ACOUSTIC SCENE CLASSIFICATION BASED ON LIGHTWEIGHT CNN WITH EFFICIENT CONVOLUTIONS

Technical Report

Guoqing Feng

Tianjin University
School of Electrical and Information, 92 Weijin Road
Tianjin, 300072, China
335789715@qq.com

Jinhua Liang

Tianjin University
School of Electrical and Information, 92 Weijin Road
Tianjin, 300072, China
tjuljh@tju.edu.cn

Biyun Ding

Tianjin University
School of Electrical and Information, 92 Weijin Road
Tianjin, 300072, China
1398491993@qq.com

ABSTRACT

This technical report is for the Task 1B Acoustic scene classification of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE). Targeting low complexity solutions for the classification problem in term of model size, a kind of lightweight Convolutional Neural Network (CNN) with efficient convolutions is designed. The network is constructed by a kind of improved bottleneck block based on the inverted residual linear bottleneck block. In the improved bottleneck block, the operations of Detpthwise Channel Ascent (DCA) and Group Channel Descent (GCD) are used to replace pointwise convolution to realize efficient channel transformation. The designed network is denoted by CNN-BDG in this report. CNN-BDG realizes a better performance which is 4.46% higher than the baseline model in the validation set. Besides, the parameters are reduced to about 30% compared to the baseline model.

Index Terms— Acoustic scene classification, Lightweight convolutional neural network, efficient convolution

1. INTRODUCTION

Acoustic Scene Classification (ASC) is aiming at helping people to automatically make sense of the environment through the analysis of sound. ASC task is involved in many problems in parallel. For example, in setting up a dataset, audio segmentation, personal archiving and labeling need to be done by a lot of manual effort. In the audio pre-processing, the audio needs to be denoised and the feature used to classification needs to be extracted manually. Recently, ASC is a regular task in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge series. DCASE provides a large dataset for participant to study the ASC algorithm instead of the collection of sound data [1].

Deep learning methods have been successful applied to ASC task in recent years. Among them, the Convolutional Neural Network (CNN) is one of the most popular because of the excellent performance. There are two types of CNN according to the dimension of convolution. One is 1-dimensional (1D) CNN. For example, [2], [3] used 1D CNN to achieve end-to-end environmental sound classification. The other is 2-dimensional CNN. In DCASE2019 and DCASE2020, the baseline system used Mel spectrogram as input fed into 2D CNN. Furthermore, some networks proposed for machine vision are also used in ASC tasks, such as VGG [4] and ResNet [5]. With the increasing size of CNN model, the performance of CNN becomes better and better recently. However, most applications of ASC are deployed in mobile devices, such as mobile phone, hearing-aid and so on. The efficiency of CNN model has become the maximum limitation in the actual application of ASC.

To address the problem of over-complicated CNN models, many methods of model compression and acceleration have been proposed based on the pre-trained network model optimization. For example, model pruning [6] deletes some unimportant channels or connections in the network model; parameter quantization [7] expresses the parameters of the model in a format with fewer bits; weights sharing [8] represents multiple weight parameters into one to reduce the number of parameters. However, the methods of optimizing the pre-trained network model do not fundamentally improve the network structure. The optimized performance depends on the original network structure. Different from optimizing the pre-trained network model, reducing the redundancy of the network structure by optimizing the network structure in the design stage which reduces parameters and calculations more directly and effectively with the same accuracy level.

In general, a complete CNN consists of convolution, pooling, nonlinear activation and batch regularization, etc. As a critical part of CNN, convolution extracts feature representations from input through a hierarchy architecture. But the huge calculations and parameters consumed by convolution is the main reason for the over-complicated network model. Therefore, the model complexity will be reduced by lightening the convolution operations. There are some successful efficient convolution operations, such as depthwise separable convolution [9] and group convolution [10]. The drawback of depthwise separable convolution is that it will cost huge calculations when the channel dimension changes a lot. While group convolution divide channel dimension into groups, which obstruct the information transmission between channels. The channel dimension is as important as the spatial dimension when it represents feature information. Besides, the number of channels can be scaled more

flexibly, so it is easier to control the feature richness and network complexity in channel dimension. Generally, channel transformation is realized by pointwise convolution accompanied by huge calculations and parameters. Therefore, a more efficient local channel transformation method including Depthwise Channel Ascent (DCA) and Group Channel Descent (GCD) is used to replace pointwise convolution. In addition, the novel channel transformation method is applied with the inverted residual linear bottleneck block to design an efficient network in this report.

2. RELATED WORK

2.1. Pointwise convolution

Pointwise convolution firstly proposed by [11] as a universal function approximator for feature extraction on the local patches. Pointwise convolution can increase nonlinear characteristic while maintaining the resolution of input features. Furthermore, it is easy to transform channel with transmitting the information across channels. Sequentially, pointwise convolution is widely used in networks, such as Inceptions[12], [13], MobileNets [9], [14] and so on. However, there is a drawback when pointwise convolution is used to transform channel when designing an efficient network. Pointwise convolution will cost huge calculations and parameters which results it is not completely suitable to an efficient network. For example, when the input and output channel numbers are 64 and 128, the width and height of the feature map are 32, pointwise convolution needs more than 8M FLoating point Operations (FLOPs) and 8k parameters.

2.2. Inverted residual liner bottleneck block

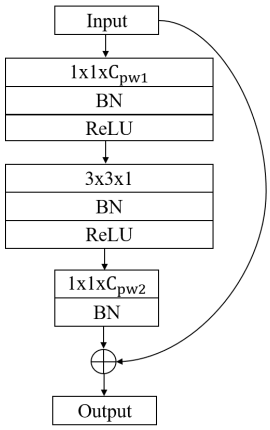

Figure 1: The structure of inverted residual linear and bottleneck block.

Inverted residual linear bottleneck block is proposed by [14] in MobileNetV2, which inherits the method of depthwise separable convolution to process spatial information and channel information separately. The structure of inverted residual linear bottleneck block is shown in Fig. 1. C_{PW1} and C_{PW2} are the output channel number of feature maps after pointwise convolution. In the inverted residual linear bottleneck block, the pointwise convolution is used for channel expansion and channel reduction to extract rich features and reduce network complexity respectively.

3. METHODOLOGY

3.1. Depthwise channel ascent

Channel transformation includes channel expansion and channel reduction. In channel expansion, pointwise convolution only considers the spatial features. While Depthwise Channel Ascent (DCA) used to replace pointwise convolution divide the channel expansion into two parts, which not only extracts spatial features but also channel features. The operation of DCA is shown in Fig. 2. In DCA, the pointwise convolution in DCA expands the channel from Cin to E1Cin where E1 is the first expansion coefficient. Then the multiple independent filters in the same channel extract spatial features from the same feature map to further expand the channel. When utilizing spatial features to increase the channel dimension, the implementation of depthwise convolution is referenced and the number of the filter in one channel is changed from 1 to E2 which is the second expansion coefficient. For convenience, it is also denoted as depthwise convolution.

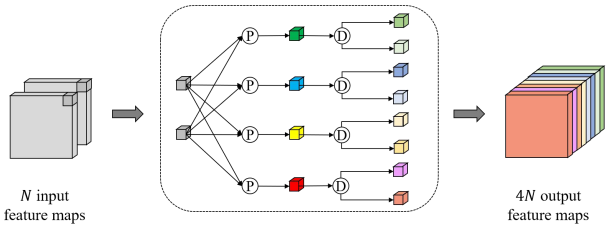

Figure 2: The operation of DCA with expansion coefficients {E_1=2, E_2=2}. Ⓟ represents the pointwise convolution. Ⓓ represents the depthwise convolution with the kernel size of 1x1.

3.2. Group channel descent

Pointwise convolution performs global linear transformation between channel in channel reduction. For efficient channel reduction, Group Channel Descent (GCD) firstly reduce the channel feature redundancy in local channel by the operation of channel compression, then a small-scale pointwise convolution is added after channel compression to transmit the information between channel. In the operation of channel compression, the feature maps are firstly divided into some groups, then the pixels at the same spatial position are summed channel-by-channel. The operation of GCD is shown in Fig. 3. The channel dimension is reduced from Cin to Cin/C1, then to Cin/(C1C2). C1 and C2 are the compression coefficients of channel compression and pointwise convolution in GCD.

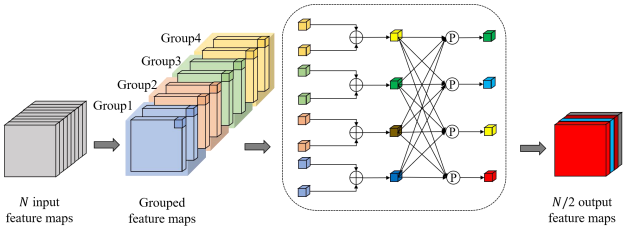

Figure 3: The operation of GCD when compression coefficients {C_1=2, C_2=1}.

3.3. Improved bottleneck block

The inverted residual linear bottleneck block is an efficient convolutional block which uses depthwise convolution to extract spatial features and pointwise convolutions to extract channel feature. However, channel transformation realized by the pointwise convolution is accompanied by huge calculations and parameters. Therefore, the pointwise convolutions in the inverted residual linear bottleneck are replaced with DCA and GCD to design a more efficient network. The structure of the improved bottleneck block with identity mapping is shown in Fig. 4. The coefficient of channel transformation is set to 6, which is consistent with [14]. When the input channel number is not equal to output channel number or the stride is not 1, there will be a DCA operation in the bypass. Note that both the original inverted residual linear bottleneck block and DCA have depthwise convolution, so only one depthwise convolution is reserved in the improved bottleneck block. The kernel size of depthwise convolution is set to 3x3.

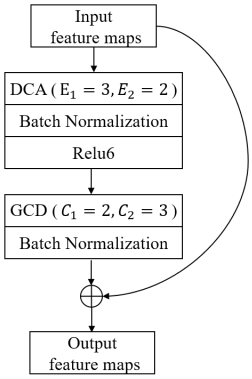

Figure 4: The structure of the improved bottleneck block with identity mapping.

3.4. Network architecture

The CNN constructed by improved bottleneck blocks with DCA and GCD (CNN-BDG) can be divided into four parts, the standard convolution block with channel number of 16, the improved bottleneck blocks with input channel number of 16, the improved bottleneck blocks with input channel number of 32, and the fully connected layer. The improved bottleneck block is repeated 2n times totally. The network architecture with n of 3 is shown in Table1. Besides, in the fully connected layer, dropout operation with drop-rate of 0.3 is used. The operations of batch normalization and Relu6 are used after standard convolution and depthwise convolution.

Table 1: The architecture of CNN-BDG with n of 3.

Operation type	Input size
Standard Conv 3x3x16	128x512x1
Improved bottleneck block1_1	64x256x16
Improved bottleneck block1_2	32x128x16
Improved bottleneck block1_3	16x64x16
Improved bottleneck block2_1	16x64x16
Improved bottleneck block2_2	8x32x32
Improved bottleneck block2_3	4x16x32
Global average pooling	4x16x32
Fully connection	1x1x32

3.5. Feature extraction

In the data preprocessing stage, the raw waves with binaural channels are first downmixed to mono. In the extraction process of the Mel spectrum, a Hamming window with a length of 1876 points (corresponding to 40ms) is used, the folding rate is 50%, and the number of Mel filter bank is set to 128 (corresponding to 128 Mel bands). The data format input to CNN is 128x512 single-channel Mel spectrum. The feature fed into the network is normalized by z-score.

3.6. Data augmentation

The raw waves in the same class are randomly mixed to generate new sound samples, which is consistent with [15]. After mixed, every raw wave generates a new raw wave. The number of train sample in the development dataset is increased from 9185 to 18370.

4. EXPERIMENTS AND RESULTS

4.1. Setup

All networks are trained 200 epochs by minimizing the cross-entropy loss with Adam optimizer. The initial learning rate is set to 0.001 and every 50 epochs decays to the original 0.2. The batch size is set to 16 in training. Model performance after each epoch is evaluated on the validation set, and best performing model is selected according to the validation set.

4.2. Results

In CNN-BDG, the number of the improved bottleneck block is selected as 6 and 10, which means n equals to 3 and 5 respectively. The results of baseline and CNN-BDGs are shown in Table2. It can be found that the parameters of CNN-BDGs is significantly fewer than the baseline model. CNN-BDG (n=3) requires about 0.3 times as many parameters as the baseline model, while the accuracy is 4.46% higher than the baseline model. The confusion matrix of CNN-BDG (n=3) is shown in Fig. 5. It is worth noted that the accuracies of CNN-BDG (n=5), CNN-BDG (n=7) and CNN-BDG (n=9) are lower than CNN-BDG (n=3) while the parameters of them are more than CNN-BDG (n=3). It may be because that more trainable parameters results in overfitting instead of performance improvement.

Table 2: The results of networks.

Model	Parameters	Size (KB)	Accuracy (%)
Baseline	115,219	450.1	87.3
CNN-BDG (n=3)	35,059	136.9	91.76
CNN-BDG (n=5)	60403	235.9	90.59
CNN-BDG (n=7)	85747	334.9	90.63
CNN-BDG (n=9)	111091	433.9	90.20

5. CONCLUSION

For low complexity solution for the ASC task, a kind of lightweight CNN is designed with efficient convolutions. The inverted residual linear bottleneck block is selected as the basic convolution block in the designed CNN-BDG. Besides, the pointwise convolutions are replaced with the operations of DCA and GCD to transform channel efficiently. CNN-BDGs realize better performance and requires fewer parameters compared to the baseline model.

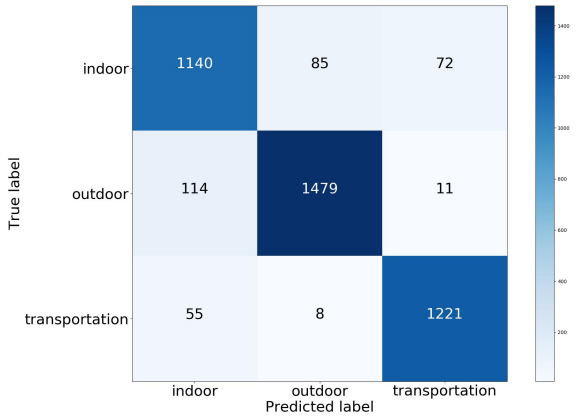

Figure 5: The confusion matrix of CNN-BDG with n of 3.

6. REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), Nov. 2018, pp. 9–13,[Online].Available: https://arxiv.org/abs/1807.09840.http://www.ieee.org/web/publications/rights/copyrightmain.html

[2] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2721–2725, 2017.C. D. Jones, A. B. Smith, and E. F. Roberts, "A sample paper in conference proceedings," in *Proc. IEEE ICASSP*, 2003, vol. II, pp. 803-806.

[3] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-End Environmental Sound Classification using a 1D Convolutional Neural Network," Expert Syst. Appl., vol. 136, pp. 252–263, 2019.

[4] O. Mariotti, M. Cord, and O. Schwander, "EXPLORING DEEP VISION MODELS FOR ACOUSTIC SCENE CLASSIFICATION," 2018.

[5] M. Liu, W. Wang, and Y. Li, "THE SYSTEM FOR ACOUSTIC SCENE CLASSIFICATION USING RESNET," 2019.

[6] R. Yu et al., "NISP: Pruning Networks Using Neuron Importance Score Propagation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9194–9203, 2018.

[7] B. Jacob et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2704–2713, 2017.

[8] G. Yang, "Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation," ArXiv, vol. abs/1902.04760, 2019.

[9] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," ArXiv, vol. abs/1704.04861, 2017.

[10] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6848–6856, 2018.

[11] M. Lin, Q. Chen, and S. Yan, "Network In Network," CoRR, vol. abs/1312.4400, 2014.

[12] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9, 2015.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[14] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520, 2018.

[15] https://github.com/iver56/audiomentations.