LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION USING BROADCASTED RESNET AND DATA AUGMENTATION

Technical Report

Wenchang Cao, Yanxiong Li, Qisheng Huang, Mingle Liu

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

ABSTRACT

This technical report describes our system for task 1 (Low-Complexity Acoustic Scene Classification with Multiple Devices) of the DCASE2022 Challenge. In our method, the BC-ResNet-Mod is used as the backbone network, and the Cross-Gradient training strategy is adopted for network training. In addition, some techniques of data augmentation are employed to enlarge the diversity of dataset, such as mix-up, spectrum correction, pitch shift and spectrum augmentation. Our system achieves the accuracy rate of 51.1% on the development dataset.

Index Terms— Acoustic scene classification, Convolution neural network, BC-ResNet

1. INTRODUCTION

Acoustic scene classification (ASC) is a task to classify each input audio recording into one class of pre-given acoustic scenes. As an important task in Detection and Classification of Acoustic Scenes and Events (DCASE), ASC has attracted a lot of attention from researchers in the community of audio and acoustic signal processing in recent years [1]-[4]. In the work of this report, we focus on the task of low-complexity ASC with multiple devices, namely, Task 1 of the DCASE2022 challenge [5]. In this task, a low-complexity model is required to classify audio recordings recorded by multiple devices (real and simulated).

In the proposed ASC method, the BC-ResNet-Mod [6] is used as the backbone of our model whose training strategy is the Cross-Gradient Training (CGT) [7]. In addition, some data augmentation techniques are adopted for further improving the performance of the proposed method. The size of our model is 125.33 KB after model compression, which is lower than the size limit of 128 KB. Evaluated on the development dataset, our system obtains classification accuracy of 51.1%.

2. DATA AUGMENTATIONS

Data augmentation is an effective way to improve the generalization ability of the ASC system. The techniques of data augmentation adopted in our system are described as follows.

2.1. Mix-up

Mix-up is an effective way for performance improvement and is easy to be implemented [8]. Mix-up makes the model behave linearly among different training samples. This linear modeling technique reduces the inadaptability of the model on training samples. Our practice is slightly different from original mix-up technique, but their overall idea is the same. We load two batches of audio recordings into memory and process them by $X' = \lambda X_i + (1-\lambda)X_j$, where X_i and X_j are two batches of audio recordings from the shuffled training data, and λ obeys the Beta distribution. Namely, λ ~ Beta (α, α) and α is set to 0.4 here.

2.2. Spectrum correction

Spectrum correction demonstrates moderate device adaptation properties [9]. However, in this work, it is adjusted before it is applied for ASC. The spectrum correction in this work aims at transforming the given input spectrum into a corrected spectrum using an ideal device (a reference device). The implementation of spectrum correction consists of two steps. First, we need a correction coefficient which can be obtained by calculating the average of *n* pairs of aligned spectra. The correction coefficient of device *A* to the reference device is the ratio of the frequency response of the reference device to the frequency response of device *A*. The frequency response of the reference device is the average of the frequency responses of multiple devices. In the second step, the corrected spectrum of device *A* can be obtained by multiplying the correction coefficient with the original spectrum of the audio recordings recorded by device *A*.

2.3. Pitch shift

Pitch shift is to resample the original audio recordings at different sampling frequencies with a certain step size. The pitches of audio recordings recorded by different devices are generally different to each other, which is helpful for ASC. Hence, pitch shifting is performed before extracting log-Mel spectrograms from audio recordings.

2.4. Spectrum augmentation

Spec augmentation modifies the spectrum by distorting the time-domain signal, masking the frequency-domain channel, and masking the time-domain channel. This enhancement method can be used to increase the robustness of the network to resist the deformation in the time domain and the loss of some segments in the frequency domain.

3. METHOD

3.1. Model architecture

3.1.1. BC-ResNet-Mod

The BC-ResNet-Mod is used as the backbone of our model, whose architecture is shown in Figure 1. The audio feature is fed into a 5×5 convolution layer with a 2×2 stride, and further transformed at several BC-Resblock [10] and max pooling layers. Before average pooling, a 1×1 convolution layer is adopted to increase the number of channels of the feature maps to 256. As a

result, the representation ability of the feature maps can be improved. Finally, the prediction vector is output by a fully-connected layer. In Figure 1, the numbers in each bracket (e.g., [40, 64, 22]) indicate the shape of the feature maps output by the module, which respectively represents the number of channels (e.g., 40 in [40, 64, 22]), length (e.g., 64 in [40, 64, 22]) and width (e.g., 22 in [40, 64, 22]) of the feature map from left to right. The last fully-connected layer outputs a 10-dimensional vector.

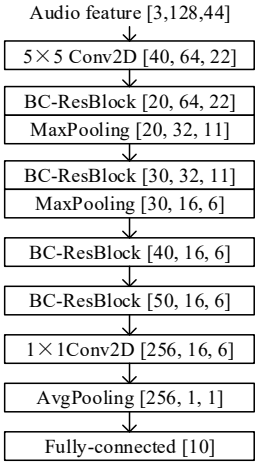

Figure 1. The architecture of the network backbone.

3.1.2. ResNorm

Since the experimental data come from different devices, there are obvious differences in the audio samples collected by different devices. The difference is more obvious on the frequency axis of the log-mel spectrum. The experiments in [11] show that the FreqIN (instance normalization by frequency) can effectively improve the robustness of the model to different devices. FreqIN is defined by

$$FreqIN(x) = \frac{x - \mu_{nf}}{\sqrt{\sigma_{nf}^2 + \epsilon}}, \tag{1}$$

where μ_{nf} and σ_{nf} denote mean and standard deviation of the input feature $x \in \mathbb{R}^{N \times C \times F \times T}$, respectively, and ϵ is a minimum number that prevents division by zero. μ_{nf} and σ_{nf} are defined by

$$\mu_{nf} = \frac{1}{CT} \sum_{c=1}^{C} \sum_{t=1}^{T} x_{ncft}, \tag{2}$$

$$\sigma_{nf}^2 = \frac{1}{CT} \sum_{c=1}^{C} \sum_{t=1}^{T} (x_{ncft} - \mu_{nf})^2, \tag{3}$$

where *N*, *C*, *F*, and *T* denote batch size, number of channels, frequency dimension, and time dimension, respectively.

Due to the introduction to the FreqIN, some information contained in categories may be lost during the procedure of regularization. An identity of original feature maps is added to make the information more comprehensive, as given by

$$\mathrm{ResNorm}(x) = \lambda \cdot x + \mathrm{FreqIN}(x) \tag{4}$$

where λ denotes a constant coefficient.

3.1.3. Cross-gradient training

The CGT makes the model have better generalization ability by superimposing the disturbance related to recording device's information on the data and expanding the recording device's information in the training set. The CGT parallelly trains a scene and a device classifier on examples perturbed by loss gradients of each other's objectives. The loss value in the device classification part represents the device information. Therefore, the gradient of the loss value with respect to the original input data is the mapping of the device information to the category space. After the gradient is superimposed on the original input, the corresponding device information will change.

3.2. Model compression

Two techniques are used to implement model compression, including quantization aware training and knowledge distillation.

3.2.1. Quantization aware training

The computational complexity is measured in terms of parameter count and MMACs (million multiply-accumulate operations). Maximum number of parameters is 128K, and the used variable type is fixed into INT8, counting all parameters (including the zero-valued ones). Maximum number of MMACs per inference: 30 MMACs (million MACs). Considering of the parameter limitation required by the organizer of DCASE 2022, we utilize quantization aware training to obtain a model with better performance. Quantization aware training is to insert a fake quant module into the model to simulate the rounding and clamping operations of the quantization model in the reasoning process. As a result, the adaptability of the model to the quantization effect can be improved, and higher accuracy of the quantization model can be obtained in the training process. In this process, all calculations (including model forward and back propagation calculation and pseudo quantization node calculation) are realized by floating-point calculation, and are quantized to the real INT8 model after the training is completed.

3.2.2. Knowledge distillation

Knowledge distillation can be used to generate a small model using the supervision information of a large model with better performance. A pretrained model [12] is used as the teacher model, and is then finetuned on the DCASE data. Afterwards, the trained teacher model is used to guide the training of the small model.

4. EXPERIMENTS

4.1. Experimental setup

4.1.1. Datasets

The development dataset consists of training subset and validation subset. The development dataset contains audio recordings from 10 cities and 9 devices: 3 real devices (A, B, C) and 6 simulated devices (S1-S6). Audio recordings recorded by devices B, C, and S1-S6 are composed of audio segments that are randomly selected from simultaneous recordings. Hence, all of these audio recordings overlap with the audio recordings from device A, but not necessarily with each other. The total amount of audio recordings in the development dataset is 64 hours. Some devices appear only in the validation subset.

The file information of all audio recordings are as follows: 44.1 kHz sampling rate, and mono channel. Audio recordings are first divided into frames via a Hamming window whose length is 2048 with 50% overlapping. Short-time Fourier transform is then performed on each frame for obtaining linear power spectrum which is finally smoothed with a bank of triangular filters for producing log-Mel spectrogram. The center frequencies of these triangular filters are uniformly spaced on the Mel-scale. In addition, to enhance the discriminative ability of audio feature, the delta and

delta-delta coefficient of the log-Mel spectrogram are calculated. Then, they are stacked along channel axis and used as the input feature of the model. The final size of input features is: 128 × 423 × 3, where 128, 423 and 3 represent numbers of frequency-band, frame and channel, respectively.

4.1.2. Implementation details

All experiments in this work are conducted using the toolkit of PyTorch. The optimizer is the stochastic gradient descent, and the categorical cross-entropy loss is used. All models are trained 200 epochs with a batch size of 512. In addition, the learning rate is set to 0.1, along with a decay factor of 1e-5. At the epoch of 4, 8, 16, 32, 64, and 128, the learning rate is reset for obtaining the re-training effect. 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.

4.2. Experimental results

The validation set for the development dataset contains 29680 audio clips, and there are new devices. We calculate the overall accuracy and evaluation indexes, such as log-loss on development dataset. Table 1 shows the class-wise accuracy and log-loss obtained by our model. Table 2 presents the device-wise log-loss. Figure 2 gives the confusion matrix of our method on validation set. It can be seen that *metro*, *public square* and *street pedestrian* are easily confused to each other. The number of parameters of our model is 125.33 K. The size of our model after quantization compression is 125.33 KB, and the number of MACS is 8.637 M.

Table 1. The class-wise accuracies of proposed method

Class	Baseline %)		Proposed %)	
	Acc（%）	Log-loss	Acc（%）	Log-loss
airport	39.4	1.534	49.3	1.365
bus	29.3	1.758	60.5	1.192
metro	47.9	1.382	40.2	1.642
metro station	36.0	1.672	41.5	1.600
park	58.9	1.448	71.1	1.129
public square	20.8	2.265	31.0	1.909
shopping mall	51.4	1.385	52.3	1.339
street pedestrian	30.1	1.822	36.9	1.915
street traffic	70.6	1.025	74.2	0.888
tram	44.6	1.462	53.5	1.426
average	42.9	1.575	51.1	1.441

Table 2. The device-wise accuracies of proposed method

Device	Baseline	Proposed
A	1.109	1.204
B	1.439	1.528
C	1.374	1.297
S1	1.621	1.500
S2	1.559	1.438
S3	1.531	1.435
S4	1.813	1.445
S5	1.800	1.598
S6	1.931	1.521

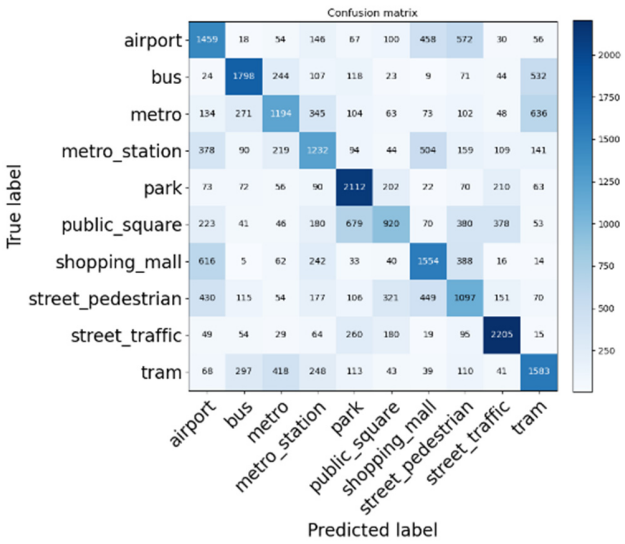

Figure 2. Confusion matrix of our method on validation set.

5. CONCLUSIONS

We proposed an ASC method using a BC-ResNet-Mod as the backbone network and adopting the strategy of CGT to training the network for alleviating device mismatch. In addition, some data augmentation techniques were adopted for further improving the performance of the proposed method. The size of our model is 125.33 KB after model compression, which is lower than the size limit of 128 KB. Evaluated on the development dataset, classification accuracy of 51.1% was obtained by the proposed method.

6. REFERENCES

[1] W. Xie, Q. He, Z. Yu and Y. Li, “Deep mutual attention network for acoustic scene classification,” *Digital Signal Processing*, vol. 123, 103450, 2022.

[2] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, “Low-complexity acoustic scene classification for multi-device audio: analysis of DCASE 2021 challenge systems,” *arXiv:2105.13734*, 2021.

[3] H.K. Chon, Y. Li, W. Cao, Q. Huang, W. Xie, W. Pang, J. Wang, “Acoustic scene classification using aggregation of two-scale deep embeddings,” in *Proc. of IEEE ICCT*, 2021, vol. 4, pp. 1341-1345.

[4] Y. Li, X. Li, Y. Zhang, W. Wang, M. Liu, and X. Feng, “Acoustic scene classification using deep audio feature and BLSTM network,” in *Proc. of ICALIP*, 2018, pp. 371-374.

[5] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, “Low-complexity acoustic scene classification in DCASE 2022 challenge,” 2022, URL: https://arxiv.org/abs/2206.03835.

[6] B. Kim, S. Chang, J. Lee, and D. Sung, “Broadcasted residual learning for efficient keyword spotting,” in *Proc. of INTERSPEECH*, 2021, pp. 4538-4542.

[7] S. Shankar, V. Piratla, S. Chakrabarti, et al., “Generalizing across domains via cross-gradient training,” *arXiv:1804.10745*, 2018.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz,

"mixup: Beyond empirical risk minimization," arXiv preprint *arXiv:1710.09412*, 2017.

[9] T. Nguyen, F. Pernkopf, and M. Kosmider, "Acoustic scene classification for mismatched recording devices using heatedup softmax and spectrum correction," in *Proc. of IEEE ICASSP*, 2020, pp. 126-130.

[10] 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.

[11] Y. Xu, Q. Kong, W. Wang and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," in *Proc. of IEEE ICASSP*, 2018, pp. 121-125.