CDNN-CRNN JOINED MODEL FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Lam Pham, Tan Doan, Dat Ngo, Hung Hong, Ha Hoang Kha

Department of Electrical and Electronics Engineering, HoChiMinh City University of Technology, Viet Nam
ldp7@kent.ac.uk, {tandoan.hcmut, datt.ngo.hcmut, honghung.dotdak}@gmail.com, hhkha@hcmut.edu.vn

ABSTRACT

This work proposes a deep learning framework applied for Acoustic Scene Classification (ASC), targeting DCASE2019 task 1A. In general, the front-end process shows a combination of three types of spectrograms: Gammatone (GAM), log-Mel and Constant Q Transform (CQT). The back-end classification presents a joined learning model between CDNN and CRNN. Our experiments over the development dataset of DCASE2019 challenge task 1A show a significant improvement, increasing 11.2% compared to DCASE2019 baseline of 62.5%. The Kaggle reports the classification accuracy of 74.6% when we train all development dataset.

Index Terms— Gammatone, log-Mel, Constant Q Transform (CQT), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN)

1. INTRODUCTION

To deal with the Acoustic Scene Classification (ASC) task, recent researches on front-end features can be separated into two main groups. The first explores one kind of time-frequency feature such as log-Mel filter, and makes effort to explore different aspects of that feature. For instance, they are multi-dimensional log-Mel spectrogram [1], wavelet spectrogram [2], auditory statistics of a cochlear filter output [3], or a kind of i-vector extraction from the traditional features like Mel-Frequency Cepstral Coefficients (MFCC) [4]. The second category attempts to combine multiple spectrograms, and that can be seen as log-Mel filter and MFCC [5], MFCC, Gammatone filter and log-Mel [6], or even a wide range of features such 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]. Inspiration from the second approach that different time-frequency features have distinct attribution, we therefore propose an effective combination of three spectrograms, gammatone (GAM) [8], log-Mel spectrogram [9] and Constant-Q Transform (CQT) [9].

Convolutional Deep Neural Network (CDNN), which was early approached for machine hearing tasks [10, 11], has become the most effective classification for ASC. In fact, most proposed DCASE2018 models show various architectures based CDNN such as [12, 13, 14, 15, 16, 17]. Additionally, Convolutional-Recurent Neural Network (CRNN), which has become a standard model applied for acoustic event detection, was also applied over LITIS dataset [18] or DCASE2017 [19], proving effective results. We, therefore, propose a joined learning model that combines both CDNN and CRNN.

In order to enhance the classification accuracy, this work also applies a data augmentation technique called mixup data, which comes from research on image classification [20]. Different from other augmentations such as added background noise [21], frequency shifting [22], or GAN network [19], this technique generates a new data by mixing two original data with different rate.

2. SYSTEM ARCHITECTURE

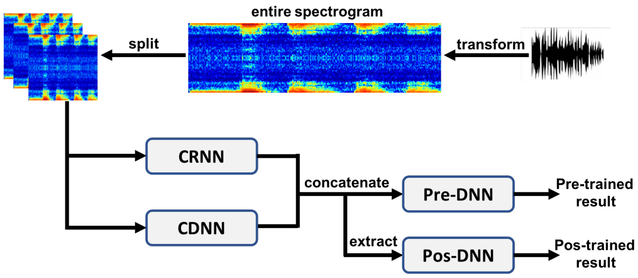

Figure 1: General System Architecture

In general, the proposed framework architecture is presented as Fig. 1. Firstly, the audio file is transferred into a two-dimensional spectrogram. In this work, we explore three types of spectrograms (GAM, log-Mel and CQT) and propose a fusion to combine all of them. Next, the entire spectrogram is split into patches with a frequency and time resolution of 128 and 128, respectively. Before feeding patches into the back-end classification, we apply mixup data augmentation technique to generate new patches, namely mixup data, which are mixed with original patches. For the back-end classifier, two parallel learning blocks, namely CDNN and CRNN, learn different attribution of the spectrogram and are concatenated before feeding into a Pre-DNN block. When the pre-trained process finishes, data concatenation between CDNN and CRNN output are extracted and fed into Pos-DNN that considered as the post-train process. The post-trained result is reported in this work.

2.1. Front-End Feature Extraction

As mentioned above, this work applies GAM [8], log-Mel [9] and CQT [9] to generate spectrograms from audio segments and Table 1 shows how we set parameters for generating spectrograms. Since

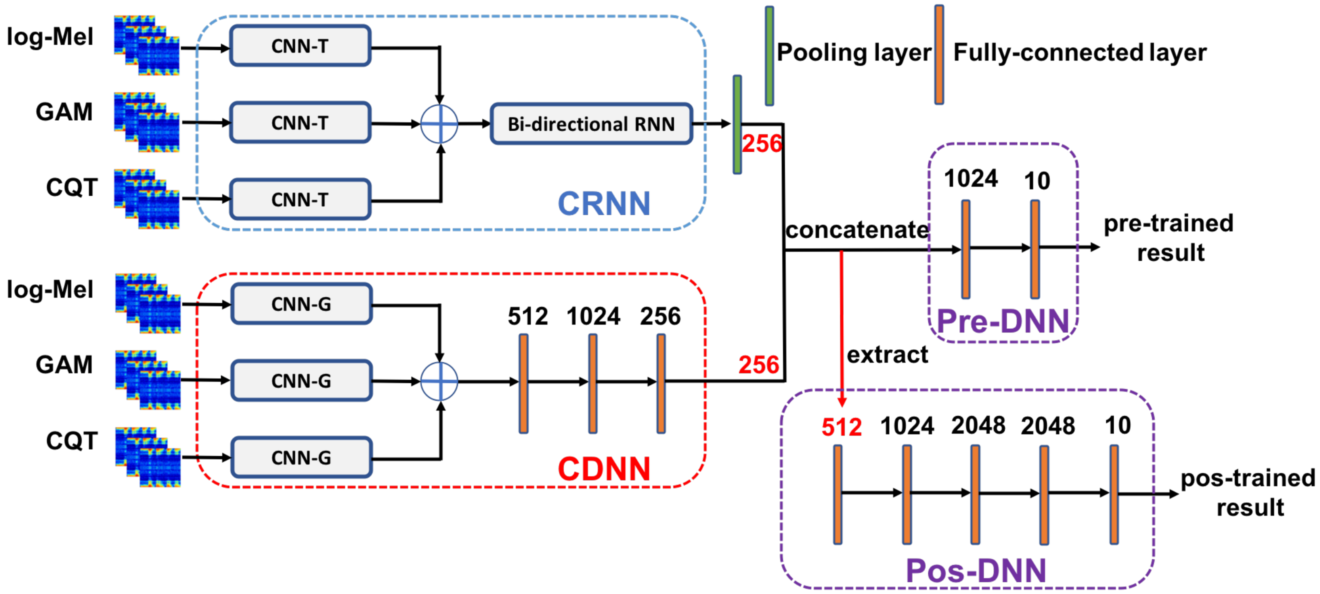

Figure 2: Back-end Classification Architecture

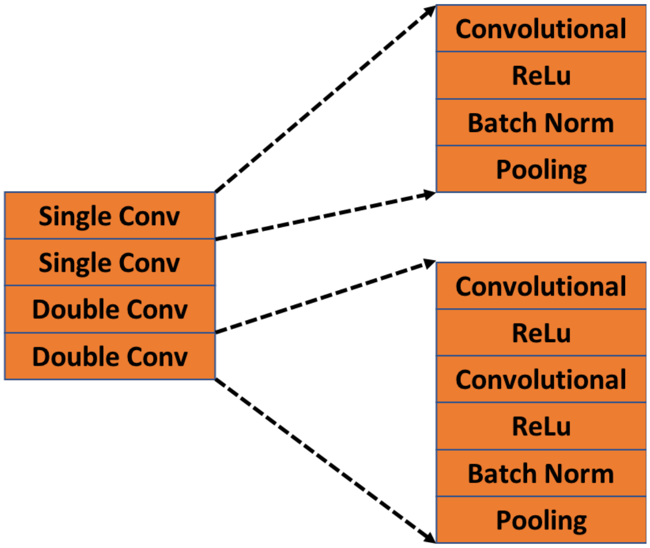

Figure 3: CNN-T or CNN-G architecture

Table 1: Setting Parameters of Spectrogram.

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

Table 2: CNN-T/G Output Shape [batch sizexFrequecy ResolutionxTime ResolutionxChannel Resolution]

Layer	CNN-G	CNN-T
Input	$n \times 128 \times 128 \times 1$	$n \times 128 \times 128 \times 1$
Single Conv	$n \times 64 \times 64 \times 32$	$n \times 64 \times 128 \times 32$
Single Conv	$n \times 32 \times 32 \times 64$	$n \times 32 \times 128 \times 64$
Doublue Conv	$n \times 16 \times 16 \times 128$	$n \times 16 \times 128 \times 128$
Double Conv	$n \times 256$	$n \times 1 \times 128 \times 256$

we aim at generating the same size of spectrograms regarding three types of transformation, the entire spectrogram shows the frequency and time resolution of 128 and 1870, respectively. Thus, every entire spectrogram is split into 14 patches with the size of 128 × 128.

2.2. Back-end Classification

The general architecture of back-end classification mentioned in Fig. 1 with the pre-trained and post-trained process is described as detailed in Fig. 2. First, three different spectrogram features with the size of 128 × 128 are fed into convolutional blocks, namely CNN-G and CNN-T, which belongs in CDNN and CRNN branches, respectively. The CDNN network that combines CNN-G and fully-connected layers shows a conventional CNN architecture. The role of three CNN-G is to transfer spectrogram features as patches into high-level features as 256-dimensional vectors before adding together. Thus, additional result goes through fully-connected layers. The CNN-T block of the CRNN network helps to transfer a spectrogram feature to a time-sequential feature that is fed into bi-directional RNN. Both CNN-T and CNN-G share a similar general architecture as described in Fig. 3, but the CNN-T pooling layers are different from that of CNN-G. In particular, with the input patch size of 128 × 128 mentioned above, the output of every convolution layer, namely single conv or double conv in Fig. 3, are different and described in Table 2. If we consider the shape of the input feature is [batch size, frequency resolution: time resolution: channel number], CNN-G scales down both the time and frequency resolution with setting pooling at [1:2:2:1] over single or double convolutional layer. At the final layer of CNN-G block, the global-mean pooling is applied to scale into $n \times 256$ (noting that n is the batch size and 256 is the number of kernel at the final double conv). However, only the frequency dimension is scaled with a pooling layer setting of [1:2:1:1] regarding CNN-T, keeping the time resolution constant. At the final layer of CNN-T, the global-mean pooling layer is used over the frequency dimension, creating a high-level feature with the shape of $n \times 1 \times 128 \times 256$. It is then reshaped to $n \times 128 \times 256$ that is considered as a sequence of 256-dimensional vectors, feeding into bi-directional RNN. The bi-directional RNN is configured by GRU cells with a dynamic sequence length setting and 128 hidden states. Thus, the output of bi-directional RNN is sent to a global-mean pooling layer where the time dimension is averaged, creating a 256-dimensional vector that is a same shape as the output of CDNN network. Eventually, outputs of CRNN and CDNN are concatenated and goes through three fully-connected layers and the final Softmax layer is used to classify. As regards the post-trained process namely Pos-DNN, more fully-connected layers are used to deeply learn the extracted high-level feature from the pre-trained process. Both the pre-trained and the post-trained process are trained at a patch-size level, using softmax layer at the final layer, built in the Tensorflow framework, using the Adam method [23] for learning rate optimisation. Batch size and learning rate are set to 100 and 0.0001 respectively.

2.3. Data Augmentation

In order to increase data variation, various types of data augmentation are explored in the 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 lables as Y_1, Y_2, we generate mixup data as follows

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

$$X_{mp2} = X_1 * (1 - \lambda) + X_2 * \lambda \tag{2}$$

$$Y_{mp} = Y_1 * \lambda + Y_2 * (1 - \lambda) \tag{3}$$

$$Y_{mp2} = Y_1 * (1 - \lambda) + Y_2 * \lambda \tag{4}$$

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

We feed both original data and generated mixup data into the back-end classification (both the pre-trained and the post-trained process), and considerably extending the training time of the model.

3. EXPERIMENTS AND RESULTS

This work follows the instruction of DCASE2019 challenge over task 1A. Therefore, the development set is separated into subsets, namely training set and testing set. Table 3 presents the experiment results over testing set on the task 1A [24] and it shows that the accuracy over every class are improved compared to the DCASE2019 baseline. Submission result over Leaderboard set on Kaggle reports the accuracy of 74.7%, improving 12.2% compared to DCASE2019 baseline of 62.5%.

Table 3: Experiment Results Over Task 1A

Class	DCASE2019 baseline	Our Method
Airport	48.4	58.9
Bus	62.3	86.3
Metro	65.1	73.9
Metro Station	54.5	68.0
Park	83.1	87.0
Public Square	40.7	52.7
Shopping Mall	59.4	71.4
Street Pedestrian	60.9	74.4
Street Traffic	86.7	90.3
Tram	64.0	74.3
Overall	**62.5**	**73.7**

4. CONCLUSION

In this work, we propose a learning model that combines three input spectrograms and explore the fusion between CRNN and CDNN for classification. The experiment results over DCASE2019 development dataset targeting task 1A review that our method are effective to improve the classification accuracy over every class. The attention layers as our future work could be added into both CDNN and CRNN since it enables us to improve performance.

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] H. Phan, O. Y. Chén, L. D. Pham, P. Koch, M. D. Vos, I. V. McLoughlin, and A. Mertins, “Spatio-temporal attention pooling for audio scene classification,” *CoRR*, 2019.

[19] 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.

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

[21] 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.

[22] 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.

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

[24] 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