A Lightweight Neural Network Combining Dilated Convolution and Depthwise Separable Convolution

Wei Sun[1,2], Xijie Zhou[1](✉), Xiaorui Zhang[2,3], and Xiaozheng He[4]

[1] School of Automation,
Nanjing University of Information Science and Technology, Nanjing 210044, China
sunw0125@163.com, cfcfwyjd@163.com
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing 210044, China
[3] Jiangsu Engineering Center of Network Monitoring,
Nanjing University of Information Science and Technology, Nanjing 210044, China
[4] Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract. Aimed to reduce the excessive cost of neural network, this paper proposes a lightweight neural network combining dilated convolution and depthwise separable convolution. Firstly, the dilated convolution is used to expand the receptive field during the convolution process while maintaining the number of convolution parameters, which can extract more high-level global semantic features and improve the classification accuracy of the network. Second, the use of the depthwise separable convolution reduces the network parameters and computational complexity in convolution operations. The experimental results on the CIFAR-10 dataset show that the proposed method improves the classification accuracy of the network while effectively compressing the network size.

Keywords: Lightweight neural network · Dilated convolution · Depthwise separable convolution · Classification accuracy · Cloud computing

1 Introduction

In recent years, convolutional neural networks have been used as an effective model in deep learning with significant progress in many fields, such as image processing, object detection and semantic segmentation. In 2012, Krizhevsky, et al. [1] first adopted deep learning algorithm and the AlexNet and won the champion of ImageNet Large Scale Visual Recognition Challenge. Since then, various convolutional neural network models have been proposed in the computer vision competition. In 2014, the Visual Geometry Group at the University of Oxford proposed the VGGNet [2], Google researchers proposed the GoogLeNet [3], and He et al. proposed the ResNet [4, 5] in 2015. These networks improve the performance of the AlexNet [6] at the cost of deeper and more complex networks to achieve higher accuracy. With the higher and higher precision for computer vision tasks, the model depth and parameters are also exponentially increasing, making these models only run on GPUs with high computing power [3]. As a consequence, existing deep neural network models cannot be deployed on these resource-constrained devices [10, 18], such as mobile phones and in-vehicle embedded devices, due to their limitations in computing power and storage capacity. The emerging cloud computing has the potential to solve this challenge.

Cloud computing technology, which combines the characteristics of distributed computing, parallel computing, and grid computing, provides users with scalable computing resources and storage space by using massive computing clusters built by ordinary servers and storage clusters built by a large number of low-cost devices. However, the currently-existing high-performance cloud computing servers are too expensive to afford for individuals and small companies.

To enhance the affordability of cloud computing, many studies propose various lightweight neural networks. Some of them aim at reducing the size of neural network through compressing model. For example, Landola et al. [3] proposed the SqueezeNet that applies a convolution kernel to convolve and dimension the upper features and a feature convolution to perform feature stacking, which greatly reduces the number of parameters of convolution layers. Zhang et al. [14] proposed the ShuffleNet, which groups multi-channel feature lines and performs convolution to avoid unsmooth information flow. Howard et al. [15] proposed the depthwise separable convolution model, named MobileNet, which convolves the features of each channel separately and uses 1×1 convolution to splice all features of different channels. These light-weight models greatly reduce the number of network parameters and computational cost. However, the classification accuracy of the compression process cannot be guaranteed because the compression implementation only uses local information of the image.

In order to address these issues, this paper proposes a lightweight neural network combining dilated convolution and depthwise separable convolution. The proposed model divides the convolution process into two processes: expansion convolution and depthwise separable convolution. Depthwise separable convolution is used to reduce network computation. However, the use of the depthwise separation convolution cannot guarantee the classification accuracy of the model [11]. To solve this problem, we introduce dilated convolution to the depthwise separable convolution architecture. The dilated convolution can increase the receptive field of the network in the convolution process without increasing the number of convolution parameters, which help extract more global features and higher-level semantic features, thus improving the classification accuracy.

2 Approach

This paper uses dilated convolution as a filter to extract the feature of the image. Compared with the traditional filters, the dilated convolution yields more full-image information without increasing the number of network parameters, where the dilated rate δ controls the size of each convolution dilation. Then, we apply depthwise separable convolution to reduce the computational complexity and size of the model. This section first presents the idea of building a joint module for dilated convolution and depthwise separable convolution, which is used to build the deep convolution network.

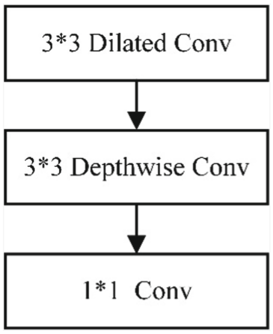


Fig. 1. Joint module

2.1 Joint Module

The joint module is the core of the proposed neural network. As shown in Fig. 1, the proposed model dilates each filter to obtain more image information without increasing the amount of calculation. The dilated filter is then used to convolve each input channel, and the final filter combines the output of different convolution channels.

Figure 2 illustrates the dilation process of 3 × 3 filter for the dilated convolution process in Fig. 1. The position of the dot mark in Fig. 2 indicates that there is a non-zero weight, and the node without the dot mark represents zero weight to that position. In Fig. 2(a), (b), and (c) represent filters with different dilated rates, respectively. The parameters of the convolution layer remain the same, and the amount of convolution process is the same too. The receptive fields of the filters (a), (b), and (c) are defined as $3 \times 3 = 9$, $7 \times 7 = 49$ and $11 \times 11 = 121$, respectively. The increase of the receptive field means that each node contains higher semantic features, which can improve the classification accuracy. To factor the influence of different dilated convolution on model accuracy, we apply hyperparameter δ to control the size of each dilated convolution. As illustrated by Fig. 2, the relationship between the receptive field and the original filter size can be represented as:

$$C = \{(S + 1)\delta + S\} \tag{1}$$

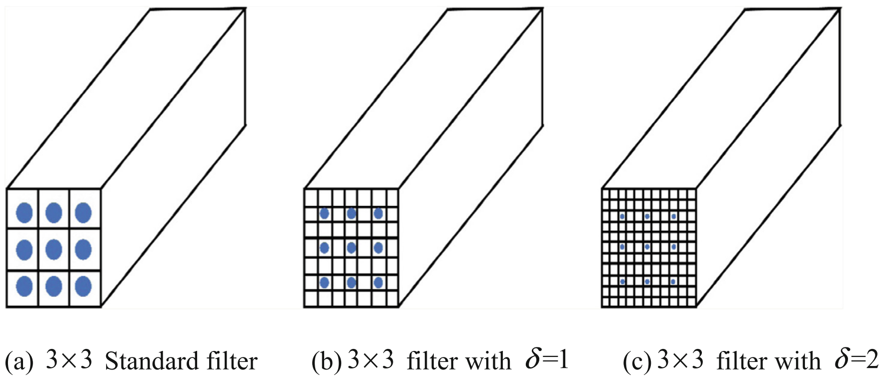
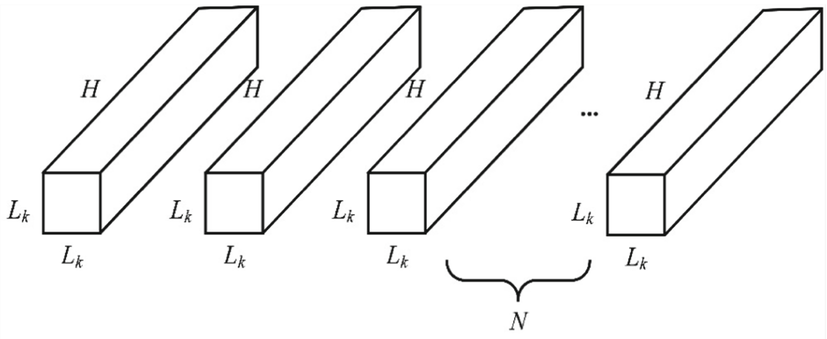

(a) 3×3 Standard filter
(b) 3×3 filter with δ=1
(c) 3×3 filter with δ=2

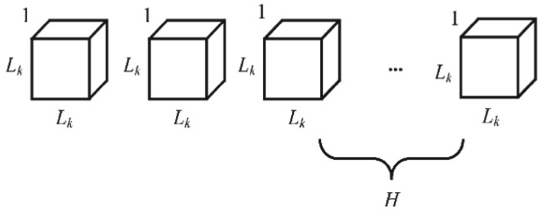
Fig. 2. Dilated convolution process

where C denotes the size of the receptive field, S the size of the initial filter, and δ the expansion rate.

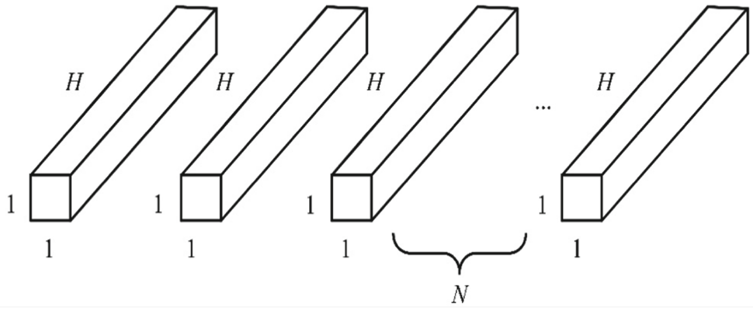
Separable convolution operation is carried out on the obtained dilated convolution filter. The size of the dilation filter is $L_k \times L_k$ with $L_k = \sqrt{C}$. Figure 3 shows the process of constructing a $L_i \times L_i \times H$ feature map and a $L_i \times L_i \times N$ feature map. This process clearly shows how to reduce the number of parameters in the model.



(a) Traditional convolution filters



(b) Depthwise convolution filters



(c) Pointwise convolution filters

Fig. 3. Depthwise separable convolution process for dilated filters

Figure 3(a), (b) and (c) represent the traditional convolution filter, depthwise convolution filter, and pointwise convolution filter, respectively. Figure 3(b) and (c) together represent a separable convolution process, where $L_i \times L_i$ is the width and height of the input feature map, N is the number of filters, $L_k \times L_k$ is the width and height of the dilated filter, and H is the number of channels. For example, a single dilated filter of $L_k \times L_k$ is firstly used to carry out convolution operations on each channel. If the number of the feature map channels is H, there are H filters with the same size to participate in the convolution operation, and the number of channels of each filter is 1. The image is then convolved by N filters with 1×1 size and H convolution channels. Figure 3 shows that a traditional convolution layer receives a $L_i \times L_i \times H$ feature map and produces a $L_i \times L_i \times N$ feature map. The amount of computation of traditional convolution is:

$$G = L_k \times L_k \times H \times N \times L_i \times L_i \tag{2}$$

The amount of computation of depthwise separable convolution is:

$$G = L_k \times L_k \times H \times L_i \times L_i + H \times N \times L_i \times L_i \tag{3}$$

Therefore, the ratio of separable convolution to the standard convolution can be represented by:

$$\frac{L_k \times L_k \times H \times L_i \times L_i + H \times N \times L_i \times L_i}{L_k \times L_k \times H \times N \times L_i \times L_i} = \frac{1}{N} + \frac{1}{L_k^2} \tag{4}$$

Equation (4) quantifies the computational reduction of separable convolution as $\frac{1}{N} + \frac{1}{L_k^2}$ compared to the conventional convolution process.

2.2 Network Architecture

To avoid the vanishing gradient problem and accelerate the network training, we apply the BN layer (Batch Normalization) and the ReLU layer to make the gradient larger [16, 17] after the joint module introduced in Sect. 2.1. This paper labels the process presented in Fig. 4 as a basic network structure.

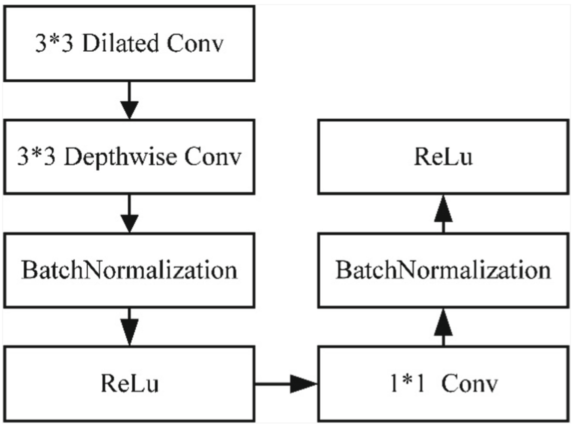


Fig. 4. Basic structure

Using only one basic structure is often not enough to form a usable neural network. Because the network is too shallow, we won't get the deep information of the image. Therefore, applying the basic structure repetitively to construct a lightweight neural network can improve the neural network performance, as shown in Fig. 5. This lightweight neural network complements the joint module to form the basic structure. Several basic network structures are combined with the average pooling layer, the full connection layer, and the Softmax layer to form the overall neural network structure. In total, the model consists of 30 layers, including one average pooling layer and one fully connected layer, 9 dilated convolution layers, 9 depthwise separable convolution layers, 9 BN layers, and one Softmax layer.

2.3 Loss Function and Optimization

We adopt cross-entropy as the loss function of neural network, using Adam as the network optimizer. The formula for cross-entropy is as follows:

$$W(p,q) = \sum_i p(i) * \log(\frac{1}{q(i)}) \tag{5}$$

where $W(p,q)$ represents cross-entropy of the distribution of the true mark, q is the predicted mark distribution of the trained model, and cross-entropy loss function measures the similarity between p and q.

Adam is considered to be robust in selecting hyperparameters. Therefore, this paper adopts adaptive Adam learning rate to optimize the proposed model.

3 Experiments

In order to verify the effectiveness of the proposed model, we constructed an experimental platform and selected a typical dataset. Then, the proposed network model was compared with other models to verify the effectiveness of the proposed model. Furthermore, we investigated the influence of the dilated convolution size on the classification accuracy of the model and verified that the classification accuracy of the proposed model.

All experiments were carried out on a computer with Intel Core i7-7700k CPU, 4.20 GHz $\times$ 8 frequency, and GTX 1080Ti graphics card. CUDA version 9.0 and cuDNN version 7.3.1 were installed. To configure the environment required for the training model. The proposed model and algorithms were compiled and operated on TensorFlow 1.12.2.

3.1 Comparison of the Proposed Network with Other Networks

To demonstrate the performance of the proposed model in network compression while ensuring accuracy of classification, we compare the proposed network with other mainstream networks and illustrate their classification accuracy based on the dataset CIFAR – 10. The comparisons are shown in Table 1.

Table 1 shows that, compared with some mainstream networks, the proposed network model achieves high accuracy on CIFAR-10 dataset. The proposed network provides

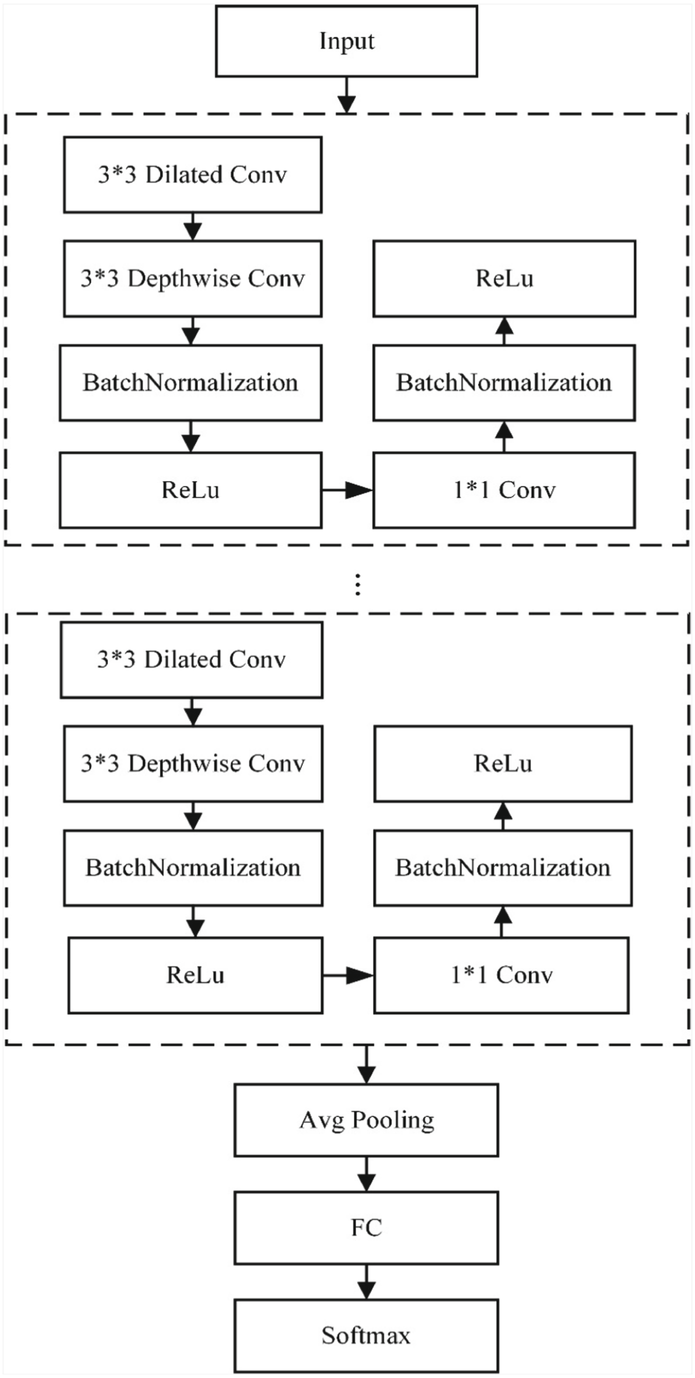


Fig. 5. Structural flow chart

Table 1. The proposed network vs popular networks

Model	CIFAR-10 accuracy	Million parameters
This paper	84.25%	3.1
1.0 MobileNet 224	83.91%	4.2
GoogleNet	83.84%	6.8
SqueezeNet	69.83%	1.25
VGG 16	86.17%	138

accurate results while greatly reducing the number of network parameters compared with MobileNet and GoogleNet. The SqueezeNet typically acquires fewer parameters, however, at the cost of low accuracy. Although the proposed network requires more parameters than SqueezeNet, it is much better in terms of classification accuracy. Because SqueezeNet sacrifices classification accuracy, it cannot meet the high-accuracy need in practical applications. Therefore, the proposed network is superior to SqueezeNet. In addition, although the VGG16 network has slightly higher accuracy in classification results than the proposed network, its model size is dozens more than the proposed model, resulting in computational difficulty when computing power is limited. Due to much fewer network parameters, the proposed network can be easily implemented on mobile devices with less storage capacity while having good classification accuracy.

3.2 Different Dilated Rate

This study applies the dilated rate to control the size of the dilated convolution, which affects the size of the receptive field, leading to the change of classification accuracy. Therefore, we compare the network classification accuracy under different dilated rates to verify the effect of the dilation rate, as summarized in Table 2.

Table 2. Classification accuracy of different dilated rates

Dilated rate	CIFAR-10 accuracy
0	82.04%
1	83.32%
2	84.25%
3	83.56%

Applying different dilated rates to the dilated convolution using the CIFAR-10 dataset, Table 2 shows that the joint dilated convolution and the depthwise separable convolution improve classification accuracy by two percent compared to those networks without joint dilated convolution. It also shows that the maximum classification accuracy is achieved when the dilated rate is 2. As the dilated rate continues to increase, the classification accuracy decreases slightly. This is because the dilated rate increases the receptive field, while the larger the receptive field, which means it may contain more global and semantic features. This observation warns us that blindly expanding the receptive field can lose a lot of local and detailed information during the convolution process, affecting the classification accuracy of small targets and distant objects.

3.3 Generalization Capability

The results in previous sections show that the proposed network performs well on the CIFAR-10 dataset. To investigate the proposed model's performance stability on other datasets, we conducted training and testing on Tiny ImageNet. The result is as follows.

Table 3. This paper network in Tiny ImageNet

Model	Tiny ImageNet accuracy
This paper	85.01%
1.0 MobileNet 224	83.81%
GoogleNet	82.94%

Table 3 shows that the proposed network has good accuracy on Tiny ImageNet. Compared with the MobileNet of Width Multiplier = 1 and Resolution Multiplier = 224, the proposed network improves the accuracy on both datasets. Compared with GoogleNet, the proposed network enhances the accuracy rate on Tiny ImageNet from 82.94% to 85.01%. Based on these comparisons, we can conclude that the proposed network can consistently improve classification accuracy, indicating a good generalization ability. The proposed model also reduces the size under the premise of ensuring accuracy, which makes it possible to achieve better classification accuracy on mobile devices.

Although the proposed network is a classification network, the proposed network can be used as the basic network of SSD or YOLO models, or transplanted to different devices, to realize real-time pedestrian detection. However, the model of pedestrian detection requires higher computational cost, which will affect the accuracy of pedestrian detection on the equipment.

4 Conclusion

This paper proposes a lightweight neural network model for joint dilated convolution and depthwise separable convolution. The joint model can reduce the computational burden with depthwise separable convolution, making it possible to apply the network to computationally-constrained devices. Meanwhile, the dilated convolution is used to increase the receptive field in the process of convolution without increasing the number of convolution parameters. It supplies global features, and higher semantic-level features can be extracted in the process of convolution. The joint model can also improve classification accuracy. Experimental results demonstrate that the proposed model makes a good compromise between the classification accuracy and the model size, while the classification accuracy of the network is guaranteed when the network is compressed. Article puts forward the lightweight of neural network can reduce the cost of cloud computing. In addition, the proposed network can be combined with Internet-of-things. For example, the depth network can be further optimized and transplanted in Android mobile devices, embedded devices such as MCU or FPGA. Such applications will convey significant impacts on human life, work, health, and many other areas of our society [21, 22].

Acknowledgement. This work is supported in part by the National Nature Science Foundation of China (No. 61304205, 61502240), Natural Science Foundation of Jiangsu Province (BK20191401), and Innovation and Entrepreneurship Training Project of College Students (201910300050Z, 201910300222).

References

1. Krizhevsky, A., Sutskever, I., Geoffrey, E.H.: ImageNet classification with deep convolutional neural networks (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Comput. Sci. (2014)
3. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
4. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
5. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks, pp. 1–6. arXiv preprint arXiv:1611.05431 (2016)
6. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size, pp. 1–8. arXiv preprint arXiv:1602.07360 (2016)
7. Ghemawat, S., Gobioff, H., Leung, S.T.: The Google file system. ACM SIGOPS Oper. Syst. Rev. **37**(5), 29–43 (2003)
8. Chang, F., Dean, J., Ghemawat, S., et al.: Bigtable: a distributed storage system for structured data. ACM Trans. Comput. Syst. (TOCS). **26**(2), 4–5 (2008)
9. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
10. Chollet, F.: Xception: deep learning with depthwise separable convolutions, p. 1. arXiv preprint arXiv:1610.02357v2 (2016)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks, pp. 1–7. arXiv preprint arXiv:1709.01507 (2017)
12. Zhang, X., Zhou, X., Lin, M., Sun, J.: ShuffleNet: an extremely efficient convolutional neural network for mobile devices. arXiv preprint arXiv:1707.01083 (2017)
13. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

14. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning quantization and Huffman coding. arXiv preprint arXiv:1510.00149v5 (2016)
15. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift, pp. 3–5. arXiv preprint arXiv:1502.03167 (2015)
16. Krizhevsky, A., Sutskever, I., Geoffrey, E.H.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012)
17. Hu, H., Peng, R., Tai, Y.W., et al.: Network trimming: a data-driven neuron pruning approach towards efficient deep architectures. In: Proceedings of the International Conference on Learning and Representation (ICLR), pp. 214–222. IEEE (2017)
18. Qiu, J., et al.: Going deeper with embedded FPGA platform for convolutional neural network. In: ACM International Symposium on FPGA (2016)
19. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices, pp. 1–2. arXiv preprint arXiv:1512.06473 (2015)