A MULTI-SPECTROGRAM DEEP NEURAL NETWORK FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan

School of Computing, University of Kent, Medway Campus, Chatham, UK
{ldp7, ivm, H.Phan, R.Palani}@kent.ac.uk

ABSTRACT

This work targets the task 1A and 1B of DCASE2019 challenge that are Acoustic Scene Classification (ASC) over ten different classes recorded by a same device (task 1A) and mismatched devices (task 1B). For the front-end feature extraction, this work proposes a combination of three types of spectrograms: Gammatone (GAM), log-Mel and Constant Q Transform (CQT). The back-end classification shows two training processes, namely pre-trained CNN and post-trained DNN, and the result of post-trained DNN is reported. Our experiments over the development dataset of DCASE2019 1A and 1B show significant improvement, increasing 14% and 17.4 % compared to DCASE2019 baseline of 62.5% and 41.4%, respectively. The Kaggle report also confirms the classification accuracy of 79% and 69.2% for task 1A and 1B.

Index Terms— Gammatone, log-Mel, Constant Q Transform (CQT), Convolutional Neural Network (CNN), Fully-connected layers.

1. INTRODUCTION

For the front-end feature extraction, it could be divided into two main approaches, mainly exploring spectrogram of audio signal. The first group only applies one kind of spectrogram mainly log-Mel, and different aspects of that feature are explored such as multi-dimensional log-Mel spectrogram [1], spectrogram based wavelet transform [2], auditory statistics conducted over cochlear filter output [3], or i-Vector from Mel-Frequency Cepstral Coefficients (MFCC) [4]. The other category explores various spectrograms likely bag-of-feature. For examples, they are log-Mel filter and MFCC [5], MFCC, Gammatone filter and log-Mel [6], or various spectrograms as Perceptual Linear Prediction (PLP), MFCC, Power Nomalized Cepstral Coefficients (PNCC), Robust Compressive Gamma-chirp filter-bank Cepstral Coefficients (RCGCC) and Subspace Projection Cepstral Coefficients (SPPCC) [7]. This work inspires from the idea that each spectrogram could contribute distinct attribution that is useful for back-end learning model. Therefore, we apply three spectrograms (Gammatone (GAM) [8], log-Mel spectrogram [9] and Constant-Q Transform (CQT) [9]) and proposed an effective way to combine them.

Regarding the back-end classifier, convolutional neural network (CNN) has become a strong approach for ASC. In fact, CNN was early approached for machine hearing tasks [10, 11], and DCASE2018 challenge showed various CNN architectures [12, 13, 14, 15, 16, 17] that achieved good results over task 1A and 1B. In this work, we propose a re-trained model that consists of a pre-trained process based CNN architecture and a post-trained model with fully-connected layers called pos-trained DNN.

Data augmentation that is useful to enhance the classification has widely applied for ASC such as adding background noise [18], frequency shifting [19], or GAN network [20]. This work also applies a type of data augmentation technique namely mixup that mixes two original data with various ratios to generate new data.

2. SYSTEM ARCHITECTURE

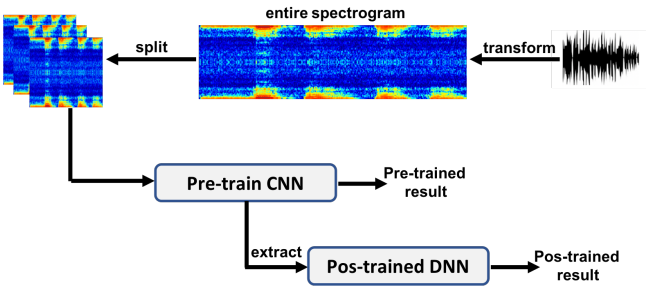

Figure 1: General System Architecture

The overall system architecture is firstly described as Fig. 1 that could be divided into two parts. While the upper shows how to generate features fed into the classification model, the lower presents two training processes, pre-trained CNN and post-trained DNN. Firstly, the audio file is transferred into a two-dimensional shape known as spectrogram and this work exploits three kinds of spectrograms (GAM, log-Mel and CQT). The whole spectrogram next is split into patches, showing the frequency and time size at 128 and 128, respectively. Thus, we apply mixup data augmentation to generate new data from these patches before feeding both original data and generated mixup data into classification. Regarding back-end learning model, the pre-trained CNN is called firstly. When the pre-trained CNN converses, the high-level feature coming from the middle layers of pre-trained CNN network are extracted and fed into pos-trained DNN. Eventually, the result of post-trained DNN is reported.

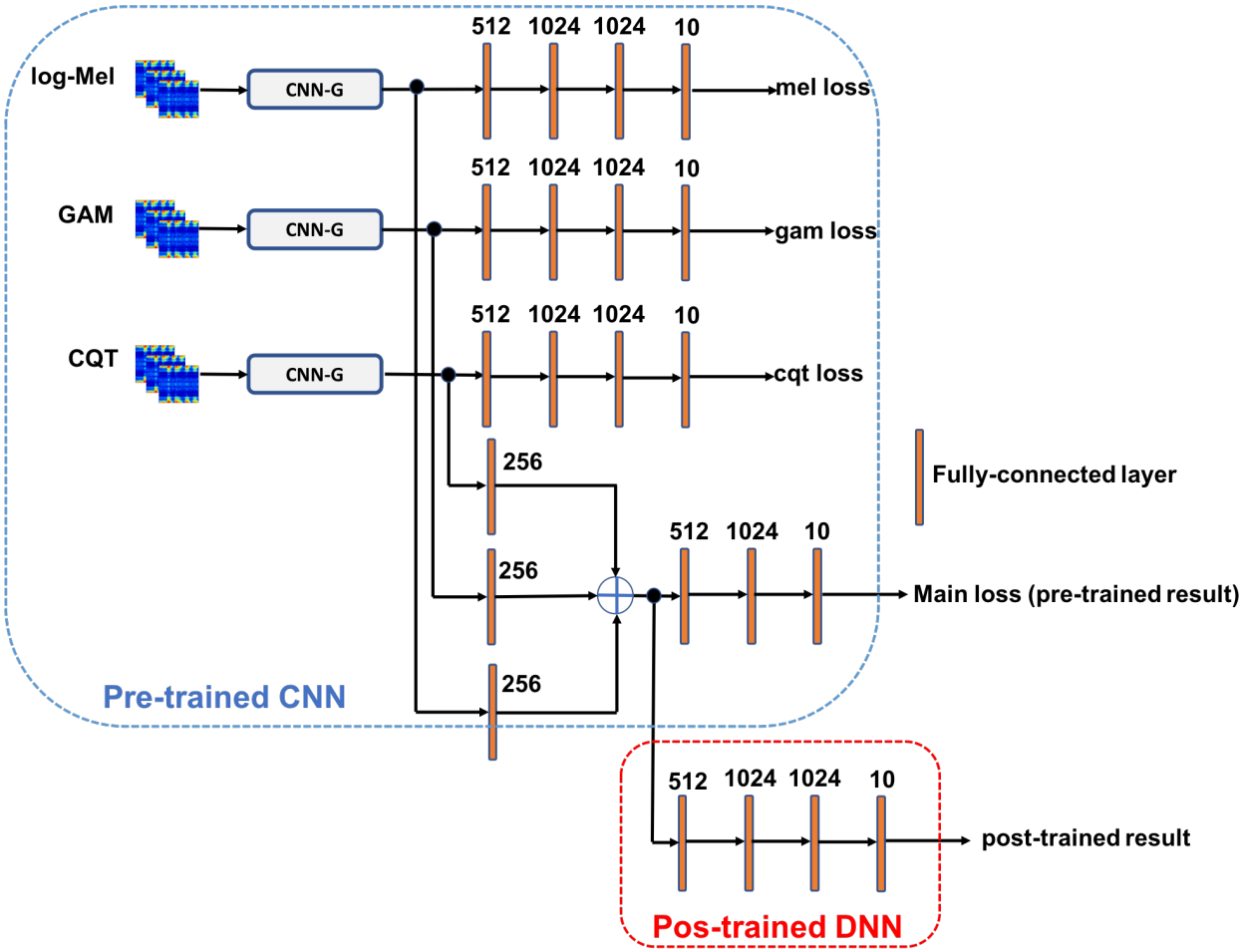

Figure 2: Re-trained Model Architecture

Table 1: Setting Parameters of Spectrogram.

Parameters	Values
Window size	1920
Hop size	256
Fast Fourier Number	4096
Frequency Min	10
Frequency resolution	128

2.1. Front-End Feature Extraction

As mentioned above, this work applies three spectrograms: GAM [8], log-Mel [9] and CQT [9] to generate input features fed into the classifier. Therefore, these spectrograms have a similar size that requires the same setting described as Table 1. By setting as Table 1, the entire spectrogram shows frequency and time resolutions of 128 and 1870 respectively, splitting into 14 patches with the size of 128×128.

2.2. Back-end Classification

The back-end classification consists of two processes that are pre-trained CNN and post-trained DNN denoted as Fig. 2. For pre-trained CNN, patches coming from three different spectrograms are fed into three parallel convolutional blocks namely CNN-G, and these CNN-Gs show the same configuration as detailed in Fig. 3. At the final convolutional layer of each CNN-G, global max/mean pooling layers are used to extract high-level features, feeding into four fully-connected layers with the configuration of $512 - 1024 - 1024 - 10$ respectively. Since we apply three different spectrograms, we have three separated networks (noting that a single network consists of one CNN-G block and four fully-connected layers), thus providing three loss functions known as $Loss_{CQT}, Loss_{GAM}$ and $Loss_{log-Mel}$. Therefore, we call these networks as log-mel, gam or cqt CNN network. Next, every high-level feature extracted from the global max/mean pooling layer of log-mel, gam and cqt CNN networks goes through a fully-connected layer with size of 256 before adding together. The addictive result is sent to three fully-connected layers and this data flow is called main CNN network. Totally, we have four loss functions, three for the log-mel, gam and cqt CNN networks and the final one

Table 2: Experiment Results Over Task 1B

Class	DCASE2019 baseline			Our Method		
	Device B	Device C	Average	Device B	Device C	Avarage
Airport	18.3	24.1	21.2	25.9	31.3	28.6
Bus	40.4	70.0	55.2	88.8	96.2	92.5
Metro	50.7	36.1	43.4	50.0	55.5	52.7
Metro Station	28.7	36.1	30.0	46.2	48.1	47.1
Park	45.2	57.0	51.1	85.1	94.4	89.7
Public Square	22.8	11.3	17.0	20.3	31.4	25.9
Shopping Mall	63.5	64.8	64.2	61.1	72.2	66.7
Street Pedestrian	37.0	37.6	37.3	53.7	46.2	50.0
Street Traffic	77.0	86.5	81.8	90.7	96.2	93.5
Tram	12.0	12.6	12.3	31.4	51.8	41.6
Overall	**39.6**	**43.1**	**41.4**	**55.3**	**62.3**	**58.8**

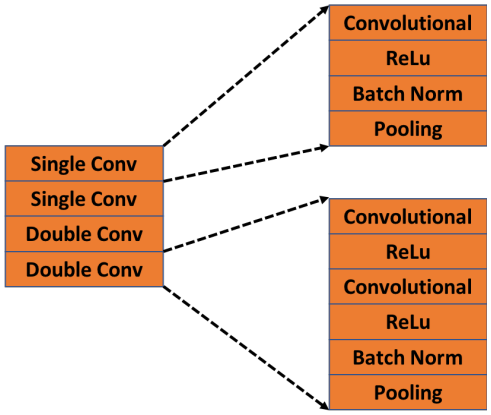

Figure 3: CNN-G configuration

Table 3: Experiment Results Over Task 1A

Class	DCASE2019 baseline	Our Method
Airport	48.4	69.4
Bus	62.3	90.8
Metro	65.1	71.4
Metro Station	54.5	64.8
Park	83.1	87.3
Public Square	40.7	63.8
Shopping Mall	59.4	73.0
Street Pedestrian	60.9	69.2
Street Traffic	86.7	91.8
Tram	64.0	81.9
Overall	**62.5**	**76.2**

for the main CNN called $Loss_{MAIN}$. The final loss function is computed as the equation (1) that focuses on the classification accuracy of the main CNN network

$$Loss = (Loss_{CQT} + Loss_{GAM} + Loss_{log-Mel})/3 + Loss_{MAIN} \quad (1)$$

Once pre-trained CNN converses over the training set, all patches of both training set and test set are fed into pre-trained CNN. Thus, the additive result in main CNN network mentioned above are extracted, creating input feature for post-trained DNN. Regarding the post-trained DNN, it is constructed by four fully-connected layers. Both pre-trained CNN and post-trained DNN are trained at a patch-size level, using softmax at final layers for classification, built in the Tensorflow framework, using the Adam method [21] for learning rate optimisation. Batch size and learning rate are set to 100 and 0.0001 respectively. Eventually, the post-trained DNN result, conducted over the entire time-frequency spectrogram, will yield the final classification accuracy.

2.3. Ensemble Model

Basing on the back-end classification mentioned above, we propose two different types of global pooling that are global max and global mean pooling at the final layers of CNN-G blocks. Therefore, the final classification accuracy is the fusion of two results of post-trained DNN due to global max and mean input features. If we consider P_{MEAN} and P_{MAX} are scores of post-trained DNN, the final accuracy is computed by averaging;

$$\bar{P} = (\bar{P}_{MEAN} + \bar{P}_{MAX})/2 \quad (2)$$

2.4. Data Augmentation

In order to increase data variation, various types of data augmentation are explored in ASC task. This work also applies a kind of data augmentation, called mixup, to enhance the performance. Let's consider original data as X_1, X_2 and expected labels as Y_1, Y_2, mixup data is generated by

$$X_{mp1} = X_1 * \lambda + X_2 * (1 - \lambda) \quad (3)$$

$$X_{mp2} = X_1 * (1 - \lambda) + X_2 * \lambda \quad (4)$$

$$Y_{mp} = Y_1 * \lambda + Y_2 * (1 - \lambda) \quad (5)$$

$$Y_{mp2} = Y_1 * (1 - \lambda) + Y_2 * \lambda \quad (6)$$

with $\lambda \in U(0, 1)$ is random coefficient.

To generate λ, we apply two distribution functions, beta distribution and uniform distribution. We feed both original data and generated mixup data into classifiers, and considerably extending the training time of model. In this work, both pre-trained CNN (mixup

on patch size) and post-trained DNN (mixup on global mean/max pooling vector) processes apply this augmentation technique to improve the performance.

3. EXPERIMENTS AND RESULTS

Table 3 presents the experiment results on the task 1A [22] and it shows that the accuracy over every class are improved compared to DCASE2019 baseline. Next, Table 2 shows the results over the task 1B[22]. Since task 1B is valued on Device B & C, only accuracy on device B and C are recorded.

4. CONCLUSION

In this paper, we propose a deep learning model that combines three input spectrogram and explore re-trained method with pre-trained CNN and post-trained DNN. The experiment results over DCASE2019 development dataset targeting task 1A and 1B review that our method are effective to improve the classification accuracy over every class.

5. REFERENCES

[1] Y. Yin, R. R. Shah, and R. Zimmermann, "Learning and fusing multimodal deep features for acoustic scene categorization," in *ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 1892–1900.

[2] S. Waldekar and G. Saha, "Wavelet transform based mel-scaled features for acoustic scene classification," in *INTERSPEECH*, 2018, pp. 3323–3327.

[3] H. Song, J. Han, and D. Shiwen, "A compact and discriminative feature based on auditory summary statistics for acoustic scene classification," in *INTERSPEECH*, 2018, pp. 3294–3298.

[4] 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," DCASE2016 Challenge, Tech. Rep., September 2016.

[5] J. Li, W. Dai, F. Metze, S. Qu, and S. Das, "A comparison of deep learning methods for environmental sound detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 126–130.

[6] H. Phan, L. Hertel, M. Maass, P. Koch, R. Mazur, and A. Mertins, "Improved audio scene classification based on label-tree embeddings and convolutional neural networks," *IEEE Transactions On Audio, Speech, And Language Processing*, vol. 25, no. 6, pp. 1278–1290, 2017.

[7] S. Park, S. Mun, Y. Lee, and H. Ko, "Score fusion of classification systems for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[8] D. P. W. . Ellis. (http://www.ee.columbia.edu/dpwe/resources/matlab/ gammatonegram) Gammatone-like spectrogram.

[9] McFee, Brian, R. Colin, L. Dawen, D. PW.Ellis, M. Matt, B. Eric, and N. Oriol, "librosa: Audio and music signal analysis in python," in *Proceedings of The 14th Python in Science Conference*, 2015, pp. 18–25.

[10] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*, no. 2635, Apr. 2015, pp. 559–563.

[11] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, W. Xiao, and H. Phan, "Continuous robust sound event classification using time-frequency features and deep learning," *PloS one*, vol. 12, no. 9, p. e0182309, 2017.

[12] R. Zhao, K. Qiuqiang, Q. Kun, D. Mark, and W. Bjorn, "Attention-based convolutional neural networks for acoustic scene classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, November 2018, pp. 39–43.

[13] O. Mariotti, M. Cord, and O. Schwander, "Exploring deep vision models for acoustic scene classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, November 2018, pp. 103–107.

[14] L. Yang, X. Chen, and L. Tao, "Acoustic scene classification using multi-scale features," in *Detection and Classification of Acoustic Scenes and Events 2018*, November 2018, pp. 29–33.

[15] T. Nguyen and F. Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Detection and Classification of Acoustic Scenes and Events 2018*, November 2018, pp. 34–38.

[16] H. Zeinali, L. Burget, and J. Cernocky, "Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge," in *Detection and Classification of Acoustic Scenes and Events 2018*, November 2018, pp. 202–206.

[17] C. Roletscheck, T. Watzka, A. Seiderer, D. Schiller, and E. André, "Using an evolutionary approach to explore convolutional neural networks for acoustic scene classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, November 2018, pp. 158–162.

[18] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2721–2725.

[19] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[20] S. Mun, S. Park, D. Han, and H. Ko, "Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane," DCASE2017 Challenge, Tech. Rep., September 2017.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840