ACOUSTIC SCENE CLASSIFICATION USING VARIOUS PRE-PROCESSED FEATURES AND CONVOLUTIONAL NEURAL NETWORKS

Technical Report

Hyeji Seo, Jihwan Park, and Yongjin Park

Advanced Robotics Research Lab, LG Electronics, Seoul, Korea
{hyeji.seo, jihwan.park, yongjinn.park}@lge.com

ABSTRACT

In this technical report, we describe our acoustic scene classification algorithm submitted in DCASE 2019 Task 1a. We focus on various pre-processed features to categorize the class of acoustic scenes using only stereo microphone input signal. In the front-end system, the pre-processed and spatial information are extracted from the stereo microphone input. Residual network, subspectral network, and conventional convolutional neural network (CNN) are used for back-end systems. Finally, we ensemble all of the models to take advantage of each algorithm. By using proposed systems, we achieved a classification accuracy of 80.4%, which is 17.9% over than the baseline system.

Index Terms— DCASE 2019 challenge, acoustic scene classification, convolutional neural network, ensemble

1. INTRODUCTION

Recently, most of the people carry their own smart devices like mobile phone, smartwatch, and tablet PCs. Those smart devices have one or more microphones to hear users voice. However, those microphones can hear not only the human voice but also environmental sounds. Thus, when we analyze recorded sounds of smart devices, which can automatically recognize acoustic scenes and events, even if users do not listen to the surrounding sound. Acoustic scenes and event detection and classification can be helpful in the situation when visual information is hard to be achieved. Also, it can help people with hearing impairment by giving them proper acoustic information.

Unfortunately, it is difficult to detect scenes and events, because the field of sounds is sensitive to environments. Various sounds can be transformed by changing time and frequency domain properties in nature. Even though, as microphones have various specifications, sounds from various microphones show plenty of different properties. To overcome these problems, deep learning technologies are applied in various ways. Yuma Sakashita proposed an ensemble of spectrograms based on adaptive temporal divisions [1], and Matthias Dorfer *et al.* suggested fully convolutional neural networks (CNNs) with I-vectors [2]. They took top ranks on DCASE2018 Task 1a with their proposed deep learning technologies.

In DCASE 2019, Task 1a aims to classify acoustic scenes recorded in various locations. All data are recorded with the same devices to guarantee the same specification of microphones. However, the training set and test set is divided by recording locations from each city. With this database, we extract various features to achieve the best performance. The first feature is log mel spectrogram from mono, harmonic, and percussive signals. The second one is low-frequency spectrograms, and the third feature is log mel spectrogram from dereverberated signal. We also use the nearest neighboring filters, the generalized cross-correlation - phase transform (GCC-PHAT), which is well-used in deep beamforming and interaural time difference (ITD) as feature vectors. The last features are log mel features of average and subtracted signals from 2 channels. With those features, we build 3 layers CNNs and subspectral network to get 8 models of 8 combinations of features and networks. By averaging 8 models, we can get the final ensemble model. The following sections are organized to explain the proposed deep learning system.

2. PROPOSED SYSTEM

In this section, we introduce our acoustic scene classification (ASC) system. The system consists of front-end systems, back-end systems, and data augmentation. In the front-end systems, robust features for ASC are extracted from the unprocessed and pre-processed stereo microphone input signal. Then, in the back-end systems, the proposed neural network model predicts a proper class of acoustic scenes. The proposed ASC system is depicted in Fig. 1.

2.1. Front-end Systems

In this section, we will describe the pre-processed features and spatial information. These are extracted from the stereo microphone input signal by using harmonic-percussive sound separation (HPSS), dereverberation, low pass filtering, etc. The pre-processed features are known helpful to reduce the mismatch between training and test set. Also, spatial information can be used to classify acoustic scenes which have a point sound source or not.

First, we perform dereverberation and denoising to overcome the mismatch between the trained model and unseen data. Reverberation is caused by the reflection of sound waves. Since each of acoustic scenes has different reverberation property, reverberation is an import clue for ASC. We use log mel spectrogram of mono, harmonic and percussive component of the dereverberated signal for ASC. For dereverberation, we use NARA-WPE, a python package for weighted prediction error dereverberation [3]. Also, we perform the nearest neighboring filtering for denoising. In DCASE 2018 challenge, the nearest neighbor filters were applied for ASC to emphasize and smooth similar patterns of sound events in a scene [4]. We apply both non-local median filter and means filter to chromagram features, which closely relates to the twelve pitches that can

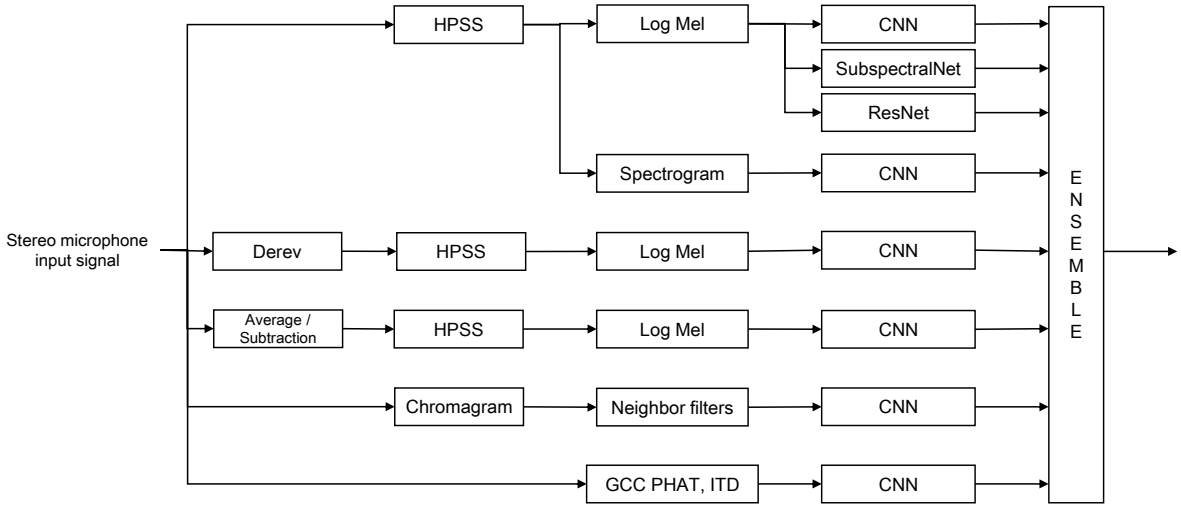

Figure 1: The block diagram of the proposed ASC system

capture harmonic and melodic characteristics. Chromagram, non-local median filtered and means filtered chromagram are used for input features.

Second, we perform HPSS, which is known effective for ASC. In DCASE 2017, Han *et al.* reported that HPSS has shown greater performance than original signal [5]. Many acoustic sounds are mixtures of harmonic and percussive signals. Harmonic signals are sparse in frequency and percussive signals contain localized information. Therefore, by decomposing the original signal to the harmonic and percussive component, we can get an import factor for classifying acoustic scenes. After separating a harmonic and percussive component from signals, we extract log mel spectrogram to extract feature vectors.

Third, we use spatial information to improve ASC performance by using stereo microphone input characteristics. To use binaural characteristics, we use inter-channel information of the left and the right channel [6]. The average and subtraction of 2 channels are an important clue for assuming the dominant component of the scene. It was shown that binaural information could help to classify acoustic scenes in DCASE 2017 challenge [5]. We use log mel spectrogram of the inter-channel information as feature vectors to improve classification accuracy. In [7], beamforming weights are predicted by deep neural networks with GCC-PHAT features and shown satisfactory results. Since GCC allows neural networks to predict the direction of arrival (DOA) reliably, it can also be helpful for predicting acoustic scenes reliably. We use the center 128 elements of GCC-PHAT values for feature vectors. Besides, ITD for each frame is concatenated with GCC-PHAT and used for input feature vectors.

Lastly, lower bands of the spectrogram are used to classify scenes. After examining the spectrogram of each of the scenes, we found that most of the import components for ASC exist below 3000 Hz. So we use lower bands of the spectrogram as feature vectors to examine specifically below 3000 Hz. Examining lower spectrogram in detail helps to classify scenes correctly.

2.2. Back-end Systems

With those front-end systems, we build 3 types of CNNs to train various pre-processed feature vectors. Conventional CNN, subspectralnet, and residual network are used for back-end systems. First, we use conventional 3 layers CNN, which is described in Table 1, to train features like images. In DCASE 2018, CNNs were used for ASC and events detection and has shown more expectional performance than convolutional recurrent neural network (CRNN) and other structures. Batch normalization is also applied to improve the performance and stability of the system. Second, we apply subspectralnet, sub-spectrograms based CNN architecture for ASC, which is published in ICASSP 2019 [8]. To capture more enhanced features, subspectralnet divide the spectrogram into several sub-spectrograms. We divide 200 dims of log mel spectrogram with 30 sub-spectogram sizes and 10 mel bin hop size. Then, each sub-spectrogram is feedforward to 3 layers CNNs and concatenated to a global classifier to determine the scene. Each sub-spectrogram is classified using their specific sub-band information and the global classifier determines the scene by discerning information at the global level. The structure of sub-networks is almost the same as the conventional CNN, which has 3 convolutional layers, described in Table 1. The global classifier consists of 2 dense layers which have 1024 and 10 units. Third, we use the residual network to train high-level features effectively [9]. Residual learning enables training deep CNN by introducing shortcut connections $F(x) + x$, which mean their outputs are added to the outputs of the stacked layers. We use for ASC has 8 convolutional layers and two dense layers of the residual network. All convolutional layers have the kernel size of (3,3) and filter sizes are set to (32, 32, 64, 64, 128, 128). Dense layers are the same as the conventional CNN.

2.3. Data Augmentation

To improve performance in unknown condition, data augmentation is important to increase the generality of the system. We perform

Table 1: The architecture of conventional CNN

Layer	Description
Conv2D	kernel size = (7, 7), # of filters = 32
BatchNormalization	batch size = 128
Activation	ReLU
Maxpool2D	pool size = (5, 5)
Conv2D	kernel size = (7, 7), # of filters = 64
BatchNormalization	batch size = 128
Activation	ReLU
Maxpool2D	pool size = (2, 10)
Conv2D	kernel size = (7, 7), # of filters = 128
BatchNormalization	batch size = 128
Activation	ReLU
Maxpool2D	pool size = (2, 2)
Dense	# of units = 1024, activation = ReLU
Dense	# of units = 10, activation = Softmax

mixup for data augmentation [10]. Since multiple sounds occur simultaneously in a real-world environment, the mixup augmentation scheme is appropriate for classifying the real-world sound scenes. Mixup augmented data is obtained as follows:

$$\widehat{x} = \lambda x_1 + (1 - \lambda)x_2, \quad (1)$$

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

where (x_1, y_1) and (x_2, y_2) are two acoustic scenes randomly chosen from the training data and $\lambda \in (0, 1)$. λ is acquired from the beta distribution $\beta(0.1, 0.9)$. One-hot labels are added with λ, labels of augmented data are represented as multi events labels.

3. EXPERIMENTS AND RESULTS

All audio samples were set to 48kHz sampling rate and 24-bit resolution. We performed two kinds of normalization methods, peak normalization, and standard normalization. Peak normalization was performed before the feature extraction to adjust the signal based on the highest sound level. After the feature extraction, features were normalized using standardization, which makes features have zero-mean and unit variance. Features are extracted with frame size 40 ms and hop size 20 ms in submission 1 and 2 and frame size 40 ms and hop size 15 ms with submission 3 and 4. Log mel spectrogram features are extracted with a bin size of 128 and 200 according to the models they are forwarded to. Our submission systems were trained with all development data set and among the training data, 20% of the data is augmented data. We used python deep learning library keras to train and test CNNs [11]. ADAM optimizer was used with an initial learning rate of 0.001 and a mini-batch size of 64. Training epoch was first set to 200, but if validation accuracy does not increase over 30 epochs, then we stopped the training. After all the training was finished, we chose the model which has the highest accuracy among all epochs. Table 1 and 2 show the result of our algorithms.

4. SUBMISSION

We used mean probabilities to ensemble the multiple models. We submitted 4 results for DCASE 2019 Task 1a. Submission 1 is an ensemble of 8 systems, which described in Table 2. Submission 2 is an ensemble of 21 systems. In contrast with submission 1, all features are trained separately. For example, mono, harmonic and percussive components of log mel spectrogram are trained separately, and we obtain 3 models. Also, there are gcc-phat model and ITD model separately, which was one inter-channel features model in submission 1. In this way, we train 21 models. Submission 3 and submission 1 are pretty much the same in algorithms, except submission 3 has hop size of 15 ms. Also, submission 4 and submission 2 are pretty much the same in algorithms, except submission 3 has a hop size of 15 ms.

Table 2: Accuracy for each system

Front-end system	Back-end system	Accuracy
Baseline		62.5
HPSS (Log-mel spec)	Conventional CNN	74.2
HPSS (Log-mel spec)	SubspectralNet	76.0
HPSS (Log-mel spec)	ResNet	71.9
Dereverbed & HPSS (Log-mel spec)	Conventional CNN	74.9
Low frequncy spectrogram	Conventional CNN	69.7
Nearest neighbor filters	Conventional CNN	44.8
Inter-channel feautres	Conventional CNN	72.1
GCC-PHAT & ITD	Conventional CNN	52.2
Ensemble (Proposed system)		80.4

Table 3: Class-wise accuracy for the development dataset

Class	Baseline	Proposed
Airport	48.4	79.5
Bus	62.3	88.6
Metro	65.1	78.6
Metro station	54.5	75.2
Park	83.1	92.0
Public square	40.7	68.2
Shopping mall	59.4	77.0
Street pedestrian	60.9	74.4
Street traffic	86.7	91.8
Tram	64.0	79.6
Average	62.5	80.4

5. CONCLUSION

In this paper, we proposed ensemble systems of convolutional neural networks with various pre-processed features for DCASE Task 1a. The systems consist of front-end systems and back-end systems. We ensembled those systems to improve performance and achieved a classification accuracy of 80.4%, which is 17.9% over than the baseline system.

6. REFERENCES

[1] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," in *DCASE2018 Challenge*, 2018.

[2] M. Dorfer *et al.*, "Acoustic Scene Classification with Fully Convolutional Neural Networks and I-Vectors," in *DCASE2018 Challenge*, 2018.

[3] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A Python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *ITG Fachtagung Sprachkommunikation*, 2018, pp. 1-5.

[4] T. Nguyen and F. Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *DCASE2018 Challenge*, 2018.

[5] Y. Han, J.Park and K. Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," in *DCASE2016 Challenge*, 2017.

[6] K. Tan, X. Zhang, and D. Wang, "Real-time Speech Enhancement Using an Efficient Convolutional Recurrent Network for Dual-microphone Mobile Phones in Close-talk Scenarios," in *Proc. IEEE ICASSP*, 2019, pp. 5751-5755.

[7] X. Xiao *et al.*, "Deep beamforming networks for multi-channel speech recognition," in *Proc. IEEE ICASSP*, 2016, pp. 5745-5749.

[8] S. S. R. Phaye, E. Benetos, and Y. Wang, "SubSpectralNet – Using Sub-spectrogram Based Convolutional Neural Networks for Acoustic Scene Classification," in *Proc. IEEE ICASSP*, 2019, pp. 825-829.

[9] K. He, X. Zhang, S. Ren, and J. Sun "Deep Residual Learning for Image Recognition," in arXiv:1512.03385, 2015.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in arXiv:1710.09412, 2017.

[11] F. Chollet *et al.*, "Keras," `http://keras.io`, 2015.