Acoustic Scene Classification Using Deep Audio Feature and BLSTM Network

Yanxiong Li, Xianku Li, Yuhan Zhang, Wucheng Wang, Mingle Liu, Xiaohui Feng
School of Electronic and Information Engineering
South China University of Technology
Guangzhou, China

***Abstract*—Although acoustic scene classification has been received great attention from researchers in the field of audio signal processing, it is still a challenging and unsolved task to date. In this paper, we present our work of acoustic scene classification for the challenge of the Detection and Classification of Acoustic Scenes and Events 2017, i.e., DCASE2017 challenge, using a feature of Deep Audio Feature (DAF) for acoustic scene representation and a classifier of Bidirectional Long Short Term Memory (BLSTM) network for acoustic scene classification. We first use a deep neural network to generate the DAF from Mel frequency cepstral coefficients, and then adopt a network of BLSTM fed by the DAF for acoustic scene classification. When evaluated on the official datasets of the DCASE2017 challenge, the proposed system outperforms the baseline system in terms of classification accuracy.**

Keywords—acoustic scene classification; bidirectional long short term memory network; deep audio feature

I. INTRODUCTION

Acoustic Scene Classification (ASC) is to determine a test audio recording belongs to which pre-given class of acoustic scenes, e.g. park, home, office. ASC is definitely useful for multimedia retrieval [1-3], audio-based surveillance and monitoring [4, 5]. The problem of ASC has been received great attention from the signal processing community with many evaluation campaigns [6-10], and is not effectively solved because of various factors, such as heavy background noise, large variations of time-frequency properties within each type of acoustic scenes [11].

The overall performance of one ASC system mainly depends on two modules: feature extraction and classifier building. Almost all of the previous works focused on these two modules for improving the performance [12]. The previous features included log Mel-band energy, Mel Frequency Cepstral Coefficients (MFCCs), spectral flux, spectrogram, Gabor filterbank, cochleograms, I-vector, histogram of gradients features [8, 13, 14]. Most back-end classifiers mainly consist of Gaussian Mixture Model (GMM), Deep Neural Network (DNN), random forest, decision tree, gradient boosting, support vector machine, hidden Markov model [9, 14]. For example, a bag-of-frame method [11] was considered as a baseline system for the challenges of both DCASE2013 [7] and DCASE2016 [15], which used MFCCs in combination with GMM. Eghbal-Zadeh et al [13] presented an I-vector extraction scheme for ASC using both left and right audio channels, and used a deep convolutional neural network trained on spectrograms of audio excerpts. Valenti et al [16] used a convolutional neural network fed by the feature of log-Mel spectrogram to classify acoustic scenes. Another method for ASC applied non-negative matrix factorization to spectro-temporal representation for decomposing features into activations of spectro-temporal elements [17].

Although many systems were proposed for ASC, to the best of our knowledge, no system combines the feature of DAF for acoustic scene representation with the classifier of BLSTM for acoustic scene classification to date. In our system submitted to the DCASE2017 challenge, we propose to build a DNN for extracting the DAF based on MFCCs, and then feed the DAF into a classifier of BLSTM for ASC. Compared to traditional features (e.g., MFCCs), the proposed feature of DAF is a deep transformed and compact representation of the original high-dimensional inputs fed to the input layer of the DNN, and thus can more effectively characterize the property differences among various classes of acoustic scenes. On the other hand, BLSTM has emerged as a scalable and state-of-the-art classifier for several classification problems related to sequential data [18]. Considering the high context correlation among the acoustic scenes and the advantage of BLSTM in capturing sequence information, we propose to use BLSTM based classifier fed by the proposed feature of the DAF to realize the task of ASC. Hence, main contributions of this study are to extract a feature of DAF for representing the property of each acoustic scene and propose a method for ASC by combining the DAF with the BLSTM network.

The rest of the paper is organized as follows. Section II describes the proposed system and Section III presents experimental results and discussions. Finally, conclusions are drawn in Section IV.

II. THE PROPOSED SYSTEM

The proposed system mainly consists of two modules: DAF extraction and BLSTM classification, as depicted in Fig. 1.

The work is funded by the national natural science foundation of China (61771200), the open project program of the national laboratory of pattern recognition (201800004), the fundamental research funds for the central universities (2015ZZ102, 2015ZM145), and the S&T Planning Project of Guangdong (2015A010103006). We thank the support of NVIDIA Corporation with the donation of the Titan GPU used for this research.

Fig. 1 The proposed system for ASC.

A. DAF Extraction

The DAF is used for representing the properties of different acoustic scenes, whose extraction is illustrated in Fig. 2. Each audio recording is split into frames for extracting MFCCs, and then a DNN based feature extractor is built for extracting bottleneck feature (i.e., DAF) based on the input feature of MFCCs. The DAF is output from the bottleneck layer of the DNN.

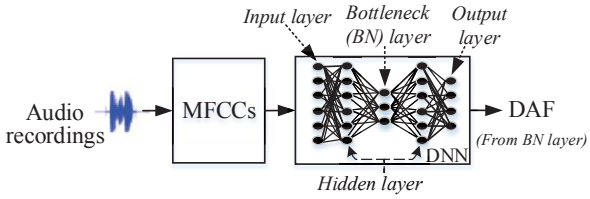

Fig. 2 The DAF extraction.

MFCCs is one of the most widely-used features for ASC [9], which is used as a component for extracting DAF here. The extraction of MFCCs is depicted in Fig. 3. Audio recording of each acoustic scene is first split into overlapping frames and is windowed by a Hamming window. Then, the Fast Fourier Transform (FFT) is performed for computing the power spectrum which is then smoothed with a bank of Mel filters. The center frequencies of these filters are uniformly spaced on the Mel-scale. Finally, outputs of logarithmic filter-banks are transformed into MFCCs by performing the Discrete Cosine Transform (DCT). The details of MFCCs extraction is introduced in [12]. The MFCCs is then fed to the DNN for extracting the DAF.

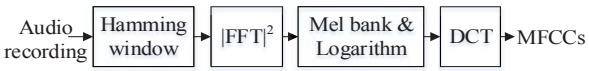

Fig. 3 The extraction of MFCCs.

The bottleneck layer is generally the narrowest layer of a DNN, whose activation signals can be used as a compact representation of the original high-dimensional inputs [19]. We generate a feature representation from the bottleneck layer of the DNN, called bottleneck feature whose extraction is depicted in Fig. 4.

For extracting the bottleneck feature, we first extract MFCCs with 13 dimensions from each audio recording. To model the dynamic properties of acoustic scenes, adjacent frames are also taken into consideration. A context of 31 frames of MFCCs is built and then the DCT with 16 bases is carried out on the MFCCs for being fed into the input layer of the DNN. Hence, the neuron number of the input layer of the DNN is 208 (i.e., 13×16). The number of hidden layer has direct impacts on the performance of the DAF for ASC, and thus its settings will be discussed in the experiments. The number of neuron of bottleneck layer, N_b (as depicted in Fig. 4), i.e., the dimension of bottleneck feature, is experimentally tuned on the development data and set to 50 (obtaining the best performance) in the experiments. The number of neuron of output layer is generally equal to the number of class needed to be identified, and thus depends on the specific task (e.g., 15 acoustic scenes here). The DNN based DAF extractor is trained using the development data and then the DAF is extracted from each audio recording of the test data using the trained DNN based feature extractor. It should be noted that the DNN here is generated as a feature extractor instead of a classifier.

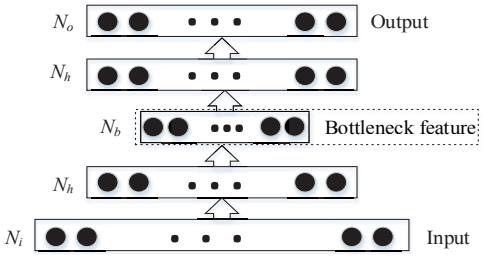

Fig. 4 The extraction of bottleneck feature. N_o, N_h, N_b and N_i stand for neuron numbers in output, hidden, bottleneck and input layers, respectively.

B. BLSTM Classification

A Recurrent Neural Network (RNN) has feedback connections, and works efficiently and flexibly for time-series signals, such as audio signal. Due to the exploding and vanishing gradient problem, a simple RNN is not easy to train, and unable to deal with long-range dependencies [20]. Hidden units of gated RNN are gate-based. Two common classes of Gated RNNs are LSTM and Gated Recurrent Units (GRUs), and the LSTM is widely used. The introductions to LSTM and GRU are given in [21] and [22], respectively.

LSTM is very flexible in classifying sequential data and is good at exploiting and storing information for long periods of time. This advantage is achieved from the use of special purpose built memory cells units [21]. Each memory block is composed of three gate units, i.e., input, output, and forget gates. Each of the three gates respectively have the ability to write, read and reset the functionality of the cell [23]. Forget gates are particularly shown to be essential for very long input sequence [24]. Although LSTM can have access to context for long periods of time, both LSTM and RNN can only obtain information from the previous context, and they cannot utilize the future context. As for the classification of acoustic scene, it needs to exploit information in both the previous and future contexts. Bidirectional RNN (BRNN) possesses this benefit by processing the sequence with two separate hidden layers in both forward and backward directions [25]. BLSTM is a combination of LSTM and BRNN [23]. Therefore, the BLSTM not only can exploit context for long periods of time, but also can have access to the context in both previous and future directions. Considering the advantage of the BLSTM in capturing sequence information, we use it as the classifier for ASC in this study.

III. EXPERIMENTS

A. Experimental Setup

The experiments are performed using the deep learning toolkit of TensorFlow [26]. Experimental data have two subsets: development dataset and evaluation dataset [10], and consist of 15 everyday environments: lakeside beach, inside bus, city center, cafe/restaurant, inside car, forest path, grocery store, home, library, metro station, office, park, residential area, train and tram. The development dataset is composed of the complete TUT Acoustic Scenes 2016 dataset (both development and evaluation datasets of the DCASE2016 challenge). For each acoustic scene, there are 312 audio segments (52 minutes of audio) in the development dataset. The evaluation data consists of 15 hours of binaural audio. There is about one hour recorded in each of 15 everyday environments. There are a total of 270 audio recordings of average length 3.5 minutes. The performance metric for ASC adopted in the DCASE2017 challenge is *classification accuracy* which is defined by: the number of correctly classified audio segments among the total number of audio segments. Each audio segment is considered an independent test sample.

The baseline system for ASC of the DCASE2017 challenge was based on a multilayer perceptron architecture which used the feature of log Mel-band energies. A context with 5 frames was adopted, resulting in a feature vector length of 200. A neural network containing two dense layers of 50 hidden neurons per layer and 20% dropout is built with 200 epochs of training. Classification decision is based on the network output layer which is of softmax type [10]. The results obtained by the baseline system are provided by the organizer of the DCASE2017 challenge.

To obtain better performance, we need to tune the parameters of both the DNN used for the DAF extraction and the BLSTM classifier for acoustic scene classification. We will only discuss the setting of key parameters, i.e., the setting of hidden layer of the DNN for the DAF extraction, since the novelty of this study is to propose a feature of the DAF. Main configurations for the DAF extraction and BLSTM classifier building are listed in Table 1.

TABLE I. THE PARAMETERS SETTINGS FOR THE DAF EXTRACTION AND BLSTM CLASSIFIER BUILDING

DAF extraction	
MFCC	Dimension: 13, frame length/overlap: 40/20 ms.
DNN	The dimension of DAF (i.e., neurons of bottleneck layer): 50, neurons of input layer: 208, neurons of output layer: 15, learning rate: 0.001, maximum iterations: 3000, batch size: 256, context size: 31 frames, weight decay: 0.1, dropout: 0.8, output layer function: Sigmoid.
BLSTM building	
BLSTM	Cell number: 400, learning rate: 0.001, iterations: 300, batch size: 256, unrolled steps: 10, training algorithm: back-propagation through time, initial forget bias: 1.

B. Results and Discussions

The number of hidden layers of the DNN directly influences the performance of the DAF for acoustic scene classification. We tune the settings of hidden layers (including bottleneck layer) for the DAF extraction on the development dataset. The results of acoustic scene classification under the conditions of different settings of hidden layers are listed in Table 2. The digit of 50 in the first column of Table 2 denotes the number of neurons of bottleneck layer, whereas the digits of both 100 and 200 are the numbers of neurons of other hidden layers. For example, [50, 100] denotes that the DNN has 2 hidden layers (including bottleneck layer), and the number of neurons of bottleneck layer is 50 while other hidden layer has 100 neurons. As shown in Table 2, the proposed system achieves 82.1% of classification accuracy (i.e., the highest value) when the parameter of hidden layer of the DNN is set to [200, 100, 50, 100, 200]. This setting of hidden layer parameters is adopted for the proposed system when it is evaluated on evaluation dataset.

TABLE II. IMPACTS OF HIDDEN LAYER SETTINGS ON THE PERFORMANCE OF THE DAF WHEN EVALUATED ON THE DEVELOPMENT DATASET

Hidden layer settings	Classification accuracy (%)
[50]	76.5
[50, 100]	77.7
[100, 50]	77.9
[100, 50, 100]	81.7
[200, 50, 100, 200]	81.9
[200, 100, 50, 100, 200]	**82.1**
[200, 100, 50, 100, 100, 200]	80.4
[200, 100, 100, 50, 100, 100, 200]	79.7

After setting the parameters of the proposed system according to Tables 1 and 2, this sub-section compares the proposed system with the baseline system. Table 3 shows results obtained by our system and the baseline system [10]. When evaluated on the development dataset, the proposed system achieves an overall average classification accuracy of 82.1% which is higher than 73.8% obtained by the baseline system. When evaluated on the evaluation dataset, the proposed system obtains an overall average classification accuracy of 67.2% which is higher than 61.0% yielded by the baseline system. That is, our system outperforms the baseline system for ASC with the improvements of 8.3% and 6.2% when evaluated on the development dataset and the evaluation dataset, respectively.

It can also be seen from Table 3 that classification accuracies for different classes of acoustic scenes are of significant differences. The reasons are probably that: 1) data unbalance (the difference of data amount of different acoustic scenes is notable); 2) similar properties between some acoustic scenes, e.g. train and tram, park and forest path. In conclusion, the datasets of ASC adopted in the DCASE2017 challenge is very complex and thus this task is quite challenging.

TABLE III. ASC RESULTS OBTAINED BY DIFFERENT SYSTEMS ON BOTH THE DEVELOPMENT DATA AND THE EVALUATION DATA.

Acoustic scene	Classification accuracy (%)			
	Development data		Evaluation data	
	Baseline	Ours	Baseline	Ours
Beach	77.6	90.4	40.7	63.9
Bus	83.7	76.7	38.9	64.8
Cafe/Restaurant	55.1	74.3	43.5	43.9
Car	86.2	88.2	64.8	78.5
City center	88.5	89.6	79.6	88.9
Forest path	83.3	90.7	85.2	84.3
Grocery store	63.1	79.4	49.1	62.0
Home	74.5	83.7	76.9	89.8
Library	60.6	73.2	30.6	65.7
Metro station	88.5	78.6	93.5	79.6
Office	97.4	90.6	73.1	63.0
Park	64.4	83.2	32.4	44.4
Residential area	62.8	84.1	77.8	55.6
Train	38.1	73.4	72.2	68.5
Tram	82.7	75.7	57.4	55.6
Overall	**73.8**	**82.1**	**61.0**	**67.2**

Although this work is a preliminary study for ASC, the results have proved the effectiveness of the proposed system. It should be noted that this study focused only on the task of ASC. But, the proposed system can be extended to other similar tasks of the DCASE2017 challenge, e.g. acoustic event detection.

IV. CONCLUSIONS

We have proposed a system of ASC for the DCASE2017 challenge by using a feature of DAF and an effective classifier of BLSTM. The proposed system outperformed the baseline system on both development and evaluation datasets. The results have shown that the proposed system is effective for solving the problem of ASC. The future work includes: 1) exploring other deep-transformed features for more effectively representing the properties of different acoustic scenes; 2) studying on the design and parameters settings of classification models for further improving the performance; 3) investigating the technique of data augmentation to generate additional data for training deep neural network, since participants of the DCASE2017 challenge are constrained to use a relatively small training dataset as a rule.

REFERENCES

[1] Y. Li, Q. He, S. Kwong, T. Li, and J. Yang, "Characteristics-based effective applause detection for meeting speech," *Signal Process.*, vol. 89, no. 8, pp. 1625-1633, 2009.

[2] Y. Li, Q. Wang, X. Zhang, W. Li, X. Li, J. Yang, X. Feng, Q. Huang, and Qianhua He, "Unsupervised classification of speaker roles in multi-participant conversational speech," *Computer Speech and Language*, vol. 42, pp. 81-99, 2017.

[3] Y. Li, Q. Wang, X. Li, X. Zhang, Y. Zhang, A. Chen, Q. He, and Q. Huang, "Unsupervised detection of acoustic events using information bottleneck principle," *Digital Signal Process.*, vol.63, pp. 123-134, 2017.

[4] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: a system for detecting anomalous sounds," *IEEE T. ITS*, vol. 17, no. 1, pp. 279-288, Jan. 2016.

[5] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio surveillance: a systematic review," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1-46, 2016.

[6] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," *Lecture notes in computing science*, vol.4122, pp. 311-322, 2007.

[7] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M.D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE T. Multimedia*, vol. 17, no. 10, pp. 1733-1746, Oct. 2015.

[8] T. Virtanen, A. Mesaros, T. Heittola, M.D. Plumbley, P. Foster, E. Benetos, and M. Lagrange, "Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop," 2016.

[9] J. Schröder, N. Moritz, J. Anemüller, S. Goetze, and B. Kollmeier, "Classifier architectures for acoustic scenes and events: implications for DNNs, TDNNs, and perceptual features from DCASE 2016," *IEEE/ACM T. ASLP*, vol. 25, no. 6, pp. 1304-1314, Jun. 2017.

[10] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen. "DCASE 2017 challenge setup: tasks, datasets and baseline system," in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, Nov. 2017.

[11] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE T. ASLP*, vol. 23, no. 1, pp. 20-31, 2015.

[12] Y. Li, X. Zhang, H. Jin, X. Li Q. Wang, Q. He, and Q. Huang, "Using multi-stream hierarchical deep neural network to extract deep audio feature for acoustic event detection," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 897-916, Jan. 2018.

[13] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: A hybrid approach using binaural I-vectors and deep convolutional neural networks," in *Proc. of Detection and Classification of Acoustic Scenes and Events 2016*, Sep. 2016.

[14] A. Rakotomamonjy and G. Gasso, "Histogram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM T. ASLP*, vol. 23, no. 1, pp. 142-153, Jan. 2015.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. of the 24th Eur. Signal Process. Conf. 2016*, Sep. 2016, pp. 1128-1132.

[16] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Proc. of Detection and Classification of Acoustic Scenes and Events 2016*, Sep. 2016.

[17] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *Proc. of IEEE ICASSP*, Mar. 2016, pp. 6445-6449.

[18] X. Li, H. Xianyu, J. Tian, W. Chen, F. Meng, M. Xu, and L. Cai, "A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction," in *Proc. of IEEE ICASSP*, Mar. 2016, pp. 544-548.

[19] D. Yu, and M.L. Seltzer, "Improved bottleneck features using pre-trained deep neural networks," in *Proc. of INTERSPEECH*, 2011, pp.237-240.

[20] R. Pacanu, T. Mikolov, and Y. Bengio, "On the difficulties of training recurrent neural networks," in *Proc. of ICML*, no.2, pp. 1310-1318, 2013.

[21] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of IEEE ICASSP*, pp. 6645-6649, 2013.

[22] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. of the Conf. on Empirical Methods in Natural Lang. Process.*, pp. 1724-1734, 2014.

[23] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. of IEEE ICASSP*, pp. 4869-4873, Apr. 2015.

[24] F.A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451-2471, 2000.

[25] M. Schuster, and K.K. Paliwal, "Bidirectional recurrent neural networks," *IEEE T. SP*, vol. 45, no. 11, pp. 2673-2681, 1997.

[26] https://www.tensorflow.org/