MULTI-CHANNEL FEATURE USING INTER-CLASS AND INTER-DEVICE STANDARD DEVIATIONS FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Soonshin Seo, Changmin Kim*, and Ji-Hwan Kim†*

Sogang University, Dept. of Computer Science and Engineering, Seoul, Republic of Korea
{ssseo, kchangmin, kimjihwan}@sogang.ac.kr

ABSTRACT

In this technical report, we describe our acoustic scene classification methods submitted to detection and classification of acoustic scenes and events challenge 2020 task 1a. Our proposed methods aim to maximize the differences between acoustic scene classes and minimize the differences between various devices. We obtained the inter-class and inter-device standard deviations of the training data and applied them to the log-mel spectrogram features. These features are added to the channel of the original log-mel spectrogram. In addition, we applied class-wise random masking for the frequency domain with small standard deviations. Then, masked features are divided into quarters on the frequency axis. They are trained using four-pathway residual convolutional neural networks. Our proposed methods achieved an overall accuracy of 72.7% for the official development dataset, which was an improvement by 18.6% over the official baseline.

***Index Terms*—** Acoustic scene classification, standard deviation, multi-channel, masking, convolutional neural networks

1. INTRODUCTION

Task 1 of the 2020 detection and classification of acoustic scenes and events (DCASE) challenge pertains to acoustic scene classification. In particular, task 1a aims to classify 10 acoustic scenes under conditions recorded on various devices. In other words, it focuses on solving the generalization property of the classification model to minimize the heterogeneity between devices.

Classes of acoustic scenes are composed of airports, bus travels, underground metro travels, metro stations, urban parks, public squares, indoor shopping malls, pedestrian streets, streets with medium traffic level, and tram travels [1]. The classed of devices are composed of four real devices (A–D) and 11 simulated devices (S1–S11). The evaluation dataset is an open set in which new devices (D, S7–S11) that are not in the development dataset are added.

* Equal contribution
† Corresponding author
The code is available at `https://github.com/sunshines14/DCASE2020`

2. MOTIVATIONS

2.1. Previous approaches

To minimize heterogeneity between devices, various acoustic scene classification models have been proposed in the DCASE challenge [2–6]. Past entries used convolutional neural networks (CNNs), which mainly used to process audio signals as images. The audio signal is converted to acoustic features, such as the log-mel spectrogram and mel-frequency cepstral coefficients. The converted signal is two-dimensional feature vectors comprising temporal and frequency axes and are used as an input to the CNNs.

In particular, McDonnell *et al*. [3], who achieved the second highest performance in DCASE 2019 task 1b, proposed the following: 1) They used delta and delta-delta features for log-mel spectrogram 2) They divided the features separately into two features based on the frequency axis. 3) They trained the separated features in a two-pathway residual CNNs. These methods demonstrated relatively high performances for new devices not included in the training dataset. However, they demonstrated relatively low performances for scene classes such as public squares and pedestrian streets.

2.2. Proposed approaches

We aim to maximize interclass variability and minimize inter-device variability. Each of the 10 acoustic scene classes and multiple devices were patterned on the temporal and frequency axes, respectively [3, 7]. We used the standard deviation to determine the distribution of features on the frequencies of each class and device. In addition to the log-mel spectrogram, features reflecting class-wise and device-wise distributions were used for training.

Furthermore, we assumed that the frequency domain with a low standard deviation was unnecessary information that caused confusion in the training. Therefore, we used class-wise random masking for the low standard deviation domain. Specifically, we expect to improve the classification accuracy for confusing scene classes, such as public squares and pedestrian streets. In addition, based on the proposed methods of McDonnell *et al*., the parameters of the channel and residual pathway were changed.

3. PROPOSED METHODS

3.1. Feature preprocessing

The development dataset comprised a training dataset and an evaluation dataset. The training and evaluation datasets comprised 13,962 and 2,968 audio clips, respectively [1]. All audio clips featured a 44.1 kHz sampling rate, 2 bytes per sample, and a mono channel. In addition, to calculate the log-mel spectrogram feature, we used 2,048 fast Fourier transform points and 1,024 hop-lengths. We extracted the power spectrum using the LibROSA library and applied the log-mel scale.

Consequently, we obtained a log-mel spectrogram with 128 frequency bins and 423 frames. In addition, the delta and delta-delta features of the log-mel spectrogram were calculated and added to the channel. To obtain the same temporal size, we applied padding to the delta and delta-delta features. Therefore, the log-mel spectrogram measured 128 × 423 × 3.

3.2. Multi-channel feature

We calculated the class-wise averages for 13,962 training datasets and 128 frequency bin axes. Therefore, 128 average values were generated for each class. From the generated average value, the overall mean value for all classes on the 128 frequency axes was obtained. Next, the variance and standard deviation were obtained for 128 frequencies in order. The same process was applicable to the devices.

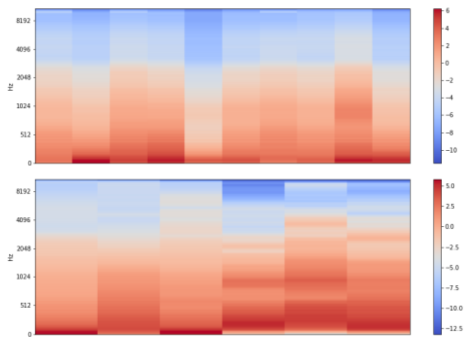
Figure 1 : Overall means of the spectrogram (The top represents class-wise domain, the bottom represents device-wise domain)

Figure 1 and Figure 2 show the spectrogram for which the overall mean and overall standard deviation of the class and device were obtained. We calculated the log-mel spectrogram by multiplying the overall standard deviation of the class and dividing the overall standard deviation of the device. Hence, for the log-mel spectrogram, two more features with class-wise and device-wise standard deviations applied were generated. By applying the same to the delta and delta–delta features, multi-channel features were constructed. Therefore, the modified log-mel spectrogram measured 128 × 423 × 9.

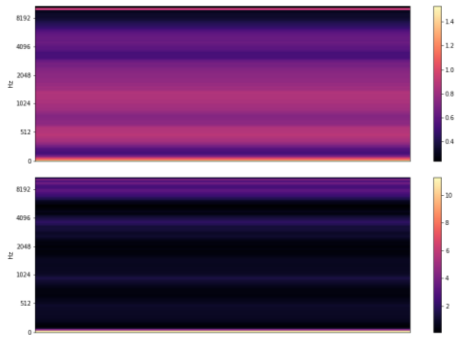
Figure 2 : Overall variances of the spectrogram (The top represents class-wise domain, the bottom represents device-wise domain)

Figure 3 shows the spectrograms of each channel of the multi-channel feature. Using these multi-channel feature, the differences between different classes in the training process is maximized, and the differences between the same devices are minimized.

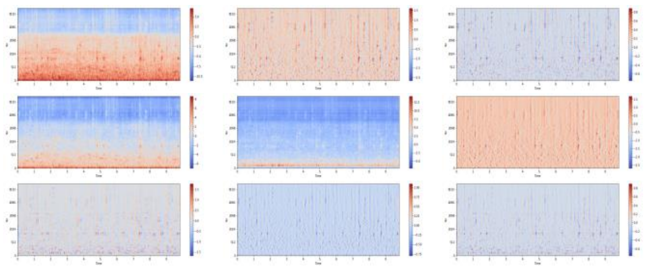
Figure 3: Multi-channel feature applied class-wise and device-wise standard deviations (row 1: log-mel spectrogram, delta feature, delta-delta feature, row 2: class-wise features, row 3: device-wise features)

3.3. Class-wise random masking

Frequency domains with low standard deviations are likely to confuse classes when training. Therefore, we considered these domains as unnecessary information and used the masking method. On the spectrogram depicting the overall standard deviation value, 64 frequency bins were selected in the order of low value. Then, shuffling is applied to the selected frequency bins. Finally, for five among 64 frequency bins, all temporal frame values were masked with 0. This process was applied to all channels. Consequently, multi-channel features using class-wise random masking were generated.

3.4. Four-pathway residual CNNs

Our model is based on the model proposed by McDonnell *et al*. We divided the multi-channel feature into four sections, i.e., the frequency bin sections of (0–32), (32–64), (64–96), and (96–128), which were used as the input of the model. Each input size featured 32 × 423 × 9. The divided inputs were trained using four residual layers. Each residual layer comprised two residual blocks, and each residual block comprised a batch normalization, nonlinear activation function, and convolutional layer [8]. Therefore, it was composed of 17 convolutional layers, including the first convolutional layer. The kernel size in all convolutional layers was 3 × 3. In addition, when down-sampling occurred between the residual layers, zero padding was applied [3]. The residual CNNs composed of four pathways independently trained each separated features.

Then, the divided feature were concatenated to 128 frequency dimensions. The concatenated features were fed into two residual layers. Finally, it was transformed into 10 dimensions using batch normalization and global average pooling. The 10 dimensions generated the output probability values for each acoustic scene class using the softmax function.

Furthermore, we changed various training parameters, such as the leaky ReLU and global max pooling. However, we confirmed that their performances were worse than those of previous parameters.

3.5. Additional training methods

Regularization methods were effective in preventing overfitting. For all convolutional layers, a weight decay value of 5 × 0.0001 was used. In addition, when generating training data, both mixup augmentation [9] and temporal crop augmentations [3] were applied. We did not use additional training datasets other than the official training dataset.

3.6. Training parameters

All experiments in this paper were conducted using Keras [10]. The optimizer used the stochastic gradient descent with a 0.9 momentum weight. For the loss function, the categorical cross-entropy loss was used. All our models were trained for 1,022 epochs with a batch size of 32. Considering the multi-channels and masking, an additional epoch was required for convergence. In addition, the learning rate was 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 was reset to obtain the re-training effect [11]. We used the checkpoint with the highest validation accuracy as the best model.

4. RESULTS AND SUBMISSION

We used the evaluation dataset of the official development dataset to evaluate the performance of the proposed model [1]. This dataset comprises 2,968 audio clips. In addition, it is an open set in which new devices (S4–S6) that are not in the training set are added. Using this dataset, the overall accuracy (based on recall) and log-loss (based on multiclass cross-entropy) were measured. As in Table 1, we tested the proposed methods by varying the number of channels, applying masking, and varying the number of model pathways. In addition, as in Tables 2 and Table 3, we tested class-wise and device-wise experiments on our proposed model, respectively.

Consequently, four single models (A–D) were selected for submission as follows: A) 5-channel-mask-2-pathway, B) 7-channel-mask-2-pathway, C) 7-channel-mask-4-pathway, and D) 9-channel. The overall accuracy of models A, B, C, and D was 72.7%, 71.0%, 72.2%, and 71.7%, respectively. The overall accuracy of the official baseline model [12] for the same evaluation dataset was 54.1%. Therefore, our best performance showed an absolute improvement of approximately 18.6% over the official baseline.

Table 1 : The performances of proposed models

Model	Channels	Masking	Pathway	Acc.	Loss
A	5	Y	2	**72.7**	1.307
B	7	Y	2	71.0	1.301
C	7	Y	4	72.2	1.408
D	9	N	2	71.7	**1.292**

Table 2 : The class-wise accuracies of proposed models

Class / Model	A	B	C	D
airport	59.1	52.7	58.1	**59.5**
bus	82.2	**82.5**	81.5	80.5
metro	**78.8**	75.4	78.5	72.4
metro station	75.1	73.4	**76.4**	73.7
park	**92.3**	91.6	89.2	90.9
public square	51.2	**55.2**	53.5	47.5
shopping mall	71.4	71.0	**71.7**	71.0
street pedestrian	53.2	45.5	46.5	**56.2**
street traffic	89.2	89.2	**90.9**	88.9
tram	74.7	73.0	75.3	**76.0**

Table 3 : The device-wise accuracies of proposed models

Device / Model	A	B	C	D
A	78.5	79.4	**81.2**	79.7
B	**72.2**	71.1	72.1	72.1
C	**78.1**	77.2	76.0	76.6
S1	**73.0**	72.1	70.3	70.9
S2	**69.7**	64.8	67.9	65.8
S3	72.7	72.4	**74.8**	74.5
S4	**73.0**	67.3	71.5	68.8
S5	69.4	67.6	**70.9**	70.0
S6	**66.1**	65.2	63.9	66.1

5. CONCLUSION

We propose acoustic scene classification models for DCASE 2020 task 1a. We calculated the class-wise and device-wise standard deviations and generated multi-channel features. The generated features was applied with class-wise random masking for the low standard deviation domain. In addition, the residual CNNs was used by separating the frequency axis into four parts. Our proposed method showed an overall accuracy of 72.7% for the official development data, which improved by 18.6% over the official baseline.

6. ACKNOWLEDGMENT

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (1711103127, Development of Human Enhancement Technology for auditory and muscle support).

7. REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proc. DCASE 2018 Workshop*, 2018, 9-13.

[2] M. Kosmider, “Calibrating neural networks for secondary recording devices,” in *Proc. DCASE 2019 Workshop*, 2019, pp. 25-26.

[3] M. D. McDonnell, and W. Gao, “Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths,” in *Proc. IEEE ICASSP*, 2020, pp. 141-145.

[4] L. Pham, T. Doan, D. Ngo, H. Hong, and H. H. Kha, “CDNN-CRNN joined model for acoustic scene classification, in *DCASE 2019 technical reports*, *2019.*

[5] K. Koutini, H. Eghbal-zadeh, and G. Widmer, “CP-JKU submissions to DCASE 19: Acoustic scene classification and audio tagging with receptive-field-regularized CNNs,” in *DCASE 2019 Workshop*, 2019.

[6] T. Nguyen, and F. Pernkopf, “Acoustic scene classification with mismatched recording devices using mixture of experts layer,” in *IEEE ICME*, 2019, pp. 1666-1671.

[7] S. Perez-Castanos, J. Naranjo-Alcazar, P. Zuccarello, M. Cobos, and F. J. Ferri, “CNN depth analysis with different channel inputs for acoustic scene classification,” *arXiv preprint arXiv:1906.04591*, 2019.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE CVPR*, 2016, pp. 770-778.

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

[10] F. Chollet, “Keras: Deep learning for humans,” *https://github.com/keras-team/keras*, 2020.

[11] I. Loshchilov, and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.

[12] J. Cramer, H. H. Wu, J. Salamon, and J. P. Bello, “Look, listen and learn more: Design choices for deep audio embeddings,” in *IEEE ICASSP*, 2019, 3852–3856.