Wider or Deeper Neural Network Architecture for Acoustic Scene Classification with Mismatched Recording Devices

Lam Pham
Lam.Pham@ait.ac.at
Austrian Institute of Technology
Austria

Khoa Tran
tdkhoa@dut.udn.vn
Da Nang University
Viet Nam

Dat Ngo
dn22678@essex.ac.uk
University of Essex
UK

Hieu Tang
hieutq10@fpt.edu.vn
FPT University
Viet Nam

Son Phan
son.pl@amanotes.com
Amanotes Company
Viet Nam

Alexander Schindler
Alexander.Schindler@ait.ac.at
Austrian Institute of Technology
Austria

ABSTRACT

In this paper, we present a robust and low complexity model for Acoustic Scene Classification (ASC), the task of identifying the scene of an audio recording. We firstly construct an ASC model in which a novel inception-residual-based network architecture is proposed to deal with the issue of mismatched recording devices. To further improve the model performance but still satisfy the low footprint, we apply two techniques of ensemble of multiple spectrograms and model compression to the proposed ASC model. By conducting extensive experiments on the benchmark DCASE 2020 Task 1A Development dataset, we achieve the best model performing an accuracy of 71.3% and a low complexity of 0.5 Million (M) trainable parameters, which is very competitive to the state-of-the-art systems and potential for real-life applications on edge devices.

KEYWORDS

Deep learning, convolutional neural network (CNN), acoustic scene classification (ASC), data augmentation, model complexity, inception.

ACM Reference Format:
Lam Pham, Khoa Tran, Dat Ngo, Hieu Tang, Son Phan, and Alexander Schindler. 2022. Wider or Deeper Neural Network Architecture for Acoustic Scene Classification with Mismatched Recording Devices. In *International Conference on Multimedia (ACM Multimedia Asia 2022), December 13–16, 2022, Tokyo, Japan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3551626.3564962

1 INTRODUCTION

The Acoustic Scene Classification (ASC) task, one of main topics in 'Machine Hearing' research field [14], has attracted much research attention recently. Indeed, not only more and more ASC datasets such as Litis Rouen [32], ESC50 [30], DCASE Task 1 [3], or Crowded Scenes [26] have been published, but various ASC systems, leveraging deep neural networks, have been also proposed (i.e. The literature review section in [22] summarizes state-of-the-art ASC systems as well as updated machine learning and deep learning techniques applied for ASC).

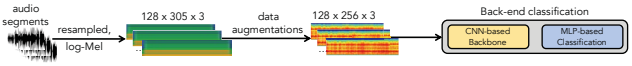

Figure 1: The high-level architecture of the proposed ASC model.

Regarding ASC challenges, they mainly come from different noise resources, various sounds in real-world environments, occurring as single sounds, continuous sounds or overlapping sounds, or dynamic energy of sound events in a sound scene recording. These challenges drive ASC research community to focus on analyzing frequency bands [11, 17, 29] rather than specific sound events [33]. However, the new issue of mismatched recording devices firstly mentioned in DCASE 2018 Task 1B challenge [3] further increases ASC challenge as this issue causes energy distribution at certain frequency bands of spectrograms from the same class significantly different (i.e. In Figure 1 of [31], Mel-based spectrograms from the same sound scene of 'on Tram' show different as they are from three different recording devices). To deal with the mismatched recording devices, ensemble of different spectrogram inputs [18, 19, 23–25, 27, 28] or ensemble of multiple classification models [5, 20] are mainly approached. However, ensemble methods present large footprint models, which is challenging to implement on edge devices or real-time applications.

This paper aims at developing an ASC model which is not only robust to deal with ASC challenges mentioned recently but also presents a low complexity with less than 1M parameters. To this end, we firstly construct an ASC model in which a novel neural network, a shallow and wide inception-residual-based architecture, is presented. The proposed ASC model is then compared with other deep learning based models using benchmark architectures such as VGGish networks (e.g., VGG16, VGG19) or residual based architectures (e.g., Resnet, DenseNet, MobileNet, or Xception) to evaluate whether a wider and shallow network or a deeper architecture is effective for ASC, specially with the mismatched recording device issue. We then apply two techniques: (1) ensemble of multiple spectrograms and (2) model compression to the proposed model,

ACM Multimedia Asia 2022, December 13–16, 2022, Tokyo, Japan

ACM ISBN 978-1-4503-9478-9/22/12.
https://doi.org/10.1145/3551626.3564962

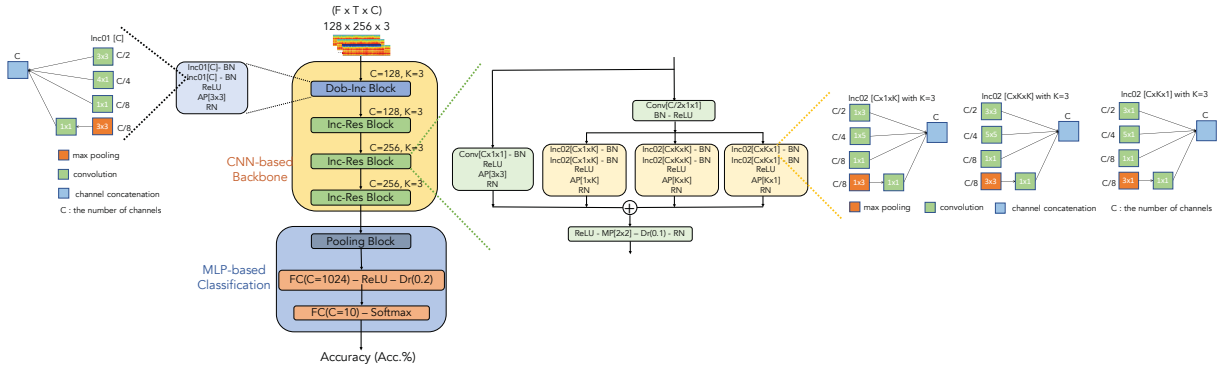

Figure 2: The proposed novel inception-residual-based deep neural network for the back-end classification.

achieve a low footprint ASC model but still perform robust and competitive to state-of-the-art systems.

2 THE PROPOSED ASC SYSTEM

We firstly construct our ASC model which presents a high-level architecture in Figure 1. As Figure 1 shows, the proposed ASC model can be separated into three main steps: The front-end feature extraction, the online data augmentations, and the back-end classification.

2.1 The front-end feature extraction

The audio recordings are firstly re-sampled to 32,000 Hz. Then, they are transformed into log-Mel spectrograms using Librosa [15]. By setting Hann window size, the hop size, the filter number to 2048, 1024, 128, respectively and applying delta, delta-delta on each spectrogram, we generate a log-Mel spectrogram of 128×305×3 from one 10-second audio segment. Notably, the channel dimension is 3, which causes by concatenating the original log-Mel spectrogram, delta, and delta-delta.

2.2 The online data augmentations

In this paper, we apply three data augmentation methods of Random Cropping [35], Specaugment [21], and Mixup [36, 37], respectively. In particular, the temporal dimension of log-Mel spectrograms of 128×**305**×3 is randomly cropped to 128×**256**×3 (e.g. Random Cropping method). Then, ten continuous and random frequency or temporal bins of the cropped spectrograms are erased (e.g. Specaugment method). Finally, the spectrograms are randomly mixed together using different ratios from Uniform or Beta distributions (e.g. Mixup method). All of three data augmentation methods are applied on each batch of spectrograms during the training process, referred to as the online data augmentations.

2.3 The back-end classification

As Figure 2 shows, the proposed back-end classification can be separated into two main parts: CNN-based deep neural network backbone and multilayer perceptron (MLP) based classification. In particular, the proposed CNN-based backbone comprises four blocks: one Inception Block and three Inc-Res Blocks as described at the upper part of Figure 2, which makes use of inception-based (e.g., Inception Block) or both inception-based and residual architectures (e.g., three Inc-Res Blocks). Three Inc-Res Blocks share the same network architecture, but channel numbers increases from 128, to 256 at two final Inc-Res Blocks. Four blocks of the CNN-based backbone are performed by Inception layers (Inc01[Channel] in Inception Block, Inc02[Channel×Kernel Size] in Inc-Res Blocks), Convolutional layer (Conv[Channel×Kernel Size]), Bach Normalization (BN) [7], Dropout (Dr(Drop Ratio) [34], Rectified Linear Unit (ReLU) [16], Max Pooling (MP[Kernel Size]), Average Pooling (AP [Kernel Size]), Residual Normalization (RN($\lambda = 0.4$)) inspired from [8]).

Regarding two Inc01 layers used in Inception Block as shown in the left part of Figure 2, we use fixed kernel sizes of [3×3], [1×1], and [4×1] (Note that using the kernel [4×1] helps to focus on frequency bands). Meanwhile, kernel sizes used in Inc02 layers in three Inc-Res Blocks as shown in the right part of Figure 2 are defined by kernel size K. By using different kernel sizes of [K×1], [K×K], and [1×K], then applying AP layers with the same kernels, and finally adding output of these AP layers together, the network can learn the distribution of energy in certain frequency bands effectively, which strengthens the network to tackle the issue of mismatched recording devices.

The MLP-based classification as shown in the lower part of Figure 2 performs a Pooling Block and two fully connected layer blocks. At Pooling Block, we extract three types of features from: (1) global average pooling across the channel dimension, (2) global max pooling across temporal dimension, and (3) global average pooling across frequency dimensions. We then concatenate these features before feeding into fully connected blocks. While the first fully connected layer (FC[Channel]) combines with ReLU and Dr, the second fully connected layer uses Softmax layer for classifying into $C = 10$ scene categories.

To further evaluate whether a wider or deeper neural network architecture is effective for ASC with the issue of mismatched recording devices, we replace the proposed CNN-based backbone by different benchmark network architectures of VGG16, VGG19, MobileNetV1, MobileNetV2, ResNet50V2, ResNet101V2, ResNet152V2, DenseNet121, DenseNet169, DenseNet201, and Xception which are available from Keras Application API [1]. In other words, only the layers before the global pooling layer of these benchmark networks are used. These reused layers are then connected with the MLP-based classification of the proposed ASC model to perform end-to-end network architectures. These network architectures

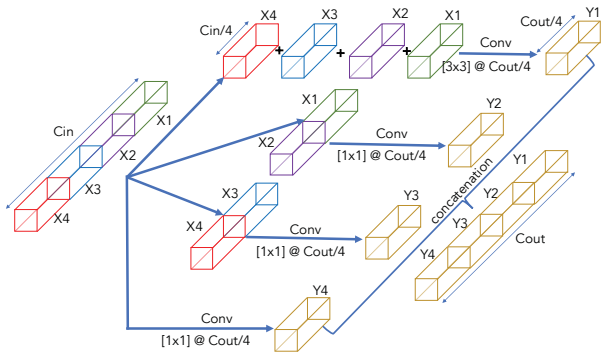

Figure 3: Channel deconvolution for reducing trainable parameters.

are then evaluated and compared with the proposed ASC model. Notably, the steps of the front-end feature extraction and the online data augmentations are retained during evaluating these network architectures.

3 ENSEMBLE METHOD AND MODEL COMPRESSION TO IMPROVE ASC MODEL

3.1 Ensemble of multiple spectrograms to improve the model accuracy

As mentioned in Section 1, an ensemble of different input spectrograms is a rule of thumb to enhance an ASC system performance. We, therefore, evaluate this ensemble strategy in our paper. In particular, we use three spectrograms of log-Mel, Constant Q Transform (CQT) [15], and Gammatone filter (Gam) [2]. By using the same settings mentioned in Section 2.1, all spectrograms present the same size of 128×305×3. For each type of spectrogram, we apply the same data augmentation methods mentioned in Section 2.2 and the proposed model presented in Section 2.3 for classification, referred to as CQT-model, log-Mel-model, and Gam-model, respectively. We then fuse the probability results by using PROD late fusion. In particular, we conduct experiments over individual network with different spectrogram inputs, then obtain predicted probability of each network as $\mathbf{p_s} = (\bar{p}_{s1}, \bar{p}_{s2}, ..., \bar{p}_{sC})$, where C is the category number and the s^{th} out of S networks evaluated. Next, the predicted probability after PROD fusion $\mathbf{p_{prod}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by:

$$\bar{p_c} = \frac{1}{S}\prod_{s=1}^{S} \bar{p}_{sc} \; for \; 1 \le s \le S \tag{1}$$

Finally, the predicted label $\hat{y}$ is determined by

$$\hat{y} = argmax(\bar{p}_1, \bar{p}_2, ..., \bar{p}_C) \tag{2}$$

3.2 Model compression techniques to reduce the model complexity

To deal with the issue of large footprint model when using ensemble of multiple spectrograms, we apply the model compression techniques: channel deconvolution and channel reduction. Regarding the channel deconvolution technique inspired from [9, 12], convolutional layers Inc02 using kernel sizes of [K×K] are re-constructed as shown in Figure 3. By using the channel deconvolution, the number of trainable parameters used in a convolutional layer with kernel [K×K] is reduced to nearly 1/8.5 of the original number. Next, we further reduce the model complexity by decreasing the channel numbers at all convolutional layers as shown in Table 1 (i.e. Only one inception layer is used in Inception Block and Inc-Res Blocks). Generally, we evaluate four cases of channel deconvolution and channel reduction, referred to as Red01, Red02, Red03, and Red04, which helps to reduce the complexity of the proposed ASC model from 4.3M to 1.6M, 0.46M, 0.17M, and 0.1M of trainable parameters, respectively.

Table 1: Channel deconvolution and channel reduction techniques to achieve low complexity models

	proposed model	Red01	Red02	Red03	Red04
Inception Block	2×128	128	64	32	32
Inc-Res Block	2×128	128	64	32	32
Inc-Res Block	2×256	128	64	32	32
Inc-Res Block	2×256	128	64	32	32
FC layer	1024	1024	1024	1024	None
FC layer	10	10	10	10	10
Parameters (M)	4.3	1.6	0.46	0.17	0.1

4 EXPERIMENTS AND DISCUSSION

4.1 Dataset and Evaluation Metric

DCASE 2020 Task 1A Development set [6]:The dataset comprises 23040 segments (duration of each is 10 seconds) with a total recording time of 64 hours. The dataset was recorded from three real devices namely A, B, and C with 40 hours, 3 hours, and 3 hours, respectively. Additionally, synthesized audio recordings namely from S1 to S6 with 3-hour recording time for each are added. As audio recordings are from both real and synthesized devices, this dataset is ideal to evaluate ASC task with the issue of mismatched recording devices.

We follow DCASE challenges, then separate the DCASE 2020 Task 1A Development set into Training and Evaluating subsets for training and evaluating processes, respectively (Note that audio recordings from S4, S5, and S6 are not presented in Training subset to evaluate unseen samples). We also obey DCASE challenges, then use Accuracy (Acc.%) as the metric for evaluating our proposed systems in this paper. To compare the model complexity, we compute the number of parameters (Million) used by evaluating models.

4.2 Model Implementation

As using the Mixup data augmentation method, labels are not one-hot encoding format. Therefore, we use Kullback–Leibler divergence (KL) loss [13] shown in Eq. (3) below.

$$Loss_{KL}(\Theta) = \sum_{n=1}^{N} \mathbf{y}_n \log\left\{\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right\} + \frac{\lambda}{2}||\Theta||_2^2 \tag{3}$$

where Θ are trainable parameters, constant λ is set initially to 0.0001, N is batch size set to 100, $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ denote expected and predicted results. We construct and train deep learning networks proposed with Tensorflow. We set epoch number=100 and using Adam method [10] for optimization. While a learning rate of 0.0001 is set for the first 80 epochs with data augmentation methods, a low learning rate of 0.000001 is set for the next 20 epoches without any data augmentation method.

Table 2: Compare the proposed ASC model to DCASE baseline and benchmark neural networks

Performances	DCASE Baseline	Proposed Model	MobileV1	MobileV2	VGG16	VGG19	ResNet50V2	ResNet152V2	DenseNet121	DenseNet201	Xception
A(%)	70.6	77.3	74.2	71.0	68.3	67.1	74.1	74.0	74.1	74.8	75.2
B(%)	60.6	70.5	60.1	56.3	54.5	56.1	57.9	60.8	63.1	58.3	62.0
C(%)	62.6	75.7	63.7	60.6	61.5	61.5	63.7	67.8	63.7	68.6	68.4
S1(%)	55.0	69.7	57.2	52.5	55.3	49.8	60.2	52.5	62.0	57.2	60.1
S2(%)	53.3	70.6	51.4	55.2	54.4	51.4	54.1	52.8	58.9	56.3	54.7
S3(%)	51.7	71.8	55.4	52.5	53.5	52.0	55.6	57.0	60.2	59.6	62.2
unseen-S4(%)	48.2	61.5	43.8	41.3	43.8	38.3	45.6	47.6	51.7	51.5	50.4
unseen-S5(%)	45.2	66.1	44.7	46.1	45.3	44.4	52.0	44.3	53.8	48.7	49.4
unseen-S6(%)	39.6	58.8	32.6	29.5	40.1	31.4	31.7	31.9	40.8	35.7	35.2
Average(%)	54.1	69.1	53.3	51.6	53.3	50.8	55.1	54.0	58.7	56.7	57.9
Parameters(M)	5.0	4.3	4.3	3.5	15.3	20.6	25.7	60.5	8.1	20.3	23.0
Memory(MB)	19.2	16.6	16.4	13.7	58.2	254.8	98.0	230.6	30.9	77.5	87.6

Table 3: Performance comparison among single and ensemble models with or without model compression

Single Models	Acc.(%)	Parameters (M)
CQT-model	**60.8**	4.3
CQT-model w/ Red01	58.7	1.6
CQT-model w/ Red02	60.2	0.46
CQT-model w/ Red03	61.0	0.17
CQT-model w/ Red04	58.2	0.1
Gam-model	**65.8**	4.3
Gam-model w/ Red01	63.5	1.6
Gam-model w/ Red02	64.3	0.46
Gam-model w/ Red03	63.7	0.17
Gam-model w/ Red04	61.9	0.1
log-Mel-model	**69.1**	4.3
log-Mel-model w/ Red01	67.3	1.6
log-Mel-model w/ Red02	67.4	0.46
log-Mel-model w/ Red03	64.7	0.17
log-Mel-model w/ Red04	65.6	0.1
Ensemble Models	**Acc.(%)**	**Parameters (M)**
CQT, log-Mel, Gam-models	**73.6**	12.9
CQT, log-Mel, Gam-models w/ Red01	72.9	4.8
CQT, log-Mel, Gam-models w/ Red02	72.0	1.4
CQT, log-Mel, Gam-models w/ **Red03**	**71.3**	**0.5**
CQT, log-Mel, Gam-models w/ **Red04**	**70.9**	**0.3**

4.3 Experimental results and discussion

As experimental results on DCASE 2020 Task 1A dataset are shown in Table 2, our proposed ASC model outperforms benchmark network architectures across recording devices. Further analyze performance of benchmark network architectures, it indicates that deeper neural networks such as VGG19, ResNet152V2 or DenseNet201 present low performance than the lower complexity networks such as VGG16, ResNet50V2, or DenseNet121 from the same architecture groups. This proves that a wider and shallow neural network is more effective rather than a deeper architecture for ASC task with mismatched recording devices.

Although applying model compression techniques helps to significantly reduce the model complexity, it affects the accuracy performance of single models as shown in Table 3. By using both ensemble of multiple spectrograms and model compression techniques (e.g. channel deconvolution and channel reduction), we can achieve ASC models which show a balance between the accuracy performance and the model complexity. Indeed, ensembles of three spectrograms using Red03 and Red04 achieve 71.3% with 0.5M and 70.9% with 0.3M respectively, which satisfies the target low footprint model with less than 1M parameters.

Table 4: Compare our proposed best models to 5 best systems from DCASE 2020 Task 1A challenge

Top-5 models [4]	Acc.(%)	Parameters (M)
Top-1 (ensemble)	84.2	341
Top-2 (ensemble)	75.0	-
Top-3 (ensemble)	74.4	39
Top-4 (ensemble)	73.3	225
Top-5 (ensemble)	73.1	13
Our system (ensemble)	**73.6**	**12.9**
	72.9	**4.8**
	72.0	**1.4**
	71.3	**0.5**
	70.9	**0.3**

Table 4 compares our best performance models with the top-five systems submitted to DCASE 2020 Task 1A challenge [3]. Our proposed model (e.g. the ensemble of CQT-model, Gam-model, and log-Mel-model) achieves the top-4 ranking which records an accuracy of 73.6% and present lower model footprint.

5 CONCLUSION

This paper has presented a novel inception-residual-based neural network for ASC task with mismatched recording devices. By conducting intensive experiments over the benchmark DCASE 2020 Task 1A Development dataset, it is indicated that the novel network presenting a wider and shallow architecture is more effective for ASC rather than deeper architectures. Additionally, our proposed ensemble of multiple spectrograms and model compression (e.g., Red03) help to achieve an accuracy of 71.3% and low footprint of 0.5M trainable parameters, which shows a balance between the model performance and the model complexity. These results also prove that our proposed ASC models are competitive to the state-of-the-art systems and validates ASC application on edge devices.

ACKNOWLEDGMENTS

The work described in this paper is performed in the H2020 project STARLIGHT ("Sustainable Autonomy and Resilience for LEAs using AI against High priority Threats"). This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101021797.

REFERENCES

[1] François Chollet et al. 2015. Keras. https://keras.io.
[2] D. P. W. Ellis. 2009. Gammatone-like spectrogram. http://www.ee.columbia.edu/dpwe/resources/matlab/gammatonegram.
[3] Detection and Classification of Acoustic Scenes and Events. 2018. (DCASE). https://dcase.community/challenge2018.
[4] Detection and Classification of Acoustic Scenes and Events. 2020. (DCASE Task 1A Results). https://dcase.community/challenge2020/task-acoustic-scene-classification-results-a.
[5] Wei Gao, M McDonnell, and STEM UniSA. 2020. Acoustic scene classification using deep residual networks with focal loss and mild domain adaptation. *Tech. Rep., DCASE Challenge* (2020).
[6] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. 2020. Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions. In *Proc. DCASE*. 56–60.
[7] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*. 448–456.
[8] Byeonggeun Kim, Seunghan Yang, Jangho Kim, and Simyung Chang. 2021. QTI submission to DCASE 2021: Residual normalization for device imbalanced acoustic scene classification with efficient design. *Tech. Rep., DCASE Challenge* (2021).
[9] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. 2016. Compression of deep convolutional neural networks for fast and low power mobile applications. In *ICLR*.
[10] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2015).
[11] Khaled Koutini, Florian Henkel, Hamid Eghbal-zadeh, and Gerhard Widmer. 2020. Low-complexity models for acoustic scene classification based on receptive field regularization and frequency damping. In *Proc. DCASE*. 86–90.
[12] Khaled Koutini, Florian Henkel, Hamid Eghbal-zadeh, and Gerhard Widmer. 2020. Low-complexity models for acoustic scene classification based on receptive field regularization and frequency damping. In *Proc. DCASE*. 86–90.
[13] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.
[14] Richard F Lyon. 2017. *Human and machine hearing: extracting meaning from sound.* Cambridge University Press.
[15] Brian McFee, Raffel Colin, Liang Dawen, D.P.W. Ellis, McVicar Matt, Battenberg Eric, and Nieto Oriol. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of The 14th Python in Science Conference*. 18–25.
[16] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*.
[17] Ritika Nandi, Shashank Shekhar, and Manjunath Mulimani. 2021. Acoustic scene classification using kervolution-based SubSpectralNet. In *Proc. INTERSPEECH*. 26–30.
[18] Dat Ngo, Hao Hoang, Anh Nguyen, Tien Ly, and Lam Pham. 2020. Sound context classification basing on join learning model and multi-spectrogram features. *arXiv preprint arXiv:2005.12779* (2020).
[19] Truc Nguyen and Franz Pernkopf. 2019. Acoustic Scene Classification with Mismatched Devices Using CliqueNets and Mixup Data Augmentation. In *Proc. INTERSPEECH*. 2330–2334.
[20] Kenneth Ooi, Santi Peksi, and Woon-Seng Gan. 2020. Ensemble of pruned low-complexity models for acoustic scene classification. In *Proc. DCASE*. 130–134.
[21] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* (2019).
[22] Lam Pham. 2021. Robust Deep Learning Frameworks for Acoustic Scene and Respiratory Sound Classification. *arXiv preprint arXiv:2107.09268* (2021).
[23] Lam Pham, Tan Doan, D Thanh Ngo, H Nguyen, and H Hoang Kha. 2019. Cdnn-CRNN joined model for acoustic scene classification. *Tech. Rep., DCASE Challenge* (2019).
[24] Lam Pham, Ian Mcloughlin, Huy Phan, and Ramaswamy Palaniappan. 2019. A Robust Framework for Acoustic Scene Classification. In *Proc. INTERSPEECH*. 3634–3638.
[25] Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan, and Yue Lang. 2019. Bag-of-Features Models Based on C-DNN Network for Acoustic Scene Classification. In *In Proc. AES*.
[26] Lam Pham, Dat Ngo, Phu X Nguyen, Truong Hoang, and Alexander Schindler. 2021. An Audio-Visual Dataset and Deep Learning Frameworks for Crowded Scene Classification. In *Proc. CBMI*. 23–28.
[27] Lam Pham, Huy Phan, Truc Nguyen, Ramaswamy Palaniappan, Alfred Mertins, and Ian McLoughlin. 2021. Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework. *Digital Signal Processing* 110 (2021), 102943.
[28] Huy Phan, Huy Le Nguyen, Oliver Y. Chén, Lam Pham, Philipp Koch, Ian McLoughlin, and Alfred Mertins. 2021. Multi-View Audio And Music Classification. In *ICASSP*. 611–615.
[29] Sai Phaye, Emmanouil Benetos, and Ye Wang. 2019. SubSpectralNet Using Sub-spectrogram Based Convolutional Neural Networks for Acoustic Scene Classification. In *Proc. ICASSP*. 825–829.
[30] Karol J Piczak. 2015. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*. 1015–1018.
[31] Paul Primus and David Eitelsebner. 2019. Acoustic scene classification with mismatched recording devices. *Tech. Rep., DCASE Challenge* (2019).
[32] Alain Rakotomamonjy and Gilles Gasso. 2015. Histogram of gradients of time–frequency representations for audio scene classification. *IEEE Trans. Audio, Speech and Language Processing* 23, 1 (2015), 142–153.
[33] Hongwei Song, Jiqing Han, Shiwen Deng, and Zhihao Du. 2019. Acoustic Scene Classification by Implicitly Identifying Distinct Sound Events. In *Proc. INTERSPEECH*. 3860–3864.
[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
[35] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. 2020. Data Augmentation Using Random Image Cropping and Patching for Deep CNNs. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 9 (2020), 2917–2931.
[36] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Learning from between-class examples for deep sound recognition. In *International Conference on Learning Representations (ICLR)*.
[37] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, and Shuwen Liu. 2018. Mixup-based acoustic scene classification using multi-channel convolutional neural network. In *Pacific Rim Conference on Multimedia*. 14–23.