A Re-trained Model Based On Multi-kernel Convolutional Neural Network for Acoustic Scene Classification

Tuan Nguyen
Electrical and Electronics Department
Ho Chi Minh City University of Technology
HCMC, Vietnam
tuannguyen@hcmut.edu.vn

Dat Ngo
Electrical and Electronics Department
Ho Chi Minh City University of Technology
HCMC, Vietnam
datt.ngo.hcmut@gmail.com

Lam Pham
Electrical and Electronics Department
Ho Chi Minh City University of Technology
HCMC, Vietnam
lamd.pham@hcmut.edu.vn

Linh Tran
Electrical and Electronics Department
Ho Chi Minh City University of Technology
HCMC, Vietnam
linhtran@hcmut.edu.vn

Trang Hoang
Electrical and Electronics Department
Ho Chi Minh City University of Technology
HCMC, Vietnam
hoangtrang@hcmut.edu.vn

***Abstract*—This paper proposes a deep learning framework applied for Acoustic Scene Classification (ASC), which identifies recording location. In general, we apply three types of spectrograms: Gammatone (GAM), log-Mel and Constant Q Transform (CQT) for front-end feature extraction. For back-end classification, we present a re-trained model with a multi-kernel CDNN-based architecture for the pre-trained process and a DNN-based network for the post-trained process. Our obtained results over DCASE 2016 dataset show a significant improvement, increasing by nearly 8% compared to DCASE baseline of 77.2%.**

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

I. Introduction

Acoustic Scene Classification (ASC) task aims to detect recording locations, showing one of main tasks in “Machine hearing” research field [1]. In a sound scene recording, it shows various sound events, located at a wide range of frequency bands. If we refer sound event to signal and sound scene to noise, the signal-to-noise ratio is changeable. Moreover, a long-time occurring sound event could be considered as sound scene in certain contexts. For instance, *residential-area* recording shows a quiet background and short-time *engine* sound as events, but long-time occurring *engine* sound in *on bus* context is considered as a background noise. These variabilities therefore make ASC task more challenging.

In order to deal with the ASC challenges, recent papers exploit multi-input feature, showing two main trends. The first trend is using one kind of time-frequency feature such as log-Mel filter, and next exploring different aspects over this spectrogram. For example, multi-dimensional log-Mel spectrogram and wavelet scalogram were effectively exploited in [2] and [3], respectively. Besides, i-vector, extracted from Mel-Frequency Cepstral Coefficients (MFCC), was proposed by [4]. To approach raw audio signal, Song et al. [5] proposed an auditory statistics of a cochlear filter output, showing a good performance over DCASE 2016 dataset. By contrast, the second trend exploits multiple spectrograms. For instance, log-Mel filter and MFCC were combined in [6], or MFCC, Gammatone filter and log-Mel were used in [7], or even it shows various features such as Perceptual Linear Prediction (PLP), MFCC, Power Nomalized Cepstral Coefficients (PNCC), Robust Compressive Gammachirp filter-bank Cepstral Coefficients (RCGCC) and Subspace Projection Cepstral Coefficients (SPPCC) [8]. We inspire the second approach for front-end feature extraction, using different time-frequency spectrograms to enrich input feature. In this paper, we therefore apply three spectrograms, gammatone (GAM) [9], log-Mel spectrogram [10] and Constant-Q Transform (CQT) [10], proposing an ensemble of three spectrograms.

For back-end classification, Convolutional Deep Neural Network (CDNN) comes as a powerful approach for ASC. In fact, the state-of-the-art introduced various network architectures such as two parallel-CNN [11], fusion of various learning models [12], hierarchical scheme of classification with CNN in [13], or GRNN-based network proposed in [14]. In this paper, we propose a re-trained model for ASC task. In particular, we apply CDNN-based network with multi-kernel architecture for the pre-trained process. Next, we extract high-level feature from the pre-trained model, and then feeding into a DNN-based network referred to the post-trained process before reporting the final classification accuracy. 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 [15].

II. System Architecture

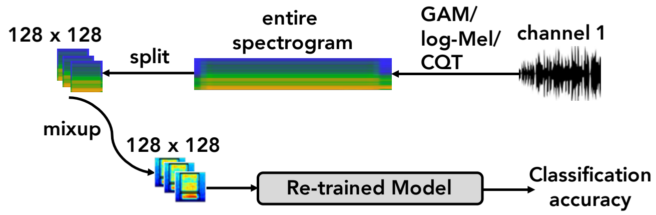

Fig. 1. General system architecture

In general, we propose the whole system as shown in Fig. 1 and Fig. 2. Firstly, the raw audio from one channel (channel 01) is transferred into an entire spectrogram, two-dimensional shape like an image. In this work, we apply three types of transformation (GAM, log-Mel and CQT), totally three spectrograms. Then, we split the whole spectrogram into small patches, showing frequency and temporal resolution at 128 and 128, respectively. Before feeding patches into the pre-trained process, we apply mixup data augmentation technique to generate new patches. After that, we feed both new and original patches into the pre-trained model.

A. Front-End Feature Extraction

As mentioned above, this paper uses GAM [9], log-Mel [10] and CQT [10] methods to transfer audio segment into spectrograms. These spectrograms are applied by the same configuration as showed in Table I to create same size. With the configuration as Table I, we obtain an entire spectrogram with size of 5120. Then, we split this entire spectrogram into 40 patches with the size of 128×128.

TABLE I
Setting Parameters of Spectrogram.

Parameters	Values
Window Size	1920
Hop Size	256
Fast Fourier Number	4096
Frequency Min	10
Frequency Resolution	128

B. Back-end Classification

The general architecture of re-trained model mentioned in Fig. 1 is described in detail in Fig. 2. In particular, the re-trained model comprise of two training processes, pre-trained and post-trained modes shown in the upper and lower parts in Fig. 2, respectively. For the pre-trained process, the model bases on CDNN architecture, comprising of four main CNN blocks showing similar architectures. Look at the **CNN-01** block for detailed analysis as shown in Table II, we firstly apply Batchnorm layer (Bn) over image patch (128x128). Next, Bn output is fed into convolutional layers (Cv) with multi-kernel setting (four different kernels setting size of [9x9], [7x7], [5x5], [3x3]). Then, every output of convolutional layers goes through Rectified Linear Unit (ReLu), BatchNorm (Bn), Average Pooling (Ap) and DropOut (Dr) layers respectively before concatenating according to channel dimension, obtaining a tensor shape (64x64x32). Next CNN blocks show similar architectures, but different from the final **CNN-04** block. At final convolutional layers, we use smaller kernel sizes of ([2x2], [3x3], [4x4] and [5x5]) due to a small frequency and temporal resolution. After these convolutional layers, we apply Global Average Pooling (Gl) over frequency and temporal dimensions to reduce the tensor shape (16x16x256) to 256-dimensional vectors. Next, we concatenate four these vectors (coming from four convolutional layers), obtaining a 1024-dimensional vector and feeding to back-end classification (three Fully-connected layers (Fc) with 2048-2048-15 configuration). After the pre-trained model finishes, we extract high-level feature (known as 1024-dimensional vector) from the pre-trained model before feeding into a DNN-based architecture, referred to the post-trained model as shown in the lower part of Fig. 2. The configuration of post-trained models shows four Fully-connected layers, setting 2048-4096-1024-15 to layers in order.

TABLE II
CNN-01 block architecture

Layer	Output
Input layer (image patch)	128×128
Bn - Cv (9×9) - Relu - Bn - Ap (2×2) - Dr (0.1)	64×64×8
Bn - Cv (7×7) - Relu - Bn - Ap (2×2) - Dr (0.1)	64×64×8
Bn - Cv (5×5) - Relu - Bn - Ap (2×2) - Dr (0.1)	64×64×8
Bn - Cv (3×3) - Relu - Bn - Ap (2×2) - Dr (0.1)	64×64×8
Output layer (tensor)	64×64×32

C. Data Augmentation

In this paper, we apply mixup data augmentation method to increase data variation, which enforces the network learning and increases Fisher criterion. Let's consider two original data as X_A, X_B and expected labels as Y_A, Y_B, we generate new data as below equations:

$$X_{mp1} = X_A * \gamma + X_B * (1 - \gamma) \quad (1)$$

$$X_{mp2} = X_A * (1 - \gamma) + X_B * \gamma \quad (2)$$

$$Y_{mp1} = Y_A * \gamma + Y_B * (1 - \gamma) \quad (3)$$

$$Y_{mp2} = Y_A * (1 - \gamma) + Y_B * \gamma \quad (4)$$

with γ is random coefficient from unit distribution.

We feed both original data and new mixup data into

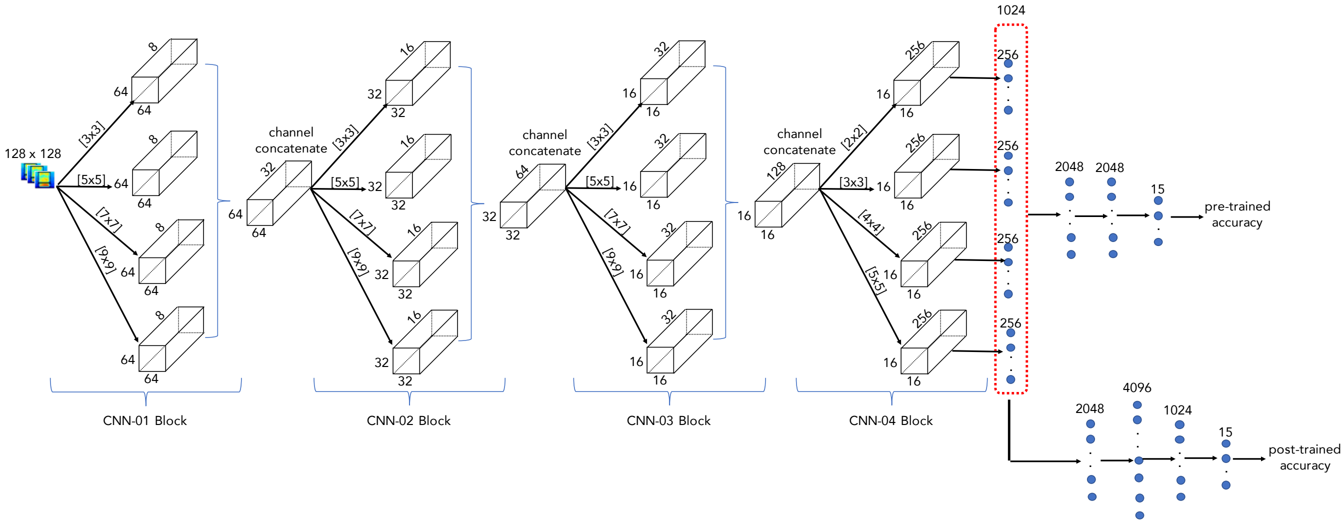

Fig. 2. Re-trained model

the classifications, patch level and vector level for pre-trained and post-trained processes, respectively.

D. Ensemble Method

As discussion in Section I above, we do experiments over separated spectrograms and exploit an ensemble model over them. In particular, If we consider a vector $X_{log-Mel}[x_1, x_2, ..., x_{15}]$, $X_{GAM}[x_1, x_2, ..., x_{15}]$, $X_{CQT}[x_1, x_2, ..., x_{15}]$ as the probability output of the post-trained process as regards log-Mel, GAM and CQT spectrograms respectively, we have a fusion strategy used in this work, which is to compute the overall per-class mean, based on;

$$\hat{y} = argmax(X_{log-Mel} + X_{GAM} + X_{CQT}) \qquad (5)$$

where $\hat{y}$ is predicted results.

E. Proposed Baseline

To explore the effect of multi-kernel architecture, we propose a baseline that shows a same number of convolutional layers as Table III, but keep kernel size at [3x3] for all convolutional layers. We compare performance of the proposed baseline and multi-kernel CDNN model (pre-trained process) as mentioned in Fig. 2 with GAM spectrogram input.

TABLE III
PROPOSED BASELINE ARCHITECTURE

Layer	Output
Input layer (image patch)	128×128
Bn - Cv (3×3) - Relu - Bn - Ap (2×2) - Dr (0.1)	64×64×32
Bn - Cv (3×3) - Relu - Bn - Ap (2×2) - Dr (0.1)	32×32×64
Bn - Cv (3×3) - Relu - Bn - Ap (2×2) - Dr (0.1)	16×16×128
Bn - Cv (3×3) - Relu - Bn - Gl - Dr (0.1)	256
Fc	512
Fc	1024
Fc	15

III. Experiments and Results

A. Dataset and setting

This paper employs the DCASE 2016 dataset [16] with totally 15 classes. In this dataset, sample rate at 44.1 kHz is used and every segment shows 30 second recording duration. The provided data comprises of two sets; a development set (Dev Set) with 78 segments each class and an evaluation set (Eva Set) with 26 segments each class for training and evaluating, respectively. We did our experiments with Tensorflow framework, using Adam method for optimization, setting learning rate, batch size and epoch number to 0.0001, 50 and 200, respectively. For mixup data augmentation, we applied for both pre-trained and post-trained processes over patch level and vector level, respectively.

B. Compared to DCASE 2016 baseline

Table IV shows a performance comparison among DCASE 2016 baseline, our proposed baseline, our pre-trained and post-trained models with three different spectrogram inputs. Compare between the pre-trained and post-trained results over three spectrograms, the post-trained result is effective to improve around 4% to 5%. For pre-trained results, our proposed baseline and multi-kernel pre-trained model over GAM spectrogram input improves by 2% and 4% compared to DCASE baseline. Multi-kernel architecture is effective to improve 2% compared to the proposed baseline with a stable kernel. As regards results over post-trained processes, log-Mel and GAM show equal results, improving by nearly 6.5% compared to DCASE 2016 of 77.2%. Meanwhile, performance over CQT are poorer with classification accuracy of 78.7%. Our ensemble method over post-trained results of three spectrograms is useful to achieve the best accuracy of 85.1%.

As regards classification accuracy over 15 classes, Fig. 3 shows post-trained results over three separated spectrograms,

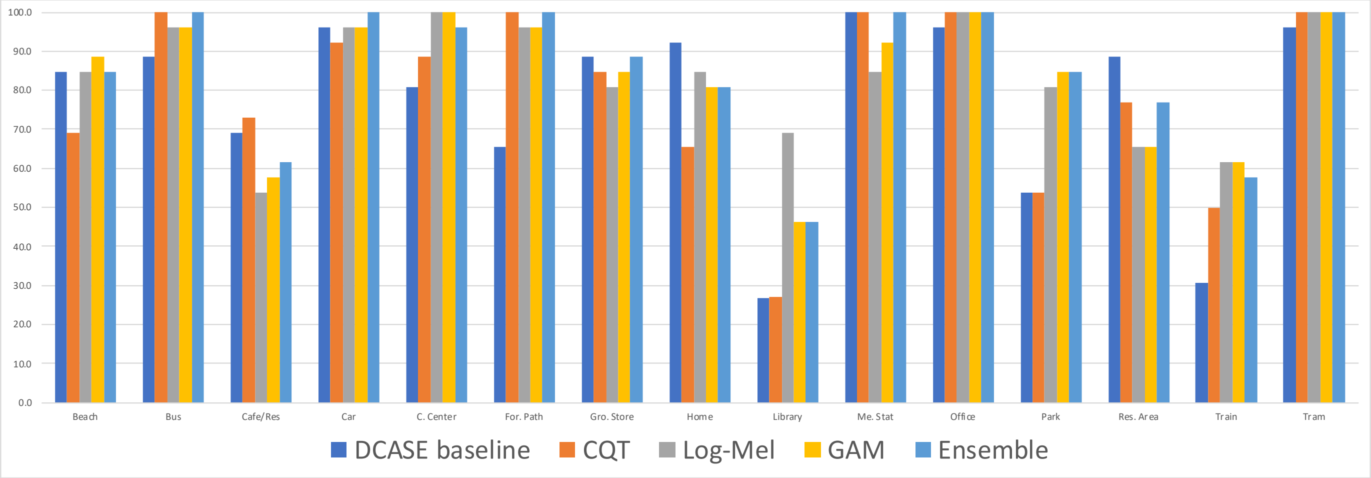

Fig. 3. System performance over 15 classes

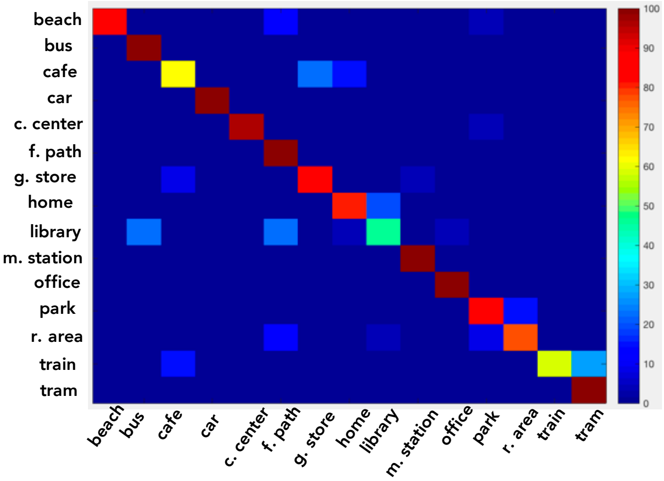

Fig. 4. Confusion matrix of our ensemble model

ensemble of them, compared to DCASE 2016 baseline. It is noted that results over CQT spectrogram show high performance over certain classes such as *bus, cafe, forest-path, metro-station, office, tram*, but poor performance over the others. GAM and log-Mel spectrogram show equal results over 15 classes, but ensemble significantly outperforms over DCASE baseline and other spectrograms. In general, ensemble is effective to improve DCASE baseline over almost classes a part from *cafe, home* and *residential-area*. Fig. 4 shows a confused matrix over ensemble results over 15 classes. The color map shows cases of inaccuracy occurring among certain classes. For instances, *train* and *tram*, *park* and *residential-area*, or *home* and *library* shows these inaccurate pairs due very similar environment. Another inaccurate results over *cafe* and *grocery-store*, showing babble noise, also prove high correlation between two environmental sounds.

TABLE IV
COMPARISON OUR SYSTEMS TO DCASE BASELINE

System	Pre/Post-trained Acc.
DCASE 2016	77.2
Baseline	79.7
CQT	74.4/78.7
log-Mel	79.0/83.6
GAM	81.3/83.8
Ensemble	82.1/**85.1**

C. *Compared to the state-of-the-art*

Compare to the state-of-the-art, we separated into two Table V and Table VI, the first table for single model comparison and the second for the best obtained accuracy. For Table V, it shows a comparison among top DCASE 2016 challenges, and our proposed single model with GAM spectrogram input achieve top-six accuracy. For recently published system as show in Table VI, we achieve top-nine accuracy over DCASE 2016 challenge and a very competitive result to the state-of-the-art. In general, ensemble models show higher performance than single models.

TABLE V
COMPARISON TO TOP-SIX SINGLE MODEL IN DCASE 2016 CHALLENGE

System	Classifier	Acc.
DCASE Baseline	CNN	77.2
Bae et al. [17]	CNN-RNN	84.1
Lee et al. [18]	CNN	84.6
Takahashi et al. [19]	DNN-GMM	85.6
Kumar et al. [20]	SVM	85.9
Valenti et al. [21]	CNN	86.2
Bisot et al. [22]	NMF-SVM	87.7
Our single model	CNN-DNN	**83.8**

IV. CONCLUSION

In this work, we propose a re-trained model that aims for Acoustic Scene Classification task, and an ensemble model over these different spectrograms. To deal with ASC

TABLE VI
COMPARISON BETWEEN THE TOP-TEN DCASE 2016 COMPETITION ARCHITECTURE ACCURACIES (TOP), RECENTLY PUBLISHED PAPERS USING DCASE 2016 DATA (MIDDLE), AND THE PROPOSED METHOD (BOTTOM) ON THE DCASE 2016 EVA DATASET.

System	Classifier	Accuracy
Bae et al. [17]	CNN-RNN	84.1
Lee et al. [18]	CNN	84.6
Lee et al. [23]	CNN ensemble	85.4
Takahashi et al. [19]	DNN-GMM	85.6
Kumar et al. [20]	SVM	85.9
Valenti et al. [21]	CNN	86.2
Marchi et al. [24]	Ensemble	86.4
Ko et al. [8]	Ensemble	87.2
Bisot et al. [22]	NMF	87.7
Eghbal-Zadeh et al. [4]	Ensemble	89.7
Shefali Waldekar et al. [3]	SVM	81.2
Seongkyu Mun et al. [25]	DNN	86.3
Juncheng Li et al. [6]	Ensemble	88.2
Rakib Hyder et al. [26]	Ensemble	88.5
Hongwei et al. [5]	SVM	89.5
Yifang Yin et al. [2]	Ensemble	91.0
Our ensemble model	Ensemble	**85.1**

challenges, we investigate whether multi-kernel architecture and ensemble of multi-spectrogram input could be effective to obtain high performance. Our extensive experiments over DCASE 2016 dataset achieve competitive results compared to both DCASE 2016 challenge and the state-of-the-art.

For future work, we will investigate the contribution of channels to classification accuracy, and how to combine various input features coming from channels and spectrograms.

REFERENCES

[1] R. F. Lyon, *Human and Machine Hearing*. Cambridge University Press, 2017.

[2] 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.

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

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

[6] J. Li, W. Dai, F. Metze, S. Qu, and S. Das, "A comparison of deep learning methods for environmental sound detection," in *Pro. ICASSP*. IEEE, 2017, pp. 126–130.

[7] 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.

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

[9] D. P. W. . Ellis. Gammatone-like spectrogram. [Online]. Available: http://www.ee.columbia.edu/dpwe/resources/matlab/gammatonegram

[10] 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.

[11] T. Lidy and A. Schindler, "Cqt-based convolutional neural networks for audio scene classification," in *Pro. DCASE2016*, 2016, pp. 60–64.

[12] G. Mafra, N. Duong, A. Ozerov, and P. Pérez, "Acoustic scene classification: An evaluation of an extremely compact feature representation," in *Pro. DCASE2016*, 2016, pp. 85–89.

[13] Y. Xu, Q. Huang, W. Wang, and M. D. Plumbley, "Hierarchical learning for dnn-based acoustic scene classification," in *Pro. DCASE2016*, 2016, pp. 105–109.

[14] M. Zöhrer and F. Pernkopf, "Gated recurrent networks applied to acoustic scene classification and acoustic event detection," in *Pro. DCASE2016*, 2016, pp. 115–119.

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

[16] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[17] S. H. Bae, I. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN," in *Pro. DCASE2016*, 2016, pp. 11–15.

[18] Y. Han and K. Lee, "Convolutional neural network with multiple-width frequency-delta data augmentation for acoustic scene classification," in *Pro. DCASE2016*, 2016.

[19] G. Takahashi, T. Yamada, S. Makino, and N. Ono, "Acoustic scene classification using deep neural network and frame-concatenated acoustic feature," in *Pro. DCASE2016*, 2016.

[20] B. Elizalde, A. Kumar, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, "Experiments on the dcase challenge 2016: Acoustic scene classification and sound event detection in real life recording," in *Pro. DCASE2016*, 2016.

[21] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "Dcase 2016 acoustic scene classification using convolutional neural networks," in *Pro. DCASE2016*, 2016, pp. 95–99.

[22] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Supervised nonnegative matrix factorization for acoustic scene classification," in *Pro. DCASE2016*, 2016, pp. 62–69.

[23] J. Kim and K. Lee, "Empirical study on ensemble method of deep neural networks for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[24] E. Marchi, D. Tonelli, X. Xu, F. Ringeval, J. Deng, S. Squartini, and B. Schuller, "The up system for the 2016 DCASE challenge using deep recurrent neural network and multiscale kernel subspace learning," DCASE2016 Challenge, Tech. Rep., September 2016.

[25] 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.

[26] R. Hyder, S. Ghaffarzadegan, Z. Feng, J. H. Hansen, and T. Hasan, "Acoustic scene classification using a CNN-supervector system trained with auditory and spectrogram image features." in *Pro .INTERSPEECH*, 2017, pp. 3073–3077.