# LOW-COMPLEXITY DEEP LEARNING SYSTEM FOR ACOUSTIC SCENE CLASSIFICATION USING TEACHER-STUDENT SCHEME AND MULTIPLE SPECTROGRAMS

Technical Report

*Lam Pham*[1], *Cam Le*[2], *Dat Ngo*[3], *Anahid Jalali,* [1], *Alexander Schindler*[1],

[1] Austrian Institute of Technology (AIT), Austria,
{Lam.Pham, SeyedehAnahid.Naghibzadehjalali, Alexander.Schindler}@ait.ac.at
[2] Ho Chi Minh University of Technology, Vietnam, cam.levt123@hcmut.edu.vn
[3] University of Essex, UK, dn22678@essex.ac.uk

## ABSTRACT

In this technical report, a low-complexity deep learning system for acoustic scene classification (ASC) is presented. The proposed system comprises two main phases: (Phase I) Training a teacher network; and (Phase II) training a student network using distilled knowledge from the teacher. In the first phase, the teacher, which presents a large footprint model, is trained. After training the teacher, the embeddings, which are the feature map of the second last layer of the teacher, are extracted. In the second phase, the student network, which presents a low complexity model, is trained with the embeddings extracted from the teacher. Our experiments conducted on DCASE 2023 Task 1 Development dataset have fulfilled the requirement of low-complexity and achieved the best classification accuracy of 57.4%, improving DCASE baseline by 14.5%.

***Index Terms—*** Mixup augmentation, Convolutional Neural Network (CNN), spectrogram, late fusion.

## 1. INTRODUCTION

To deal with the ASC challenge of mismatched recording devices, the state-of-the-art systems mainly leverage ensemble techniques: Ensemble of spectrogram inputs [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] or ensemble of different classification models [13, 14, 15]. Although these approaches prove effective to deal with the issue of mismatched recording devices and achieve potential results, they present large model complexity. This lead to challenges to apply ASC components on edge-devices. Recently, DCASE 2021 Task 1A challenge [16] focuses on dealing the issue of high-complexity model. The challenge requires the maximum model complexity of 128 KB. Furthermore, the next challenges of DCASE 2022 Task 1 and DCASE 2023 Task 1 do not allow to use pruning techniques as the pruning parameters still occupy the memory and cost the computation on edge-devices. These challenges also require the maximum MACs (Multiply-Add cumulation) of 30 M.

In this technical report, a low-complexity deep learning frameworks using teacher-student scheme and multiple spectrograms for ASC task is presented.

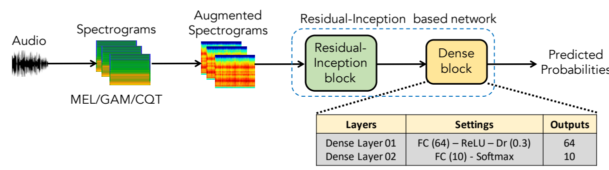


Figure 1: The high-complexity teacher network architecture.

## 2. THE PROPOSED SYSTEM

### 2.1. The teacher network architecture

As Figure 1 shows, the proposed teacher network can be separated into three main steps: the front-end feature extraction, the online data augmentation, and the convolutional neural network (CNN) based network. Initially, a raw audio signal is firstly transformed into different spectrograms of $128\times132$ by using MEL filter [17], Gammatone filter [18], or CQT [17] with the FFT number, Hanna window size, hop size, and the filter number set to 4096, 2048, 326, and 128. Next, we apply delta and delta-delta on these spectrograms to generate three-dimensional spectrograms of $128\times128\times3$ (The original spectrogram, delta, and delta-delta). We then apply the Mixup [19, 20] augmentation method on the spectrograms. We finally feed the augmented spectrograms into back-end deep learning networks for classification, referred to as the teachers. As we use three spectrograms, we train three individual teachers.

Regarding the teacher architecture, it comprises two main parts: a CNN-based backbone followed by a dense block. The CNN-based backbone, which presents a residual-inception based architecture, is reused from [21, 10]. The dense block comprises two dense layers (Dense Layer 01 and Dense Layer 02), which is shown in the lower part of Figure 2. After training the teachers, the embeddings, which are the feature map at the first fully connected layer of the dense block (FC (64)), are extracted for training the student networks. The teachers are trained using Entropy loss ($Loss_1$) as shown in Figure 2.

### 2.2. The student network architecture

A student network architecture is presented in Table 1. As we use three spectrograms, we develop three individual students which

Table 1: The low-complexity student network architecture.

| **Layers** | **Output** |
|---|---|
| Input | 128×128×3 |
| Convolution ([2×2] @ C_out1=16) - ReLU - BN - AP [2×2]- Dropout (10%) | 64×64×16 |
| Convolution ([2×2] @ C_out2=16) - ReLU - BN - AP [2×2] - Dropout (15%) | 32×32×16 |
| Convolution ([2×2] @ C_out3=16) - ReLU - BN - AP [2×2] -Dropout (20%) | 16×16×32 |
| Convolution ([2×2] @ C_out4=32) - ReLU - BN - GAP - Dropout (25%) | 32 |
| FC (64) - ReLU - Dropout (30%) | 64 |
| FC (10) - Softmax | 10 |

share the same network architecture. As the configuration shows in Table 1, three student presents 22962 trainable parameters, which occupy 88704 Byte (one parameter is presented by 32 bit) and 29267550 MACs on an edge device. Training the students is presented in Figure 2 with two loss functions ($Loss_2$ and $Loss_3$). The $Loss_3$ is traditional Entropy loss which is applied to the final layer (Softmax layer) for classification. Meanwhile, the $Loss_2$ is mean squared error (MSE) which is applied to the fully connected layer (FC (64)) of the student and the 64-dimensional embeddings extracted from the teacher. During training the student, the Mixup data augmentation is not applied and the ratio of $Loss_2$ and $Loss_3$ is empirically set to 1:1.

As we apply three spectrograms of CQT, GAM, and Mel, we fuse the probability results obtained from three individual students. In particular, we conduct experiments over individual student network with each spectrogram input, then obtain predicted probability of each network as $\mathbf{\bar{p}_s} = (\bar{p}_{s1}, \bar{p}_{s2}, ..., \bar{p}_{sC})$, where $C$ is the category number and the $s^{th}$ out of $S$ networks evaluated. Next, the predicted probability after PROD fusion $\mathbf{p_{f-prod}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by:

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

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

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

## 3. EVALUATION SETTING AND RESULTS

### 3.1. TAU Urban Acoustic Scenes 2022 Mobile development dataset [22]

This report presents the results on DCASE 2023 Task 1 Development set, which was proposed in DCASE 2023 challenge [23]. In this challenge, the limitation of model size is set to 128 KB of trainable parameters and the maximum MACs is set to 30 M, 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 2023 Task 1 challenge [23], use two subsets known as Training (Train.) and Evaluation (Eval.)

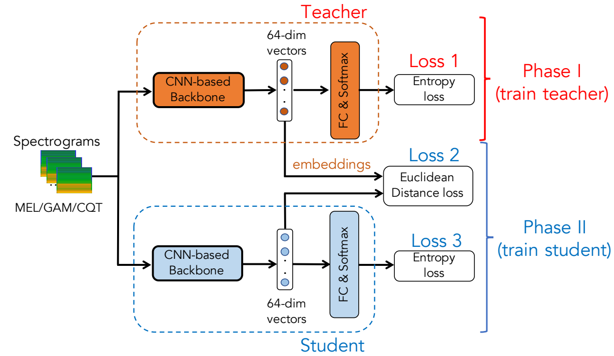


Figure 2: Training the student network using knowledge distillation.

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.

### 3.2. Network Implementation

All the CNN based architectures in this paper are conducted by Tensorflow frameworks. Training these architectures uses Adam algorithm for the optimization. We run all experiments on the GPU GeForce RTX 208.

### 3.3. Experimental Results

The experimental results are presented in Table 2. As Table 2 shows, results on GAM and MEL are competitive and outperform the records of DCASE baseline and CQT spectrogram. The ensemble of three models without and with using knowledge distillation achieve accuracy of 56.8% and 57.4%, respectively. The best model using ensemble of multiple spectrograms and knowledge distillation improves the DCASE baseline by 14.5% The log-loss score of this system presents 1.333 which is less than the DCASE score of 1.575. However, this system requires more memory of 88.7 MB compared with DCASE baseline of 46.5 MB.

## 4. CONCLUSION

We have presented a low-complexity system for ASC task, which leverages teacher-student scheme and multiple spectrogram inputs. Our proposed low-complexity ASC system achieves an accuracy of 57.4%, a log-loss score of 1.333, 88.7 KB memory occupation, and 29.27 M MACs.

## 5. REFERENCES


[1] L. Pham, I. Mcloughlin, H. 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.

Table 2: Performance (Acc.%/Log loss) comparison among DCASE baseline, the individual students without distillation, ensemble of three students without distillation, the individual students with distillation, the ensemble of three students with distillation

| | DCASE baseline | CQT w/o dis. | GAM w/o dis. | MEL w/o dis. | Ens. Students w/o dis. | CQT w/ dis. | GAM w/ dis. | MEL w/ dis. | Ens. Students w/ dis. |
|---|---|---|---|---|---|---|---|---|---|
| Airport | 39.4/1.534 | 46.6/2.091 | 49.1/1.694 | 45.7/1.801 | 57.1/1.327 | 52.6/2.112 | 38.2/1.789 | 51.7/1.656 | 62.3/1.306 |
| Bus | 29.3/1.758 | 63.0/1.793 | 61.8/1.559 | 68.6/1.554 | 75.4/0.880 | 67.7/1.700 | 68.0/1.583 | 55.9/1.612 | 77.5/0.841 |
| Metro | 47.9/1.382 | 24.9/2.130 | 33.7/1.745 | 42.1/1.668 | 43.8/1.363 | 36.6/2.095 | 33.4/1.801 | 37.1/1.708 | 45.8/1.305 |
| Metro station | 36.0/1.672 | 20.7/2.113 | 35.3/1.853 | 37.0/1.848 | 44.4/1.510 | 25.0/2.109 | 41.8/1.930 | 51.4/1.755 | 47.0/1.453 |
| Park | 58.9/1.448 | 56.9/2.037 | 67.6/1.716 | 76.4/1.554 | 76.6/1.118 | 66.6/2.052 | 74.3/1.650 | 73.1/1.478 | 78.9/1.037 |
| Public square | 20.8/2.265 | 16.5/2.123 | 37.1/1.915 | 37.8/1.890 | 40.9/1.590 | 12.4/2.163 | 32.4/1.950 | 31.9/1.869 | 39.9/1.646 |
| Shopping mall | 51.4/1.385 | 38.0/2.108 | 73.8/1.686 | 61.3/1.747 | 69.6/1.209 | 23.1/2.131 | 75.5/1.741 | 60.6/1.619 | 63.7/1.278 |
| Street pedestrian | 30.1/1.822 | 19.4/2.279 | 23.4/1.927 | 15.2/2.026 | 24.2/2.140 | 15.9/2.219 | 27.6/2.004 | 27.9/1.911 | 23.4/2.134 |
| Street traffic | 70.6/1.025 | 56.6/2.082 | 73.1/1.519 | 75.5/1.425 | 77.7/0.936 | 50.1/2.108 | 77.1/1.458 | 73.5/1.552 | 77.6/0.948 |
| Tram | 44.6/1.462 | 34.7/2.083 | 29.0/1.688 | 43.1/1.651 | 58.0/1.413 | 36.6/2.006 | 50.9/1.749 | 45.3/1.692 | 58.2/1.384 |
| Average | 42.9/1.575 | 37.7/2.084 | 50.7/1.730 | 50.2/1.716 | 56.8/1.349 | 38.6/2.070 | 51.9/1.765 | 50.8/1.685 | 57.4/1.333 |
| Memory (KB) | 46.5 | 28.8 | 28.8 | 28.8 | 88.7 | 28.8 | 28.8 | 28.8 | 88.7 |
| MACs (M) | 29.23 | 9.75 | 9.75 | 9.75 | 29.27 | 9.75 | 9.75 | 9.75 | 29.27 |


[3] H. Phan, O. Y. Chén, L. Pham, P. Koch, M. De Vos, I. McLoughlin, and A. Mertins, “Spatio-temporal attention pooling for audio scene classification,” in *Proc. INTERSPEECH*, 2019, pp. 3845–3849.

[4] L. Pham, I. Mcloughlin, H. Phan, and R. Palaniappan, “A robust framework for acoustic scene classification,” in *Proc. INTERSPEECH*, 2019, pp. 3634–3638.

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

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

[7] Y. Sakashita and M. Aono, “Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions,” DCASE Challenge, Tech. Rep., 2018.

[8] H. Phan, H. Le Nguyen, O. Y. Chén, L. Pham, P. Koch, I. McLoughlin, and A. 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, H. 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, p. 102943, 2021.

[10] L. Pham, K. Tran, D. Ngo, H. Tang, S. Phan, and A. Schindler, “Wider or deeper neural network architecture for acoustic scene classification with mismatched recording devices,” in *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, 2022.

[11] L. Pham, A. Schindler, M. Schütz, J. Lampert, S. Schlarb, and R. King, “Deep learning frameworks applied for audio-visual scene classification,” *arXiv preprint arXiv:2106.06840*, 2021.

[12] L. Pham, A. Schindler, H. Tang, and T. Hoang, “Dcase 2021 task 1a: Technique report,” *Tech. Rep., DCASE2021 Challenge*, 2021.

[13] S. Phaye, E. Benetos, and Y. Wang, “SubSpectralNet using sub-spectrogram based convolutional neural networks for acoustic scene classification,” in *Proc. ICASSP*, 2019, pp. 825–829.

[14] Z. Ren, K. Qian, Y. Wang, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, “Deep scalogram representations for acoustic scene classification,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662–669, 2018.

[15] H. Song, J. Han, S. Deng, and Z. Du, “Acoustic scene classification by implicitly identifying distinct sound events,” in *Proc. INTERSPEECH*, 2019, pp. 3860–3864.

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

[17] B. McFee, R. Colin, L. Dawen, D. 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.

[18] D. P. W. . Ellis, “Gammatone-like spectrogram,” 2009. [Online]. Available: http://www.ee.columbia.edu/dpwe/resources/matlab/gammatonegram

[19] 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.

[20] Y. Tokozume, Y. Ushiku, and T. Harada, “Learning from between-class examples for deep sound recognition,” in *ICLR*, 2018.

[21] L. Pham, D. Salovic, A. Jalali, A. Schindler, K. Tran, C. Vu, and P. X. Nguyen, “Robust, general, and low complexity acoustic scene classification systems and an effective visualization for presenting a sound scene context,” *arXiv preprint arXiv:2210.08610*, 2022.

[22] A. Mesaros, T. Heittola, and T. 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.

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