DNN BASED MULTI-LEVEL FEATURES ENSEMBLE FOR ACOUSTIC SCENE CLASSIFICATION

Jee-weon Jung, Hee-soo Heo*, Hye-jin Shim, and Ha-jin Yu†*

School of Computer Science, University of Seoul, South Korea

ABSTRACT

Acoustic scenes are defined by various characteristics such as long-term context or short-term event, making it difficult to select input features or pre-processing methods suitable for acoustic scene classification. In this paper, we propose an ensemble model which exploits various input features that vary in their degree of pre-processing: raw waveform without pre-processing, spectrogram, and i-vector a segment-level low dimensional representation. We tried to effectively perform combination of deep neural networks that handle different types of input features by using a separate scoring phase by using Gaussian models and support vector machines to extract scores from individual system that can be used as a confidence measure. Validity of the proposed framework is tested using the detection and classification of acoustic scenes and events 2018 dataset. The proposed framework showed accuracy of 73.82% using the validation set.

Index Terms— Acoustic scene classification, DNN, metric learning

1. INTRODUCTION

Acoustic scene classification (ASC) is a task of increasing demand with its applicability towards various machines and intelligent systems. Three noticeable features can be observed by analyzing the past editions of detection and classification of acoustic scenes and events (DCASE) competitions: (a) deep neural networks (DNNs) are mainly used with various architectures, (b) various features such as spectrogram, MFCC, CQCC are being used, and (c) ensemble of two or more classifiers are used with majority voting or score-sum.

Despite this active research, choosing an appropriate feature for ASC task remains a difficult problem. One of the main factors that make this problem more difficult may be the fact that the features that are appropriate for representing each scene in the ASC task can be different. For example, segment-level features such as an i-vector may be useful for classifying scenes where the characteristics appear over a long period of time. Frame-level features such as sepctrograms can be used to classify scenes where characteristics occur in a particular frequency band at short intervals. In addition, raw waveform without any pre-processing can be used when considering characteristics that are difficult to be represented by existing features. Therefore, in order to consider all the characteristics of various kinds of features, we trained DNNs which input each type of feature and combined results from multiple DNNs.

Another problem is that the two most used ensemble methods, majority voting and score-sum, both do not include confidence measures. Majority voting actually 'vote' classifiers' and score-sum uses a softmax activation of the output layer as confidence score which actually cannot represent confidence [1].

In this paper, we make the following contributions:

1. Exploit features that can be more useful for classifying different scenes.
2. Train Gaussian models and support vector machines to extract scores with confidence.

Specifically, three features are individually studied for ASC task. The first feature is raw waveform which is directly input to the DNN with pre-emphasis as its pre-processing. Another is spectrogram which is widely used for ASC task with convolutional neural networks (CNNs). The last is i-vector, a segment-level low dimensional representation, also known to suit ASC task. Gaussian mixture models (GMMs) and support vector machines (SVMs) are used as back-end classifiers to get a confidence score for each class given an embedding. The overall proposed framework is depicted in Figure 1.

The remainder of this paper is organized as follows. Section 2 describes the three systems with different features used in this study. Section 3 presents metric learning scheme with the proposed running mean and its variations. Experimental settings and system specifications are given in Section 4 with experimental results. Section 5 introduces relevant works and the paper is concluded in Section 6.

2. SYSTEM DESCRIPTION

In this section, we describe each system used for ensemble according to its input features.

2.1. Raw waveform based system

Recently, systems that show promising results with DNNs that directly input raw waveforms have been proposed in various tasks [2, 3]. Through visualization of raw waveform based DNN models, it has been shown that the kernels of 1-d convolutional layers detect specific frequency bands [4]. Many raw waveform systems aim to extract features that suit the objective defined by the loss function of DNN better than existing acoustic feature extracting techniques through extracting most useful frequency bands. In this work, we use the RWCNN model proposed by Jung et al. [3] with modifications. The used raw waveform system consists of convolutional blocks and fully connected layers: each convolutional block consists of 1-d convolutional layer followed by layer normalization, leaky rectified unit activation and max pooling. Modifications and

*These authors have equal contribution.

† Corresponding author.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(2017R1A2B4011609)

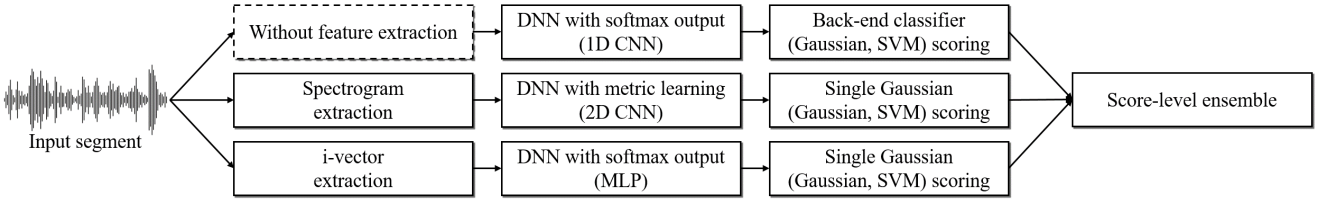

Figure 1: Illustration of the overall framework.

detailed description of the raw waveform based system is present in Section 3.3.

2.2. Spectrogram based system

Spectrogram is a widely used feature in speech signal processing systems such as speech recognition or speaker recognition. We expected that the detection of events occurring in a specific frequency band using the spectrogram will contribute to perform the ASC task. 2d convolutional neural network (CNN) was used to embed the spectrogram extracted from each segment so that it could be used for the metric learning. Cosine similarity between each sample and centroid of each class is used for the metric learning. We used max feature map (MFM) based architecture among various CNN variations [5]. In the MFM based architecture, max operation is applied to multiple featuremaps to calculate the output of each layer. In this case, we expected that the appropriate filter size will be searched while filters are trained to classify each scene, through competing between the filters of different sizes.

2.3. i-vector based system

In contrast to raw waveform or spectrogram, the i-vector (identity vector) is a low-dimensional representation of a given segment using factor analysis [6]. Thus, regardless of the length of a given segment, one vector with fixed dimensionality is extracted. The i-vector was originally proposed for speaker verification, but it has been shown in previous DCASE challenges to well perform in ASC task as well [7].

2.4. back-end scoring

support vector machine (SVM) with rbf kernel and sigmoid kernel, and single Gaussian model with diagonal and full covariance were used as back-end classifier. The classifiers were trained to classify the acoustic scene by using embeddings from DNNs. We expected that using back-end classifier for scoring instead of softmax output, may lead ensemble of multiple DNNs more efficient.

3. EXPERIMENTAL SETTINGS

Experiments in this paper utilize soundfile and scipy python modules for raw waveform and spectrogram extraction [8]. Keras deep learning toolkit [9] with tensorflow back-end [10, 11] was used for DNN training and decoding. The scikit-learn module was used for Gaussian model and SVM scoring [12].

3.1. Dataset

All experiments in this paper uses task 1-a among the DCASE 2018 dataset. Task 1-a of the DCASE 2018 dataset comprises 8,640 audio segments recorded in 48 k sampling rate, 24 bit resolution, stereo, and divided into length of 10 seconds. 4 fold cross-validation was conducted based on the provided meta data about the place of recordings. The development set and the validation set does not have audio segments from identical place. We only report the accuracy on the first fold in this paper.

3.2. Feature configurations

Stereo raw waveforms are used as input feature to the DNN with pre-emphasis resulting in feature shape of (48,000 * 10, 2).

Spectrograms were extracted by shifting 30ms window by 10ms. After extracting spectrogram represented by 721 coefficients on each window, only 300 coefficients of low frequency bands were used; we empirically confirmed that low frequency bands are more useful for ASC. Finally, a spectrogram of size 499×300 was extracted from each segment of 10 second.

The i-vectors are extracted from a diagonal GMM with 1024 components, trained with 60- dimensional MFCC features. A total variability matrix that can extract an 200-dimensional i-vector was trained for 10 iterations. Length normalization nor linear discriminant analysis were applied and kaldi toolkit [13] was used.

3.3. System configurations

Raw waveform based DNN uses the RACNN-LSTM model from Jung et al.'s work with a few modifications for ASC task [3]. Modifications include followings: stride size of the strided convolutional layer was changed to 12 for 48 kHz sampling rate, 256 kernels used for the last convolutional layer, and stereo audio input instead of mono.

Spectrograms based DNN comprises two fully connected layers following three MFM layers. Fully connected layers contain 256 nodes activated by leaky ReLU function. L2 normalization [14] was applied to the output of the last fully connected layer. The configuration of MFM based system is shown in Table 1. In each MFM layer, the output is calculated through the max operation between featuremaps generated by filters of different sizes.

The i-vector based DNN comprises 4 fully connected layers. In this system, DNN performs only as a feature enhancer because i-vector is already sophisticated feature in segment-level. 4 fully connected layers each have 512 units, and l-2 regularization is applied.

Table 1: Configuration of MFM based CNN system.

layer	output shape	kernel sizes
1^{st} MFM	499×300×32	5×5, 7×7, 9×9, 11×11
Max pooling	166×60×32	3×5
2^{nd} MFM	166×60×64	3×3, 5×5, 7×7, 9×9
Max pooling	55×12×64	3×5
3^{rd} MFM	55×12×64	3×3, 5×5, 7×7, 9×9
Max pooling	1×3×64	55×4
Average pooling	1×3×64	55×4
Concatenating	1×3×128	
Flatten	384	

3.4. Results

Table 2: Classification accuracy (%) of the individual systems and ensemble system with 4 classifiers.

system \ classifier	All w/o weight	All w weight	Gaussian w weight	SVM w weight
raw-waveform	67.15	68.10	67.91	66.56
spectrogram	66.24	66.20	66.44	66.44
i-vector	63.74	63.93	65.17	63.66
Ensemble	73.82	73.23	73.15	72.71

4. REFERENCES

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[2] D. Palaz, M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4295–4299.

[3] J. Jung, H. Heo, I. Yang, H. Shim, and H. Yu, "A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[4] J. Lee, J. Park, K. Kim, Luke, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," *arXiv preprint arXiv:1703.01789*, 2017.

[5] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *arXiv preprint arXiv:1511.02683*, 2015.

[6] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[7] B. Elizalde, H. Lei, G. Friedland, and N. Peters, "An i-vector based approach for audio scene detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[8] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed ¡today¿]. [Online]. Available: http://www.scipy.org/

[9] F. Chollet *et al.*, "Keras," https://github.com/keras-team/keras, 2015.

[10] A. Martn, A. Ashish, B. Paul, B. Eugene, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf

[11] A. Martin, B. Paul, C. Jianmin, C. Zhifeng, D. Andy, D. Jeffrey, D. Matthieu, G. Sanjay, I. Geoffrey, I. Michael, K. Manjunath, L. Josh, M. Rajat, M. Sherry, M. G. Derek, S. Benoit, T. Paul, V. Vijay, W. Pete, W. Martin, Y. Yuan, and Z. Xiaoqiang, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[14] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.