LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION USING DEEP MUTUAL LEARNING AND KNOWLEDGE DISTILLATION FINE-TUNING

Technical Report

Shilong Weng, Liu Yang, Binghong Xu

School of Computer Science and Cyber Engineering
Guangzhou University, Guangzhou, China
2112106223@e.gzhu.edu.cn, yangliupu@gmail.com, 2112006235@e.gzhu.edu.cn

ABSTRACT

This technical report describes our submission for task 1 *low-complexity acoustic scene classification* of the DCASE 2023 challenge. To enhance the generalization to unseen devices, the reassembled 10-second audio is convolved with a microphone impulse response randomly selected from the Microphone Impulse Response Project library before fed into models. Then a ResNet38 teacher model pre-trained on AudioSet and three low-complexity BC-Res2Net student models are involved in Deep Mutual Learning to further improve the performance of the teacher model, and obtain a well-initialized student model as well. Next, we use Knowledge Distillation fine-tuning to teach the student model to learn from the well-performing teacher model while maintaining the predictive performance of the teacher model. Finally, the student model is quantized by Post-Training Static Quantization to implement inference computations using 8-bit integers.

Index Terms— Acoustic scene classification, impulse response, deep mutual learning, knowledge distillation fine-tuning

1. INTRODUCTION

Low-complexity acoustic scene classification (ASC) is a regular task in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge of 2023[1], which aims to classify a test recording into one of ten predefined acoustic scene categories with limited computational and memory allowance. Moreover, this task requires to achieve generalization across various devices. The low-complexity and generalization requirements of the task are characterized by three key points:

- The memory for model parameters must be capped at 128K, regardless of the parameter type utilized. This constraint enables participants to make an efficient trade-off between model memory and parameter type.
- The computational consumption for a single inference must be limited to 30 million multiplicative accumulation operations (MMACs).
- The audio data used in the task was recorded by a variety of devices. Synthetic data for several mobile devices was also generated based on the recorded audio. Consequently, in addition to test the generalization capabilities of the model to different cities and unseen audio, this challenge also aims to assess how well the model can adapt to different and unseen recording devices.

In this report, an effective data processing method and a model training framework constituted by deep mutual learning (DML) and knowledge distillation (KD) fine-tuning are presented to address the low-complexity and device robust ASC problem.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Microphone Impulse Response

The dataset utilized for this task is the *TAU Urban Acoustic Scenes 2022 Mobile development dataset*[2]. It is derived from the *TAU Urban Acoustic Scenes 2020 Mobile development dataset* by cropping the original 10-second audio files into 1-second clips, and the sampling rate was 44.1 kHz. We borrowed the CP-JKU scheme [3] and reassembled all the training audio into 10-second segments according to the segment identifiers. Then the audio was downsampled to 32 kHz.

It is known that enhancing the diversity of training data can efficiently promote the generalization capability of the model to various recording devices. The challenge organizers created synthetic data for 11 mobile devices based on the original recordings. Sonowal *et al.* [4] found that using microphone impulse responses (IRs) from Microphone Impulse Response Project (MicIRP) library [5] to augment the training data of the DCASE 2020 Task 1a set could bring performance improvements. Therefore, we randomly simulate 'new' recording devices during training by convolving the reassembled audio signals with a diverse set of microphone IRs. Suppose that X_{ir} is a microphone IR randomly selected from MicIRP library whose download link is provided by the challenge organizers. Then an original audio signal X is convolved with X_{ir} to obtain an unknown simulated device audio output X_{un}, which can be formulated as,

$$X_{\mathrm{un}} = X * X_{ir}, \tag{1}$$

where $*$ stands for convolution operator. Furthermore, to compress the waveform of X_{un} and control the difference in amplitude range of the audio before and after convolution, the following dynamic range compression is employed,

$$\begin{aligned} x^{\mathrm{diff}} &\triangleq \max(X) - \min(X), \\ x^{\mathrm{diff}}_{\mathrm{un}} &\triangleq \max(X_{\mathrm{un}}) - \min(X_{\mathrm{un}}), \\ \hat{X}_{\mathrm{un}} &= X_{\mathrm{un}} \cdot \frac{x^{\mathrm{diff}}}{x^{\mathrm{diff}}_{\mathrm{un}}}. \end{aligned} \tag{2}$$

Source code: https://github.com/wsdragon2010/GZHU_DCASE2023_TASK1

We use 68 IRs of vintage microphones in total, which means synthetic audio data that recorded by 68 'new' devices is included in the training data. This operation makes the model ignore the characteristics of the recording devices and pay more attention to the classification information of the audio, so as to effectively reduce the dependence of the model on specific recoding devices, while maintaining the validity of model to original audio. The convolution with IRs described in (1) and (2) is applied with a probability of 0.5 to training dataset, so as to ensure that both the original audio and the simulated audio are fed into the model during training.

After convolution, each 10-second recording was randomly cropped into a 1-second snippet and fed to the model in a single epoch. That is to say, only one-tenth of the available data can be seen by the model, which can increase the diversity of the training data to a certain extent as well.

2.2. Data augmentation

On the basis of audio reassembling and convolution with IRs, two kinds of data augmentation techniques are applied to the training data sequentially.

The first one includes time shifting and time-frequency masking operations. We randomly shift an audio clip by a time interval shorter than 1 second forward. Signal values beyond the original time range on the right side of axis are used to fill in the area on the left side. To extract temporal and spectral features from the audio data, we apply short-time Fourier transform (STFT) to the shifted audio using a Hanning window of size 2048 and a hop size of 1024 samples for student model, and a Hanning window of size 800 and a hop size of 320 samples for teacher model. Then mel filter banks are applied with 256 mel bins for both student and teacher models, followed by a logarithmic operation to obtain the log mel spectrograms of the audio. Finally, we apply the time-frequency masking to the log mel spectrograms, and the maximum size of each masking band is set as 8 for the time domain and 40 for the frequency domain, respectively. The application probability of both time shifting and time-frequency masking is 0.7.

The second kind of data augmentation techniques includes mixup [6] and mixstyle[7]. We compare the effects of mixup and mixstyle in improving the model performance. The weight parameters of both mixup and mixstyle are chosen as $\alpha = 0.3$, and their application probabilities are 0.7 and 0.6, respectively. Experimental results show that mixstyle outperforms mixup, thus for both teacher model and student model, we apply mixstyle to the training data.

3. MODEL TRAINING FRAMEWORK USING DML AND KD FINE-TUNING

A novel framework that combines DML with KD fine-tuning is proposed for model training. Three low-complexity student models and a pre-trained teacher model are employed. The goal of DML is to further improve the performance of teacher model and obtain a well-initialized student model. Then we employ KD fine-tuning to transfer the knowledge learned by the well-performing teacher model to the low-complexity student model. This process aims to improve the classification performance of the student model.

3.1. Deep Mutual Learning

DML trains two or more networks which are denoted as $\mathrm{Model} = \{\mathrm{model}_1, \cdots, \mathrm{model}_N\}$ simultaneously. In the proposed model

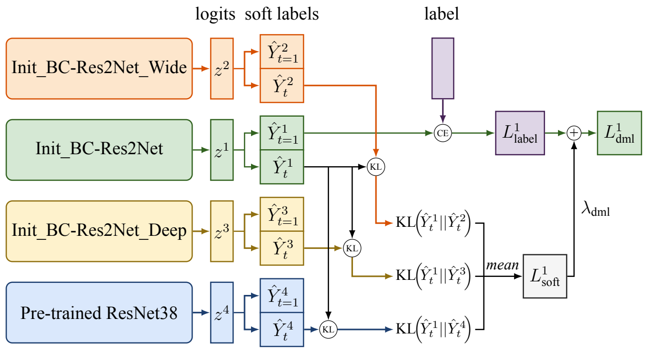

Figure 1: Diagram of DML training. It includes three BC-Res2Net student models and a pre-trained ResNet38 teacher model. CE denotes the computation of cross-entropy. Only the calculation process of total loss for the model BC-Res2Net is displayed.

training framework, the number of networks $N = 4$. At each iteration, every network learns from the other networks. Figure 1 illustrates the schematic diagram of DML training. Note that for convenience, Figure 1 only displays how model_1, which is denoted as Init_BC-Res2Net in this figure, learns from other models. Specifically, each network outputs its own probability distribution and passes the distribution as a soft label to the other networks. Then the other networks update their losses and parameters on the basis of both the soft and hard labels. The soft label for each model output is calculated as

$$\begin{aligned} \hat{y}^n_m &= \log\left(\frac{z^n_m/t}{\sum_{j=1}^{M} z^n_j/t}\right), \ m = 1, \cdots, M, \\ \hat{Y}^n_t &= [\hat{y}^n_1, \cdots, \hat{y}^n_M], \ n = 1, \cdots, N, \end{aligned} \tag{3}$$

where z^n_m is the logits of the nth model output on the mth category, and M is the total number of categories. t denotes the temperature and is employed to control the degree of smoothing of the soft labels. $\hat{y}^n_m$ represents the predicted soft probability of the nth model on the mth category, and $\hat{Y}^n_t$ are the soft labels of the nth model.

For the nth model, the hard label loss L^n_{label} is obtained by cross-entropy, and the soft label loss is calculated by Kullback-Leibler (KL) divergence as

$$L^n_{\text{soft}} = \frac{1}{N-1} \sum_{\substack{1 \le l \le N \\ l \ne n}} \mathrm{KL}\left(\hat{Y}^n_t || \hat{Y}^l_t\right), \ n = 1, \cdots, N. \tag{4}$$

Finally, the total loss of the nth model in the DML process is the weighted sum of its hard label loss and soft label loss, i.e.,

$$L^n_{\text{dml}} = L^n_{\text{label}} + \lambda_{\text{dml}} L^n_{\text{soft}}, \tag{5}$$

where λ_{dml} is the weight of the soft label loss.

It is worth noting that DML does not require additional knowledge source and it extracts knowledge directly through interactions between networks. It can effectively improve the performances of all networks involved in learning. More importantly, the interactions among the output soft labels of the models enable DML to avoid overfitting and enhance the robustness of all the models.

Three student models and one pre-trained teacher model are employed for DML training. After the DML process, a well-performing teacher model and a properly initialized student model are obtained for the following KD fine-tuning.

3.2. Student Model

The student model employed in the proposed model training framework is based on the Broadcast Residual Network (BC-ResNet)[8]. BC-ResNet was a deep neural network developed for efficient keyword detection, and it utilized both residual learning and broadcast mechanism. In the employed student model, the ResNet part in BC-ResNet is replaced by Res2Net[9], and the new model is referred to as BC-Res2Net[10]. Res2Net can extract features within different receptive fields and in multiple scales at a lower computational cost by adding small blocks of residuals to the original residual cell structure. Besides, a simple but yet effective module called Residual Normalization (ResNorm) is added to BC-Res2Net to reduce the reliance on various devices[11].

Three models are utilized as student models in DML, including a BC-Res2Net with the number of channels $C = 24$, a wider BC-Res2Net with $C = 80$, which is denoted as BC-Res2Net_wide, and a deeper BC-Res2Net named BC-Res2Net_deep, in which $C = 24$, and the number of BC-Res2Block and ResNorm within each module is doubled. The purpose of adding BC-Res2Net_wide and BC-Res2Net_deep to the DML is to allow BC-Res2Net to learn some information contained in deeper and wider networks, therefore to compensate for the limitations due to its lack of width and depth.

Denoting the number of Mel bins, and the number of time steps as F and T, Table 1 shows the overall architecture of BC-Res2Net and the size of the output feature map in each block.

Table 1: Architecture of BC-Res2Net as the student model.

Block	Output Size
input	$(1, F, T)$
ResNorm Conv2D (5×5)	$(2C, F/2, T/2)$
BC-Res2Block × 1 ResNorm MaxPool$(2, 2)$	$(C, F/4, T/4)$
BC-Res2Block × 1 ResNorm MaxPool$(2, 2)$	$(1.5C, F/8, T/8)$
BC-Res2Block × 3 ResNorm	$(2C, F/8, T/8)$
BC-Res2Block × 3 ResNorm	$(2.5C, F/8, T/8)$
Conv2D (5×5), Group=2.5C Conv2D (1×1) Mean	$(4C, 1, 1)$
Conv2D (1×1)	(10)

3.3. Teacher Model

We use ResNet38 trained by Kong *et al.* [12] on AudioSet [13] as the pre-trained teacher model. To adapt ResNet38 to the ASC task, we implement migration learning on it. ResNet38 is a deep audio neural network trained with 1.9 million audio clips and an ontology of 527 sound classes. Residual networks help ResNet38 to alleviate the vanishing gradient problem that commonly encountered in training very deep networks. The large number of sound classes can provide a comprehensive representation of unique sounds. Therefore, ResNet38 has demonstrated high accuracy rates in real-world sound classification tasks.

3.4. Knowledge Distillation Fine-tuning

KD has been widely used in various fields as a model compression tool. When training a student model, the probability distributions of the teacher model's predictions on the input audio samples, which are also known as soft labels, are utilized as an additional target. Therefore, KD allows the student model to imitate the output of the teacher model as much as possible, leading to improved generalization capacity and, to some extent, increased fitting speed of the student model.

We employ ResNet38 and BC-Res2Net trained by DML as the teacher model and initialized student model for KD fine-tuning. Soft labels and soft label loss are calculated in a similar way to DML as expressed in (3) and (4). Denoting the soft label loss in KD fine-tuning by L_{dist}, the total loss in KD can be calculated as

$$L_{\text{kd}} = L_{\text{label}} + \lambda_{\text{kd}} L_{\text{dist}}, \tag{6}$$

where λ_{kd} is the weight of the soft label loss.

4. EXPERIMENTAL SETUP AND RESULTS

4.1. Training Setup

The learning rate during the experiments is fixed at 1e-4 for individual training of the student and teacher models in the process of DML training and KD fine-tuning. Adam optimizer is utilized, and our experimental results indicate that the type of optimizer does not have a significant impact on the outcomes.

In the Res2Net structure of the student model, we set the scale size to be 4. During the DML training and KD fine-tuning, the temperature t is both set to 3 to generate soft labels, ensuring that the labels are smooth and not too much information is lost at the same time. In the DML training, the weight of soft label loss is set as $\lambda_{\text{dml}} = 1$. This is due to the fact that each student model is trained from scratch, and the purpose of DML training is to promote the performance of the pre-trained ResNet38 teacher model. Therefore, we do not want a model to put great influence on another. However, in the KD fine-tuning, the weight of soft label loss is set to be a large value $\lambda_{\text{kd}} = 50$ since we want the BC-Res2Net student model to learn as much as possible from the representations of the well-performing teacher model.

4.2. Results

Table 2 shows the accuracy performance of the student model BC-Res2Net and the teacher model ResNet38 on the provided evaluation set under different experimental settings. It is clear that the training method that includes convolution with IRs, mixstyle, DML training and KD fine-tuning performs the best.

Note that using DML followed by KD fine-tuning achieves better performance than using either one alone. In particular, using only KD performs almost the same to not using both of them. Experimental results reveal that using DML training enables the student model to converge more quickly with improved performance during KD fine-tuning. Conversely, using KD individually tends to result in unstable student performance and makes the model sensitive to the weight parameter λ_{kd}, sometimes leading to reduced model performance. This highlights the necessity of DML training, and the combination of DML and KD fine-tuning provides a fast and more effectively way to improve the performance of low-complexity student model.

Table 2: Accuracy performances of the student model and the teacher model under different training settings. Here, BC-Res2Net has a width of C = 24. Conv_IR indicates whether the input audio is convolved with IRs.

Model	Conv_IR	Mixup	Mixstyle	DML	KD	Real Devices	Seen Devices	Unseen Devices	Overall
student model BC-Res2Net	✗	✗	✗	✗	✗	63.71	48.17	36.90	49.59
	✓	✗	✗	✗	✗	**66.99**	55.73	45.45	56.05
	✗	✓	✗	✗	✗	62.91	52.09	42.68	52.55
	✗	✗	✓	✗	✗	65.40	55.34	47.05	55.93
	✓	✗	✓	✗	✗	64.34	56.90	52.21	57.81
	✓	✗	✓	✓	✗	61.04	57.34	**56.09**	58.16
	✓	✗	✓	✓	✓	65.28	**59.53**	54.66	**59.82**
	✓	✗	✓	✗	✓	63.26	57.88	52.41	57.85
teacher model ResNet38	✗	✗	✗	✗	-	70.30	52.28	44.61	55.73
	✓	✗	✗	✗	-	74.07	61.14	58.24	64.48
	✗	✓	✗	✗	-	72.76	54.04	48.63	58.47
	✗	✗	✓	✗	-	74.15	61.04	56.81	64.00
	✓	✗	✓	✗	-	74.59	67.96	64.09	68.88
	✓	✗	✓	✓	-	**76.09**	**71.10**	**69.97**	**72.39**

Table 3: Results of the four submitted models on the provided evaluation set.

submitted model ID	Real Devices			Seen Simulated Devices			Unseen Simulated Devices			Complexity		Overall		
	A	B	C	S1	S2	S3	S4	S5	S6	Params	MMAC	Q_ACC	ACC	Log Loss
1	**71.77**	**59.19**	**64.48**	58.41	**59.25**	60.50	57.13	56.95	49.37	76,906	23.97	59.67	59.82	1.146
2	70.82	59.18	64.47	**59.12**	58.91	**61.30**	57.64	**57.97**	50.00	76,906	23.97	**59.93**	-	-
3	66.84	55.22	60.77	56.86	56.26	58.61	58.27	57.83	**50.43**	76,906	23.97	57.89	58.15	1.158
4	66.55	55.53	60.70	57.39	56.06	58.52	**59.33**	57.45	50.48	76,906	23.97	58.00	-	-

4.3. Quantification

In this task, we utilize Post-Training Static Quantization in PyTorch [14] to convert all the parameters and computations involved in the final low-complexity BC-Res2Net student model to int8 type. The information loss is minimized during the quantization process while the overall model performance is maintained. "*fbgemm*" is employed as the observer, and a portion of the training data is used as calibration set.

5. SUBMISSIONS

The final results on the provided evaluation set are reported in Table 3. Listed below is a detailed description of the four submissions.

- *submission* 1: Models trained using the proposed training framework of DML and KD fine-tuning and augmentation techniques including Conv_IR and mixstyle.
- *submission* 2: On the basis of *submission* 1, the test data is augmented using the test time augmentation (TTA)[15] technique by performing 10 random Conv_IR on the test data and averaging the fusion of the 10 inference results. The randomness of convolution in TTA makes it impossible to ensure consistency between the test data input to the model before and after quantization during inference. Therefore, we only present quantized result of *submission* 2 in Table 3.
- *submission* 3: The best student model obtained by using DML training individually and augmentation methods including Conv_IR and mixstyle. This model performs best on unseen devices as shown in Table 2.
- *submission* 4: On the basis of *submission* 3, using TTA technique to augment the test data under the same setting with *submission* 2. Again, we only present the quantized result.

6. CONCLUSION

In this technical report, we describe the data processing method and a model training framework to tackle with task 1 of the DCASE 2023 challenge. We augment the audio training dataset by simulating recoding devices in the form of convolving the reassembled audio signals with randomly selected microphone IRs. Then a model training framework composed of DML training and KD fine-tuning is proposed to compress the knowledge of a well-performing ResNet38 teacher model into a low-complexity BC-Res2Net student model. Experimental results indicate that DML is of vital importance for improving student model performance. Finally, 8-bit quantization is applied to the student model.

7. REFERENCES

[1] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," https://arxiv.org/abs/2206.03835, 2022.

[2] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, pp. 56–60.

[3] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE22: distilling knowledge for low-complexity convolutional neural networks from a patchout audio transformer," DCASE2022 challenge, Tech. Rep., 2022.

[4] S. Sonowal and A. Tamse, "Novel augmentation schemes for device robust acoustic scene classification," in *Interspeech*, 2022, pp. 4182–4186.

[5] "Microphone impulse response project." [Online]. Available: http://micirp.blogspot.com/

[6] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR)*, 2018.

[7] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *International Conference on Learning Representations (ICLR)*, 2021.

[8] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Interspeech*, 2021.

[9] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.

[10] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Multi-scale architecture and device-aware data-random-drop based fine-tuning method for acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, 2022.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "Domain generalization on efficient acoustic scene classification using residual normalization," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, 2021.

[12] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[13] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[14] https://pytorch.org/docs/stable/quantization.html.

[15] I. Kandel and M. Castelli, "Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset," *Health information science and systems*, vol. 9, pp. 1–22, 2021.