Mathematical Biosciences and Engineering

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

MBE, 19(12): 12232–12246.
DOI: 10.3934/mbe.2022569
Received: 11 June 2022
Revised: 08 August 2022
Accepted: 16 August 2022
Published: 22 August 2022

Research article

CED-Net: A more effective DenseNet model with channel enhancement

Xiangqun Li[1,2], Hu Chen[1], Dong Zheng[1] and Xinzheng Xu[1,2,*]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
[2] Key Laboratory of Opto-technology and Intelligent Control, Ministry of Education, Lanzhou Jiaotong University, Lanzhou 730070, China

* **Correspondence:** Email: xxzheng@cumt.edu.cn; Tel: +8615952151616; Fax: 051683591726.

Abstract: In recent years, deep convolutional neural network (CNN) has been applied more and more increasingly used in computer vision, natural language processing and other fields. At the same time, low-power platforms have more and more significant requirements for the size of the network. This paper proposed CED-Net (Channel enhancement DenseNet), a more efficient densely connected network. It combined the bottleneck layer with learned group convolution and channel enhancement module. The bottleneck layer with learned group convolution could effectively increase the network's accuracy without too many extra parameters and computation (FLOPs, Floating Point Operations). The channel enhancement module improved the representation of the network by increasing the interdependency between convolutional feature channels. CED-Net is designed regarding CondenseNet's structure, and our experiments show that the CED-Net is more effective than CondenseNet and other advanced lightweight CNNs. Accuracy on the CIFAR-10 dataset and CIFAR-100 dataset is 0.4 and 1% higher than that on CondenseNet, respectively, but they have almost the same number of parameters and FLOPs. Finally, the ablation experiment proves the effectiveness of the bottleneck layer used in CED-Net.

Keywords: convolutional neural network; channel enhancement; Squeeze and Excitation; CondenseNet; bottleneck layer

1. Introduction

With the advent of the era of big data, deep learning technology has become a research hotspot in the field of artificial intelligence. It has shown great advantages in image recognition, speech recognition, natural language processing and other fields. The problem of sequence labeling is the most common problem in natural language. Shao et al. [1] assign semantic labels in input sequences, exploiting encoding patterns in the form of latent variables in conditional random fields to capture latent structure in observed data. Lin et al. [2] proposed an attentional segmentation recurrent neural network (ASRNN), which relies on a hierarchical attentional neural semi-Markov conditional random field (semi-CRF) model for sequence labeling tasks.

Convolutional neural networks (CNN) have been widely used in computer vision recognition tasks. Djenouri et al. [3] proposed a technique for particle clustering for object detection (CPOD), built on top of region-based methods, using outlier detection, clustering, particle swarm optimization (PSO), and deep convolutional networks to identify smart object data. Shao et al. [4] proposed an end-to-end multi-objective neuroevolution algorithm based on decomposition and dominance (MONEADD) for combinatorial optimization problems to improve the performance of the model in inference. From 2010 to 2017, the ImageNet Large Scale Visual Recognition Challenge has been held for seven years. The image classification accuracy of the champions has increased from 71.8% to 97.3%. The emergence of AlexNet in 2012 was a milestone in deep learning field. After that, the ImageNet dataset accuracy has been significantly improved by novel CNNs, like VGG [5], GoogleNet [6], ResNet [7,8], DenseNet [9], SE-Net [10], and automatic neutral architecture search [11–13].

However, it is necessary to consider high accuracy, platform resources, and the efficiency of systems in real-world applications, e.g., automatic drive systems, intelligent robot systems, and mobile device applications. Moreover, most of the best-performing CNNs need to run on a high-performance graphics processing unit (GPU). So, real-world tasks have driven the development of more lightweight CNNs, to allow CNN to be used in more low-performance devices [14,15], like Xception [16], MobileNet [17], MobileNet V2 [18,19], ShuffleNet [20], ShuffleNet V2 [21] and CondenseNet [22]. Group convolution and depth-wise separable convolution [23] are crucial in these works.

As the best paper at the CVPR 2017 conference, DenseNet beat the best performing ResNet on ImageNet without group convolution or depth-wise separable convolution. Subsequently, the SE-Net achieved the best results in the history of ImageNet in ILSVRC2017, but there are still too many parameters in SE-Net. Following these works, Huang et al. [9] have proposed Learned Group Convolutions to improve DenseNet connection and convolution methods. Inspired by these jobs, we study using Squeeze-and-Excitation block (SE-block) to improve the lightweight CNN. Furthermore, we explore how to design the structure of the convolutional layer to enhance the network's performance.

We propose a more efficient network, CED-Net, which combines bottleneck layer with learned group convolution and SE block. Learned group convolution can crop the network channel during the training phase. And the SE block can recalibrate the feature channel to enhance the channel beneficial to the network. Through experiments, we demonstrate that CED-Net is superior to other lightweight network in terms of accuracy, the number of parameters, and FLOPs.

2. Related works

2.1. Model compression and efficient network architectures

In the past few years, designing CNNs by adjusting an optimal depth to balance accuracy and performance was a very active field. Most recent work has been many progresses in algorithm optimization exploration, including pruning redundant connections [24–27], using low-accuracy or quantized weights [28,29], or designing efficient network architectures.

Early researchers proved pruning redundant and quantization are effective methods because deep neural networks often have a substantial number of redundant weights that can be pruned or quantized without sacrificing (and sometimes even improving) accuracy. For CNNs, different pruning techniques may lead to varying levels of granularity [30]. Fine-grained pruning, e.g., independent weight pruning [31], generally achieves a high degree of sparsity. Coarse grained pruning methods such as filter-level pruning earn a lower degree of sparsity, but the resulting networks are much more regular, facilitating efficient implementations.

Recently researchers have explored the structures of the efficient network that can be applied on mobile devices such as MobileNet V2, ShuffleNet V2, and NasNet. In these networks, depth-wise separable convolutions play a vital role, which can reduce a large number of network parameters without significantly reducing the accuracy. However, according to the Howard et al. [17,18], a large amount of depth-wise separable convolutions will decrease the computational speed of the network. Therefore, CED-Net uses a more efficient group convolution and densely connected architecture to reduce the number of parameters of the network. Furthermore, because many deep-learning libraries efficiently implement group convolutions, they save a lot of computational time in theory and practice.

In addition, the bottleneck layer proposed in ResNet can effectively reduce parameters for multilayer network. Our experiments show that CED-Net can achieve higher accuracy and fewer parameters than CondenseNet of the same structure when layers are deeper.

2.2. Learned group convolution and squeeze-and-excitation

Huang et al. [9], as the best paper for CVPR2017, proposed a densely connection network that is better than the previous champion ResNet on the ImageNet. After that, CondenseNet achieved the same accuracy with only half of the number of parameters of DenseNet. In CondenseNet, learned group convolution plays a key role; it can train the network with sparsity inducing regularization for a fixed number of iterations. Subsequently, it prunes away unimportant filters with low magnitude weights. Because many deep-learning libraries efficiently implement group convolutions, they save a lot of computational time in theory and practice.

Moreover, the Squeeze-and-Excitation structure that shines on ILSVRC2017 has been experimented on by most famous networks. Squeeze and Excitation are two very critical operations. First, it is used to model the interdependencies between feature channels explicitly. It is a new “channel recalibration” strategy. Specifically, by automatically learning the importance of each feature channel, SE-Net enhances the proper channel and suppresses useless channels. Most of the current mainstream networks are constructed based on superimposed basic blocks. It can be seen that the SE module can be embedded in almost all network structures, so CED-Net achieves more efficient performance by embedding the SE module.

3. Channel enhancement dense network

In this section, we first introduce the structure and function of the bottleneck layer. Next, we explore how SE Block as a channel enhancement block can improve the performance of CED-Net. Finally, we describe the network details of CED-Net for CIFAR dataset.

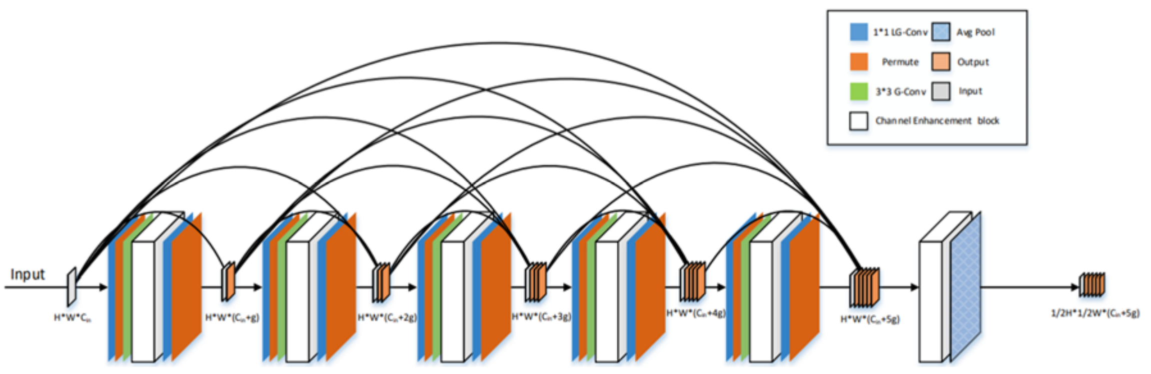

Figure 1. A 5-layer dense block with channel enhancement and bottleneck layer.

As shown in Figure 1, H, W, C_{in} are the height, width, and the number of channels of the input image, respectively, and g is the growth coefficient of the channel. CED-Net consists of multiple dense blocks for feature extraction. The dense block is shown in Figure 2(c). It consists of two 1 × 1 LG-Conv (Learned Group Convolution) layers and one 3 × 3 G-Conv (Group Convolution) layer. Each 1 × 1 LG-Conv layer uses a permute operation for channel shuffling to reduce accuracy. BN-ReLU nonlinearly activates the input and output in the dense block. And use the AvgPool layer for down sampling.

3.1. Bottleneck layer with learned group convolution

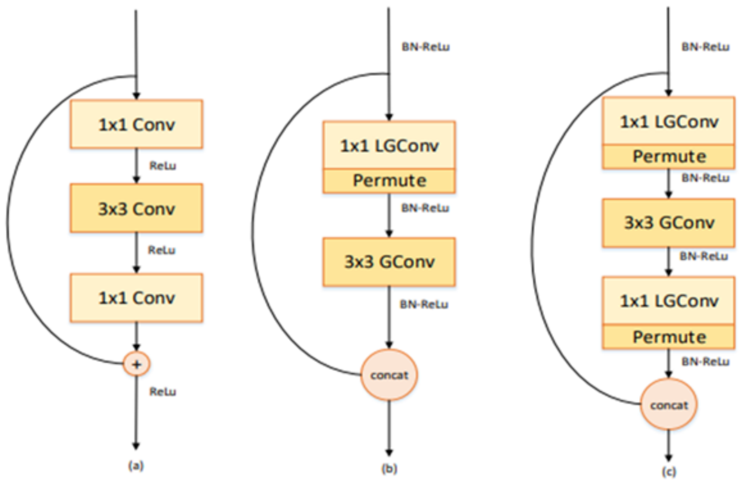

Figure 2. Different networks' bottleneck layer or dense block. (a) ResNet. (b) CondenseNet. (c) CED-Net.

The bottleneck layer is proposed in ResNet, and the detailed structure is shown in Figure 2(a). The three-layer bottleneck structure consists of 1 × 1, 3 × 3, and 1 × 1 convolutional layers, where two 1 × 1 convolutions are used to reduce and increase (restore) dimensions. The 3 × 3 convolutional layer can be seen as a bottleneck for a smaller input/output dimension. We replace the 1 × 1 standard convolution with the learned group convolution, and the 3 × 3 standard convolution is replaced with the group convolution. Unlike ResNet, the CED-Net replaces element-wise addition with channel concatenation. Because it can use the semantic information of different scale feature maps to achieve better performance by increasing the channel, the element addition operation does not take up too much memory during network transmission. Still, it may introduce extra noise that will lose some feature map information.

Figure 2(b) shows the structure used in CondenseNet. The Permute layer, enabling shuffling between channels, is designed to reduce the adverse effects of the introduction of 1 × 1 LG-Conv. But there are still many parameters in a deep network with the bottleneck layer. Figure 2(c) shows part of the structure used by CED-Net. This structure has fewer parameters than that in Figure 2(b). Expressly, the condense factor and bottleneck factor in CED-Net are set to 4 and reduced by half compared to CondenseNet. This is to reduce the parameters caused by adding a 1 × 1 LG-Conv layer.

One dense layer used in CED-Net is of quadratic time complexity ($\Theta(25G^2/4+4CG)$) concerning the number (C) of input channels and the number (G) of output channels. Compared with ordinary 3 × 3 convolution ($\Theta(9CG)$), as a result of C is much greater than G with the deepening of network layers, CED-Net reduces the time complexity by half.

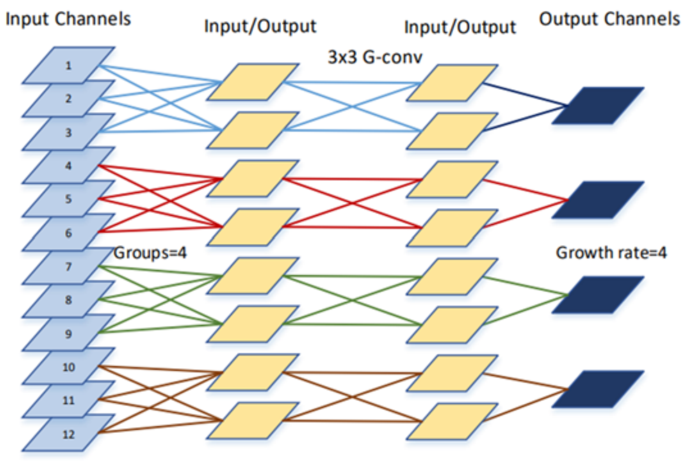

Figure 3. Bottleneck layer with Learned Group Convolutions.

Figure 3 shows how channels change the process of the bottleneck layer based on learning group convolution. The parameters and calculation amount are 1/4 of the standard bottleneck layer. Based on the image classification comparing experiments on the CIFAR dataset, we can conclude that our structure can increase the classification accuracy by 0.4% when the number of parameters and the amount of calculation is almost the same as CondenseNet (see Section 4). When network layers are deeper (depth is 272), the number of parameters and the amount of calculation of CED-Net are smaller than the CondenseNet of the same depth. Still, the classification accuracy is higher than that of CondenseNet.

3.2. Channel enhancement with Squeeze and Excitation

In CED-Net, since the network is a densely connected structure, the input data of each convolution layer has a large amount of channel information. And the output after convolution is the sum of all previous channel information. This has led to the entanglement of information and spatial relevance. Furthermore, in lightweight networks, group convolution can significantly reduce the amount of computation by ensuring that each convolution operation is only on the corresponding input channel group. However, if multiple sets of convolutions are stacked together, there is a side effect: A channel output is only derived from a few numbers of input channels. This would reduce the information flow between channel groups and express information.

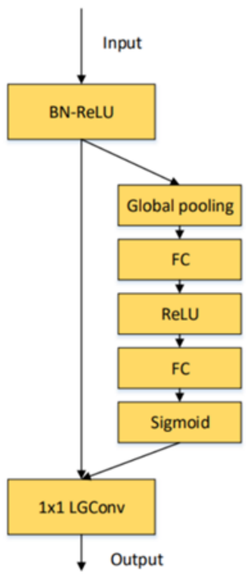

Figure 4. Channel Enhancement with Squeeze-and-Excitation.

Therefore, we use the channel permute (see Figure 2(c)) and the Squeeze-and-Excitation block to make the information between the groups more circulated to allow the network to focus on more helpful information. As shown in Figure 4, Squeeze-and-Excitation blocks can improve the representation of the network by increasing the interdependence between convolution feature channels. The detailed process is divided into two steps: Squeeze and Excitation.

Squeeze. CNNs all have the problem that due to the nature of convolutional calculations, each convolution filter can only focus on specific spatial information. To alleviate this problem, the Squeeze, as a global description operation, encodes the global spatial information into the channel descriptor and calculates the mean of each channel through global average pooling.

$$z_c = F_{sq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i,j) \tag{1}$$

As shown in Eq (1), where Z_c is the output of the squeeze layer, W, H are the width and height of the input feature map of the current layer. u_c is the input feature map, and $F_{sq}(*)$ can represent the global information of the entire feature map. The global average pooling used in this paper squeezes the feature map into a value to indicate the importance of the corresponding channel.

Excitation. To take advantage of the information obtained by the squeeze operation, the excitation operation needs to meet two criteria to achieve full capture of channel dependencies. First, it must be able to learn nonlinear interactions between channels. And second, it must learn a non-mutually exclusive relationship. Specifically, the gate mechanism is parameterized by concatenating two fully connected (FC) layers above and below the nonlinear (ReLU) and then activated with the sigmoid function.

$$s = F_{ex}(z, W) = \sigma\big(g(z, W)\big) = \sigma\big(W_2 \theta(W_1 z)\big) \tag{2}$$

where θ is the ReLU function.$W_1 \in R^{\frac{C}{r}\times C}$, $W_2 \in R^{C\times\frac{C}{r}}$ are the weights of the dimensionality reduction layer and the dimensionality increase layer, respectively. Where r is the dimensionality reduction rate, and C is the number of channels. To limit the complexity of the model and increase the generalization, a "bottleneck" is formed by a two-layer FC layer around a nonlinear map, where r sets 16. Finally, after obtaining the so-called gate, by multiplying the channel gates by the corresponding feature maps, you can control the flow of information for each feature map.

We embed the Squeeze-and-Excitation block into the 3 × 3 G-Conv layer because the number of input/output feature channels in the first 3 × 3 G-Conv is the same and smaller. The Squeeze-and-Excitation block can effectively enhance the effective channel after feature extraction without extra parameters. According to the research results of Hu et al., this method can balance the accuracy of the model and the number of parameters.

Algorithm 1 Image classification based on CED-Net

```
Input: In = datasets $(x_1, y_1)$, $(x_2, y_2)$, …, $(x_m, y_m)$
Output: Op = Classification accuracy: $(y_1, y_2, \dots, y_n)$
    Set: CED-Net feature extraction: $G_k(\cdot)$, $k \in (0, n)$
    for x = 1 : m do
        $Softmax(G_k(x_i)) = \frac{e^{g_i}}{\sum_k^n e^{g_k}}$
        $i \in [1,m]$ ,where $g_i$ is one class value in $G_k(\cdot)$.
        Return Op
    end for
```

3.3. Network design

CED-Net can guarantee good performance while maintaining lightweight models because of the effective combination of bottleneck layer structure and channel enhancement blocks. An important difference between CED-Net and other network architectures is that CED-Net has a very narrow layer. The relatively small channel growth rate is sufficient to obtain the most advanced results on the test dataset. This can increase the proportion of features from the later layers relative to features from the previous layers. So, we set the channel growth rate of a dense connection layer to 4. And we found that if the number of early layers is set too deep, it will significantly increase the FLOPs of the network.

Architectural details. The model used in our experiments has three dense blocks. Before the data enters the first dense block, the input image would go through a 3 × 3 standard convolution which output channels are 16 and stride size is 2. In the dense layer, the number of channel enhancement blocks should be set according to the growth rate, the input channels, and the output channels, see Eq (3).

$$n = \frac{Cout - Cin}{g} \tag{3}$$

where g is the growth rate, C_{in} is the input channels, n is the number of channel enhancement blocks, and C_{out} is the output channels. For example, in the experiment, we set the growth rate to 8, 16, and 32, and the channels of dense layer output is 256, 756, and 1696 respectively, so the number of channel enhancement blocks in the dense layer are all 30.

Table 1. Network structure of CED-Net on CIFAR.

Layers	Output Size	Output Channels	Repeat	Stride
3 × 3 Convolution	32 × 32	16	1	1
Dense bottleneck block	32 × 32	256 (g = 8)	30	1
Avg pooling	16 × 16		1	2
Dense bottleneck block	16 × 16	736 (g = 16)	30	1
Avg pooling	8 × 8		1	2
Dense bottleneck block	8 × 8	1696 (g = 32)	30	1
Global avg pooling	1 × 1	1696	1	8
Fully connected	1 × 1	10	1	

Table 2. Network structure of CED-Net on ImageNet.

Layers	Output Size	Output Channels	Repeat	Stride
3 × 3 Convolution	112 × 112	64	1	2
Dense bottleneck block	112 × 112	96 (g = 8)	4	1
Avg pooling	56 × 56		1	2
Dense bottleneck block	56 × 56	192 (g = 16)	6	1
Avg pooling	28 × 28		1	2
Dense bottleneck block	28 × 28	448 (g = 32)	8	1
Avg pooling	14 × 14		1	2
Dense bottleneck block	14 × 14	1088 (g = 64)	10	1
Avg pooling	7 × 7		1	2
Dense bottleneck block	7 × 7	2112 (g = 128)	8	1
Global avg pooling	1 × 1	2112	1	7
Fully connected	1 × 1	1000	1	

For each convolutional layer with a kernel size of 3 × 3, each side of the input is zeros-padded to keep the feature size fixed. In general, we add the batch normalization layer and the ReLU function after the last dense layer and then use the global average pooling to compress the feature map into one dimension as the input of the Softmax layer. The exact network configuration is shown in Tables 1 and 2.

The training process of CED-Net is shown in Algorithm 1. (x_i, y_i) in the input represent the images and label of the ith batch respectively. For each batch, we use softmax to obtain the output Yi of CED-Net. Finally, the image features G_k of n categories are obtained.

4. Experiments

This section conducted experiments on the CIFAR10, CIFAR-100, and the ImageNet (ILSVRC 2012) datasets. First, we compared them with other advanced convolutional neural networks, such as VGG16, ResNet-101, and DenseNet. Then, we conducted ablation experiments to CED-Net, mainly comparing three networks, the primary network of CED-Net-128, the optimization network with only the bottleneck layer, and the network with only the channel enhancement block. Through these experiments, we verify the effectiveness of our improved method. Next, we will introduce the data set and the evaluation indicators of the experiment.

4.1. Dataset

The CIFAR-10 and CIFAR-100 datasets consist of colored natural images with 32 × 32 pixels. CIFAR-10 consists of images drawn from 10 classes and CIFAR-100 from 100 classes. The training and test sets contain 50,000 and 10,000 images, respectively, and we picked up 5000 training images as a validation set. We adopt a standard data augmentation scheme (mirroring/shifting) and image zero-padded with 4 pixels per side, and then randomly cropped to generate a 32 × 32 image. The image is flipped horizontally at a probability of 0.5 and normalized by subtracting the channel average and dividing by the channel standard deviation.

The ImageNet datasets consist of 224 × 224 pixels colored natural images with 1000 classes. The training and validation sets contain 1,280,000 and 50,000 images, respectively. We adopt the data-augmentation scheme at training time and perform a rescaling to 256 × 256 followed by a 224 × 224 center crop at test time before feeding the input image into the networks.

4.2. Evaluation criteria

We evaluate CED-Net on three criteria:

4.2.1. Classification accuracy

Accuracy is the most common metric. It is the number of samples that are paired divided by the number of all samples. Generally speaking, the higher the accuracy is, the better the classifier will be:

$$accuracy = (TP + TN)/(P + N) \tag{4}$$

where P (positive) is the number of positive examples in the sample, and N (negative) is the number of negative examples. TP (true positives) is the number of samples that are positive examples that are correctly classified. TN (true negatives) is the number of samples that are actually negative that are correctly classified.

4.2.2. Model parameter

For a single convolutional kernel we have:

$$parameters = k^2 \times Cin \times Cout \tag{5}$$

where k is the convolution filter’s size, Cin is the input channels, and Cout is the output channels;

4.2.3. FLOPs (floating-point operations)

To measure the amount of calculation of the model, we compute the number of FLOPs of each layer. For convolutional kernels, we have:

$$FLOPs = 2HW\ (k^2\ Cin + 1)Cout \tag{6}$$

where H, W are height and width. For fully connected layers, we compute FLOPs as:

$$FLOP\,s = (2I - 1)O \tag{7}$$

where I is the input dimensionality and O is the output dimensionality.

4.2.4. Other criteria

To further prove the stability of CED-Net, we added the interpretation and comparison of precision, recall and F-measure in the ablation experiment:

$$precision = TP/(TP + FP) \tag{8}$$

$$recall = TP/(TP + FN) \tag{9}$$

$$F - measure = 2 * precision * recall/(precision + recall) \tag{10}$$

4.3. The result of image classification

4.3.1. Training details

We train all models with stochastic gradient descent (SGD) using similar optimization hyper-parameters [23–30]. And we set the Nesterov momentum weight to 0.9 without damping and use a weight decay of 0.0001. All models are trained with mini-batch size 128 for 200 epochs on the training datasets. We use the cosine annealing learning rate curve, starting from 0.1 and gradually reducing to 0.

4.3.2. Results on CIFAR

In this part, we train CED-Net and other advanced convolutional neural networks on the CIFAR-10 and CIFAR100 datasets. We compared these models under the above three evaluation criteria. See Table 3 for a detailed list.

Table 3. The classification accuracy on CIFAR-10 and CIFAR-100.

Model	Params	FLOPs	CIFAR-10	CIFAR-100
VGG-16	14.73 M	314 M	92.64	72.23
ResNet-101	42.51 M	2515 M	93.75	77.78
ResNeXt-29	9.13 M	1413 M	94.82	78.83
MobileNet V2	2.30 M	92 M	94.43	68.08
DenseNet-121	6.96 M	893 M	94.04	77.01
CondenseNet-86	0.52 M	65 M	94.48	76.36
CondenseNet-182	4.20 M	513 M	95.87	80.13
CED-Net-128	0.69 M	75 M	94.89	77.35
CED-Net-272	5.32 M	649 M	96.31	80.72

In Table 3, we show the results of comparing 128-layer CED-Net and 272-layer CED-Net with other state-of-the-art CNN architectures. All models were trained in 200 epochs in the experiment. The results show that after introducing the bottleneck layer structure and channel enhancement blocks to CED-Net, the CondenseNet increases the accuracy by 0.4–0.5% with minimal parameters and FLOPs cost compared with the same number of stacked blocks n datasets. Moreover, compared to the more advanced MobileNet V2, CED-Net is more accurate without using depth-wise separable convolutions. And the parameter amount is 1/4 of it, and the FLOPs are also more minor.

4.3.3. Results on ImageNet

In this part, we train CED-Net and other advanced convolutional neural networks on the ImageNet datasets. We compared these models under the above four evaluation criteria. See Table 4 for a detailed list.

Table 4. The classification accuracy on ImageNet.

Model	Params	FLOPs	Top-1	Top-5
VGG-16	138.36 M	15.48 G	71.93	90.67
ResNet-101	44.55 M	7.83 G	80.13	95.4
MobileNet V2	3.5 M	0.3 G	71.8	91
DenseNet-121	7.98 M	2.87 G	74.98	92.29
CondenseNet	4.8 M	0.53 G	73.8	91.7
SE-Net	115 M	20.78 G	81.32	95.53
CED-Net-115	9.3 M	1.13 G	78.65	93.7

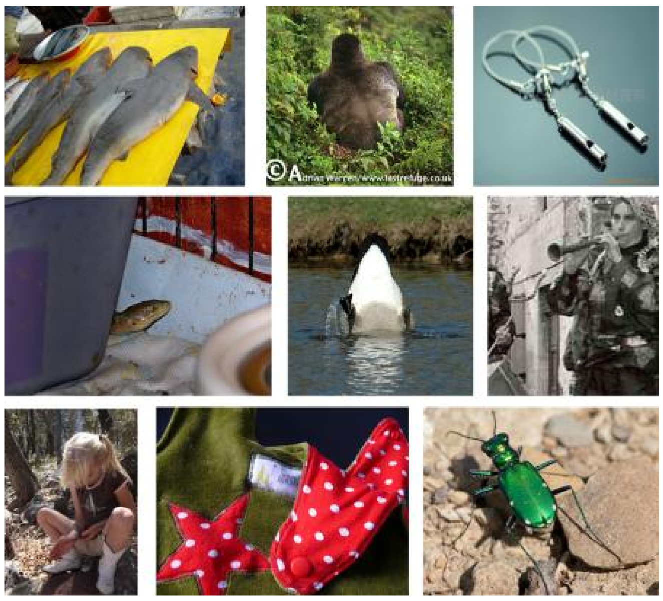

Figure 5. ImageNet misclassified pictures.

In Table 4, we show the results of comparing 115-layer CED-Net with other CNN architectures. The results show that the accuracy of Top-1 and Top-5 is improved by 4.85 and 2%, respectively, compared with the same depth of CondenseNet. At the same time, the dense bottleneck block used in the CED net is more complex. Compared with DenseNet, CED-Net increases the number of parameters by 16.5% but reduces the amount of calculation by 39.4%; the accuracies of Top-1 and top-5 are improved by 3.67 and 1.41%, respectively. Compared with SE-Net, CED-Net reduces the Top-1 accuracy by 2.67%, but the parameter quantity is only 8.1% of SE-Net.

Some misclassification images are shown in Figure 5. There may be unavoidable interference information in these pictures; Also, it may be that the network model constructed in this paper does not learn a sufficient number of diverse features and cannot correctly identify each picture with different features.

4.3.4. Redundant LG-Conv

In the dense bottleneck block shown in Figure 2, we use the learned group revolution before and after the 3 × 3 group convolution, which means that there are two consecutive learned group convolutions between the two 3 × 3 group convolutions. The two index layers used have redundancy, but we think it is necessary. These redundancies can improve the learned group revolution's generalization performance and help subsequent feature extraction. But this design increases the amount of calculation and parameters of the intermediate convolution.

4.3.5. Ablation experiments

In this part, we performed a CED-Net ablation experiment. We trained four models on the CIFAR-10 dataset, CEDNet-128a with no bottleneck layer and channel enhancement block, CED-Net-128b with convolutional layer structure changed to bottleneck layer, CED-Net-128c with channel enhancement block based on CondenseNet-86 and CED-Net-128 that we proposed in this paper.

Table 5. The result of ablation experiments of CIFAR-10.

Model	Params	FLOPs	Accuracy	Precision	Recall	F-measure
CondenseNet[a]	0.52M	65.82M	94.48	94.50	94.48	94.49
CED-Net-128[b]	0.59M	75.04M	94.75	94.75	94.75	94.75
CED-Net-128[c]	0.66M	67.04M	94.74	94.76	94.74	94.75
CED-Net-128	**0.69M**	**75.41M**	**94.89**	**94.89**	**94.89**	**94.88**

Note: [a]The basic model of CED-Net same as CondenseNet-86 without bottleneck layer and channel enhancement block; [b]The basic model of CED-Net only add a bottleneck layer; [c]The basic model of CED-Net only add channel enhancement block.

In Table 5, CondenseNet-86 is our basic model. It can be seen that when we turn the structure of CondenseNet into a bottleneck layer, the parameters and FLOPs of the network are only slightly improved, and the accuracy can be increased by about 0.3%. When we added the channel enhancement block to CondenseNet-86, we saw not much increase in FLOPs. But the parameters are raised, and the accuracy can be improved by about 0.3%. In our CED-Net-128, the accuracy rate has been significantly improved, and the channel enhancement block mainly causes the increase in parameters.

The bottleneck layer structure causes an increase in FLOPs. In addition, the Accuracy, Precision, Recall and F-measure of each model are very close, which prove that the four models have extracted stable features.

5. Conclusions

This paper introduces CED-Net: a more efficient densely concatenated convolutional neural network based on feature enhancement block and bottleneck layer structure, which increases accuracy by learning group convolution and feature reuse. To make the reasoning effective, the pruned network can be converted to a network with conventional group convolution, which is effectively implemented in most deep learning libraries. In our experiments, CED-Net outperformed its underlying network CondenseNet and other advanced convolutional neural networks such as Mobilenet V2 and ResNeXt in terms of computational efficiency at the same accuracy level. Moreover, CED-Net has a much simpler structure with higher accuracy. We anticipate further research in CED-Net to combine this framework to the Neural Architecture Search (NAS), so as to design more lightweight Convolutional Neural Network models. We hope our work will draw more attention toward a broader view of using lightweight architecture for deep learning.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61976217), the Opening Foundation of Key Laboratory of Opto-technology and Intelligent Control, Ministry of Education (KFKT2020-3), the Fundamental Research Funds of Central Universities (No. 2019XKQ YMS87), Science and Technology Planning Project of Xuzhou (No. KC21193).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. Shao, J. C. W. Lin, G. Srivastava, A. Jolfaei, D. Guo, Y. Hu, Self-attention-based conditional random fields latent variables model for sequence labeling, *Pattern Recognit. Lett.*, **145** (2021), 157–164. https://doi.org/10.1016/j.patrec.2021.02.008
2. J. C. W. Lin, Y. Shao, Y. Djenouri, U. Yun, ASRNN: A recurrent neural network with an attention model for sequence labeling, *Knowl. Based Syst.*, **212** (2021), 106548. https://doi.org/10.1016/j.knosys.2020.106548
3. Y. Djenouri, G. Srivastava, J. C. W. Lin, Fast and accurate convolution neural network for detecting manufacturing data, *IEEE Trans. Ind. Inf.*, **17** (2021), 2947–2955. https://doi.org/10.1109/TII.2020.300149
4. Y. Shao, J. C. W. Lin, G. Srivastava, D. Guo, H. Zhang, H. Yi, et al., Multi-objective neural evolutionary algorithm for combinatorial optimization problems, *IEEE Trans. Neural Networks Learn. Syst.*, **2021** (2021), 1–11. https://doi.org/10.1109/TNNLS.2021.3105937

5. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv:1409.1556.
6. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 1–9. https://doi.org/10.48550/arXiv.1409.4842
7. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778. https://doi.org/10.48550/arXiv.1512.03385
8. S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 5987–5995. https://doi.org/10.1109/CVPR.2017.634
9. G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 2261–2269. https://doi.org/10.1109/CVPR.2017.243
10. J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 7132–7141. https://doi.org/10.1109/CVPR.2018.00745
11. B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le, Learning transferable architectures for scalable image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 8697–8710. https://doi.org/10.48550/arXiv.1707.07012
12. C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L. J. Li, et al., Progressive neural architecture search, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 19–34. https://doi.org/10.48550/arXiv.1712.00559
13. E. Real, A. Aggarwal, Y. Huang, Q. V. Le, Regularized evolution for image classifier architecture search, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 4780–4789. https://doi.org/10.1609/aaai.v33i01.33014780
14. A. Banitalebi-Dehkordi, Knowledge distillation for low-power object detection: A simple technique and its extensions for training compact models using unlabeled data, in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, (2021), 769–778. https://doi.org/10.1109/ICCVW54120.2021.00091.
15. A. Goel, C. Tung, Y. H. Lu, G. K. Thiruvathukal, A Survey of methods for low-power deep learning and computer vision, in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, (2020), 1–6. https://doi.org/10.1109/WF-IoT48130.2020.9221198
16. F. Chollet, Xception: Deep learning with depthwise separable convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 1251–1258. https://doi.org/10.48550/arXiv.1610.02357
17. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, preprint, arXiv:170404861.
18. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 4510–4520. https://doi.org/10.48550/arXiv.1801.04381
19. F. Zhang, Q. Li, Y. Ren, H. Xu, Y. Song, S. Liu, An expression recognition method on robots based on mobilenet V2-SSD, in *2019 6th International Conference on Systems and Informatics (ICSAI)*, IEEE, (2019), 118–122. https://doi.org/10.1109/ICSAI48974.2019.9010173

20. X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 6848–6856. https://doi.org/10.1109/CVPR.2018.00716
21. N. Ma, X. Zhang, H. T. Zheng, J. Sun, Shufflenet v2: Practical guidelines for efficient CNN architecture design, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 116–131. https://doi.org/10.48550/arXiv.1807.11164
22. G. Huang, S. Liu, L. Van der Maaten, K. Q. Weinberger, Condensenet: An efficient densenet using learned group convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 2752–2761. https://doi.org/10.48550/arXiv.1807.11164
23. R. Zhang, F. Zhu, J. Liu, G. Liu, Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis, *IEEE Trans. Inf. Forensics Secur.*, **15** (2020), 1138–1150. https://doi.org/10.1109/TIFS.2019.2936913.
24. S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding, preprint, arXiv:151000149.
25. H. Lin, A. Kadav, I. Durdanovic, H. Samet, H. P. Graf, Pruning filters for efficient convnets, preprint, arXiv:1608.08710.
26. Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), 2736–2744. https://doi.org/10.48550/arXiv.1708.06519
27. Y. He, P. Liu, Z. Wang, Z. Hu, Y. Yang, Filter pruning via geometric median for deep convolutional neural networks acceleration, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), 4340–4349. https://doi.org/10.48550/arXiv.1811.00250
28. M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: ImageNet classification using binary convolutional neural networks, in *European Conference on Computer Vision*, (2016), 525–542. https://doi.org/10.1007/978-3-319-46493-0_32
29. Y. Jiao, S. Li, X. Huo, Y. K. Li, Synchronous weight quantization-compression for low-bit quantized neural network, in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, (2021), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533393
30. H. Mao, S. Han, J. Pool, W. Li, X. Liu, Y. Wang, et al., Exploring the regularity of sparse structure in convolutional neural networks, preprint, arXiv:170508922.
31. W. Yin, G. Dong, Y. Zhao, R. Li, Coresets application in channel pruning for fast neural network slimming, in *2021 International Joint Conference on Neural Networks (IJCNN)*, (2021), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533343