ACOUSTIC SCENE CLASSIFICATION USING LIGHTWEIGHT RESNET WITH ATTENTION

Technical Report

Wenchang Cao, Yanxiong Li, Qisheng Huang

School of Electronic and Information Engineering,
South China University of Technology, Guangzhou, China,
wenchangcao98@163.com, eeyxli@scut.edu.cn, 839508665@qq.com

ABSTRACT

This technical report describes our system for the subtask A (Low-Complexity Acoustic Scene Classification with Multiple Devices) of Task1 (Acoustic Scene Classification) of the DCASE2021 Challenge. Due to the limited space-complexity of the model, we choose ResNet with depthwise separable convolution as our backbone network, and introduce the attention mechanism to the network. In addition, some data augmentation techniques, such as Mixup, Spectrum correction, are adopted for expanding the diversities of dataset. Our system achieves the accuracy rate of 72.4% on the development dataset, and the model size meets the requirement of subtask A.

Index Terms— Acoustic scene classification, convolution neural network, lightweight ResNet

1. INTRODUCTION

Acoustic scene classification (ASC) is a task to classify each input audio recording into one class of pre-given acoustic scenes. In the dataset of Task 1 of DCASE2021 challenge [1], audio recordings are recorded with three different recording devices in 12 different cities. A relatively small number of audio recordings of the dataset are synthesized from the original audio recordings. In this report, we introduce our work on the subtask: Low-Complexity Acoustic Scene Classification with Multiple Devices [2]. In this subtask, the ASC system should not only have good generalization ability and robustness, but also meet the requirements of low space-complexity.

We use the depthwise separable convolution [3] as the basic module to build the convolutional neural network, and introduce the attention mechanism to our network. In addition, we use several data augmentation techniques to mitigate the problem of generalization capability of our system when it is evaluated on the audio recordings acquired by different recording devices.

2. EXPERIMNET SETUP

2.1. Acoustic Feature

The development dataset includes a training subset and an evaluation subset. The training and evaluation subsets consist of 13962 and 2968 audio recordings, respectively. The file information of all audio recordings are as follows: 44.1 kHz sampling rate, 2 bytes per sample, and mono channel. In addition, to calculate the log-mel spectrogram feature, we use 2048 fast Fourier transform points with 1024 hop-lengths. We extract the power spectrum using the LibROSA library and use the log-mel scale. Consequently, we obtain a log-mel spectrogram with 128 frequency bins. In addition, the delta and delta-delta coefficient of the log-mel spectrogram are calculated and added to the channel. To obtain the same temporal size, we apply padding to the delta and delta-delta coefficient. Therefore, the size of log-mel spectrogram is: 128 × 423 × 3, where 128, 423 and 3 stand for the numbers of frequency, time and channel, respectively.

2.2. Data Augmentations

Due to the limited space-complexity of the model, we believe that data augmentation is an important way to improve the generalization ability of the system. We manipulate the data as follows:
1) Mixup: According to [4], mixup is an effective way for performance improvement and is easy to be implemented. We set the coefficient of mixup to 0.4 and randomly mix the data of two adjacent batches and their corresponding labels.
2) Spectrum correction: It is proposed in [5] and demonstrates moderate device adaptation properties. However, the method of spectrum correction in this work needs some adjustment before it can be applied to ASC. The spectrum correction in this work aims to transform the given input into a spectrum with an ideal device as a reference. We here employ spectrum correction as a data augmentation technique. Inspired by [6], we first average the spectrum of all devices except for the device A, which is one of ways to create a reference device spectrum. Then, we obtain additional data by correcting the spectrum of the device A.
3) Pitch shift and speed change: For each training audio recording, we randomly change their pitch and speed.
4) Mix audios: Inspired by [6], 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 differences among devices.

2.3. Model Design

As shown in Figure 1, our model is based on a lightweight residual network with depthwise separable convolution. The depthwise separable convolution module can greatly reduce the size of the network’s parameters, which is suitable for the model with low space-complexity. Inverted Residual Block is composed of several convolutional layers as shown in the Figure 2. After each convolutional layer, batch normalization (BN) layer is added to accelerate the performance of the model and the convergence speed. After the BN layer, we use the ReLU activation function. Considering the differences among log-mel energy, first-order difference, and second-order difference, we add channel attention mechanism into the model. In addition, we notice that different scenes might have noteworthy parts in frequency or time, so we also serialize spatial attention after channel attention, as done in [7]. The combination of channel attention and spatial attention module is called CASAM block. The CASAM block will be described in detail below.

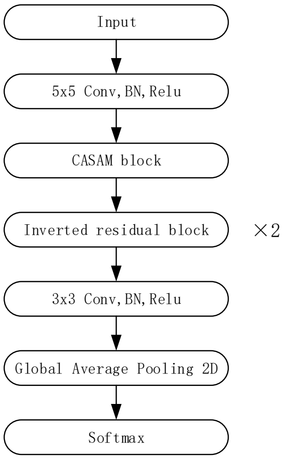

Figure 1. Proposed Model

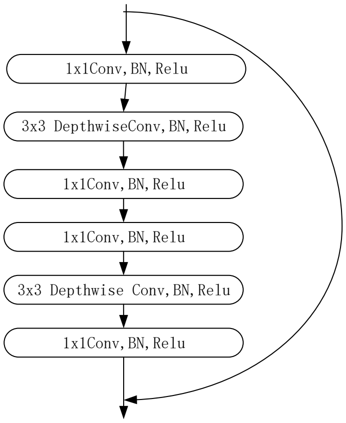

Figure 2. Inverted Residual Block

In Figure 3, the input is a feature map **F** with dimension of H×W×C. We perform global average pooling and maximum pooling for obtaining two channel descriptions with dimension of 1×1×C.

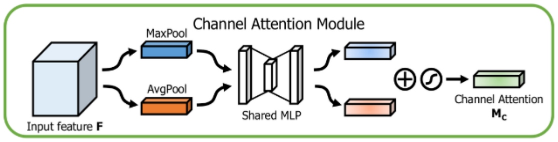

Figure 3. Channel Attention Module

Then, they are respectively fed into a two-layer neural network, i.e., multi-layer perceptron (MLP). The number of neurons in the first layer of the MLP is C/r, and the activation function is ReLU. The number of neurons in the second layer of the MLP is C. Note that this two-layer neural network is shared. Then, the weight coefficient **M**c is obtained by adding the two features after a Sigmoid activation function. Finally, the new-scaled feature can be obtained by multiplying the weight coefficient with the original feature **F**.

In Figure 4, the input feature map **F**' with dimension of H×W×C is fed into the spatial attention module for implementing average pooling and maximum pooling on one channel dimension to obtain two channel descriptions with dimension of H×W×1. These two channel-descriptions are concatenated together according to the channel. Finally, the concatenation of these two descriptions is fed into a 7×7 convolution layer and the activation function of Sigmoid for obtaining the weight coefficient **M**S. The new feature map after scaling is obtained by multiplying the weight coefficient **M**S with the feature map **F**'.

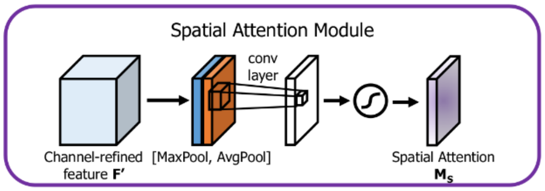

Figure 4. Spatial Attention Module

2.4. Train

All experiments in this work are conducted using Keras. The optimizer is the stochastic gradient descent. For the loss function, the categorical cross-entropy loss is used. All our models are trained for 600 epochs with a batch size of 16. In addition, the learning rate is set to 0.1, along with a decay factor of 1 × 0.00001. At epoch 2, 6, 14, 30, 126, 254, and 511, the learning rate is reset for obtaining the re-training effect, while different retraining from scratch can improve the training speed of the model and the training results remain consistent. We use the checkpoint with the highest validation accuracy as the best model.

3. QUANTIZATION FOR MODEL COMPRESSION

In subtask A, the limit of space-complexity for the model is 128 KB excluding zero parameters. That is, the model contains at most 32768 parameters when using float-point operation with 32 bits. In order to meet the requirement, we use a post-training quantization method, which is provided by Tensorflow2 [8], for compressing our model. Quantization not only reduces the model size but also improves hardware accelerator latency with little degradation in final classification accuracy. And it will not cause significant impact on the performance of the system.

4. RESULTS

The validation subset of the development dataset contains 2968 audio tracks, and contains new recording devices. We calculate the overall accuracy and evaluation indexes such as logloss on development dataset. We submitted four systems, system 1 and system 2 employ 5×5 convolution kernel. The training epochs of system 1 and system 2 are different. System 1 was trained with 600 epochs, while system 2 is trained with 800 epochs. System 3 and system 4 use the convolutional kernel size of 3×3, and both systems are trained with 600 epochs. At the same time, there are four inverted residual blocks in system 4. The specific results are shown in the Table 1. Table 2 presents the device-wise accuracy. Our largest model is system 4, whose number of non-zero parameters is 50238 and whose size after quantization compression is 102.9 KB.

Table 1. Class-wise accuracies obtained by the proposed method (in %).

Class	Baseline	Sys 1	Sys 2	Sys 3	Sys 4
airport	40.5	51.7	58.8	55.1	49.3
bus	47.1	80.8	81.1	82.5	82.2
metro	51.9	67.7	67.3	67.0	68.7
metro station	28.3	63.6	70.0	72.7	75.1
park	69.0	83.8	86.2	83.5	87.5
public square	25.3	61.6	52.9	56.6	62.0
shopping mall	61.3	86.2	66.7	79.1	79.1
street pedestrian	38.7	56.6	45.5	55.6	55.2
street traffic	62.0	86.5	90.6	85.2	84.5
tram	53.0	77.7	77.4	79.7	80.4
average	**47.7**	**71.6**	**69.6**	**71.7**	**72.4**

Table 2. Device-wise accuracies obtained by proposed method (in %).

Device	Sys 1	Sys 2	Sys 3	Sys 4
A	80.9	84.8	83.6	82.1
B	73.9	69.9	72.3	76.3
C	75.1	76.6	77.5	73.6
S1	68.2	66.7	72.4	71.5
S2	66.7	67.3	64.8	67.3
S3	72.4	73.0	73.0	73.6
S4	68.8	66.7	68.2	70.0
S5	72.1	63.9	69.7	72.1
S6	66.7	57.9	63.6	65.2

5. CONCLUSIONS

We proposed an ASC method using a lightweight residual network with both channel and spatial attentions. In addition, some data augmentation techniques were adopted for further improving the performance of the proposed method. The size of our model is 102.9 KB after model compression, which is lower than the size limit of 128 KB. Evaluated on the development dataset, classification accuracy of 72.4% was obtained by the proposed method.

6. REFERENCES

[1] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. *Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions.* In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020). 2020. Submitted. URL: https://arxiv.org/abs/2005.14623.

[2] Irene Martín-Morató, Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. *Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems.* 2021. arXiv:2105.13734.

[3] Sandler M, Howard A , Zhu M, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” arXiv preprint arXiv:1710.09412, 2017.

[5] T. Nguyen, F. Pernkopf, and M. Kosmider, “Acoustic scene classification for mismatched recording devices using heatedup softmax and spectrum correction,” in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 126–130.

[6] Hu, Hu and Yang, et.al (2020), “Device-Robust Acoustic Scene Classification Based on Two-Stage Categorization and Data Augmentation,” DCASE2020 Challenge, Tech. Rep.

[7] Woo S, Park J, Lee J Y, et al. CBAM: Convolutional Block Attention Module[J]. Springer, Cham, 2018.

[8] M. Abadi, A. Agarwal, P. Barham, et.al, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/