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

Shilong Weng[1], *Liu Yang*[*] [1], *Binghong Xu*[1], *Xing Li*[2]

[1]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China
[2]vivo Mobile Commun co Ltd, China
2112106223@e.gzhu.edu.cn, yangliupu@gmail.com, 2112006235@e.gzhu.edu.cn, li.xing2@vivo.com

ABSTRACT

In this paper, a novel model training framework constituted by deep mutual learning (DML) and knowledge distillation (KD) fine-tuning is proposed for low-complexity acoustic scene classification (ASC). The model training phase consists of two stages. In the first stage, a ResNet38 teacher model pre-trained on AudioSet and three low-complexity BC-Res2Net student models with different widths and depths are involved in DML to enhance the teacher model performance, and attain a well-initialized student model. In the second stage, we utilize KD fine-tuning to teach this student model to learn from the high-performing teacher model while maintaining the predictive performance of the teacher model. Experimental results on *TAU Urban Acoustic Scenes 2022 Mobile development dataset* demonstrate the effectiveness of the proposed framework as well as its superiority over using KD alone under the same configurations.

Index Terms— Acoustic scene classification, deep mutual learning, knowledge distillation fine-tuning, ResNet38, BC-Res2Net

1. INTRODUCTION

Low-complexity acoustic scene classification (ASC) aims to classify a given recording into a predefined acoustic scene category by a well-designed system. It has received increasing interest because it enables deployment of classification systems on a wide range of edge devices with limited computational capacity and memory resources. This paper focus on the low-complexity ASC task in the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 challenge [1]. The low-complexity and generalization requirements of this task are characterized by three key points:

P1. Audios were recorded by a variety of devices in different cities, and synthetic data for several mobile devices was also generated based on the recorded audio.

P2. The memory for model parameters must be capped at 128K, regardless of the parameter type utilized.

P3. The computational consumption for a single inference must be limited to 30 million multiply-accumulate operations (MMACs).

For P1, augmentation schemes are frequently employed to enhance the generalization capacity of the system on recordings from unseen devices [2, 3, 4]. For P2 and P3, most low-complexity ASC approaches are based on model compression techniques and can be assorted into four classes, including feature selection [5, 6], pruning [7, 8, 9], designing efficient network architectures [10, 11, 12] and knowledge distillation (KD) [13, 14, 15]. KD has been widely utilized to derive efficient and lightweight student models by training them to emulate large and high-performing teacher models. Motivated by the concept of KD, Zhang et al. [16] presented a model training strategy called deep mutual learning (DML), in which multiple student models could learn collaboratively and teach each other throughout the training process, aiding in discovering a wider and more robust minima that generalized better to test data. The DML strategy has been applied in various fields [17, 18] and proven to be useful.

In this paper, we propose an effective model training framework that consists of DML and KD fine-tuning, in which DML plays a vital role in preparing both the teacher model and student model for the following KD fine-tuning. As opposed to solely using student models in [16], the proposed framework incorporates one pre-trained teacher model and three student models of same type but with different widths and depths during the DML stage.

The remainder of this paper is organized as follows. Section 2 describes the methodology for preprocessing and augmenting data prior to input into models. In Section 3, the proposed model training framework that consists of DML and KD fine-tuning is presented to obtain a low-complexity ASC system. Section 4 describes the experimental setup and presents the experimental results. Finally, Section 5 concludes this study.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Data preprocessing

The dataset utilized for this task is the *TAU Urban Acoustic Scenes 2022 Mobile development dataset* [19]. 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 from [20] and reassembled all the training audio into 10-second segments according to the segment identifiers. Then the audio was downsampled to 32 kHz.

2.2. Microphone Impulse Response and Augmentation

To enhance the diversity of training data and promote the generalization capability of the ASC model to various recording devices, we simulate "new" recording devices by randomly convolving the reassembled 10-second audio signals with the freely available microphone impulse responses (IRs) from the Microphone Impulse Response Project (MicIRP) library [21] as suggested in [3]. Totally

This work was partially funded by the National Natural Science Foundation of China under grant 61801133. [*]Corresponding author.

68 IRs of vintage microphones are utilized, which means synthetic audio data that recorded by 68 “new” devices is included in the training data. The probability of the audio in training dataset being convolved with IRs is set as 0.5, in order to ensure both the original audio and the simulated audio are fed into the model during training.

Then 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.

Furthermore, 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. 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 models, 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 [22] and mixstyle [23]. 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.

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. As shown in Fig. 1(a), three low-complexity student models and a pre-trained teacher model are employed in DML. The goal of DML is to further improve the performance of teacher model and attain a well-initialized student model. Then we utilize KD fine-tuning to transfer the knowledge of the high-performing teacher model to the low-complexity 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 framework, the number of networks $N = 4$. At each iteration, every network learns from the other networks. Fig. 1 (b) illustrates the schematic diagram of DML. Note that for convenience, Fig. 1(b) only displays how model_1, which is denoted as Init_BC-Res2Net, learns from other models. For the nth model, denoting its logit on the mth category as z_m^n, then its predicted soft probability on the mth category can be calculated by comparing z_m^n with the other logits [13],

$$\hat{y}_m^n = \frac{\exp\left(z_m^n / T_{\mathrm{dml}}\right)}{\sum_{j=1}^{M} \exp\left(z_j^n / T_{\mathrm{dml}}\right)},\ m = 1, \cdots, M, \tag{1}$$

where M is the total number of categories, and T_{dml} is a temperature utilized to control the degree of smoothing of the soft probability. When $T_{\mathrm{dml}} = 1$, (1) degenerates into softmax operation.

The output probability distribution of the nth model can be written as

$$\hat{Y}_{T_{\mathrm{dml}}}^n = \left[\hat{y}_1^n, \cdots, \hat{y}_M^n\right],\ n = 1, \cdots, N, \tag{2}$$

and is passed to the other networks as a soft label. The soft label loss of the nth model is computed as

$$L_{\mathrm{soft}}^n = \frac{1}{N-1} \sum_{\substack{1 \le l \le N \\ l \ne n}} \mathrm{KL}\left(\hat{Y}_{T_{\mathrm{dml}}}^n \middle|\middle| \hat{Y}_{T_{\mathrm{dml}}}^l\right). \tag{3}$$

The hard label loss of the nth model is obtained by cross-entropy. 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.,

$$L_{\mathrm{dml}}^n = L_{\mathrm{label}}^n + \lambda_{\mathrm{dml}} L_{\mathrm{soft}}^n, \tag{4}$$

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

Note that DML does not require additional knowledge source and it extracts knowledge directly through interactions among 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. After the DML process, a high-performing teacher model and a properly initialized student model are obtained for the following KD fine-tuning.

3.2. Student Model

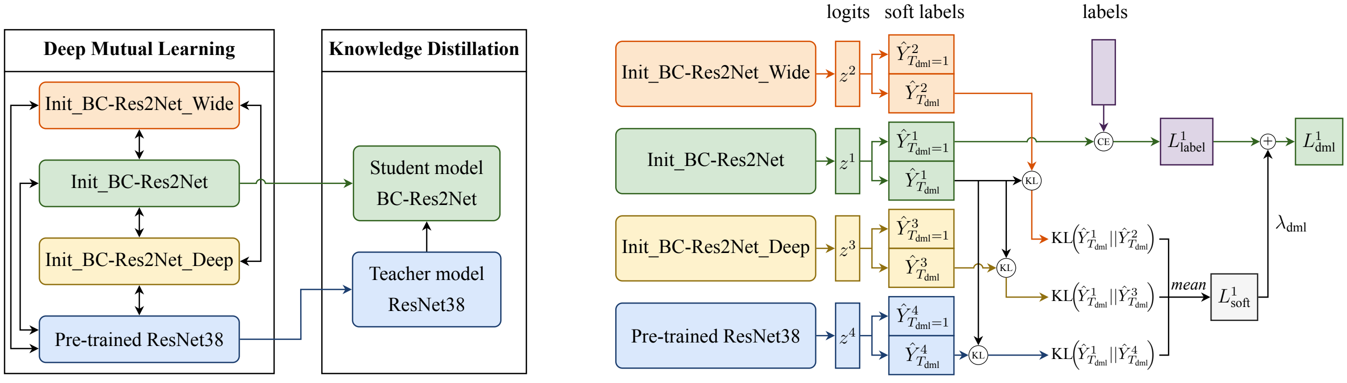
The student model employed in the proposed model training framework is based on the Broadcast Residual Network (BC-ResNet) [24]. BC-ResNet was a deep neural network developed for efficient keyword detection, and it utilized both residual learning and broadcast mechanism. In the student model employed in the proposed framework, the ResNet part in BC-ResNet is replaced by Res2Net [25], and the new model is referred to as BC-Res2Net [26]. By adding small blocks of residuals to the original residual cell structure, Res2Net can extract features within different receptive fields and in multiple scales at a lower computational cost. In addition, a simple but effective module called Residual Normalization (ResNorm) is added to BC-Res2Net to reduce the system reliance on various devices [11].

Three student models with different widths and depths are utilized 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 DML is to allow BC-Res2Net to learn specific information contained in deeper and wider networks, thereby compensating for its limitations 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 the employed BC-Res2Net and the size of the output feature map in each block.

3.3. Teacher Model

We use ResNet38 trained by Kong et al. [27] on AudioSet [28] as the teacher model. 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


Deep Mutual Learning
Init_BC-Res2Net_Wide
Init_BC-Res2Net
Init_BC-Res2Net_Deep
Pre-trained ResNet38
Knowledge Distillation
Student model
BC-Res2Net
Teacher model
ResNet38
logits
soft labels
labels
z^2
z^1
z^3
z^4
$\hat{Y}^2_{T_{\text{dml}}=1}$
$\hat{Y}^2_{T_{\text{dml}}}$
$\hat{Y}^1_{T_{\text{dml}}=1}$
$\hat{Y}^1_{T_{\text{dml}}}$
$\hat{Y}^3_{T_{\text{dml}}=1}$
$\hat{Y}^3_{T_{\text{dml}}}$
$\hat{Y}^4_{T_{\text{dml}}=1}$
$\hat{Y}^4_{T_{\text{dml}}}$
CE
KL
L^1_{label}
L^1_{dml}
λ_{dml}
$\mathrm{KL}(\hat{Y}^1_{T_{\text{dml}}}||\hat{Y}^2_{T_{\text{dml}}})$
$\mathrm{KL}(\hat{Y}^1_{T_{\text{dml}}}||\hat{Y}^3_{T_{\text{dml}}})$
$\mathrm{KL}(\hat{Y}^1_{T_{\text{dml}}}||\hat{Y}^4_{T_{\text{dml}}})$
mean
L^1_{soft}

(a) (b)

Figure 1: Diagram of the proposed model training framework. (a) DML prepares both student model and teacher model for the following KD fine-tuning. (b) Three BC-Res2Net student models and a pre-trained ResNet38 teacher model are involved in DML. For convenience, only the process by which Init_BC-Res2Net learns from other models is displayed. Ⓒⓔ denotes the computation of cross-entropy.

Table 1: Architecture of BC-Res2Net as a 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)

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 increased fitting speed of the student model.

For KD fine-tuning, we utilize the DML trained ResNet38 and BC-Res2Net as the teacher model and the initialized BC-Res2Net student model, respectively. Soft labels and soft label loss are calculated in a similar way to DML as expressed in (2) and (3). Denoting the soft label loss of the student model in KD fine-tuning by L_{dist}, the total loss in KD can be calculated by

$$L_{\text{kd}} = L_{\text{label}} + \lambda_{\text{kd}} L_{\text{dist}}, \tag{5}$$

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 both the student and teacher models in the process of DML and KD fine-tuning[1]. Adam optimizer is utilized, and our experimental results indicate that the type of optimizer does not have a significant impact on the outcomes.

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

4.2. Results

The performances of our student model BC-Res2Net and teacher model ResNet38 are evaluated on the test set provided by DCASE 2023 challenge and illustrated in Table 2. During the experiments, we followed the official data partitioning principle [29].

It can be seen from Table 2 that mixstyle outperforms mixup for both student model and teacher model, which means mixstyle is more competent to enhance device generalization. What is more, using the combination of DML and KD fine-tuning produces superior results compared to using DML alone. Clearly, the training

[1] Source code is available at https://github.com/wsdragon2010/GZHU_DCASE2023_TASK1

Table 2: Accuracy and log loss performances of student model and teacher model on test set under different configurations. BC-Res2Net has a width of C = 24. "Conv_IR" indicates whether the input audio is convolved with IRs in a certain probability. "Real", "Seen" and "Unseen" represent real devices, seen simulated devices and unseen simulated devices, respectively.

Model	**Configuration**					**Acc. (%)**				**Log Loss**
	Conv_IR	Mixup	Mixstyle	DML	KD	Real	Seen	Unseen	Overall	
student model BC-Res2Net	✗	✗	✗	✗	✗	63.71	48.17	36.90	49.59	2.586
	✓	✗	✗	✗	✗	**66.99**	55.73	45.45	56.05	1.581
	✗	✓	✗	✗	✗	62.91	52.09	42.68	52.55	1.558
	✗	✗	✓	✗	✗	65.40	55.34	47.05	55.93	1.289
	✓	✗	✓	✗	✗	64.34	56.90	52.21	57.81	1.202
	✓	✗	✓	✓	✗	61.04	57.34	**56.09**	58.16	1.158
	✓	✗	✓	✓	✓	65.67	**60.50**	54.61	**60.26**	**1.131**
teacher model ResNet38	✗	✗	✗	✗	-	70.30	52.28	44.61	55.73	3.442
	✓	✗	✗	✗	-	74.07	61.14	58.24	64.48	1.645
	✗	✓	✗	✗	-	72.76	54.04	48.63	58.47	1.307
	✗	✗	✓	✗	-	74.15	61.04	56.81	64.00	1.544
	✓	✗	✓	✗	-	74.59	67.96	64.09	68.88	1.138
	✓	✗	✓	✓	-	**76.09**	**71.10**	**69.97**	**72.39**	**0.836**

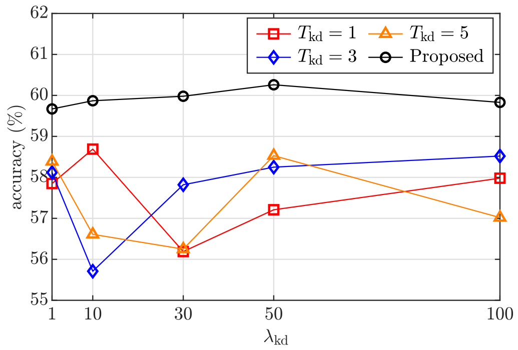


Figure 2: Accuracy of using KD alone with various λ_{kd} and T_{kd}, comparing to the performance of the proposed framework.

framework that includes convolution with IRs, mixstyle, DML and KD fine-tuning performs the best.

Fig. 2 illustrates the accuracy performance of the student model by using KD alone with various values of weight parameter λ_{kd} at different temperatures T_{kd}. By comparing with the performance of the proposed framework while $T_{\mathrm{kd}} = 3$, $T_{\mathrm{dml}} = 3$ and $\lambda_{\mathrm{dml}} = 1$, it can be observed from Fig. 2 that regardless of the parameter tuning, using KD alone can not pass the performance beyond our proposed framework. Besides, experiments reveal that DML enables the student model to converge more quickly with improved performance during KD fine-tuning. Conversely, using KD alone tends to result in unstable student performance and makes the model sensitive to the weight parameter λ_{kd}, as shown in Fig. 2. This highlights the necessity of DML. To summarize, the combination of DML and KD fine-tuning provides a fast and effective way to improve the performance of low-complexity ASC system.

To demonstrate the effectiveness of the proposed model training framework, we compare our student model BC-Res2Net with the student model denoted as "RFR-CNN" that employed in [14], and compare our teacher model ResNet38 with the teacher models referred to as "PaSST-Ensemble" and "PaSST & CP-ResNet Ensemble" in [15]. Note that "PaSST-Ensemble" is the fusion of 6 different PaSST models, and "PaSST & CP-ResNet Ensemble" uses the fusion results of 6 different PaSST models and 6 different CP-ResNet models, while we utilize only one teacher model, i.e., ResNet38. As shown in Table 3, our student model outperforms "RFR-CNN" by approximately 0.5% while having less parameters and MMACs. Moreover, our teacher model exhibits an absolute advantage over the two teacher models in [15] by almost 4% in terms of overall classification accuracy.

Table 3: Performance comparison of various student models and teacher models.

Model	**Params**	**MMACs**	**Acc. (%)**
RFR-CNN [14], 2022	127,046	29.06	59.76
BC-Res2Net (Ours)	**76,906**	**23.97**	**60.26**
PaSST-Ensemble [15], 2023	-	-	63.63
PaSST & CP-ResNet Ensemble [15], 2023	-	-	68.31
ResNet38 (Ours)	73,804,121	9,179.52	**72.39**

5. CONCLUSION

In this paper, we tackle with the low-complexity ASC task in DCASE 2023 challenge. We present a novel model training framework that consists of DML and KD fine-tuning. DML helps both teacher model and student model prepare for the following KD fine-tuning, then KD is used to compress the knowledge of a high-performing ResNet38 teacher model into a low-complexity BC-Res2Net student model in an optimal manner. Experimental results demonstrate that DML plays a critical role in enhancing the final performance of the proposed low-complexity ASC system. Next, we aim to apply the proposed training framework to newer and stronger models in an attempt to achieve even better performance.

6. 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," 2022.

[2] J. Han, M. Matuszewski, O. Sikorski, H. Sung, and H. Cho, "Randmasking augment: a simple and randomized data augmentation for acoustic scene classification," in *ICASSP*, 2023, pp. 1–5.

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

[4] H. Hu, C.-H. H. Yang, X. Xia, X. Bai, X. Tang, Y. Wang, S. Niu, L. Chai, J. Li, H. Zhu, F. Bao, Y. Zhao, S. M. Siniscalchi, Y. Wang, J. Du, and C.-H. Lee, "A two-stage approach to device-robust acoustic scene classification," in *ICASSP*, 2021, pp. 845–849.

[5] L. P. Schmidt, B. Kiliç, and N. Peters, "Feature selection using alternating direction method of multiplier for low-complexity acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[6] C. Paseddula and S. V. Gangashetty, "Acoustic scene classification using single frequency filtering cepstral coefficients and DNN," in *IJCNN*, 2020, pp. 1–6.

[7] S. Tofigh, M. O. Ahmad, and M. Swamy, "A low-complexity modified thinet algorithm for pruning convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 29, pp. 1012–1016, 2022.

[8] J. Kim, K. Yoo, and N. Kwak, "Position-based scaled gradient for model quantization and pruning," in *NIPS*, 2020, pp. 20 415–20 426.

[9] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *ICLR*, 2021.

[10] M. Aswathy and K. Suresh, "RQNet: residual quaternion CNN for performance enhancement in low complexity and device robust acoustic scene classification," *IEEE Trans. Multimedia*, pp. 1–13, 2023.

[11] B. Kim, S. Yang, J. Kim, and S. Chang, "Domain generalization on efficient acoustic scene classification using residual normalization," in *DCASE 2021 Workshop*, 2021.

[12] A. Singh and M. D. Plumbley, "Low-complexity CNNs for acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[13] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[14] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE 2022 Workshop*, 2022.

[15] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: efficient acoustic scene classification with CP-Mobile," DCASE 2023 challenge, Tech. Rep., 2023.

[16] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4320–4328.

[17] F. Xudong, G. Xiaofeng, K. Ping, L. Xianglong, and Z. Yalou, "Pedestrian detection and tracking with deep mutual learning," in *ICCWAMTIP*, 2021, pp. 217–220.

[18] R. Masumura, M. Ihori, A. Takashima, T. Tanaka, and T. Ashihara, "End-to-end automatic speech recognition with deep mutual learning," in *APSIPA ASC*, 2020, pp. 632–637.

[19] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: generalization across devices and low complexity solutions," in *DCASE 2020 workshop*, 2020, pp. 56–60.

[20] 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," DCASE 2022 challenge, Tech. Rep., 2022.

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

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," in *ICLR*, 2018, pp. 1–13.

[23] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *ICLR*, 2021.

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

[25] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2021.

[26] 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 *DCASE 2022 Workshop*, 2022.

[27] Q. Kong, M. Yin Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.

[28] 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 *ICASSP*, 2017, pp. 776–780.

[29] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *DCASE 2018 workshop*, 2018, pp. 9–13.