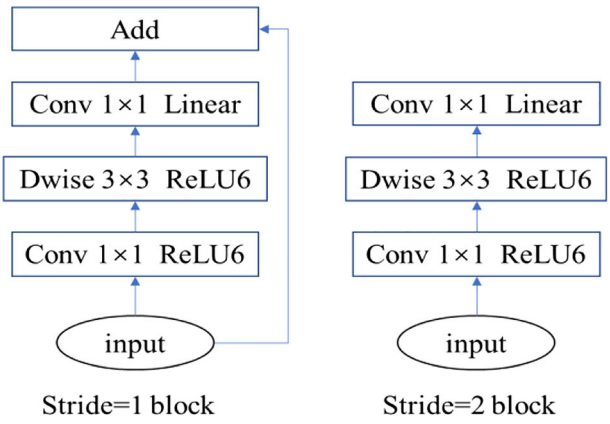


**Fig. 6** Block structures of depthwise separable convolution in MobileNetV2

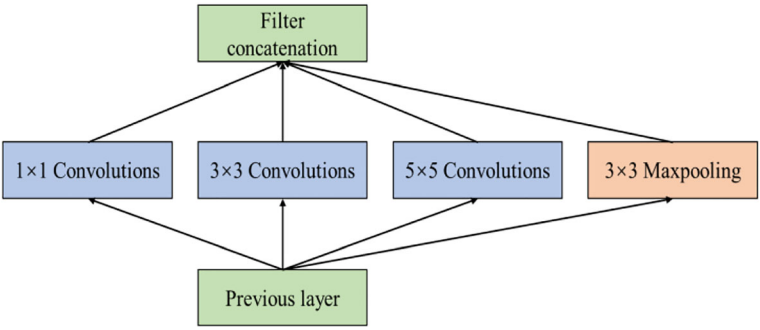


**Fig. 7** The basic Inception structures

As shown in Fig. 9, when the step size is set as 1, the asymmetric convolution kernels such as 1 × 3 and 3 × 1 are instead of the symmetrical convolution kernels 3 × 3 in the model, which reduces network parameters and increases network depth. Considering that low resolution image feature input application scene, to make full use of Dwise layer information, an additional 2 × 2 convolution kernel is added and the convolution results are concatenated together. Convolution kernels with different sizes are correspond to different levels of receptive fields, and they are stitched together to fusion the characteristics information from different scale. Since the application scenario is low-resolution image feature, no additional convolution needed to fill the network balance the width and depth of the network.

When the step size is set as 2, the Dwise convolution results is concatenated with 2 × 2 Max pooling for efficient information extraction. The pooling operation is performed as a supplement of down-sampling to reduce network overfitting.

### 3.2.2 Modifying activation function to avoid information loss

The Dwise layers are filled with Inception structure to broaden the width of the network, and small expansion factors are allowed, which optimizes network performance at a lower cost. However, ReLu6 is used as the activation function for the raise of dimension before Dwise layer in MobileNet V2, a small expansion factor used may cause the activation space to collapse and lose a lot of useful information in low-resolution application scenarios10. In summary, when Inception structure uses a smaller expansion factor, linear activation function replaces ReLU6 during the process of increasing the dimension to ensure information integrity. Linear function and ReLU6 function are shown in Eq. 2 and Eq. 3, and the specific process is shown in Fig. 10.

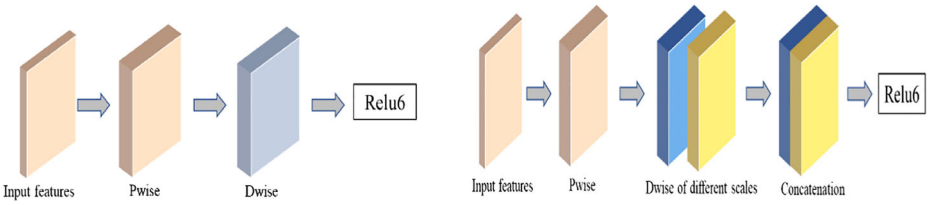


**Fig. 8** The improvement of Dwise layer

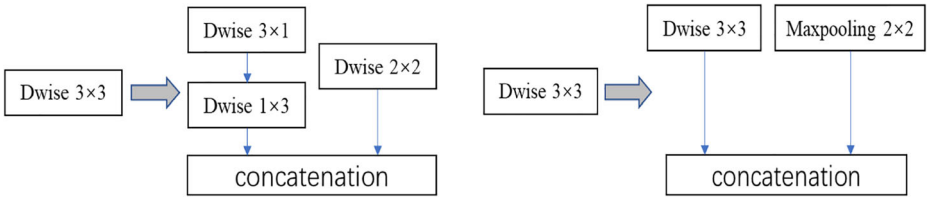


**Fig. 9** Flow of Dwise layer with Inception structure

$$f_{\text{Linear}}(x) = x \tag{2}$$

$$f_{\text{ReLU6}}(x) = \min(6, \max(0, x)) \tag{3}$$

### 3.2.3 Interlayer connection to improve classification accuracy

Any feature information is an important element of low-resolution features, so it should be fully extracted and utilized when CNN model is constructed for network performance improvement. Hereinbefore, the concatenation operation is applied to reuse the features on Dwise layer of depthwise separable convolution, but the network doesn't enhance the reuse of information through interlayer connections. Under the premise of ensuring the maximum information transmission between the layers of the network, DenseNet [11] directly connects all layers to reuse feature to reduce the gradient disappearance and strengthens the transmission of features, which can make better use of the feature information. On this basis, we reuse the feature information of each layer once for sparse connections, which not only realizes feature reuse, but also reduces unnecessary operation overhead.

When the size of feature map is very small after down-sampling, the effect of interlayer connection is not significant, which indicates that interlayer connection should be implemented only when the size of feature map is large. There are five bottleneck structures in Tailored MobileNet V2, where input feature scales of the first four bottlenecks are relatively large, and their interlayer connections is beneficial to feature reuse. In the structure shown in Fig. 11, the features of the first convolution layer are concatenated with the features after bottleneck1. Bottlenenck2 need to be performed twice (bottlenenck2.1 and bottlenenck2.2), and their respective features are concatenated together. Also, Bottlenenck3 and Bottlenenck4 perform the similar operations.

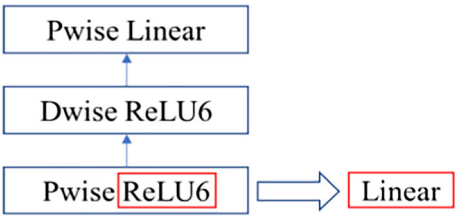


**Fig. 10** Modification of activation function of inverse residual structure

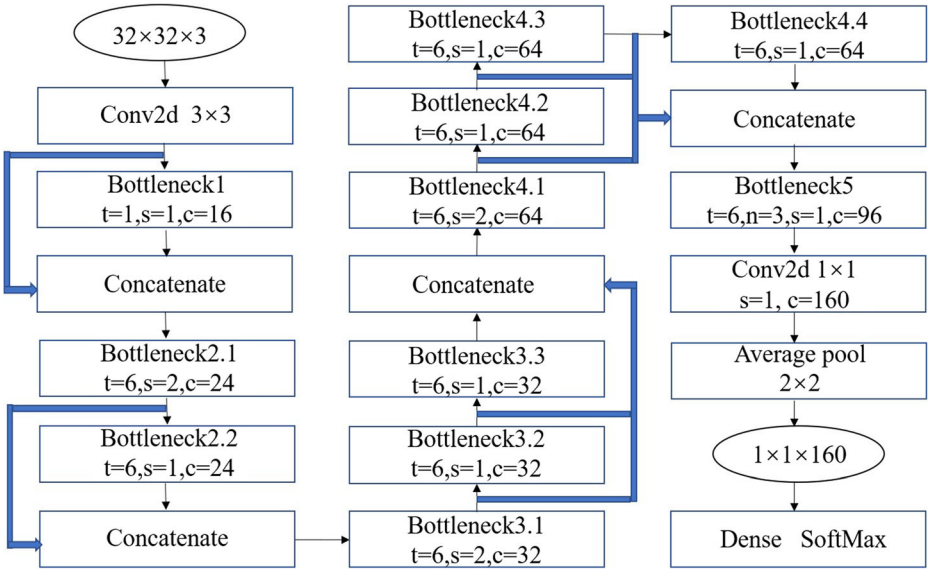


**Fig. 11** Structural diagram of network connection between layers

### 3.2.4 Different expansion factors adopted to remove network redundancy

The above three aspects of model improvement improve the classification accuracy, but also increase the number of network parameters. Excessive parameters can improve the function fitting performance of the training set, but it is not conducive to improving the prediction effect of new data. Hence, the following improvement aims to remove redundant parts of the network to reduce the number of parameters.

According to the network structure analysis of MobileNet V2, the last four bottleneck expansion factors of Tailored MobileNet V2 are set to 6. In terms of parameters, the total number of Tailored MobileNet V2 is 619,546, in which bottleneck 1 has fewer parameters and the parameters of the last four bottlenecks are 17,760, 45,504, 211,456 and 343,872. The proportions of parameters are shown in the Fig. 12.

For low resolution applications scenarios, the feature maps down sampled to 8 × 8 and 4 × 4 in latter four bottlenecks perform the scale-up operation with expansion factors of 6. However, the down-sampled images often have rich semantic information. If the same expansion factor as the higher resolution image is used, it will obviously increase the redundancy of the network so as to bring unnecessary computation. For example, when the expansion factor is 6, the computational cost is twice as much as the cost when the expansion factor is 3. Considering that the Dwise layer is filled with multi-scale characteristic information, smaller expansion factors are allowed to ensure the network performance.

The decreasing expansion factors $t$ is adopted to removes the redundancy of the network model during down sampling. Through this method, the main structure is retained, which can reflect the input image feature information more effectively for higher classification accuracy. With the reduction of the size of the feature maps, the dimensions are increased by using the expansion factors 6, 5, 3 and 1 respectively, as shown in Fig. 13.

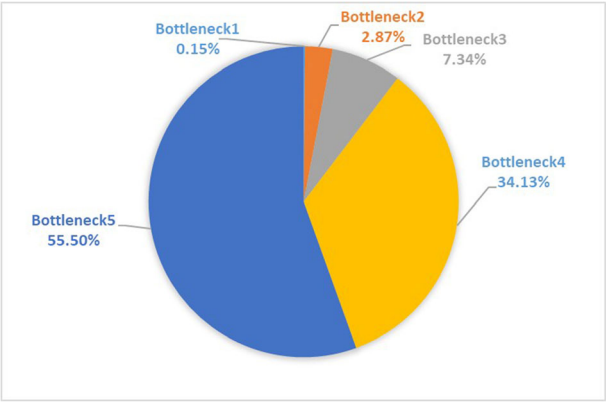


**Fig. 12** The proportion of parameters in each Bottleneck

## 4 Experiments and evaluations

### 4.1 Datasets and training settings

#### 4.1.1 Datasets for evaluation

This paper optimizes the MobileNet V2 model and build a new model called Low-res MobileNet. Low-res MobileNet relies on small datasets, and the application scenario is mainly classification task of low-resolution image. Therefore, CIFAR-10, CIFAR-100 and CINIC-10 are selected as the evaluation datasets, and 32 × 32 low resolution images are involved in experiments.

The CIFAR-10 dataset includes 60,000 32 × 32 RGB images, which are divided into 10 categories, and each category contains 6000 images (5000 for model training and 1000 for model testing). As a small dataset that only occupied about 100 M, CIFAR-10 contains objects and noise information of various scales and sizes in the real world, which is in line with the background of computer vision applications.

The CIFAR-100 dataset includes 60,000 32 × 32 RGB images, which are divided into 100 categories, and each category contains 600 images (500 for model training and 100 for model testing). Compared with CIFAR-10, CIFAR-100 has more image types and fewer samples of each type, which makes it more difficult to classify.

The CINIC-10 [5] dataset combines the images in CIFAR-10 dataset with the down-sampled images in ImageNet dataset to obtain a dataset with 270,000 images of 32 × 32 × 3 size. The dataset scale of CINIC-10 is 4.5 times that of CIFAR-10. CINIC-10 dataset is

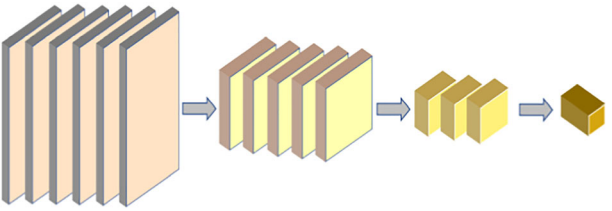

**Fig. 13** The feature maps when using degressive expansion factors

divided into three equal parts, which are used as training set, verification set and test set, respectively.

#### 4.1.2 Training settings

PyTorch is selected as the experimental platform, and the graphics card is GeForce GTX 1060 6G. For CIFAR-10 and CIFAR-100, batch size is set to 64 and epoch is set to 200. For CINIC-10, batch size is set to 64 and epoch is set to 300. The above parameters allow the model to be fully trained. In order to reduce the influence of hyper-parameters, Adam optimizer is used to optimize the loss function while verifying the effect of the four improvement measures. In other experiments, SGD optimizer is used to optimize the loss function. The initial learning rate is set to 0.1 and reduced to 0 in cosine mode. The loss function chooses the SoftMax cross-entropy function [32] shown in Eq. (4), where $y_i$ and $p_i$ represents the distribution of real probability and the prediction probability from network output, respectively, and $K$ represents the number of categories.

$$\mathrm{L} = -\sum_{i=1}^{K} y_i \log(p_i) \tag{4}$$

### 4.2 Experiment results of tailored MobileNet V2 architecture

20% of the training set of CIFAR-10 is used as the verification set and the rest is used for training. Tailored MobileNet V2 and its improvements were tested on this verification set.

#### 4.2.1 Experiment results of tailored MobileNet V2

The test results of the Tailored MobileNet V2 model are shown in Fig. 14. It can be seen that the image recognition accuracy reaches 86.97% after 200 iterations. The network model occurred the over fitting phenomenon, and the loss function rises in the later stage during the validation process, which indicates that there still exists the possibility of improvement for Tailored MobileNet V2.

#### 4.2.2 Experiment results of improving tailored MobileNet V2

Four architecture improvements for Tailored MobileNet V2 are verified in CIFAR-10 verification set respectively. The experimental results are presented in Table 2.

Improvement 1 is to fill Dwise layer. It can be seen from Table 2 that the classification accuracy of the improved model after adding the Inception structure is 0.61% higher than that of Tailored MobileNet V2, which indicates that the filling of Dwise layer can strengthen the feature extraction ability of the model.

Improvement 2 is to modify the activation function. It can be seen from Table 2 that the accuracy of the improved model after adopting Line function is 0.73% higher than that of Tailored MobileNet V2, which indicates that linear function is used during dimension upgrading process can effectively avoid the information loss.

Improvement 3 is to create the connection between layers. It can be seen from Table 2 that the classification accuracy of this improved model is 0.77% higher than that of Tailored

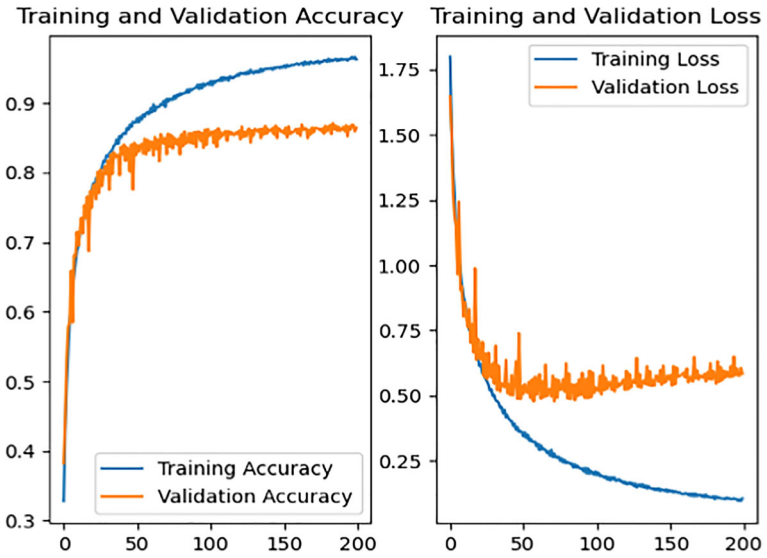


**Fig. 14** The accuracy and loss function of Tailored MobileNet V2

MobileNet V2, which indicates that inter layer feature reuse can effectively improve the performance of the model.

Improvement 4 is to modify the expansion factors. It can be seen from Table 2 that when the expansion factors of 6, 5, 3 and 1 are adopted respectively, the accuracy of the improved model with small expansion factors can still reach to 87.22%, which is equivalent to Tailored MobileNet V2. The number of parameters and calculation of the model are reduced to 0.23 M and 14.34 M respectively, which is only 37.1% and 67.1% of that of Tailored MobileNet V2.

Low-res MobileNet is the model after four architecture improvements, and its classification accuracy reaches to 88.81%, which is 1.84% higher than that of Tailored MobileNet V2. The parameters of the Low-res MobileNet model are 0.36 M, accounting for only 58.1% of the Tailored MobileNet V2 parameters. Due to the feature reuse, the computational complexity of Low-res MobileNet model is slightly higher than that of Tailored MobileNet V2, with an increase of 4.08 M. The experimental results of Low-res MobileNet are shown in Fig. 15. It can be seen that the over-fitting phenomenon of the optimized model has been greatly alleviated.

**Table 2** The results of improvements of Tailored MobileNet V2

| Model | Parameter (M) | FLOPs (M) | Accuracy (%) |
|---|---|---|---|
| Base (Tailored MobileNet V2) | 0.62 | 21.38 | 86.97 |
| Improvement1 | 0.86 | 28.30 | 87.58 |
| Improvement2 | 0.62 | 21.38 | 87.70 |
| Improvement3 | 0.76 | 30.62 | 87.74 |
| Improvement4 | 0.23 | 14.34 | 87.22 |
| All improvements (Low-res MobileNet) | 0.36 | 25.46 | 88.81 |

### 4.3 Network performance test performed on CIFAR-10

In CIFAR-10 dataset, validation set is used for model validation and preliminary evaluation (as shown in Fig. 15), and test set is used for model testing and performance evaluation (as shown in Fig. 16). Low-res MobileNet model achieves 89.38% classification accuracy on test set (slightly higher than that on validation set), which indicates that the model has excellent network generalization.

The classification accuracy, training parameters, FLOPs, total memory (read + write) and the speed of forward propagation are compared in Table 3. To adequately analyze the network performances, each model is also conducted on data enhancement technology.

The networks involving in performance comparison in Table 3 include large CNN networks (VGG-16, Inception V1/V3, and Xception) and typical lightweight CNN networks (SqueezeNet, ShuffleNet V1/V2 and MobileNet V1/V2).

In terms of classification performance in Table 3, Low-res MobileNet without data enhancement is slightly lower than large networks such as VGG-16 but higher than the ShuffleNet V1/V2 and MobileNet V1/V2. Also, the accuracy of each model with data enhancement has been improved, and the accuracy of Low-res MobileNet has reached to 93.32%. For FLOPs, Low-res MobileNet is much lower than the VGG-16, Inception V1, Inception V3 and Xception, and lower than SqueezeNet, ShuffleNet V1/V2 and MobileNet V1/V2. A small number of FLOPs meets the computation requirements of mobile terminal devices with limited computing resources. In terms of the number of parameters, Low-res MobileNet is much smaller than the other models. It is worth noted that a large number of modules stacked on Inception and Xception models makes the network memory insufficient. In the case of limited storage resources, this is not conducive to deploying the network model to resource-constrained hardware. As far as the read/write memory is concerned, Low-res MobileNet needs less space than other lightweight networks do. In terms of running speed, Low-res MobileNet is prior to other networks.

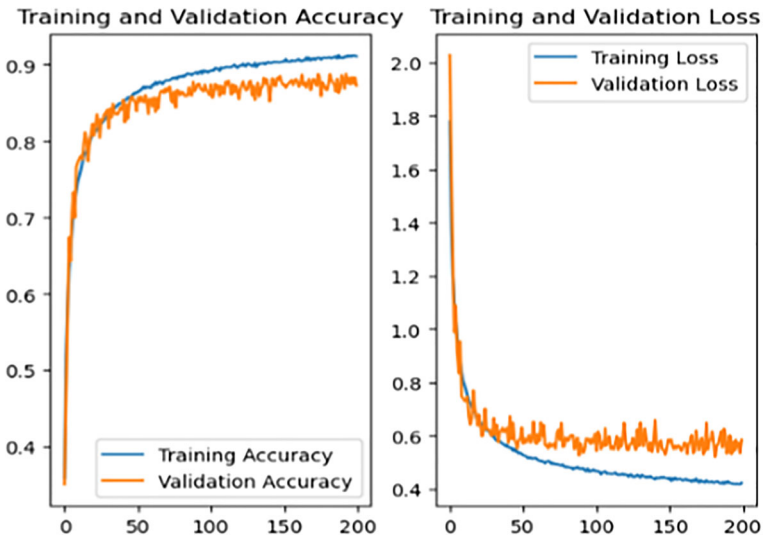


**Fig. 15** The results of Low-res MobileNet running on the CIFAR-10 verification set

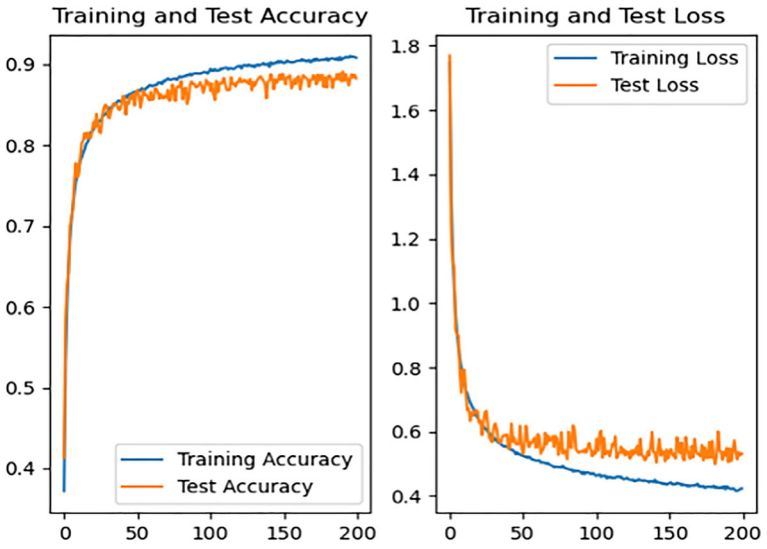


**Fig. 16** The results of Low-res MobileNet running on the CIFAR-10 test set

To sum up, Low-res MobileNet only consumes 25.46 M FLOPs with 0.36 M parameters, and the classification accuracy meets the application requirements of real-time image processing.

### 4.4 Network performance test performed on CINIC-10 and CIFAR-100

To demonstrate the generalization of Low-res MobileNet, the following tests are further tested on CINIC-10 [5] and CIFAR-100 datasets. The experimental results are shown in Table 4.

Low-res MobileNet is respectively tested in CINIC-10 dataset using two methods: (1) Train set for network training & test set for network testing. (2) Combine training set and validation

**Table 3** Comparison of training results based on CIFAR-10 dataset

| Model | Classification accuracy (%) | | Parameter (M) | FLOPs (M) | Memory R+W(MB) | Batches/s. |
|---|---|---|---|---|---|---|
| | Without data enhancement | Data enhancement | | | | |
| VGG-16 | 90.62 | 94.62 | 14.7 | 314.43 | 62.77 | 32 |
| Inception V1 | 91.99 | 96.13 | 6.2 | 1533.47 | 104.22 | 6 |
| Inception V3 | 92.03 | 97.24 | 22.3 | 3403.58 | 199.78 | 3 |
| Xception | 89.67 | 94.81 | 21.0 | 1135.92 | 136.33 | 7 |
| SqueezeNet | 84.39 | 90.52 | 0.7 | 54.1 | 11.54 | 35 |
| ShuffleNet V1(g=3) | 84.35 | 89.94 | 0.9 | 40.46 | 17.18 | 31 |
| ShuffleNet V2(2×) | 86.95 | 92.02 | 5.3 | 183.86 | 37.57 | 21 |
| MobileNet V1 | 84.04 | 89.61 | 3.2 | 47.18 | 18.55 | 32 |
| MobileNet V2 | 86.27 | 91.83 | 2.3 | 94.60 | 35.03 | 21 |
| Low-res MobileNet | 89.38 | 93.32 | 0.36 | 25.46 | 12.46 | 35 |

**Table 4** Comparison of training results based on CINIC-10 and CIFAR-100

| Model | Classification accuracy (%) | | | Parameter (M) | FLOPs(M) | Memory R+W(MB) | Batches/s. |
|---|---|---|---|---|---|---|---|
| | CINIC-10 [5] | | CIFAR-100 | | | | |
| | Method 1 | Method 2 | | | | | |
| VGG-16 | 84.75 | 87.77± 0.16 | 72.93 | 14.7 | 314.43 | 62.77 | 32 |
| Inception V1 | 88.46 | 91.17±0.12 | 77.16 | 6.2 | 1533.47 | 104.22 | 6 |
| Inception V3 | 88.68 | 91.70±0.14 | 77.19 | 22.3 | 3403.58 | 199.78 | 3 |
| Xception | 85.87 | 87.21±0.08 | 75.03 | 21.0 | 1135.92 | 136.33 | 7 |
| SqueezeNet | 81.92 | 84.64±0.11 | 69.44 | 0.7 | 54.1 | 11.54 | 35 |
| ShuffleNet V1(g=3) | 80.92 | 84.89±0.13 | 70.12 | 0.9 | 40.46 | 17.18 | 31 |
| ShuffleNet V2(2×) | 83.56 | 86.53±0.17 | 71.82 | 5.3 | 183.86 | 37.57 | 21 |
| MobileNet V1 | 80.45 | 82.00± 0.16 | 66.04 | 3.2 | 47.18 | 18.55 | 32 |
| MobileNet V2 | 83.04 | 86.27±0.11 | 68.11 | 2.3 | 94.60 | 35.03 | 21 |
| Low-res MobileNet | 84.77 | 87.08±0.14 | 71.60 | 0.36 | 25.46 | 12.46 | 35 |

set for network training & test set for network testing. In the two methods, the training samples of CINIC-10 are 1.8 times and 3.6 times of CIFAR-10 respectively, which can test the model performance under different training samples [21, 26, 32]. The training samples has been trained five times independently in Method 2, so both mean and standard deviation are listed to reflect the classification accuracy.

CIFAR-100 has more types and the sample size of each type is small, which makes the network training difficult. The performance of Low-res MobileNet under a small number of samples can be observed by testing on this dataset.

As shown in Table 4, in CINIC-10 dataset, the accuracy of Low-res MobileNet is still higher than that of SqueezeNet, ShuffleNet V1/V2 and MobileNet V1/V2, and is roughly equivalent to VGG-16 and Xception. Due to the increased difficulty of training, the performance of Low-res MobileNet in CIFAR-100 dataset is slightly worse than that in CINIC-10 dataset, but it is still better than most of the lightweight networks, and is equivalent to ShuffleNet V2. Remarkably, the parameters and FLOPs of Low-res MobileNet are less than those of other networks, and the network has advantages over other networks in terms of memory and running speed.

## 5 Conclusion

Limited by hardware resources and application scenarios, it is laborious to deploy large-scale CNN models on mobile and embedded devices for real-time images process. As a lightweight network specially developed for mobile applications, MobileNet V2 performs well in network performance. However, it is not adaptive to low resolution image analysis. Hence, a series of network improvements are implements in MobileNet V2 to construct Low-res MobileNet model. Inception structure and interlayer connection are adopted to increasing the classification accuracy, while the expansion factors are adjusted to reduce computational effort. Different datasets are used to verify the model and evaluate the performance. Numerous

experiment results reflect that the Low-res MobileNet network has outstanding generalization and high classification accuracy, which is competent in real-time image classification task in resource-constrained scenarios. Compared with the other lightweight network, Low-res MobileNet outperforms in less storage resource about 2 MB memory and lower computation load about 25.46 M FLOPs, which allows the lightweight model to be easily deployed to mobile and embedded devices with limited hardware resources.

**Acknowledgments** This research work was supported by National Natural Science Foundation of China (61001049) and Beijing Natural Science Foundation (4172010).

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

## References

1. Bai L, Lyu Y, Huang X (2021) RoadNet-RT: High Throughput CNN Architecture and SoC Design for Real-Time Road Segmentation. In IEEE Transactions on Circuits and Systems I: Regular Papers (vol. 68, no. 2, pp. 704–714) https://doi.org/10.1109/TCSI.2020.3038139
2. Cheng G, Zhou PC, Han JW (2018) Duplex metric learning for image set classification. IEEE Trans Image Process 27(1):281–292
3. Cheng G, Yang CY, Yao XW, Guo L, Han JW (2018) When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs. IEEE Trans Geosci Remote Sens 56(5):2811–2821
4. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251–1258). https://doi.org/10.1109/cvpr.2017.195
5. Darlow L N, Crowley E J, Antoniou A, Storkey A (2018) CINIC-10 is not ImageNet or CIFAR-10. arXiv preprint arXiv:1810.03505
6. Gu K, Xia ZF, Qiao JF, Lin WS (2020) Deep Dual-Channel neural network for image-based smoke detection. IEEE Transactions on Multimedia 22(2):311–323
7. Gu K, Liu HY, Xia ZF, Qiao JF, Lin WS, Thalmann D (2021) PM2.5 monitoring: use information abundance measurement and wide and deep learning. IEEE Transactions on Neural Networks and Learning Systems 32(10):4278–4290
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90
9. Howard AG, Zhu M, Chen B, Kalenichenko D (2019) MobileNets: efficient convolutional neural networks for Mobile vision applications. Appl Intell 50(1):107–118
10. Huang G, Liu S, Laurens van der Maaten (2017) CondenseNet: An Efficient DenseNet using Learned Group Convolutions arXiv preprint arXiv: 1711.09224
11. Huang G, Liu Z, Laurens V D M, et al (2017) Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700–4708). https://doi.org/10.1109/cvpr.2017.243
12. Iandola F N, Han S, Moskewicz M W, Ashraf K, Dally W J, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv preprint arXiv:1602.07360

13. Jia Y, Shelhamer E, Donahue J, et al (2014) Caffe: convolutional architecture for fast feature embedding. In ACM Conf Multimedia (pp. 675–678). https://doi.org/10.1145/2647868.2654889
14. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25(2):1106–1114
15. Liao X, Li KD, Zhu XS, Liu KJR (2020) Robust detection of image operator chain with two-stream convolutional neural network. IEEE Journal of Selected Topics in Signal Processing 14(5):955–968
16. Liao X, Yu YB, Li B, Li ZP, Qin Z (2020) A new payload partition strategy in color image steganography. IEEE Transactions on Circuits and Systems for Video Technology 30(3):685–696
17. Lin M, Chen Q, Yan S (2014) Network in network. In Int. Conf. Learning Representations (pp:1–10)
18. Lobov SA, Mikhaylov AN, Shamshin M, Makarov VA, Kazantsev VB (2020) Spatial properties of STDP in a self-learning spiking neural network enable controlling a Mobile robot. Front Neurosci 14:88–98
19. Ma M N, Zhang X Y, Zheng H T, Sun J (2018) Shufflenet V2: practical guidelines for efficient CNN architecture design. In European Conf Comput Vision (pp:122–138). https://doi.org/10.1007/978-3-030-01264-9_8
20. Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) ESPNet: efficient spatial pyramid of dilated convolutions for semantic segmentation. In European Conf Comput Vision (pp. 561–580). https://doi.org/10.1007/978-3-030-01249-6_34
21. Roccetti M, Delnevo G, Casini L, Mirri S (2021) An alternative approach to dimension reduction for pareto distributed data: a case study. Journal of Big Data 8:39–62
22. Sakib S, Fouda MM, Fadlullah ZM, Nasser N, Alasmary W (2021) A proof-of-concept of ultra-edge smart IoT sensor: a continuous and lightweight arrhythmia monitoring approach. IEEE Access 9:26093–26106. https://doi.org/10.1109/ACCESS.2021.3056509
23. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) MobileNetV2: inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510–4520). https://doi.org/10.1109/CVPR.2018.00474
24. Shelhamer E, Long J, Darrell T (2015) Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431–3440). https://doi.org/10.1109/cvpr.2015.7298965
25. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Computer Science 9:55–56
26. Sun YM, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. Int J Pattern Recognit Artif Intell 23(4):687–719
27. Szegedy C, Liu W, Jia Y, et al, (2015) Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1–9). https://doi.org/10.1109/CVPR.2015.7298594
28. Tan M, Le Q (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In proceedings of the 36th international conference on machine learning 97, 6105-6114
29. Yang SM, Wang J, Deng B, Liu C, Li HY, Fietkiewicz C, Loparo KA (2019) Real-time neuromorphic system for large-scale conductance-based spiking neural networks. Ieee Transactions on Cybernetics 49(7):2490–2503
30. Yang SM, Deng B, Wang J, Li HY, Lu ML, Che YQ, Wei XL, Loparo KA (2020) Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. Ieee Transactions on Neural Networks and Learning Systems 31(1):148–162
31. Yang SM, Gao T, Wang J, Deng B, Lansdell B, Linares-Barranco B (2021) Efficient spike-driven learning with dendritic event-based processing. Front Neurosci 15:601109. https://doi.org/10.3389/fnins.2021.601109
32. Zhang ZL, Sabuncu MR (2018) Generalized cross entropy loss for training deep neural networks with Noisy labels. Neural Information Processing Systems 31:1–11
33. Zhang X, Zhou X, Lin M, Sun J (2018) ShuffleNet: an extremely efficient convolutional neural network for Mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6848–6856). https://doi.org/10.1109/CVPR.2018.00716
34. Zhou N, Liang R, Shi W (2021) A lightweight convolutional neural network for real-time facial expression detection. IEEE Access 9:5573–5584. https://doi.org/10.1109/ACCESS.2020.3046715