THUEE SUBMISSION FOR DCASE 2020 CHALLENGE TASK1A

Technical Report

Xinxin Ma[1,2], *Yunfei Shao*[1], *Yong Ma*[2], *Wei-Qiang Zhang*[1]

1. Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
2. School of Physics and Electronic Engineering, Jiangsu Normal University, Xuzhou 221116, China

mxxjsnu@163.com, shaoyf@mail.tsinghua.edu.cn, may@jsnu.edu.cn, wqzhang@tsinghua.edu.cn

ABSTRACT

In this report, we described our submission for the task1a of Detection and Classification of Acoustic Scenes and Events (DACSE) 2020 Challenge: Acoustic Scene Classification with Multiple Devices. Our methods are mainly based on two types of deep learning models: ResNet and Mini-SegNet. In our submissions, we designed two classification systems. Firstly, we applied spectrum correction to combat mismatched frequency responses, and further proposed in log-mel domain. Then these features are fed to ResNet or Mini-SegNet models for feature learning. In order to prevent overfitting, we adopted mixup augmentation, ImageDataGenrator and temporal crop augmentation for data augmentation. Besides, we tried an ensemble of multiple subsystems to enhance the generalization capability of our system. In our work, our final system achieved an average of 75.02% on different devices in the Development dataset.

Index Terms—DCASE2020, acoustic scene classification, log-mel spectrogram, ResNet, Mini-SegNet

1. INTRODUCTION

Sounds carry a great deal of information about our environments, from individual physical events to sound scenes as a whole. The problem of sensing and understanding the environment in which a sound is known as Acoustic Scene Classification (ASC) [1]. It is a multi-class classification task recognizing the recorded environment sounds specific acoustic scenes that characterize either the location or situation such as park, metro station, tram, etc. ASC has been applied to smartphones, tablets, robots, and cars for customized services. For example, if a car "hears" children yelling from behind a corner, it can slow down to avoid a possible accident. A smartphone could automatically change its ringtone to be most appropriate for a romantic dinner, or an evening in a noisy pub.

In recent years, the research on acoustic scene classification tasks has become more and more diversified. Initialized in 2013 [2], the DCASE challenge has been successfully held by the audio and acoustic signal processing (AASP) technical committee. As one of the substantial tasks, acoustic scene classification has been extensively practiced in every challenge. DCASE 2018 and 2019 proposed the mismatch in different recording devices A, B, C and D. In 2020 [3], the task of acoustic scene classification has been divided into 2 subtasks. Among them, the subtask A works on the dataset collected with mismatched recording devices. This task has its own dedicated dataset called "TAU Urban Acoustic Scenes 2020 Mobile". The goal is to create a model capable of predicting acoustic scenes using audio recording from low quality devices and simulated devices. Additionally, the dataset contains a fair amount of examples from a high quality devices (referred to as A), but only a limited number from the targeted low quality devices (referred to as B and C) and simulated devices (referred to as S1-S6). A gap in amount and quality of the recorded data causes overfitting on data of devices A. Especially, a part of the evaluation set is a compressed version of recorded audio data from device D and simulated devices S7-S11. This brings ASC closer to real-world conditions, but also presents a huge challenge.

This year's task1A is more challenging than previous years, because the audio data is not only come from real recording equipment, but also simulated with a variety of devices. To deal with challenge, we learn from the work of Michal Kosmider [4] et al, try to apply spectrum correction to adjust the varying frequency response of the recording devices. Correction is applied to the short-time Fourier transform (STFT) of the audio recording, which means that audio can be inverted back to waveform after the correction has been applied or directly transformed into spectrogram. This can improve the classification accuracy to certain extent, which we found in our work.

In this report, we describe two systems for task1A in the DCASE 2020 Challenge. These systems consist of two important stages. Firstly, mono audio signals are converted to time-frequency representations, scaled by spectrum correction, and zero mean and unit variance normalization. Secondly, the log-mel feature are fed to ResNet or Mini-SegNet models for feature learning. The output layer includes a dense layer of *C* classes and a softmax for classification. Meanwhile, ensemble methods are applied to combine several features and CNN settings to enhance the generalization capability of our work. We did not use any additional data to that provided by the challenge organizers.

The rest of the paper is organized as follows. Section 2 presents the proposed ASC systems, including audio preprocessing

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219. The corresponding author is Wei-Qiang Zhang.

and spectrum correction, the two convolutional neural networks, and data augmentation. Section 3 provides experiments and the performance of the proposed approach. Finally, conclusion is provided in Section 4.

2. METHODS

This section introduces the applied audio preprocessing methods. It also describes the details utilized process flow and ConvNet architecture.

2.1. Audio preprocessing and Spectrum Correction

The spectrum correction proposed in [4] scales the frequency response of the recording devices. Spectrum correction is implemented in two steps. First, the correction coefficients are computed from the spectrum of *n* aligned pairs of recordings. All records are then transformed using the calculated coefficients. In view of our experimental comparison, we only use 750 samples of data from each device A, B, C to determine the reference spectrum and the coefficients of each device. The spectrum coefficients are expressed as vectors, i.e. one coefficient per frequency bin. We use the corresponding coefficient to scale the spectrum bin of each device. The correction is applied by multiplying the Short Time Fourier Transform of the signal by the correction coefficients on the frequency axis of each time point.

The sampling rate is 44.1 kHz. The audio segments are 10s in length. The STFT use a Hanning window. The window size and hop size are 2048 and 1024 samples, respectively, and the HTK formula to define the mel scale [5]. Our implementation used python, and the Librosa library [6].

After spectrum correction, 128 Mel filters are used to further present the spectrum in the log-mle domain. Then, zero mean and unit variance normalization is applied to the log-mel feature. Therefore, we extract the log-mel energy of 128 frequency bins and 431 temporal frames per segment.

2.2. Neural Network

2.2.1. ResNet--Splitting of high and low frequencies

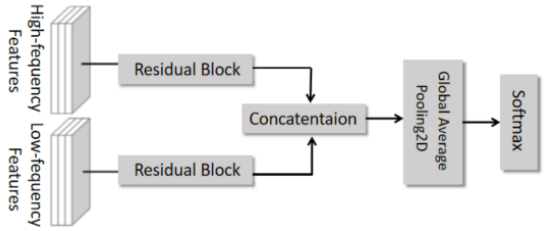

Figure 1: The architecture of ResNet. It consists of two pathways in the residual network. Each pathway is made up of several residual block. After these stacks, each pathway is connected. Then, followed by global average pooling layer, and softmax.

After spectrum correction, the spectrogram is further proposed in log-mel domain using 128 mel filters. In this part, we learn from the work of Mark D. Mc Donnell [7], think that the frequency features to be learned for high frequencies are likely to different to those for low frequencies. So, just like their work, we also try to divided 128 dimensional features into two dimensions, 0 to 63 and 64 to 127 dimensions. Then two pathways are used in the residual network: one is high frequency and the other is low frequency. Before network output, these two pathways just fuse two convolutional layers.

The architecture of ResNet is illustrated in Figure 1. The whole network input has 128 frequency dimensions, but these dimensions are immediately divided in the network, so that the dimensions 0 to 63 are processed by another residual block with 17 convolutions and dimensions 64 to 127. All kernel in these paths are 3 × 3. After these stacks, each channel connection forms 128 frequency dimensions, which are then operated by two 1 × 1 convolutional layers. The second layer is reduced to the number of ten acoustic scene categories. Subsequently, followed by a batch normalization layer, a global average pooling layer, and softmax.

2.2.2. Mini-SegNet--Semantic segmentation

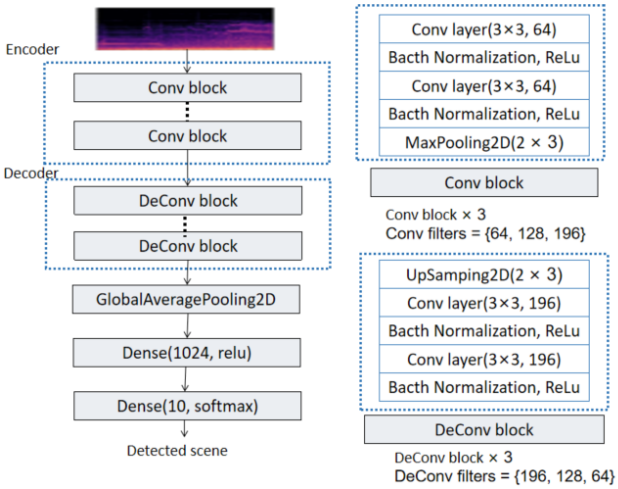

Figure 2: The architecture of Mini-SegNet. It consists of encoder and decoder module. Each module consists of two convolutional layers whose convolution kernel is 3 × 3. Differently, the encoder module is down-sampled by the max-pooling, and the decoder module use up-sampling to restore sampling information.

We think that the acoustic scene is composed of some basic units (acoustic events), just as language governs the syntax of phonemes and words. These acoustic events contain some semantic information, which has a certain internal relationship with the discrimination of acoustic scene. Therefore, we designed encoder-decoder network similar to SegNet [8] for image semantic segmentation, which we term Mini-SegNet.

The proposed network is illustrated in Figure 2. It is mainly composed of encoder and decoder module. In the encoder module, consists of three Conv block. Each block contains two Convolutional layers, followed by batch normalization, ReLU, and max-pooling. The decoder module is similar and consists of three DeConv block. Each block, up-samping is performed first, then followed by Convolutional layers, batch normalization, ReLU. Finally, global max pooling is applied, and two dense layers are utilized to output final predictions.

2.3. Data augmentation

Like many entries in previous DCASE challenges, we combined mixup [9], ImageDataGenrator and temporal crop augmentation.

In mixup, we randomly select a pair of samples from training data. Let x_1, x_2 be the features, and y_1, y_2 be the one-hot labels respectively, the data is mixed as follows:

$$x = \lambda x_1 + (1-\lambda)x_2 \quad (1)$$

$$y = \lambda y_1 + (1-\lambda)y_2 \quad (2)$$

where the parameter λ is a random variable with Beta distribution $B(0.4, 0.4)$.

ImageDataGenerator is mainly used in image classification, it is a kind of image generator. At the same time, it can also enhance the data in batch, expand the size of data set, and enhance the generalization ability of the model. In our work, it is implemented with width shift, height shift. We additionally used crop augmentation in the term poral axis: each of the two samples combined using mixup were first cropped independently and randomly from 431 dimensions down to 400.

3. EXPERIMENTSAFTER

3.1. Experiment setup

We used stochastic gradient descent, with a batch size of 32, momentum of 0.9, and the cross-entropy loss function. At the same time, we using a warm restart learning rate schedule, its maximum value of 0.1 after 2, 6, 14, 30, 126 and 254 epochs, and then decays according to a cosine pattern to 1×10^{-5}. In our work, each network has trained for 254 epochs. It is shown by [10] and verified by [11] that this approach this can provide improvements in accuracy on image classification relative to using steeped schedules.

3.2. Inference and Results

TAU Urban Acoustic Scenes 2020 Mobile Development dataset, it contains from 10 cities and 9 devices: 3 real devices (A, B, C) and 6 simulated devices (S1-S6). The dataset is provided with a training/test in which 70% of the data for each device is included for training, 30% for testing. Some devices (S4, S5, S6) appear only in the test subset. We report the performance of our system using this train/test setup in order to allow comparison of system on the development set.

The DCAS 2020 Task1A challenge is evaluated using accuracy calculated as the average of the class-wise accuracy, also known as "macro-average accuracy". Because the data sets come from different devices, and the train/test setup. We describe the results of our systems in device A (Dev. A), device B (Dev. B), device C (Dev. C), average S1-S3 (Ave. S1-S3), and average S4-S6 (Ave. S4-S6), etc. Instead table 1 shows results for a single neural network trained in various configurations using the official train-test split. To a certain spectrum correction can combat mismatched frequency responses, to improve the recognition accuracy. Compared with the DCASE 2020 task1A baseline system, our two networks structures are greatly improved.

Table 1: Accuracy on the development dataset for both architectures (for device A, B, C, average B&C, average S1-S3, and average S4-S6) with and without spectrum correction. For example, Resnet-no represents no spectrum correction, Resnet-co represents spectrum correction.

System	Average	Dev.A	Dev.B	Dev.C	Ave.BC	Ave.S1-S3	Ave.S4-S6
Baseline	54.1%	70.6%	60.6	62.6	61.6	53.33	44.33
Resnet-no	63.26%	74.84%	59.39%	67.87%	63.63%	62.32%	60.10%
Resnet-co	**68.92%**	**80.60%**	70.90%	77.27%	**74.09%**	**66.46%**	**64.04%**
Segnet-no	62.86%	76.06%	66.67%	**77.57%**	72.12%	61.51%	53.63%
Segnet-co	65.02%	74.84%	**72.12%**	74.24%	73.18%	64.24%	57.07%

3.3. Model ensemble and submissions

In our work, we have made more attempts on the above two basic networks. For example, try different kernel size, more acoustic features, and add attention mechanism, etc. After that, these works will be described in detail and analyzed further in a dedicated article.

In the final submission, we ensembled our various experimental schemes to further improve the system generalization ability. Model ensemble is successful in boosting the system's performance according to previous work. We ensemble our models using linear combination as follows:

$$y_{\text{ensemble}} = \sum_{n=1}^{N} w_n y_n + b \quad (3)$$

where N is the number of subsystems, y_n is the output score of each subsystem, w_n is the weight coefficient for each subsystem, and b is the bias.

The detailed accuracy after fusion is shown table 2, we submitted three prediction results using different weights:

1) Zhang_THUEE_task1a_1.output.csv: ensembled 11 subsystems, achieved 74.95% on the development dataset.

2) Zhang_THUEE_task1a_2.output.csv: achieved our highest average accuracy of 75.02% on different devices in the development dataset.

3) Zhang_THUEE_task1a_3.output.csv: ensembled 8 subsystems, achieved 74.34% on the development dataset.

Table 2: After ensembled all the kinds of subsystems, accuracy of different devices on the development dataset (for device A, B, C, average B&C, average S1-S3, and average S4-S6).

Ensemble	Average	Dev.A	Dev.B	Dev.C	Ave.BC	Ave.S1-S3	Ave.S4-S6
task1a_1	74.95%	82.12%	76.67%	81.82%	79.24%	74.34%	70.30%
task1a_2	**75.02%**	82.12%	76.67%	81.82%	79.24%	74.34%	70.51%
task1a_3	74.34%	81.82%	76.67%	81.52%	79.09%	74.34%	68.69%

4. CONCLUSION

In this report, we present our methods and techniques used in the task1A of DCASE 2020 challenge. We used spectrum correction to combat mismatched frequency responses. We applied two types of deep learning model including ResNet and Mini-SegNet. Besides, we adopted mixup, ImageDataGenrator and temporal crop augmentation for data augmentation. In our submission, we proposed ensembles of many CNN structures in order to enhance the classification accuracy of subtask 1A of DCASE2020 challenge. Combining different neural network further improves the results. While the best average accuracy of a single basic neural network on all devices is only about 68.92%. After the ensemble of our experimental scheme, our final best system achieved 75.02% on the development set.

5. REFERENCES

[1] Virtanen T, Plumbley M D, Ellis D, "Computational Analysis of Sound Scenes and Events || Approaches to Complex Sound Scene Analysis," 2018.
[2] http://dcase.community/
[3] http://dcase.community/challenge2020/task-acoustic-scene-classification
[4] Michal Kosmider, "Calibrating neural networks for secondary recording devices," Tech. Rep., DCASE2019 Challenge, June 2019.
[5] S. J. Yong, D. Kershaw, J. Odell, D. Ollason, V. Valtchev and P. Woodoland, The HTK Book version 3.4. Cambridge University Press, 2006.
[6] B. McFee, C. raffle, D. Liang, D. P. Ellis, et al., "librosa: Audio and music signal analysis in python." In processing of the 14th python in science conference, 2015, pp.18-25.
[7] Mark D, Mc Donnel, Wei Gao, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," Tech. Rep., DCASE2019 Challenge, June 2019.
[8] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017:1-1.
[9] H. Zahng, M. Cisse, Y. N. Dauphin, and D. Loped-paz, "mixup: beyond empirical risk minimization," arxiv preprint arxiv:1710.09412, 2017.
[10] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with restarts," CoRR, vol. abs/1608.03983, 2016. [online]. Available: http://arxiv.org/abs/1609.03983
[11] M. D. McDonnel, "Training wide residual networks for deployment using a single bit for each weight," 2018, in Proc. ICLR 2018; arxiv: 1802.08530.