Low-complexity deep learning frameworks for acoustic scene classification

Technical Report for DCASE 2022 Task 1 challenge (AIT-Essex-Teams)
Lam Pham, Dat Ngo, Anahid Jalali, Alexander Schindler

Abstract— In this report, we presents low-complexity deep learning frameworks for acoustic scene classification (ASC). The proposed frameworks can be separated into four main steps: Front-end spectrogram extraction, online data augmentation, back-end classification, and late fusion of predicted probabilities. In particular, we initially transform audio recordings into Mel, Gammatone, and CQT spectrograms. Next, data augmentation methods of Random Cropping, Specaugment, and Mixup are then applied to generate augmented spectrograms before being fed into deep learning based classifiers. Finally, to achieve the best performance, we fuse probabilities which obtained from three individual classifiers, which are independently-trained with three type of spectrograms. Our experiments conducted on DCASE 2022 Task 1 Development dataset have fullfiled the requirement of low-complexity and achieved the best classification accuracy of 60.1%, improving DCASE baseline by 17.2%.

Clinical relevance— Mixup data augmentation, Convolutional Neural Network (CNN), pruning, quantization, spectrogram, Gammatone filter.

I. Introduction

To deal with one of the main ASC challenges, mismatched recording devices, a variety of methods have been proposed, which mainly make use of ensemble techniques: Ensemble of spectrogram inputs [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] (i.e., This approach uses multiple spectrogram inputs but only one model architecture) or ensemble of different classification models [12], [13], [14] (i.e., This approach uses only one spectrogram input, but explores the spectrogram by different model architectures). Although these approaches help to achieve good results, they show very large footprint models, which causes challenging to apply on edge-devices. Indeed, all top-10 systems proposed in recent DCASE challenges in 2018, 2019, 2020 present large architectures of deep neural networks, requiring larger than 2 MB of memory to store trainable parameters. Recently, DCASE 2021 and DCASE 2022 Task 1A challenges [15] focus on dealing the issue of high-complexity model, then require the maximum model complexity of 128 KB. Notably, DCASE 2022 Task 1A challenge does not allow to use pruning techniques as the pruning parameters still occupy the memory and cost the computation on edge-devices.

In this report, we introduces robust and low-complexity deep learning frameworks for ASC task. In particular, to deal with the ASC challenge of mismatched recording devices, we propose an ensemble of multiple spectrogram inputs, using Mel filter [16], Gammatone [17] filter, and CQT [16]. For each network used for training an individual spectrogram input, we deal with the issue of model complexity by apply-ing multiple techniques of channel reduction, decomposed convolution, and quantization.

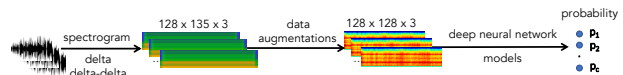## II. The proposed low-complexity deep learning framework

Fig. 1. High-level architecture of an ASC system.

A. Propose low-complexity ASC models

A framework for ASC task with high-level architecture is shown in Fig. 1. Initially, a draw audio signal is firstly transformed into a spectrogram of 128×135 by using MEL filter [16], Gammatone filter [17], or CQT [16] with the FFT number, Hanna window size, hop size, and the filter number set to 4096, 2048, 306, and 128. We then apply delta and delta-delta on the spectrogram, generate a new spectrogram of $128\times135\times3$ (i.e., The channel dimension is 3 which causes by concatenating the original spectrogram, delta, and delta-delta).

Next, we apply data augmentation methods on the spectrograms. In this report, we apply three data augmentation methods of Random Cropping [18], Specaugment [19], and Mixup [20], [21]. In particular, the temporal dimension of spectrograms of $128\times\mathbf{135}\times3$ is randomly cropped into $128\times\mathbf{128}\times3$ (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 augmentation methods.

We then fed augmented spectrograms into a back-end deep learning networks for classification. We propose three models in this report: (M1) a model with four convolutional layers as shown in Table III, (M2) a model with six convolutional layers as shown in Table I, and (M3) a model with eight convolutional layers as shown in Table II. As using limited filter numbers, proposed models present low complexity architectures, in which M3 presents the largest one.

Inspired by [22], the convolutional layer used in these models is decomposed to sub convolutional computation as

TABLE I

LOW-COMPLEXITY M2

Network architecture	Output
Convolution ([3×3] @ C_out1=16) - ReLU - BN - Dropout (10%)	128×128×16
Convolution ([3×3] @ C_out2=16) - ReLU - BN - AP [2×2] - Dropout (10%)	64×64×16
Convolution ([3×3] @ C_out3=32) - ReLU - BN - Dropout (10%)	64×64×32
Convolution ([3×3] @ C_out4=32) - ReLU - BN - AP [2×2] - Dropout (10%)	32×32×32
Convolution ([3×3] @ C_out5=64) - ReLU - BN - AP [2×2] - Dropout (10%)	16×16×64
Convolution ([3×3] @ C_out6=64) - ReLU - BN - GAP - Dropout (10%)	64
FC - Softmax	C = 10

TABLE II

LOW-COMPLEXITY M3

Network architecture	Output
Convolution ([3×3] @ C_out1=16) - ReLU - BN - Dropout (10%)	128×128×16
Convolution ([3×3] @ C_out2=16) - ReLU - BN - AP [2×2] - Dropout (10%)	64×64×16
Convolution ([3×3] @ C_out3=32) - ReLU - BN - Dropout (10%)	64×64×32
Convolution ([3×3] @ C_out4=32) - ReLU - BN - AP [2×2] - Dropout (10%)	32×32×32
Convolution ([3×3] @ C_out5=64) - ReLU - BN - AP [2×2] - Dropout (10%)	32×32×64
Convolution ([3×3] @ C_out6=64) - ReLU - BN - AP - Dropout (10%)	16×16×64
Convolution ([3×3] @ C_out6=128) - ReLU - BN - AP - Dropout (10%)	16×16×128
Convolution ([3×3] @ C_out6=128) - ReLU - BN - GAP - Dropout (10%)	128
FC - Softmax	C = 10

TABLE III

LOW-COMPLEXITY M1

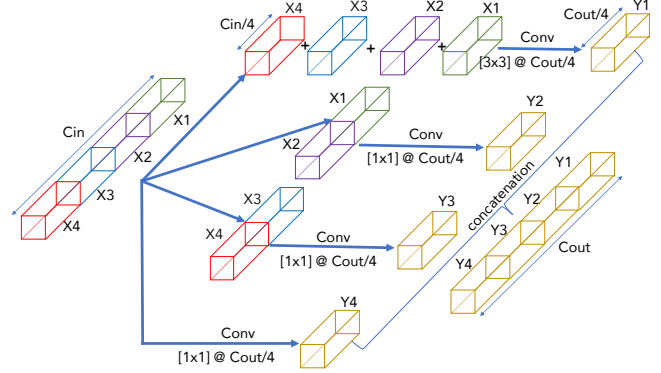| Network architecture	Output
Convolution ([3×3] @ C_out2=16) - ReLU - BN - AP [2×2] - Dropout (10%)	64×64×16
Convolution ([3×3] @ C_out4=32) - ReLU - BN - AP [2×2] - Dropout (10%)	32×32×32
Convolution ([3×3] @ C_out5=64) - ReLU - BN - AP [2×2] - Dropout (10%)	16×16×64
Convolution ([3×3] @ C_out6=128) - ReLU - BN - GAP - Dropout (10%)	128
FC - Softmax	C = 10

Fig. 2. Decomposed convolution technique applied to a convolutional layer.

shown in Figure 2. By decomposing a convolutional layer into four sub-convolutional layers, the model complexity is reduced to nearly 1/8.5. Additionally, 8-bit quantization method is applied on each model which further reduces the model size to 1/4. As a result, three proposed low-footprint models report the model complexity of 10.6 KB, 10.0 KB, and 36.8 KB for M1, M2, and M3, respectively.

As using multiple spectrograms as a rule of thumb to improve ASC performance [1], [9], [2], [4], an ensemble of three spectrograms of log-Mel, Gammatone, and CQT is conducted in this work. As we need to train independently three models for three different spectrogram inputs, the final complexity of the proposed frameworks (i.e., each framework comprises three models of M1, M2 or M3) are approximately 31.8 KB, 30 KB, and 110 KB respectively, which meets DCASE 2022 Task 1 challenge requirement (i.e., the final model must be less than 128 KB).

To fuse probability results obtained from three spectrograms, we conduct experiments over individual networks with each spectrogram input, then obtain predicted probability of each network as $\bar{\mathbf{p}}_\mathbf{s} = (\bar{p}_{s1}, \bar{p}_{s2}, ..., \bar{p}_{sC})$, where C is the category number and the sth out of S networks evaluated. Next, the predicted probability after PROD fusion $\mathbf{P}_{\mathbf{f-prod}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by:

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

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

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

III. EVALUATION SETTING AND RESULTS

A. TAU Urban Acoustic Scenes 2022 Mobile, development dataset [23]

This dataset is referred to as DCASE 2022 Task 1 Development, which was proposed for DCASE 2022 challenge [24]. In this challenge, the limitation of model complexity is set to 128 KB of trainable parameter, not allow to use pruning techniques, and evaluate on 1-second audio segment. The dataset is slightly unbalanced, being recorded

across 12 large European cities: Amsterdam, Barcelona, Helsinki, Lisbon, London, Lyon, Madrid, Milan, Prague, Paris, Stockholm, and Vienna. It consists of 10 scene classes: airport, shopping mall (indoor), metro station (underground), pedestrian street, public square, street (traffic), traveling by tram, bus and metro (underground), and urban park. The audio recordings were recorded from 3 different physical devices namely A (10215 recordings), B (749 recordings), C (748 recordings). Additionally, synthetic data for mobile devices was created based on the original recordings, referred to as S1 (750 recordings), S2 (750 recordings), S3 (750 recordings), S4 (750 recordings), S5 (750 recordings), and S6 (750 recordings).

To evaluate, we follow the DCASE 2022 Task 1 challenge [24], use two sub-sets known as Training (Train.) and Evaluation (Eval.) from the Development set for training and testing processes, respectively. Notably, two of 12 cities and S4, S5, S6 audio recordings are only presented in the Eval. subset for evaluating the issue of mismatched recording devices and unseen samples.

B. Deep learning framework implementation

We use Tensorflow framework to construct all deep learning models proposed in this report and Adam method [25] for optimization. The training and evaluating processes are conducted on GPU Titan RTX 24GB. As we use Mixup for data augmentation method, labels are not one-hot encoding format. Therefore, Kullback–Leibler divergence (KL) loss [26] is used as 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 $\hat{\mathbf{y}_i}$ denote expected and predicted results.

TABLE IV

PERFORMANCE (ACC.% AND MACs()) COMPARISON AMONG DCASE
BASELINE, THE PROPOSED ENSEMBLE OF MULTIPLE SPECTROGRAMS
WITH MODELS M1, M2, OR M3

Performance	DCASE baseline	Ensemble at M1	Ensemble at M2	Ensemble at M3
Airport	39.4	52.5	60.7	57.2
Bus	29.3	69.4	62.5	71.3
Metro	47.9	44.9	36.2	44.9
Metro station	36.0	50.6	28.3	40.9
Park	58.9	81.1	67.2	83.9
Public square	20.8	23.8	37.7	50.8
Shopping mall	51.4	74.2	60.7	60.5
Street pedestrian	30.1	27.9	23.0	34.1
Street traffic	70.6	81.2	78.6	78.8
Tram	44.6	50.5	59.1	78.0
Average	42.9	55.6	51.4	60.1
Parameters (K)	-	33.8	31.9	115.9
MACs (M)	-	0.9	0.75	0.9

C. Performance comparison between DCASE baseline and the proposed models

As Table IV shows, all proposed ensembles in three models (M1, M2 and M3) outperform DCASE baseline across individual categories. Ensemble at M3 model achieves the best overall performance of 60.1% which improves DCASE baseline by 17.2%. Notably, 'street_pedestrian' class shows low performance compared with the others, which need to further investigate.

IV. CONCLUSION

We have just presented low-complexity frameworks for ASC task, which makes use multiple spectrogram inputs and model compression techniques. While the ensemble of multiple spectrograms helps to tackle different ASC challenges of mismatched recording devices or lacking of input to improve the performance, multiple techniques of model compression help to achieve low-complexity models less than 128 KB.

REFERENCES

[1] L. Pham, I. Mcloughlin, Huy Phan, R. Palaniappan, and A. Mertins, "Deep feature embedding and hierarchical classification for audio scene classification," in *International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.

[2] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and Y. Lang, "Bag-of-features models based on C-DNN network for acoustic scene classification," in *Proc. AES*, 2019.

[3] Huy Phan, Oliver Y Chén, Lam Pham, Philipp Koch, Maarten De Vos, Ian McLoughlin, and Alfred Mertins, "Spatio-temporal attention pooling for audio scene classification," *arXiv preprint arXiv:1904.03543*, 2019.

[4] Lam Pham, Ian Mcloughlin, Huy Phan, and Ramaswamy Palaniappan, "A robust framework for acoustic scene classification," in *Proc. INTERSPEECH*, 2019, pp. 3634–3638.

[5] D. Ngo, Hao Hoang, A. Nguyen, Tien Ly, and L. Pham, "Sound context classification basing on join learning model and multi-spectrogram features," *ArXiv*, vol. abs/2005.12779, 2020.

[6] Truc Nguyen and Franz Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Proc. DCASE*, 2018, pp. 34–38.

[7] Yuma Sakashita and Masaki Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," Tech. Rep., DCASE Challenge, 2018.

[8] Huy Phan, Huy Le Nguyen, Oliver Y. Chén, Lam Pham, Philipp Koch, Ian McLoughlin, and Alfred Mertins, "Multi-view audio and music classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 611–615.

[9] L. Pham, Huy Phan, T. Nguyen, R. Palaniappan, A. Mertins, and I. Mcloughlin, "Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework," *Digital Signal Processing*, vol. 110, pp. 102943, 2021.

[10] Lam Pham, Khoa Dinh, Dat Ngo, Hieu Tang, and Alexander Schindler, "Wider or deeper neural network architecture for acoustic scene classification with mismatched recording devices," *arXiv preprint arXiv:2203.12314*, 2022.

[11] Lam Pham, Alexander Schindler, Mina Schütz, Jasmin Lampert, Sven Schlarb, and Ross King, "Deep learning frameworks applied for audio-visual scene classification," *arXiv preprint arXiv:2106.06840*, 2021.

[12] Sai Phaye, Emmanouil Benetos, and Ye Wang, "SubSpectralNet using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *Proc. ICASSP*, 2019, pp. 825–829.

[13] Zhao Ren, Kun Qian, Yebin Wang, Zixing Zhang, Vedhas Pandit, Alice Baird, and Bjorn Schuller, "Deep scalogram representations for acoustic scene classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662–669, 2018.

[14] Hongwei Song, Jiqing Han, Shiwen Deng, and Zhihao Du, "Acoustic scene classification by implicitly identifying distinct sound events," in *Proc. INTERSPEECH*, 2019, pp. 3860–3864.

[15] Dase Community, "DCASE 2022 Task 1A Description," https://dcase.community/challenge2022/task-low-complexity-acoustic-scene-classification#description.

[16] Brian McFee, Raffel Colin, Liang Dawen, D.P.W. Ellis, McVicar Matt, Battenberg Eric, and Nieto Oriol, "librosa: Audio and music signal analysis in python," in *Proceedings of The 14th Python in Science Conference*, 2015, pp. 18–25.

[17] D P W (2009) Ellis, "Gammatone-like spectrogram," 2009.

[18] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara, "Data augmentation using random image cropping and patching for deep cnns," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2917–2931, 2019.

[19] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[20] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, and Shuwen Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.

[21] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada, "Learning from between-class examples for deep sound recognition," in *ICLR*, 2018.

[22] Lam Pham, Hieu Tang, Anahid Jalali, Alexander Schindler, and Ross King, "A low-complexity deep learning framework for acoustic scene classification," *arXiv preprint arXiv:2106.06838*, 2021.

[23] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018, pp. 9–13.

[24] Detection and Classification of Acoustic Scenes and Events Community, *DCASE 2022 challenges*, http://dcase.community/challenge2022.

[25] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[26] Solomon Kullback and Richard A Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.