INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A Robust Framework For Acoustic Scene Classification

Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan

School of Computing, The University of Kent, Medway, Kent, United Kingdom
{ldp7, ivm, H.Phan, R.Palani}@kent.ac.uk

Abstract

Acoustic scene classification (ASC) using front-end time-frequency features and back-end neural network classifiers has demonstrated good performance in recent years. However a profusion of systems has arisen to suit different tasks and datasets, utilising different feature and classifier types. This paper aims at a robust framework that can explore and utilise a range of different time-frequency features and neural networks, either singly or merged, to achieve good classification performance. In particular, we exploit three different types of front-end time-frequency feature; log energy Mel filter, Gammatone filter and constant Q transform. At the back-end we evaluate effective a two-stage model that exploits a Convolutional Neural Network for pre-trained feature extraction, followed by Deep Neural Network classifiers as a post-trained feature adaptation model and classifier. We also explore the use of a data augmentation technique for these features that effectively generates a variety of intermediate data, reinforcing model learning abilities, particularly for marginal cases. We assess performance on the DCASE2016 dataset, demonstrating good classification accuracies exceeding 90%, significantly outperforming the DCASE2016 baseline and highly competitive compared to state-of-the-art systems.

Index Terms: Machine hearing, acoustic scene classification, convolutional neural network, deep neural network, spectrogram, log-Mel, Gammatone filter, constant Q transform

1. Introduction

Acoustic Scene Classification (ASC), which aims to identify recording location by analysis of background sound, constitutes one of the main tasks of the emerging research field named "machine hearning" [1]. The main challenge of this task comes from various foreground events occurring in one background sound recording. Some distinct events occur only in certain scenes, but could be mixed with other events that occur in a range of scenes. A model that learns distinct features in one scene cannot perform well on another scene when those distinct features are absent. Conversely, a model focussing only on background sound may not achieve competitive results when a background sound from one scene becomes part of a foreground event in another scene (for example, vehicle sounds occurring naturally in a city centre scene may also appear occasionally in a quiet street or beach scene).

To deal with the ASC challenges mentioned above, recently proposed ASC techniques can be separated into two main categories. The first explores some kind of time-frequency feature such as log-Mel filter, and attempts to learn different aspects of that feature. Examples include multi-dimensional log-Mel spectrogram [2], wavelet spectrogram [3], auditory statistics of a cochlear filter output [4], or a kind of i-vector extraction from the traditional features like Mel-Frequency Cepstral Coefficients (MFCC) [5]. The second group attempts to combine multiple spectrograms, such as log-Mel filter and MFCC [6], MFCC, Gammatone filter and log-Mel [7], 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) [8]. With the inspiration that different time-frequency features have distinct strengths, and thus an effective combination of features could enhance performance, this paper adopts the second approach. In summary, we aim to exploit three different time-frequency features through a two stage feature extraction and classification process. We choose three perceptually relevant features, the gammatone (GAM) [9], log-Mel spectrogram [10] and Constant-Q Transform (CQT) [10].

A convolutional neural network (CNN) appears to have become the most effective classifier for ASC, since it was first applied to machine hearing [11, 12], although recent papers have proposed numerous extensions or variants of the basic CNN architecture. In general, the best ASC methods use a pre-trained model to generate low-level features from original input features, from which another learning model is applied to train from these low-lever features. A popular approach is to use a CNN for the pre-trained model, followed by a post-trained Support Vector Machine (SVM) [13]. Similarly, a Random Forest is used to generate low-level features before applying a combined classifier (using CNN and SVM) [7] for the post-trained model. i-vector techniques, which are very effective in fields of speaker or language identification [14], were also explored by Li et al. [6] and Eghba-Zadeh et al. [5], achieving reasonable results. However, the best results are currently obtained by an ensemble of front-end features, indicating that there is complementary information in different feature types. There are also numerous kinds of back-end classifiers in use, including various combined forms. Indeed, the top classification accuracy for the DCASE2016 dataset [2, 5, 15, 13] is currently achieved by fusing different classification methods or mixing different input features like a bag-of-features input. This work continues the trend by proposing to use both CNN and DNN classifiers, pre-trained and post-trained respectively, as well as making use of an ensemble of time-frequency input features. The resulting architecture yields highly competitive accuracy on DCASE2016 and has the advantage of being relatively less complex than many state-of-the art methods.

In order to enhance performance, a wide range of data augmentation techniques have been attempted by various authors, with the inspiration of increasing the variety of input data. Indeed, various data augmentation techniques have been shown to be effective in ASC such as added background noise [16], frequency shifting [17], or GAN network [18]. This work adopts a type of data augmentation called mixup, which comes from research on image classification [19] and has only very recently applied in audio research fields [20, 21] but not yet, to our knowledge, to ASC.

http://dx.doi.org/10.21437/Interspeech.2019-1841

2. The Proposed System

2.1. Proposed Baseline Model

Since we aim to analyse an intensive ensemble coming from three time-frequency features individually as well as in combined forms, we firstly establish a unified baseline architecture, shown in Fig 1. The baseline shown uses a log-Mel spectrogram [10] for front-end time-frequency feature extraction, with the entire spectrogram split into non-overlapping patches of time resolution 128 frames and 128 frequency bins. Each 128×128 image patch is then fed separately into the classification model. For the back-end classification, a pre-training process (referred to as the CNN pre-training) trains a ***CNN block*** followed by a ***DNN block 01***, detailed in Table 1 and Table 2 (centre column), respectively. At the final layer a $softmax$ function minimises cross-entropy, based on;

$$E(\theta) = -\frac{1}{N}\sum_{i=1}^{N} y_i.log(\hat{y}_i(\theta)) + \frac{\lambda}{2}.||\theta||_2^2 \qquad (1)$$

where $E(\theta)$ is the loss function over all parameters θ of the pre-trained CNN model, constant λ is set to 0.0001, y_i and $\hat{y}_i$ are expected and predicted results, respectively.

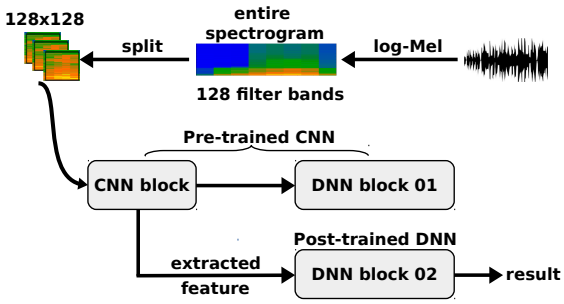

Figure 1: *The baseline proposed architecture.*

Next, we propose a DNN post-training process for ***DNN block 02***, which is described in the right hand column of Table 2. This is used to train the extracted global max or mean of all channels from the final convolutional layer of the ***CNN block*** in the pre-trained model. ***DNN block 02*** consists of four fully-connected layers, also uses a $softmax$ function at the final layer and has a similar loss function to eqn. (1) for training. Both the CNN pre-training and DNN post-training are performed at a patch-size level, built in the Tensorflow framework, using the Adam method [22] 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 in this baseline model.

2.2. Proposed Ensemble Model

As discussion above, there are various approaches to using a combination of time-frequency features, but combinations of the three spectrograms in this paper (log-Mel, GAM and CQT) have not yet been evaluated to the best of our knowledge. While log-Mel applies Mel filterbanks to Fourier transformed audio to simulate the overall frequency selectivity of the human auditory system [23], the Gammatone filter is based on the cochlea activation response of the human inner ear. As regards CQT, it is based on the geometric relationship of pitch, which may make it effective when undertaking a comparison between natural and artificial sounds, as well as being suitable for analysis of musical notes. Since these spectrograms come from different audio models, it is highly feasible that they can each contribute distinct features to back-end classification. This inspires us to exploit an investigation of the performance of ensembles of those features.

In total, our ensemble investigation spans five different models as listed in Table 3. The first three models have a similar architecture to the proposed baseline shown in Fig. 1, apart from simply replacing the spectrogram method with log-Mel, CQT and GAM respectively. The two remaining models, the ***Concat*** model and the ***Additive*** model, combine the extracted features from the ***CNN block*** as detailed in Fig. 2, by either adding or concatenating respectively.

If we consider a vector $X_{log-Mel}[X_1, X_2, ..., X_{256}]$ as the output of the ***CNN block*** for the log-Mel spectrogram, this vector goes through a fully-connected layer denoted as $ReLu(X_{log-Mel} * W_{log-Mel})$ where $W_{log-Mel}[W_1, W_2, ..., W_{256}]$ are training parameters. Thus, $addition$ and $concatenation$ functions of Tensorflow framework are called to combine outputs of ***CNN block*** before feeding into the ***DNN block 01*** and ***DNN block 02*** for

Table 1: *CNN block architecture.*

Layer	Output Shape	Kernel Size
Conv 1	128×128×32	9×9
Max pooling	64×64×32	2×2
Conv 2	64×64×64	7×7
Max pooling	32×32×64	2×2
Conv 3	32×32×128	5×5
Max pooling	16×16×128	2×2
Conv 4	16×16×256	3×3
Global max & mean pooling	256	

Table 2: *DNN block architecture.*

Layer	DNN block 01	DNN block 02
Input layer	256	256
Fully-connected	512	512
Fully-connected	1024	1024
Fully-connected	15	1024
Fully-connected		15

Table 3: *Five models based on different spectrogram input.*

System name	Front-end feature	Back-end classification
log-Mel	log-Mel	preCNN & posDNN
GAM	Gammatone	preCNN & posDNN
CQT	CQT	preCNN & posDNN
Additive	Addition of GAM, log-Mel, CQT	preCNN & posDNN
Concat	Concatenation of GAM, log-Mel, CQT	preCNN & posDNN

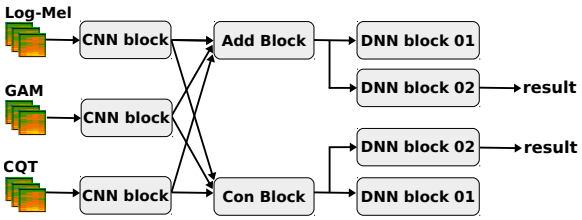

Figure 2: *Diagram of spectrogram combinations.*

pre-trained and post-trained processing, respectively.

Since we have five models with each one reporting two probability scores (namely postDNN-MAX and postDNN-MEAN, coming from max and mean pooling layers fed into the post-trained DNN respectively), we can also perform a 10-way score-level fusion of system probabilities, which we denote as the ensemble result. There is only one fusion strategy used in this work, which is to use the overall aggregated per-class mean.

As mentioned, for every model in Table 3, we have two average probability results, one from postDNN-MAX and another from the posDNN-MEAN, which are combined;

$$\bar{P}_{log-Mel} = \bar{P}_{postDNN-MAX} + \bar{P}_{posDNN-MEAN} \quad (2)$$

and thus the reported ensemble classification accuracy in this work is the unweighted sum of ten probabilities;

$$\hat{y} = argmax(\bar{P}_{log-Mel} + \bar{P}_{GAM} + \bar{P}_{CQT} + \bar{P}_{Add} + \bar{P}_{Con}) \quad (3)$$

2.3. Data Augmentation

By increasing data variation, data augmentation has shown itself effective at improving performance in ASC task. In this case we apply the mixup technique, to improve between-class training. Let X_1, X_2 and y_1, y_2 be the original inputs fed into a learning model and expected one-hot labels from two classes, respectively. From this we generate new mixup data, as follows;

$$X_{mp1} = X_1 * \lambda + X_2 * (1 - \lambda) \quad (4)$$

$$X_{mp2} = X_1 * (1 - \lambda) + X_2 * \lambda \quad (5)$$

$$y_{mp} = y_1 * \lambda + y_2 * (1 - \lambda) \quad (6)$$

$$y_{mp2} = y_1 * (1 - \lambda) + y_2 * \lambda \quad (7)$$

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

We feed both original data and generated mixup data into learning models to double batch size from 100 to 200, and considerably extending the training time of model. In this work, we apply this technique to both the pre-trained CNN (mixup on patch size) and the post-trained DNN (mixup on global mean/max pooling vector).

3. Experiment Result And Discussion

3.1. Datasets

This work employs the DCASE2016 dataset [24]. In this dataset, audio signals were recorded at a sample frequency at 44.1 kHz with a 30 s recording duration for every audio file. The data is subdivided into two sets; a development set (Dev Set) and an evaluation set (Eva Set), one for training and another for evaluating, with 15 classes as detailed in Table 4. In total, the development and evaluation sets comprise 13 hours of data.

Table 4: *DCASE2016 Dataset*

Class	Dev Set	Eva Set
Beach	78	26
Bus	78	26
Cafe/Restaurant	78	26
Car	78	26
City center	78	26
Forest Path	78	26
Grocery Store	78	26
Home	78	26
Library	78	26
Metro station	78	26
Office	78	26
Park	78	26
Residential area	78	26
Train	78	26
Tram	78	26

3.2. Results On The DCASE2016 Dataset

According to the proposed baseline architecture, different spectrograms (GAM, log-Mel and CQT models) and combined forms (additive and concatenated models) are explored. The average accuracy obtained over the evaluation set, compared to the DCASE2016 baseline [25], is reported in Table 5.

Table 5: *Performance comparison for DCASE2016 task [25].*

System	Eva Set
DCASE2016 [25]	77.2
log-Mel model	83.3
GAM model	84.6
CQT model	82.1
Additive model	85.4
Concat model	85.9
Ensemble	90.3

As can be seen, all of the single models exceed the accuracy of the baseline. Both the additive model and the concatenative model outperform single-spectrogram models, but the ensemble over 10 results (using 5 models) yields significantly better performance, a 13% improvement over evaluation set accuracy.

The confusion matrix of that system is given in Fig. 4. It is interesting to note that incorrect cases occur around three main groups, between *home* and *library* classes, *cafe/restaurant* and *grocery store*, and among *train*, *tram*, *cafe/restaurant*. In future we consider that mixup augmentation focussing on those classes may be beneficial, but we also believe that attention learning would improve performance further.

Table 6 further compares the accuracy of the top results published in the DCASE2016 challenge, and that of four state-of-the-art recently published papers. Comparing our method with the top ten results from the DCASE2016 challenge [25], the fusion methods [5, 8, 32] achieve higher performance than

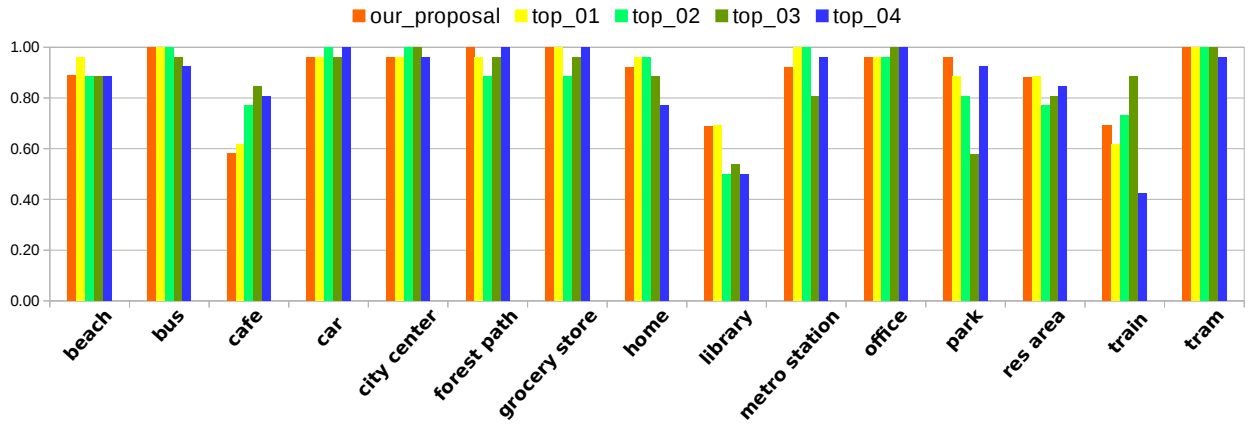

Figure 3: *Performance comparison over every class against top four results from the DCASE2016 challenge [25] on the Eva set.*

Table 6: *Comparison between the top-ten DCASE2016 competition architecture accuracies (top), recently published papers using DCASE2016 data (middle), and the proposed method (bottom) on the DCASE2016 Eva dataset.*

System	Classifier	Accuracy
Bae et al. [26]	CNN-RNN	84.1
Lee et al. [27]	CNN	84.6
Lee et al. [28]	CNN ensemble	85.4
Takahashi et al. [29]	DNN-GMM	85.6
Kumar et al. [30]	SVM	85.9
Valenti et al. [31]	CNN	86.2
Marchi et al. [32]	Ensemble	86.4
Ko et al. [8]	Ensemble	87.2
Bisot et al. [15]	NMF	87.7
Eghbal-Zadeh et al. [5]	Ensemble	89.7
Shefali Waldekar et al. [3]	SVM	81.2
Seongkyu Mun et al. [33]	DNN	86.3
Juncheng Li et al. [6]	Ensemble	88.2
Rakib Hyder et al. [13]	Ensemble	88.5
Hongwei et al. [4]	SVM	89.5
Yifang Yin et al. [2]	Ensemble	91.0
Proposed method	Ensemble	**90.3**

single models [26, 29, 31, 30]. We also list the accuracy of recently published methods [13, 6, 2] and our proposed model outperforms all DCASE2016 results as well as the most recent methods apart from the system of Yin et al. [2]. Like our system, they use three features, but require highly complex 3D CNN, 2D CNN and high resolution waveform classifiers. They thus combine extremely high network complexity with an extremely high processing rate (for waveform data). By contrast, we use multiple shallow classifiers that are relatively simple, and which operate at a much slower frame rate. These results are further explored in Fig. 3, which shows the performance of every class from those top four results of DCASE2016 challenge. As can be seen, the accuracy of our method is competitive to the four systems for most of the DCASE2016 challenge classes, with the notable exception of the *cafe/restaurant* class.

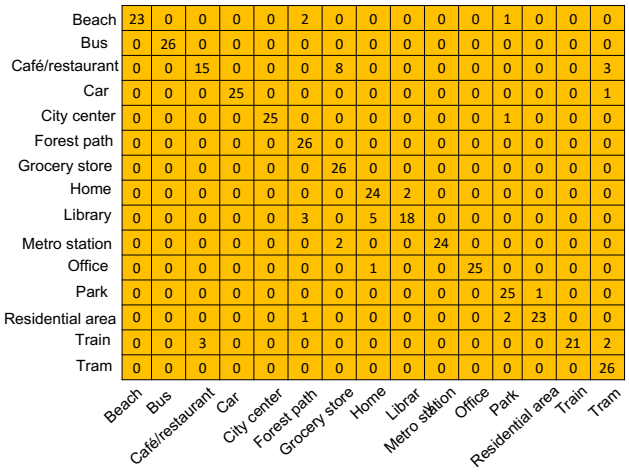

Figure 4: *Confusion matrix for the DCASE2016 Eva dataset*

4. Conclusion

This paper has presented an exploration of ensemble features and models for acoustic scene classification. Using a feature approach based upon three kinds of time-frequency transformation (namely log-Mel, Gammatone filter and constant Q transform), the back-end classification, a two-step training method, performs well. To deal with challenges implicit in the ASC task, we investigate whether different time-frequency spectrogram types can be combined, and whether the pre-/post-trained process can improve classification accuracy. In terms of result, the classification accuracy obtained from an ensemble of features through a pre-trained CNN and a post-trained DNN performs well. Evaluating on experiments using the DCASE2016 dataset, the proposed method achieves highly competitive results compared to state-of-the-art systems. In future, further experiments will be conducted on applying attention techniques as well as exploitation of emerging classifier structures from associated domains.

5. 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 *INTERSPEECH*, 2018, pp. 3323–3327.

[4] 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.

[5] 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.

[6] 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.

[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. (http://www.ee.columbia.edu/dpwe/resources/matlab/ gammatonegram) Gammatone-like spectrogram.

[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] 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. IEEE, Apr. 2015, pp. 559–563.

[12] 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.

[13] 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 *INTERSPEECH*, 2017, pp. 3073–3077.

[14] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PLoS ONE*, vol. 9, no. 7, p. e100795, 07 2014.

[15] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Supervised nonnegative matrix factorization for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[16] 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.

[17] 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.

[18] 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.

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

[20] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, and S. Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.

[21] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.

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

[23] I. V. McLoughlin, *Speech and Audio Processing: a MATLAB-based approach*. Cambridge University Press, 2016.

[24] T. Heittola, A. Mesaros, and T. Virtanen, "DCASE2016 baseline system," DCASE2016 Challenge, Tech. Rep., September 2016.

[25] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 26, no. 2, pp. 379–393, 2018.

[26] S. H. Bae, I. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN," DCASE2016 Challenge, Tech. Rep., September 2016.

[27] Y. Han and K. Lee, "Convolutional neural network with multiple-width frequency-delta data augmentation for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

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

[29] G. Takahashi, T. Yamada, S. Makino, and N. Ono, "Acoustic scene classification using deep neural network and frame-concatenated acoustic feature," DCASE2016 Challenge, Tech. Rep., September 2016.

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

[31] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[32] 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.

[33] S. Mun, S. Shon, W. Kim, D. K. Han, and H. Ko, "Deep neural network based learning and transferring mid-level audio features for acoustic scene classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 796–800.