DCASE 2022 Submission: Low-Complexity Model Based on Depthwise Separable CNN for Acoustic Scene Classification

Technical Report

Yiqiang Cai[1], He Tang[1], Chenyang Zhu[2], Shengchen Li[1], and Xi Shao[3]

[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China, {yiqiang.cai21, he.tang19}@student.xjtlu.edu.cn, shengchen.li@xjtlu.edu.cn
[2]School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China, chenyangzhu2018@163.com
[3]College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China, shaoxi@njupt.edu.cn

Abstract

The task1 of DCASE 2022 put forward higher requirements for system complexity and the new datasets also brought greater challenges. We tried to reproduce several models in previous years, but did not get a good performance. Therefore, we introduced the depthwise separable CNN method to the baseline architecture, which successfully reduces the complexity and improves the accuracy. We also used three methods of data augmentation, mixup, pitch shifting and stretching to further improve the results.

Index terms — Low-complexity acoustic scene classification, data augmentation, depthwise separable convolution, bottleneck, channel shuffle

1 Introduction

The DCASE Challenge held every year is a competitive and top-ranked data challenge in acoustic signal processing society. The task1 of this year is about acoustic scene classification (ASC) which aims to classify each input audio recording into a pre-given class of acoustic scenes, such as underground stations, street traffic or public squares. Acoustic scenes are commonly diffused with a large amount of mixed information like the sounds of people talking, car driving, noise etc. This makes accurate scene prediction difficult and also an interesting research problem. In this task, ASC systems must not only have good generalization and robustness, but also meet the requirement of low spatial complexity.

In the dataset for DCASE2022 Challenge task1, recordings of 10 different acoustic scenes from 12 cities were collected using four different devices, as well as partially synthesized data created from the original speech. Comparing with previous years, each segment from the dataset was split from 10 seconds to 1 second. It brought much greater difficulty to the task as the information of each segment was reduced to 1/10 as before, which means that the classifier needs to make predictions based on less informative features. At the same time, the requirements of complexity are stricter this year, including the limit of parameters to 128k and the limit of computation to 30MMAC.

We tried to reproduce several models in previous years, but leading to a worse performance than baseline system. Therefore, we decided to configure the baseline and make further improvement on it, and our goal is to reduce the complexity while maintaining the accuracy.

2 Data Processing

2.1 Dataset

The TAU Urban Acoustic Scene 2022 Mobile development dataset contains 230,350 samples [1]. It is worth noting that the 2022 audio files are one second long, and therefore ten times longer than the previous data files. Each sample corresponds to one of the ten classes and no sample has more than one label. This dataset includes a variety of audio samples collected from three real and six analogue devices. The main recording device was from a Zoom F8 recorder with binaural microphones, and data from the Samsung Galaxy S7 and iPhone SE were also included. The simulated devices were synthesized by processing the device

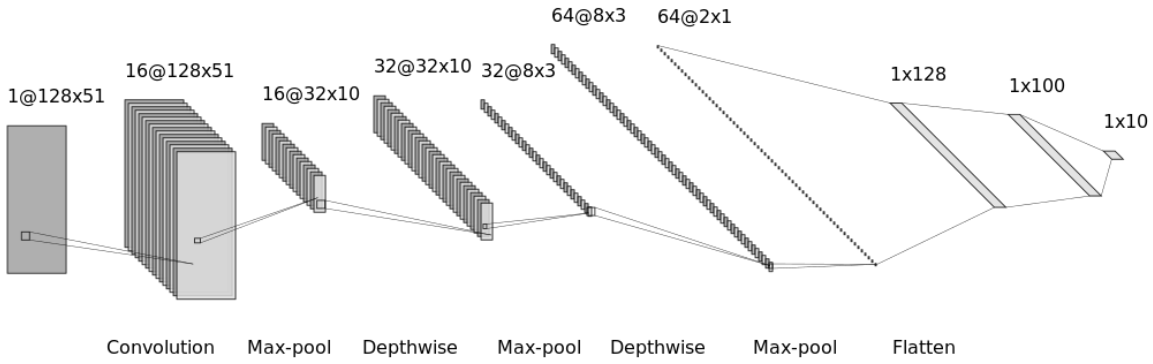

Figure 1: **Feature Maps**

data. The challenge organizers provided the basic metadata for the training/test split, with 139,970 samples in the training set and 2,9680 samples in the test set. the evaluation dataset for the TAU Urban Acoustic Scene 2021 Mobile contains 65,533 samples and also includes audio data from the GoPro Hero5 Session and five Audio data from new devices such as analogue devices

2.2 Feature Extraction

All audio segments were formatted mono, 44khz sample rate, 24 bit resolution per sample. For each 1 second input segment, 2048 FFT points were performed for every 1024 samples and the power spectrum was extracted. It means that the number of bins for a power spectrum is 51. The log-Mel filter bank features of 128 frequency bins were then extracted and the mean and variance normalised for each frequency bin. As a result, an input feature has a shape of 128 x 51 x 1.

2.3 Data Augmentation

Due to the limited complexity of the model, we believe that data expansion is an important way to increase the generalization of the system[2][3]. We operate on the data as follows:

1. Pitch shift and speed change: For each training audio recording, we randomly change their pitch and speed.
2. Mix audios: Inspired by[3] , we randomly mix two audio recordings from the same acoustic scene, with the goal of simulating more devices, smoothing the transition among devices, and reducing the variance among devices.
3. Spectrum correction: The spectrum of the reference device are obtained by averaging the spectrum of all the training devices except device A. Then use the spectrum of the reference device to correct the spectrum of device A.

3 Model Architecture

The model architecture is based on the baseline system and finalized through step-by-step modification and debugging. As shown in Figure 1 and Table 1, the model contains four components. The input component utilized a 7x7 convolutional layer followed by batchnorm and ReLu to expand the number of channels, and includes a max-pool layer. Then it is followed by two identical DW components, each DW contains 1 or 2 Depthwise Block and a max-pool layer. Finally, the output component is consistent with the baseline, which contains a flatten layer and two linear layers. To accommodate the domain diversity of acoustic scene inputs, the model uses a max-pool layer to narrow the feature map instead of dilation [4].

	Architecture	Input Shape
Input	ConvolutionBlock	128x51x1
	MaxPool2d(4, 5)	128x51x16
DW1	DepthwiseBlock	32x10x16
	(DepthwiseBlock)	32x10x16
	MaxPool2d(4, 3)	32x10x32
DW2	DepthwiseBlock	8x3x32
	(DepthwiseBlock)	8x3x32
	MaxPool2d(4, 3)	8x3x64
Output	Flatten	2x1x64
	LinearBlock	128
	Output	10

Table 1: **Architecture**

Model	Data Aug	Shuffle	MACS/M	Params/K	Log_loss	Accuracy%
Baseline	-	-	29.24	46.51	1.575	42.9
DepthwiseBlock(2)	N	N	7.31	35.93	1.617	45.2
DepthwiseBlock(2)	N	Y	7.31	35.93	1.710	44.2
DepthwiseBlock(2)	Y	N	7.31	35.93	1.410	48.4
Performance	-	-	**-75%**	**-23%**	**-10%**	**+13%**
DepthwiseBlock(1)	N	N	6.29	25.53	1.551	45.6
DepthwiseBlock(1)	N	Y	6.29	25.53	1.578	46.2
DepthwiseBlock(1)	Y	N	6.29	25.53	1.350	49.5
Performance	-	-	**-79%**	**-45%**	**-14%**	**+15%**

Table 2: **Performance**

3.1 Bottleneck

Bottleneck [5][6][7] has been widely utilized in a number of networks and obtain a good performance, which is a 1x1 group convolutional layer put at the first position. The groups of bottleneck equals to the input channels. As shown in Figure 2, the Depthwise Block in our model consists of a bottleneck and a Depthwise Convolution. The bottleneck also plays a role in the expansion of the amount of channels.

3.2 Depthwise Convolution

As the Figure 2 shows, Depthwise Block [8] [7] [9] utilizes the depthwise separable convolution. It consists of two layers, a 7x7 Depthwise convolutional layer followed by a batchnorm [10] and a non-linear activation function SiLu [11], and a 1x1 Pointwise convolutional layer followed by a batchnorm and a non-linear activation function ReLu. The Depthwise convolutional layer is actually a multi-channel group convolution [12], which means that the feature map is evenly divided to the number of the channels, and passed to the same number of convolutional layers, then the outputs are concatenated. Group convolution can effectively reduce the number of Mult-Adds to meet the requirements of this year's challenge. The Pointwise convolutional layer is a 1x1 convolutional layer served as a fully connected layer, which has the most parameters of the Depthwise Block [7].

3.3 Channel Shuffle

After the group convolution of bottleneck, the feature map of different channels are seperated and have no relations with each other. It obviously reduces the computation to a large extend, but results in the information loss between channels. The channel shuffle method [9] tries to build relationships between channels by reconstruct the information structure. It is optional in the Depthwise Block.

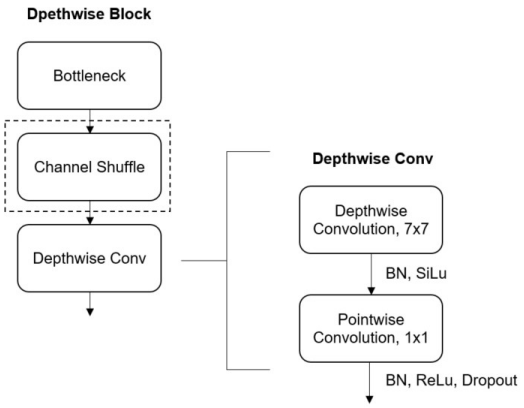

Figure 2: **Depthwise Block**

4 Result

Training and testing was carried out using the splits provided by the official development dataset. We trained the model for 200 epochs by using Adam optimizer with a learning rate of 0.001, batch size to 32. We applied two strategies for the DW components, with 1 Depthwise Block or 2 Depthwise Blocks in it. And for each strategy, data augmentation and channel shuffle used or not were the sub-strategies, which led to final results. The table 2 shows the performance of the model and baseline. Our model got better accuracy and loss as the baseline while achieved much less complexity.

5 Conclusion

In this technical report, we describe our low-complexity model for the acoustic scene classification task. The log-mel filter bank features were extracted from dataset and applied a series of data augmentations. Depthwise separable CNN, bot-

tleneck and channel shuffle methods successfully reduced the complexity of our model while improving the performance. In further works, we will keep trying to enhance the accuracy and decrease the complexity to achieve the best balance.

References

[1] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions." 2020. [Online]. Available: http://login.ez.xjtlu.edu.cn/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.2005.14623&site=eds-live&scope=site

[2] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C.-H. Lee, "Device-robust acoustic scene classification based on two-stage categorization and data augmentation." 2020. [Online]. Available: http://login.ez.xjtlu.edu.cn/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.2007.08389&site=eds-live&scope=site

[3] S. Seo and J.-H. Kim, "Mobilenet using coordinate attention and fusions for low-complexity acoustic scene classification with multiple devices," DCASE2021 Challenge, Tech. Rep., June 2021.

[4] D. Battaglino, L. Lepauloux, and N. Evans, "Acoustic scene classification using convolutional neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[5] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., June 2021.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pp. 770 – 778, 2016. [Online]. Available: http://login.ez.xjtlu.edu.cn/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.7780459&site=eds-live&scope=site

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications." 2017. [Online]. Available: http://login.ez.xjtlu.edu.cn/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1704.04861&site=eds-live&scope=site

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions." 2016. [Online]. Available: http://login.ez.xjtlu.edu.cn/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1610.02357&site=eds-live&scope=site

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," 2018, pp. 6848–6856.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," vol. 37, pp. 448–456, 07–09 Jul 2015. [Online]. Available: https://proceedings.mlr.press/v37/ioffe15.html

[11] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *CoRR*, vol. abs/1710.05941, 2017. [Online]. Available: http://arxiv.org/abs/1710.05941

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, p. 84–90, may 2017. [Online]. Available: https://doi.org/10.1145/3065386