Audio Engineering Society

Conference Paper

Presented at the Conference on
Audio Forensics
2019 June 18 – 20, Porto, Portugal

Bag-of-Features Models Based on C-DNN Network for Acoustic Scene Classification

Lam Pham[1], Ian McLoughlin[1], Huy Phan[1], Ramaswamy Palaniappan[1], and Yue Lang[2]

[1]*The University of Kent, School of Computing, Medway, UK*
[2]*Huawei Technologies Co. Ltd., Shenzhen, China*

Correspondence should be addressed to Lam Pham (ldp7@kent.ac.uk)

ABSTRACT

This work proposes bag-of-features deep learning models for acoustic scene classification (ASC) – identifying recording locations by analysing background sound. We explore the effect on classification accuracy of various front-end feature extraction techniques, ensembles of audio channels, and patch sizes from three kinds of spectrogram. The back-end process presents a two-stage learning model with a pre-trained CNN (preCNN) and a post-trained DNN (postDNN). Additionally, data augmentation using the mixup technique is investigated for both the pre-trained and post-trained processes, to improve classification accuracy through increasing class boundary training conditions. Our experiments on the 2018 Challenge on Detection and Classification of Acoustic Scenes and Events - Acoustic Scene Classification (DCASE2018-ASC) subtask 1A and 1B significantly outperform the DCASE2018 reference implementation and approach state-of-the-art performance for each task. Results reveal that the ensemble of multi-spectrogram features and data augmentation is beneficial to performance.

1 Introduction

Acoustic scene classification (ASC), identifying the type of location at which a recording was made, is sometimes considered one of the main tasks of a recently emerging research field named "machine hearing" [1]. The task is essentially to recognise environmental sounds, and is applicable to a wide range of applications including acoustic surveillance [2], robotic navigation [3], context awareness [4] and in recording analysis. Acoustic event detection is a closely related task of recognising the occurrence of events from the sounds that they make [5, 6]. Considering the attributes of a recording from a given environment, it usually contains acoustic elements of both a background sound field and various sporadic foreground events. Moore et al. [7] coined the term 'roomprint' to describe the unique sounds of a particular environment. While certain locations contain very distinct events, these can be mixed with indistinct events that also occur in a wide variety of other scenes, complicating the task of forensic audio analysis. Yet forensic acoustics is already an established field, able to associate specific events from their acoustic signatures [8], automatically and with im-

proving levels of speed and accuracy.

If the background is considered noise and the foreground events are considered signal (as is common in audio recordings), the signal-to-noise ratio in forensic audio analysis can be highly variable. Noise levels could be high or low, as can signal levels, due to diverse environments and recording conditions. These variabilities make the ASC task challenging, particularly if it only focuses on performing either background noise classification or isolated event classification.

While there are a number of established methods of evaluating both sound scene and sound event classification [9], the recent challenge on Detection and Classification of Acoustic Scenes and Events (DCASE2018) contains ASC challenge task A [10] to recognise 10 different acoustic scenes. This task will be used in Section 3, to assess performance.

Most recent ASC research, including much of that entered for DCASE2018, classifies an entire spectrogram from one sound scene recording. Techniques tend to make use of mel filter and log energy spectrograms, classified by deep learning model backends [11, 12, 13]. The size and complexity of the input spectrogram probably explains why the main contributions usually focus on constructing and training powerful learning models, but there has been little research on front-end feature extraction. For instances, while Mariotti et al. [12] experimented on a very wide range of deep learning models (VGG8, VGG10, VGG12, Resnet 18, Resnet 34, Resnet 50), attention-based pooling layers proposed by Zhao Ren et al. [11] helped to improve the quality of pooling layers compared with traditional global pooling layers. Other research published by Liping Yang et al. [13] developed a very complex CNN model called the Xception architecture in which information from a previous convolution layer is added into the next convolution layer like the Resnet architecture.

However, decreasing gains are being achieved through more complex architectures, motivating us to focus on front-end features. In particular we look at the effect of different features on classification accuracy. With the inspiration that the same sound leads to different shapes of time-frequency information when displayed using different kinds of spectrogram, we propose investigating accuracy from classification using three kinds of spectrogram. Specifically, we use the Gammatone filter spectrogram [14] (also known as a Gammatonegram), log-mel spectrogram and Constant-Q transform (CQT) [15] spectrogram.

Time-frequency shapes that are characteristic of different sound scenes have a variety of durations and also differ in frequency range and resolution, thus it is interesting to explore the extraction and analysis of different feature durations from full-size spectrograms (specifically, we evaluate an ensemble of 0.3, 0.6, 1.2 and 2.4 s non-overlapping patches). Furthermore, all recordings in the DCASE dataset have at least two audio channels (stereo). Those two channels were obtained from microphones worn in the ears of human listeners, and so can roughly be denoted 'left' and 'right'. Since it is known that humans achieve significant intelligibility gain from using stereo information (see Chapter 4 of [16]), it is likely that some of this gain could also be achieved by well-trained deep neural networks. We therefore evaluate performance from classifying these separately as well as from classifying their mean.

For back-end learning models, CNNs have proven their capability for image recognition, and since a spectrogram is effectively an image, they performed well when first evaluated for sound event detection [17]. Most currently published DCASE2018 architectures follow a trend of enhancing network architecture to better learn from input features [11, 12, 13, 18, 19, 20]. Unfortunately ever increasing complexity significantly increases computational overhead, yet yields decreasing performance gains. Thus, inspired by ideas of transferred learning models proposed in [21] or transferred low-level features [22], we believe that if we transfer results of a first training process used to model low-level features, into a second model to aggregate those features, it can improve classification accuracy without requiring an unduly complex network architecture. Indeed, one system proposed by Jee et al. in DCASE2018 [23] proves that a two-stage training process by very traditional Gaussian methods could obtain competitive results. This paper therefore proposes using a pre-trained CNN architecture with four convolution blocks (each

comprising one convolutional and one ReLU layer) followed by three fully-connected layers for feature extraction. This is then followed by a post-trained DNN architecture with four fully-connected layers to deeply learn from features extracted by the pre-trained CNN. The main contribution of this proposed back-end learning model is to explore the fusion of both pre-trained CNN and post-trained DNN results (rather than to concentrate on its other benefits, such as reduced network complexity).

To improve the learning model, various techniques of data augmentation are also evaluated, such as adding noise to training data [24, 25] or frequency shifting [25]. These techniques have also been used by Lu et al. [26] showing that pitch shifting is the most effective. Furthermore, we apply a data augmentation technique called mixup, that has recently been applied to related research areas [27, 28] and appears to yield good results.

The remaining sections of this paper describe the system, present experiment conditions and results before a discussion and final conclusion.

2 System description

2.1 High level architecture

The general architecture of the proposed system, depicted in Fig. 1, has two main processes. The first one (top half) has the role of transforming the selected audio channels into one or more types of spectrogram, and then splitting a full spectrogram into smaller patches of different sizes to form a bag-of-features. The second process (bottom half) includes a pre-trained CNN model and a post-trained DNN model. The former receives input from the patches extracted by the first process and is trained on frame-level labels from the training data set. Once the CNN is trained on the bag-of-features, the CNN final output layer activations are fed from that network into the DNN as input features. The DNN is in turn then trained on recording-level labels from the training data set. A final classification result is obtained from the post-trained DNN model.

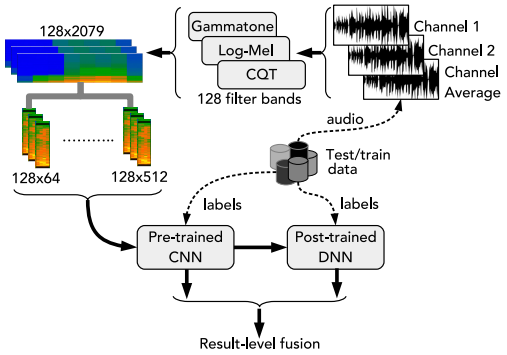

Fig. 1: Proposed system

2.2 Baseline system

Since this work analyses ensembles of different bags-of-features and different challenge tasks of DCASE2018, we establish a baseline proposal, as detailed in Table 1, to compare the effect of different systems against. Additionally, we will compare overall performance against the standard DCASE2018 baseline system of [10], also shown in Table 1.

Note that our baseline proposal uses a smaller spectrogram window size of 0.025 s (albeit extracted from a higher dimension spectrogram) and a smaller hop size of 20% respectively compared with a window size at 0.04s and 50% overlap in the DCASE2018 baseline, since we aim to extract additional useful information from the input spectrogram. Regarding the spectrogram, we use a log-Mel filter [15] with 128 bands which is much larger than the 40 Mel filter bands of the DCASE2018 baseline. Based on these parameters, the proposed spectrogram has a bigger size in both time and frequency bins of 128×2079 compared with 40×500 in the DCASE2018 baseline, although our system splits the proposed spectrogram into smaller patches with size 128×128 before feeding them into the back-end process and learning models of our proposed system.

As regards learning models, while the DCASE2018 baseline reuses the architecture of the top ranked submission of DCASE2016 [29], with only two convolution blocks (convolution, batchnorm, RELU and dropout layers) followed by one dense

Table 1: The primary operating parameters of the proposed and DCASE2018 baseline systems.

Parameters	Proposed baseline	DCASE2018 baseline
Window size	0.025s	0.04s
Overlap	80%	50%
Method	log-mel	log-mel
Bands	128	40
Spectrogram	128×2079	40×500
Features	16 patches (128×128)	the whole spectrogram
Learning model	pre-trained CNN+ post-trained DNN	CNN

layer and an output layer with softmax classification, this proposal is for an enhanced learning model with the pre-trained CNN and the post-trained DNN, listed in Table 2 and shown in Fig. 2. The pre-trained CNN model presents four convolutional blocks which include a convolutional layer followed by max pooling layer without batchnorm and dropout layers. At the final convolution block, instead of using a max pooling layer, a global-mean pooling layer is applied to enhance the accuracy based on the idea of considering the contribution of all channels output to the final convolution block as a bag-of-features and reducing noise. The role of classification is handled by the next three fully-connected layers, and at the final layer softmax function, minimizing the cross-entropy to tune parameters, denoted as θ,

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

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

Inspiration from transfer learning techniques [21, 22] indicates that if parameters in a model could be improved when they are transferred into another model and further trained, the low-level features extracted from middle layers could be, too. Therefore, we propose the post-trained DNN in Fig. 2 that is used to train the extracted global mean of channels at the final convolution layer block of the pre-trained CNN. In order to tune the parameters of the post-trained DNN, a similar loss function as in eqn. (1) is adopted. Both the pre-trained CNN and post-trained DNN are built in the Tensorflow framework, set with batch size and learning rate of 100 and 0.0001 respectively, and using the Adam method for learning rate optimisation [30].

Table 2: Detail of the pre-trained CNN architecture.

Layer	Output shape	Kernel
Conv 1	128x128x32	3x3
Max pooling 1	64x64x32	2x2
Conv 2	64x64x64	3x3
Max pooling 2	32x32x64	2x2
Conv 3	32x32x128	3x3
Max pooling 3	16x16x128	2x2
Conv 4	16x16x256	3x3
Global mean pool	256	
Fully connected 1	512	
Fully connected 2	1024	
Fully connected 3	10	

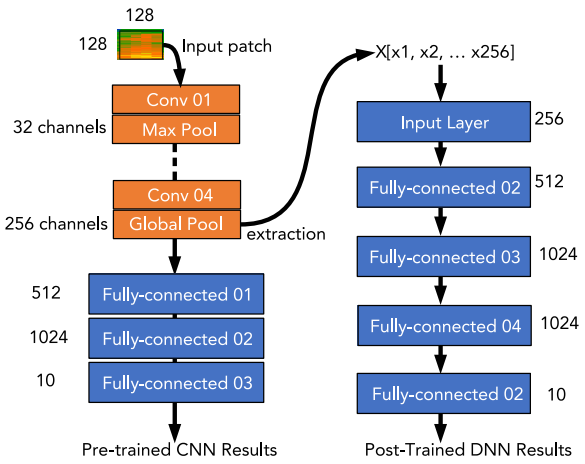

Fig. 2: Pre-trained CNN and post-trained DNN architecture, and detail of the low level feature extraction from the CNN.

Table 3: Bag-of-feature analysis

Channel	channel 01, channel 02, (channel 01 + channel 02)/2
Spectrogram transformation	Gammatone, log-Mel and CQT (each 128 filters)
Different patch sizes	0.3 s, 0.6 s 1.2 s, 2.4 s

2.3 Bag-of-features analysis

Due to the unique challenges of the ASC task as discussed in the introduction above, we consider ensemble multi-resolution models to be promising to enhance the classification accuracy over the current state-of-the-art. Indeed, almost all publications related to DCASE2018 apply a wide range of ensemble models. However, as mentioned, most authors focus on improved learning models while maintaining consistent input features. This motivates us to analyse and explore the effects on classification accuracy of different bags-of-features, divided into three groups shown in Table 3.

The first exploration focuses on the effect of different channels, including the first channel alone, the second channel alone, and the mean of both channels. The second exploration considers using three different spectrogram transformations (namely the gammatone filter [14], CQT and log-mel [15]) with the same filter dimension of 128. By using similar parameters such as window size, hop size and patch number, the amount of data fed into the back-end learning model will be similar among the three types of spectrogram. The third exploration is for different sizes of patch, as regards the number of time bins. We explore patches with fixed 128 frequency bins but different time bins of 64, 128, 256 and 512, respectively. All bag-of-features explorations are made by comparing performance (percentage classification accuracy) against the baseline systems.

2.4 Fusion methods

There are two kinds of fusion method used in this work, namely a mean strategy and a max strategy. For the former, if we consider $P^n_{CNN} = (P^n_1, P^n_1, ..., P^n_C)$, with C being the class number and the n^{th} out of N patches fed into learning model, as probability of test sound scene instance at the pre-trained CNN, the mean classification probability is denoted as $\bar{P}_{CNN} = (\bar{P}_1, \bar{P}_2, ..., \bar{P}_C)$ where,

$$\bar{P}_c = \frac{1}{N}\sum_{n=1}^{N} P^n_c \quad for \;\; 1 \leq n \leq N \tag{2}$$

and the predicted label from the pre-trained CNN is determined using,

$$\hat{y} = argmax(\bar{P}_1, \bar{P}_2, ..., \bar{P}_C) \tag{3}$$

Regarding the second strategy, the overall probability per class is defined as the maximum over all N patch classifications,

$$\bar{P}_c = max(P^n_c) \quad for \;\; 1 \leq n \leq N \tag{4}$$

and again, eqn. 3 is used to predict the overall label.

Eventually, the final fusion between results of the pre-trained CNN and the post-trained DNN is determined as the class achieving the highest overall probability,

$$\hat{y} = argmax(\bar{P}_{CNN} + \bar{P}_{DNN}) \tag{5}$$

2.5 Mixup data augmentation

Inspired by the robust learning abilities of deep networks, data augmentation (DA) which makes data contain more variability, has proven effective. DA has been widely applied to the ASC task [24, 25, 26], but in this this paper, in addition to traditional noise and pitch shift augmentation, we we apply a mixup technique for DA. This is used to provide training augmentation that reenforces two-class training. To explain this technique, consider X_1, X_2 and Y_1, Y_2 to be the input feature fed into a model and the expected one-hot labels of two different classes. respectively. Mixup data is then generated using the following equations,

$$X_{mp1} = X_1 * \lambda + X_2 * (1 - \lambda) \tag{6}$$

$$X_{mp2} = X_1 * (1-\lambda) + X_2 * \lambda \quad (7)$$

$$Y_{mp} = Y_1 * \lambda + Y_2 * (1-\lambda) \quad (8)$$

$$Y_{mp2} = Y_1 * (1-\lambda) + Y_2 * \lambda \quad (9)$$

with $\lambda \in U(0,1)$ is a random mixing coefficient that is obtained both using uniform distribution and beta distribution.

For each batch size of 100, we split into two sub-batches of size 50 each. From those, we obtain new 100 mixup data samples. Eventually, we mix both the original data and the mixup data randomly before feeding into the learning model. This work applies the mixup technique on patch sizes at the pre-trained CNN, as well as over the mean pooling vector for the post-trained DNN. It is therefore applied separately to both networks during their respective training.

3 Results and discussion

3.1 Data Set Up

Experiments in this paper are conducted on DCASE2018 task 1 subtasks A and B [10]. For subtask 1A, the audio files in the dataset are all similar wave file format recordings with sample rate 48k Hz and 10 s duration. All were recorded using the same device (namely device A), and were grouped into ten classes, with one class label per recording. However, the data is unbalanced so that the number of recordings per class is slightly uneven, as reported in Table 4. In total, the task includes 8640 audio files. Using the DCASE2018 suggested test/train split [10], recordings are separated into a training subset (6122 audio files) and a test subset (2518 audio files). In this work, the pre-trained CNN model trained first. Once the model converges on the training set, the 256-dimensional global means vectors (shown in Table 2, extracted before the fully connected layers) are fed into the post-trained DNN model for its post-training process. The final accuracy score is obtained from the post-trained DNN softmax output.

For DCASE2018 subtask 1B, all audio files from the subtask 1A mentioned above are reused. In addition, more recordings are obtained that were recorded on two different devices (namely B and C). However, the number recordings from devices B and C is much smaller, totalling 4-hours recording compared to 24 hours from device A. Final performance ranking for subtask 1B is only based on classification accuracy over device B and device C. More intensive experiments are performed using subtask 1A, to determine the most effective model, which is then trained and evaluated on subtask 1B.

Table 4: The number of recordings in each class for DCASE2018 subtasks 1A and 1B.

Task	1A Train	1A Test	1B Train	1B Test
Airport	599	265	707	301
Bus	622	242	730	278
Metro	603	261	711	297
Metro Station	605	259	713	295
Park	622	242	730	278
Public Station	648	216	756	252
Shopping Mall	585	279	693	315
Pedestrian Street	617	247	725	283
Traffic Street	618	246	726	282
Tram	603	261	711	297
Total files	6122	2518	7202	2878

3.2 Results on DCASE2018 Subtask 1A

3.2.1 Baseline comparison

The accuracy of every class reported by the DCASE2018 baseline and by the proposed methods is displayed in Table 5, with the individual contributions of the pre-trained CNN and post-trained DNN networks shown separately. In general, the pre-trained CNN result shows a lower accuracy of 58.3% compared to the 59.7% of the DCASE2018 baseline. But when we apply a deeper training process on the extracted features using the post-trained DNN, accuracy is enhanced by 2% absolute compared to the baseline DCASE2018 reference implementation.

3.2.2 Bag-of-channel ensembles

Working from the proposed baseline, the effect on classification accuracy was analysed when using the three different channel arrangements shown in Table 3. Results are presented in Table 6 and reveal that channel 1, channel 2 and their average differ

Table 5: Performance comparison between DCASE2018 baseline (Ref.) and the proposed baseline networks, presented separately, in terms of percentage classification accuracy.

Class	Ref.	preCNN	postDNN
Airport	72.9	63.7	56.2
Bus	62.9	63.2	67.4
Metro	51.2	46.4	31.4
Metro station	55.4	49.0	68.7
Park	79.1	71.1	72.3
Public square	40.4	60.6	59.3
Shopping mall	49.6	57.0	74.9
Pedestrian stree	50.0	28.3	29.5
Traffic in street	80.5	82.1	82.9
Tram	55.1	66.3	74.3
Average	**59.7**	**58.3**	**61.7**

more widely at the output of the pre-trained CNN with the highest score at 58.3% for channel 1, but the results from the post-trained DNN are similar. Ensemble models exploiting different channels from either the pre-trained CNN or the post-trained DNN improve accuracy only slightly.

Table 6: Channel effect

	preCNN	postDNN
Channel 01	58.3	61.7
Channel 02	50.8	58.0
Sum Average	53.2	60.2
Ensemble	**61.4**	**63.1**

3.2.3 Bag-of-feature-size ensembles

This section presents an experiment to determine whether ensembles of different patch sizes can improve accuracy, shown in Table 7. In general, although the pre-trained CNN and post-trained DNN results on different sizes are not significantly improved, ensemble result over all patch sizes improve performance by neatly 5% over the DCASE2018 baseline. It is more effective than channel ensembles.

3.2.4 Bag-of-spectrogram ensembles

We next determine the effect of using the three spectrogram transformation types listed in Table 3 for classification of patches sized 128×128 as in the proposed baseline. The results, shown in Table 8, indicate that the best pre-trained CNN result is 58.4%, for the Gammatone spectrogram. However the best post-trained DNN result comes from the Log-Mel filter approach with 61.7%. Noticeably, an ensemble among spectrograms at either the pre-trained CNN or the post- trained DNN helps to improve the accuracy more than 10% compared with the DCASE2018 baseline.

Table 7: Patch size effect

	preCNN	postDNN
0.3s	60.3	61.5
0.6s	58.3	61.7
1.2s	59.1	62.2
2.4s	61.1	59.1
Ensemble	**64.5**	**63.9**

Table 8: Spectrogram effect

	preCNN	postDNN
GAM	58.4	59.3
Log-mel	58.3	61.7
CQT	54.2	57.4
Ensemble	**70.4**	**71.1**

Table 9: Mixup data augmentation effect, listing preCNN/postDNN accuracy separately.

	No mixup	With mixup
GAM	58.4 / 59.3	69.1 / 70.5
Log-mel	58.3 / 61.7	62.9 / 66.4
CQT	54.2 / 57.4	59.2 / 63.0
Ensemble	**70.4 / 71.1**	**72.4 / 74.8**

3.2.5 Data augmentation effect

By intensively investigating various aspects of input features, a bag-of-spectrogram approach shows effective to improve the classification accuracy. To further enhance the performance, the mixup data augmentation mentioned above is applied for both the pre-trained and the post-trained processes. The results are described in Table 9, presented separately for three different spectrogram systems, and indicate that mixup data augmentation can significantly improve classification accuracy. In particular, both pre-trained CNN and post-trained DNN

Table 10: Results on DCASE2018 sub task 1B

	Baseline			**Our method**		
	Device B	Device C	mean(B&C)	Device B	Device C	mean(B&C)
Airport	68.9	76.1	72.5	83.3	83.3	83.3
Bus	70.6	86.1	78.3	83.3	100	91.7
Metro	23.9	17.2	20.6	55.6	44.4	50.0
Metro station	33.9	31.7	32.8	61.1	55.6	58.3
Park	67.2	51.1	59.2	84.4	77.8	86.1
Public square	22.8	26.7	24.7	22.2	38.9	30.6
Shopping mall	58.3	63.9	61.1	83.3	83.3	83.3
Street pedestrian	16.7	25.0	20.8	38.9	44.4	41.7
Street traffic	69.4	63.3	66.4	88.9	88.9	88.9
Tram	18.9	20.6	19.7	55.5	61.1	58.3
Average	**45.1**	**46.2**	**45.6**	**66.7**	**67.8**	**67.3**

results are improved by at least 4% compared to the previous results. When we fuse three post-trained DNN results (over three spectrogram inputs), we achieve the highest accuracy, improving the DCASE2018 baseline (59.7%) by over 15%. However our result, at 74.8%, does not quite match the best public leaderboard result for DCASE 2018 at the time of writing, which is 79.8% [13], which uses a vastly more complex Xception architecture (with 22 million parameters; compared to just 1.05 million for preCNN and 1.72 million for postDNN in our system).

3.3 Results on DCASE2018 Subtask 1B

Since multi-spectrogram input and mixup data augmentation both show their effectiveness compared to ensembles of channels or feature size, they are also applied to DCASE2018 subtask 1B. The results obtained are listed in Table 10, and indicate that our proposed method significantly outperforms the DCASE2018 baseline for both device B and C, by over 21% (noting that accuracy over all classes are improved), however falling 2nd on the public leaderboard after Xception [13], at time of writing.

4 Conclusion

This paper has presented several explorations of different ensemble models for classification of acoustic scenes in forensic acoustic analysis. Our approach is based on a front-end bag-of-features allied with a back end consisting of a pre-trained convolutional neural network feature extractor, feeding a deep neural network classifier. To deal with challenges implicit in the ASC task, we investigated whether features from different time-frequency spectrogram types, and whether different classification patch durations and channel splits could improve classification accuracy. The findings indicate that complimentary information in different spectrograms, different channels and to some extent in different patch durations can be exploited to increase accuracy, with by far the largest improvement coming from ensemble methods on all three spectrogram types, allied with mixup data augmentation. For future research, we plan further investigation on techniques of transfer learning and post-trained models, as well as explore different spectrogram fusions over different durations of patch, since it is likely that optimal patch durations differ for the various spectrogram types. We also believe that a combination of bag-of-features front-end processing and more effective back-end learning models will enable us to obtain better results in the ASC task.

References

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

[2] Viet, Q. N., Kang, H., Chung, S. T., Cho, S., Lee, K., and Seol, T., "Real-time audio surveillance system for PTZ camera," *International Conference on Advanced Technologies for Communications*, pp. 392–397, 2013.

[3] Chu, S., Narayanan, S., Kuo, C. . J., and Mataric, M. J., "Where am I? Scene Recognition for Mobile Robots using Audio Features," in *2006 IEEE International Conference on Multimedia and Expo*, pp. 885–888, 2006.

[4] Ravindran, S. and Anderson, D., "Audio Classification And Scene Recognition and for Hearing Aids," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 860–863, 2005.

[5] Heittola, T., Mesaros, A., Eronen, A., and Virtanen, T., "Context-Dependent Sound Event Detection," *Eurasip J.Audio, Speech, and Music Processing*, (1), pp. 1–13, 2013.

[6] Mascia, M., Canclini, A., Antonacci, F., Tagliasacchi, M., Sarti, A., and Tubaro, S., "Forensic and anti-forensic analysis of indoor/outdoor classifiers based on acoustic clues," in *23rd European Sig. Proc. Conf. (EUSIPCO)*, pp. 2072–2076, 2015.

[7] Moore, A. H., Brookes, M., and Naylor, P. A., "Roomprints for forensic audio applications," in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–4, IEEE, 2013.

[8] Maher, R. C., "Lending an ear in the courtroom: forensic acoustics," *Acoustics Today*, 11(3), pp. 22–29, 2015.

[9] McLoughlin, I., Zhang, H., Xie, Z., Song, Y., Xiao, W., and Phan, H., "Continuous robust sound event classification using time-frequency features and deep learning," *PloS one*, 12(9), p. e0182309, 2017.

[10] Mesaros, A., Heittola, T., and Virtamen, T., "A Multi-Device Dataset For Urban Acoustic Scene Classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, 2018.

[11] Zhao, R., Qiuqiang, K., Kun, Q., Mark, D., and Bjorn, W., "Attention-Based Convolutional Neural Networks For Acoustic Scene Classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, pp. 39–43, 2018.

[12] Mariotti, O., Cord, M., and Schwander, O., "Exploring Deep Vision Models For Acoustic Scene Classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, pp. 103–107, 2018.

[13] Yang, L., Chen, X., and Tao, L., "Acoustic Scene Classification Using Multi-Scale Features," in *Detection and Classification of Acoustic Scenes and Events 2018*, pp. 29–33, 2018.

[14] Ellis, D. P. W. ., "Gammatone-like spectrogram," http://www.ee.columbia.edu/dpwe/resources/matlab/ gammatonegram.

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

[16] McLoughlin, I. V., *Speech and Audio Processing: a MATLAB-based approach*, Cambridge University Press, 2016, ISBN 9-781-10708-5466.

[17] Zhang, H., McLoughlin, I., and Song, Y., "Robust Sound Event Recognition using Convolutional Neural Networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2635, pp. 559–563, IEEE, 2015.

[18] Nguyen, T. and Pernkopf, F., "Acoustic Scene Classification Using A Convolutional Neural Network Ensemble And Nearest Neighbor Filters," in *Detection and Classification of*

Acoustic Scenes and Events 2018, pp. 34–38, 2018.

[19] Zeinali, H., Burget, L., and Cernocky, J., "Convolutional Neural Networks And X-Vector Embedding For DCASE2018 Acoustic Scene Classification Challenge," in *Detection and Classification of Acoustic Scenes and Events 2018*, pp. 202–206, 2018.

[20] Roletscheck, C., Watzka, T., Seiderer, A., Schiller, D., and André, E., "Using An Evolutionary Approach To Explore Convolutional Neural Networks For Acoustic Scene Classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, pp. 158–162, 2018.

[21] Diment, A. and Virtanen, T., "Transfer Learning of Weakly Lable Audio," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 18–25, 2017.

[22] Phan, H., Hertel, L., Maass, M., Koch, P., Mazur, R., and Mertins, A., "Improved Audio Scene Classification Based on Label-Tree Embeddings and Convolutional Neural Networks," *IEEE/ACM Transactions On Audio, Speech, And Language Processing*, 25(6), pp. 1278–1290, 2017.

[23] Jee-weon, Jung, H.-s., Heo, H.-j., Shim, H.-j., and Yu, "DNN Based Multi-Level Feature Ensemble For Acoustic Scene Classification," in *Detection and Classification of Acoustic Scenes and Events 2018*, pp. 118–122, 2018.

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

[25] Salamon, J. and Bello, J. P., "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, 24(3), pp. 279–283, 2017.

[26] Lu, R., Duan, Z., and Zhang, C., "Metric learning based data augmentation for environmental sound classification," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5, 2017.

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

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

[29] Michele, Valenti, A., Diment, G., Parascandolo, S., Squartini, T., and Virtanen, "DCASE 2016 Acoustic Scene Classification Using Convolutional Neural Networks," in *Detection and Classification of Acoustic Scenes and Events 2016*, pp. 95–99, 2016.

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