DOMAIN-SPECIFIC EXTERNAL DATA PRE-TRAINING AND DEVICE-AWARE DISTILLATION FOR DATA-EFFICIENT ACOUSTIC SCENE CLASSIFICATION

Technical Report

*Dominik Karasin, Ioan-Cristian Olariu, Michael Schöpf, Anna Szymańska**

Students at Johannes Kepler Universität Linz, Linz, Austria
{k12213736, k12219769, k12213283}@students.jku.at, 247105@p.lodz.edu.pl

ABSTRACT

In this technical report, we present our submission to the DCASE 2025 Challenge Task 1: Low-Complexity Acoustic Scene Classification with Device Information. Our approach centers on a compact CP-Mobile student model distilled via Bayesian ensemble averaging from different combinations of three teacher architectures: CP-ResNet, BEATs, and PaSST—using AudioSet pretrained checkpoints for the last two. We then fine-tune the student on each recording device to improve per-device classification accuracy. To compensate for the limited 25% train-split, we pre-train both teacher and student on CochlScene and apply data augmentation, of which Device Impulse Response augmentation was particularly effective.

***Index Terms*—** Acoustic scene classification, CP-Mobile, Knowledge distillation, CochlScene, Device Impulse Response, Freq-MixStyle, CP-ResNet, PaSST, BEATs

1. INTRODUCTION

Acoustic Scene Classification (ASC) focuses on identifying acoustic scenes from raw audio. The system we developed is designed to classify 1-second audio clips into 10 predefined audio scenes. Similarly to the previous year, this year's DCASE Task 1 challenge [1] faces two low-complexity constraints: maximum memory allowance for model parameters equal to 128 kB and computational complexity at inference time restricted to 30 MMACs. Another challenge arises from the recording device mismatch. While recordings from real device A comprise 8 hours of audio, others (real devices B, C and simulated devices S1-S6) amount to 9 hours and 56 minutes in total. This year's focus is put on device information, which can be used to fine-tune the models for specific recording devices. Due to availability of recording device information in the evaluation dataset, distinct models can be used per device, while still applying the general model to unseen recording devices. An additional change was made in terms of the availability of data. Training data is restricted to 25% subset of the DCASE24 Task 1 dataset. However, it is permitted to utilize external ASC datasets for model development.

2. DATASETS

2.1. TAU Urban Acoustic Scenes 2022 Mobile dataset

Our primary dataset is the TAU Urban Acoustic Scenes 2022 Mobile dataset (TAU22) [2], an extension of the 2020 Mobile dataset [3]. In TAU22, each original 10-second clip has been split into ten 1-second, single-channel samples at 44.1 kHz. TAU22 includes recordings from multiple European cities across ten scene classes, captured with four real devices (A, B, C, and D) and supplemented by simulated devices (S1–S10). The ten classes in TAU22 are: *airport, bus, metro, metro station, park, public square, shopping mall, street pedestrian, street traffic, tram*.

The 2025 Low-Complexity Acoustic Scene Classification task provides both official development and evaluation splits. For development, only 25% of the official training set is permitted during model training [1]. This corresponds to last year's 25% train split. The development set can be further split into:

- **Development-train:** devices A, B, C and simulations S1–S3 (8.25 hours of audio)
- **Development-test:** devices A, B, C and simulations S1–S6 (9.7 hours of audio)

The evaluation dataset consists of the same ten acoustic scenes, captured by devices present in the TAU22 development dataset (A, B, C, S1-S3), one additional real device (D) and four additional simulated devices (S7-S10). Samples from known devices retain their device ID, while all recordings from the additional devices are labeled as "unknown".

2.2. CochlScene dataset

CochlScene [4] is an acoustic scene dataset, collected through crowdsourcing. It consists of 76,115 single-channel audio files with a sample rate of 44.1kHz and a length of 10 seconds. There are a total of 13 different classes, spanning acoustic scenes from urban areas in South Korea. The 13 classes in CochlScene are: *bus, cafe, car, crowded indoor, elevator, kitchen, park, residential area, restaurant, restroom, street, subway, subway station*.

2.3. AudioSet

AudioSet [5] is a large-scale multi-label audio event dataset, gathered from YouTube. It contains over 2 million ten-second audio clips, annotated by humans across a total of 632 classes. Each sample is a single-channel audio file with a sample rate of 44.1kHz. The dataset is hierarchically structured, such that categories can be subdivided into increasingly specific event labels. It is widely used as a benchmark for multi-label audio tagging, sound event detection, and pre-training of general-purpose audio feature extractors [6, 7, 8, 9].

*Thanks to JKU Institute of Computational Perception for compute resources.

3. ARCHITECTURES

3.1. Teacher models

In order to choose the most suitable models for conquering the task, we analyzed the submissions from previous years [10]. As it was shown, PaSST and CP-ResNet architectures perform effectively on TAU22 development set [11].

CP-ResNet [12] is a receptive field regularized convolutional neural network (RFR-CNN) [13]. The authors of [12] show that controlling the receptive field leads to enhanced generalization for ASC.

The Patchout faSt Spectrogram Transformer (PaSST) [7], a complex, transformer-based model with 85 million parameters, pre-trained on AudioSet (2.3), focuses on a global context. Due to its fully self-attention-based architecture, PaSST is capable of extracting broad-scale relationships across the mel spectrogram.

The third model used in our approach is BEATs [14], an iterative audio pre-training framework to learn Bidirectional Encoder representation with Audio Transformers.

3.2. Student model

For the final submission, we use CP-Mobile architecture [15], which was provided as a baseline model. The detailed architecture can be seen in the Table 1.

Blocks	Input shape	Parameters	MACs
Initial convolutions	[1, 1, 256, 65]	2,456	2,810,960
Block 1 (CPM-S)	[1, 32, 64, 17]	4,992	5,083,456
Block 2 (CPM-D)	[1, 32, 64, 17]	4,992	5,083,456
Block 3 (CPM-S)	[1, 32, 64, 17]	4,992	3,739,968
Block 4 (CPM-T)	[1, 32, 64, 9]	6,576	2,378,096
Block 5 (CPM-S)	[1, 56, 32, 9]	15,112	4,182,352
Block 6 (CPM-T)	[1, 56, 32, 9]	20,968	5,841,328
Final convolution	[1, 104, 32, 9]	1,060	299,540

Table 1: CP-Mobile architecture indicating input shape, total parameters and MACs per block

The architecture of CP-Mobile consists of CPM blocks composed of sequences of three layers: point-wise expansion, depth-wise convolution, and point-wise projection. Each layer consists of a convolutional operation with batch normalization and ReLU activation applied. This structure allows to keep the expressiveness, while reducing computational complexity of the model.
The mentioned blocks can be described as:

1. **Transition block (CPM block T)**, which increases the channel dimension and does not contain any residual connections.
2. **Standard block (CPM block S)**, which does not change the channel dimension and uses the residual connection.
3. **Spatial Downsampling (CPM block D)**, which does not change the channel dimension and uses the residual connection with average pooling.

The structure of each block can be seen in Figure 1.

The first two layers project the input data from mel spectrograms to models feature space. At the last three layers, 1x1 convolution, batch normalization, and adaptive average pooling are applied.

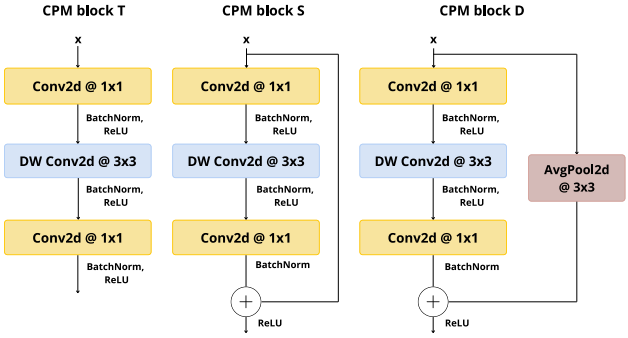

Figure 1: Visualisation of CPM blocks structures

With 61,148 parameters and 29,419,156 MACs the architecture meets the constraints when the model weights are converted to half-precision (16 bit) floating point representation for inference.

4. FEATURE EXTRACTION AND DATA AUGMENTATION

4.1. Preprocessing

We resample audio to a model-specific target sampling rate and compute log-scaled mel spectrograms. The parameters used for the STFT and log-scaled mel spectrogram vary between architectures and are listed in Table 2.

Parameter	CP-Mobile	CP-ResNet	BEATs	PaSST
Original sample rate (kHz)	44.1	44.1	44.1	44.1
Target sample rate (kHz)	32	32	16	32
FFT size	4096	4096	1024	1024
Window length (ms)	96	96	25	25
Hop length (ms)	16	24	10	16
Number of mel bins	256	256	128	128

Table 2: Preprocessing parameters for different model architectures

4.2. Data augmentations

We split the augmentation into two categories: waveform-level and spectrogram-level.

4.2.1. Waveform-level Augmentations

- **Time rolling:** Time rolling is a data augmentation technique that applies a circular shift to the waveform. A randomly selected segment of up to 0.1 seconds from the beginning or end of the audio is moved to the opposite end, simulating variations in time positioning.
- **DIR:** Device Impulse Response (DIR) augmentation [16] simulates the acoustic characteristics of new audio recording devices to improve robustness for unseen devices. This is achieved by convolving the audio waveform with one of 66 impulse responses taken from MicIRP[1]. The augmentation is applied with a probability of 70% to samples recorded with device A.

4.2.2. Spectrogram-level Augmentations

- **SpecAugment:** SpecAugment [17] is a commonly used spectrogram augmentation strategy that improves generalization by

[1]https://micirp.blogspot.com

randomly masking portions in both frequency and time domains. In our implementations we apply frequency masking with up to 48 frequency bins. Additionally, for PaSST and BEATs, time masking with a size of up to 20 time frames per spectrogram is also applied.

- **Freq-MixStyle:** Freq-MixStyle [18] is a adaption of MixStyle [19] in the frequency domain. It is applied by switching feature statistics (mean, variance) between samples within a batch along the frequency axis. Freq-MixStyle is applied with a probability of 30% for CP-Mobile, 40% for BEATs and PaSST and 80% for CP-ResNet. The mixing coefficients are drawn from a beta distribution with $\alpha = 0.4$. It is not applied during device-specific fine-tuning.

5. KNOWLEDGE DISTILLATION

Knowledge distillation (KD) [20] is a training method, where a model is not only trained on the one-hot encoded class labels directly, but also on the logits of one ore more teacher models. The teachers are usually large models with high performance. Knowledge distillation in general leads to better performing and more robust models.

Through a division of the outputs of the teacher and student models with a temperature value (τ) and subsequent application of the softmax function, softer, more informative targets are produced.

The loss function is a weighted average of a label loss (L_l), in our case the cross-entropy-loss, and the distillation loss (L_{KD}), which is the Kullback-Leibler (KL) divergence between teacher and student logits.

With λ as the weight and z_S and z_T as the output logits of the student and teacher model, the loss function is calculated as follows:

$$\text{Loss} = \lambda L_l(\delta(z_S), y) + (1-\lambda)\tau^2 L_{kd}(\delta(z_S/\tau), \delta(z_T/\tau))$$

Instead of a single teacher, we use Bayesian Ensemble Averaging (BAE) [10, 21] of several teacher models. With this, multiple teachers, possibly trained with different configurations, can be combined.

We use online KD to also apply the same data augmentation pipeline for the teacher models [22].

6. PRE-TRAINING ON AUDIOSET AND COCHLSCENE

Considering the limited size of the development dataset, we found it beneficial to pre-train (or use existing pre-trained weights for) both the teachers and the student models on external audio datasets.

For the teacher models based on a transformer architecture, PaSST and BEATs, we use publicly available checkpoints pre-trained on AudioSet. Since the classes and their number do not match the downstream training, the classification heads are discarded. In the case of PaSST we use the checkpoint `passt_s_swa_p16_128_ap476`[2] corresponding to a model trained on AudioSet with weak labels. In the case of BEATs we use the checkpoint[3] provided by the authors of [9], corresponding to a model pre-trained using self-supervised learning with patch-wise masked prediction on AudioSet and fine-tuned on AudioSet with weak labels.

We furthermore use the CochlScene dataset [4] to pre-train the models involved in the preparation of the submitted systems. Since this dataset was specifically created for ASC tasks, albeit under very different urban conditions (Asia - South Korea) and using more heterogeneous recording devices, we hypothesize that models pre-trained on it would more effectively adapt to the task at hand and generalize better to unseen recording devices. The CP-ResNet teacher and the CP-Mobile student models are trained on 1-second slices of CochlScene audio clips. For the PaSST and BEATs teacher, we use the full audio clips, matching the 10s input size of their AudioSet pre-training. Table 5 details the key hyperparameters used, and Table 3 shows the average accuracies on the CochlScene test split and the improvements in downstream accuracy. We do not use the CochlScene-retrained BEATs teacher because it does not improve the accuracy in the downstream task.

	CPM general	**CP-ResNet**	**PaSST**	**BEATs**
CochlScene avg. accuracy (%)	71.63	71.88	85.96	84.97
TAU22 avg. accuracy gain	+3.36	+6.05	+0.22	-2.67

Table 3: CochlScene pre-training accuracy impact

7. EXPERIMENTS ON TAU22

7.1. Experimental setup

We ran our experiments on three personal computers with consumer-grade graphics cards and one shared lab system with older-generation data-center GPUs. The maximum amount of dedicated GPU memory available was 24 GB on two systems, which limited the number of teachers we could ensemble for training with online Knowledge Distillation.

7.2. Device-specific training

DCASE'25 Task 1 focuses on fine-tuning the obtained model per device present during training (development-train devices: A, B, C, S1, S2, S3). The general model is used to initialize six specialized models, which are further fine-tuned on data specific to only one device. At inference time, the input is dispatched to a specialized model using the device ID, if known, otherwise to the general model. This way, one can obtain higher accuracies for devices encountered during training.

Table 4 exemplifies the improvements in accuracy achieved through device-specific training.

Model	**Setting**	**A**	**B**	**C**	**S1**	**S2**	**S3**	**Macro avg. accuracy**
CPM student	General	67.09	60.00	63.62	59.03	58.15	61.42	60.20
CPM student	Device-specific	71.36	63.98	67.20	60.94	58.55	64.03	62.00
CP-ResNet	General	66.12	59.39	64.20	55.27	57.55	60.03	59.11
CP-ResNet	Device-specific	69.42	61.98	66.78	59.85	59.55	62.03	61.00

Table 4: Comparison between CP-ResNet and CP-Mobile models before and after applying device-specific training (showing devices in the training set)

7.3. Knowledge Distillation

For training the general model, we use the temperature $\tau = 2$ and the weight $\lambda = 0.02$—values that produced good results in previous editions of the task [15]. For device-specific training, the best results were achieved with $\lambda = 0.1$.

[2] https://github.com/kkoutini/PaSST/releases/

[3] https://github.com/fschmid56/PretrainedSED/releases

We experimented with multiple combinations of teachers. The best combination we found was formed of these models: one BEATs, one general CP-ResNet, and one device-specific CP-ResNet. The output of the multi-device inner models is dispatched based on device information and then further aggregated with the other teachers in the ensemble.

KD leads to an increase in performance when applied in training the general and the device-specific models, using the above-mentioned teachers for both steps.

7.4. Training procedure

A general CP-ResNet teacher model is pre-trained from scratch on CochlScene and subsequently fine-tuned on the development dataset with the augmentations described in Section 4.2. Using the best checkpoint from the previous model, we fine-tune a multi-device CP-ResNet by mirroring the device-specific training procedure in the baseline.

The PaSST teacher model is trained on CochlScene starting from the pre-trained AudioSet checkpoint, then fine-tuned on the TAU22 dataset. When training on the TAU22 dataset, we match the shape of the mel spectrograms from the $1s$ samples to the shape that was used in the AudioSet pre-training, by repeating the content along the time dimension.

For the BEATs teacher we also use a checkpoint pre-trained on AudioSet. We tried multiple techniques to match the input size when fine-tuning, on the waveform (pad with zeros, repeat content) or on the mel spectrogram (repeat content). Simply inputting the $1s$ samples from the TAU22 dataset is much faster to train and requires less memory, while showing only a small decrease in the final accuracy.

As a starting point for training the student models, we create a CP-Mobile (CPM) checkpoint by pre-training on CochlScene. We apply KD and data augmentations to train general models as described in the previous sections. Finally, we fine-tune device-specific models, in most cases also using KD for this last stage (the exceptions are noted in Section 8).

We use the Adam [23] optimizer and a cosine learning rate scheduler with the corresponding hyperparameters adapted to each task. The specific values are captured in Table 5.

Task	Dataset	Max LR	Warm-up	Epochs	Batch size	Teacher
CPM general	TAU22	0.005	2000	150	256	○
CPM device specific	TAU22	0.0005	200	50	256	○
CPM pre-train	CochlScene (1s)	0.005	2000	100	512	○
CP-ResNet pre-train	CochlScene (1s)	0.001	2000	150	512	•
CP-ResNet general	TAU22	0.001	2000	100	256	•
CP-ResNet device-specific	TAU22	0.00001	200	50	256	•
PaSST retrain	CochlScene (10s)	0.00001	2000	25	20	•
PaSST general	TAU22	0.00001	2000	25	20	•
BEATs general	TAU22	0.00001	2000	30	80	•

Table 5: Hyperparameters used for different training tasks. The values shown were used for most runs—deviations are indicated in Section 8

From most training runs, we kept the best checkpoints based on maximizing the test macro-averaged accuracy, but we also selected some last-epoch checkpoints, as will be indicated in the description of the submitted systems in Section 8.

8. SUBMISSIONS AND RESULTS

Our submission for the challenge consists of the following four systems:

i We used the teachers, described in Section 7.3 for training both the general and device-specific student models and select the best checkpoint according to the test accuracy.

ii Equivalent to *i*, except that we always used the last checkpoint.

iii We used a PaSST and a device-specific CP-ResNet as teachers for training the general student model. For this system, we did not use KD during device-specific fine-tuning.

iv For this system, the device-specific models were cherry-picked from several runs with different hyperparameters, according to their test accuracy. The models for devices A and C, as well as the general model, are the same as for *i*. For models B and S2, λ was set to 0.02 and for S3 it was set to 0.05. We did not use KD for fine-tuning on device S1.

Tables 6 and 7 show the obtained test accuracies for all four systems and the baseline per class and device, respectively.

System	airport	bus	metro	metro station	park	public square	shopping mall	street pedestrian	street traffic	tram
Baseline	44.43	64.81	43.87	48.22	72.75	32.04	53.14	34.43	74.10	51.08
i	57.36	82.29	65.05	51.92	84.24	43.40	64.98	34.01	76.63	60.14
ii	52.94	79.26	56.30	48.32	81.31	41.52	59.06	33.27	77.51	57.67
iii	50.37	72.36	59.29	51.45	83.87	42.09	70.61	32.63	79.76	62.84
iv	57.97	82.29	64.92	51.85	84.21	44.04	64.61	34.44	76.63	60.10

Table 6: Class-wise test accuracies of each system

System	A	B	C	S1	S2	S3	S4	S5	S6	Avg.
Baseline	62.80	52.87	54.23	48.52	47.29	52.86	48.14	47.23	42.60	50.72
i	71.36	63.98	67.20	60.94	58.55	64.03	59.64	59.48	52.88	62.00
ii	69.33	58.66	64.59	55.30	56.91	61.52	55.85	55.85	50.45	58.71
iii	69.79	62.13	66.78	58.58	57.24	61.67	59.15	56.85	52.61	60.53
iv	71.36	63.98	67.60	61.48	58.55	64.03	59.64	59.48	52.88	62.11

Table 7: Device-wise test accuracies of each system

9. CONCLUSION

In this report, we present our submission for the DCASE 2025 Challenge Task 1: Low-Complexity Acoustic Scene Classification with Device Information. Our strategy combines pre-training on the CochlScene dataset, knowledge distillation from different teacher architectures, and device-specific fine-tuning. Incorporating pre-training on an external ASC dataset is a novel strategy for DCASE Task 1 submissions. The submission employed CP-Mobile as student model, which adheres to the low-complexity constraints of the challenge. Furthermore, we apply data augmentation at both waveform and spectrogram-level, to address the limited size of the train split. Compared to the baseline, our system achieves an improvement of more than 10 percentage points on the macro-averaged accuracy for the development-test set.

10. ACKNOWLEDGMENT

The work in this project was conducted as part of the course "Machine Learning and Audio: a challenge" at JKU Linz. We want to express our appreciation for the opportunity to participate in this practical challenge. We are especially grateful to the course organizers, Florian Schmid and Paul Primus, for their guidance and continued support throughout the project.

11. REFERENCES

[1] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, and G. Widmer, "Low-complexity acoustic scene classification with device information in the dcase 2025 challenge," 2025.

[2] 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," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, November 2022.

[3] 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.

[4] I.-Y. Jeong and J. Park, "CochlScene: Acquisition of acoustic scene data using crowdsourcing," *CoRR*, vol. abs/2211.02289, Nov. 2022.

[5] 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)*, Mar. 2017, pp. 776–780.

[6] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021, Brno, Czechia, August 30 - September 3, 2021*. ISCA, 2021, pp. 571–575.

[7] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Interspeech 2022*, Sept. 2022, pp. 2753–2757.

[8] F. Schmid, K. Koutini, and G. Widmer, "Dynamic Convolutional Neural Networks as Efficient Pre-Trained Audio Models," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 32, pp. 2227–2241, Mar. 2024.

[9] F. Schmid, T. Morocutti, F. Foscarin, J. Schlüter, P. Primus, and G. Widmer, "Effective pre-training of audio transformers for sound event detection," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[10] D. Nadrchal, A. Rostamza, and P. Schilcher, "Data-efficient acoustic scene classification with pre-training, bayesian ensemble averaging, and extensive augmentations," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2024 Workshop (DCASE2024)*, October 2024, pp. 91–95.

[11] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and CNNs with CP-mobile," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, 2023, pp. 161–165.

[12] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive Field Regularization Techniques for Audio Classification and Tagging With Deep Convolutional Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1987–2000, 2021.

[13] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "The Receptive Field as a Regularizer in Deep Convolutional Neural Networks for Acoustic Scene Classification," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sept. 2019, pp. 1–5.

[14] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio Pre-Training with Acoustic Tokenizers," in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, July 2023, pp. 5178–5193.

[15] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and CNNs with CP-mobile," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2023 Workshop (DCASE2023)*, 2023, pp. 161–165.

[16] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *31st European Signal Processing Conference (EUSIPCO 2024)*, 09 2023, pp. 176–180.

[17] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Interspeech 2019*, 2019, pp. 2613–2617.

[18] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech 2022*, 2022, pp. 2393–2397.

[19] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain Generalization with MixStyle," *CoRR*, vol. abs/2104.02008, Apr. 2021.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[21] J. Xu, S. Li, A. Deng, M. Xiong, J. Wu, J. Wu, S. Ding, and B. Hooi, "Probabilistic Knowledge Distillation of Face Ensembles," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 3489–3498.

[22] L. Beyer, X. Zhai, A. Royer, L. Markeeva, R. Anil, and A. Kolesnikov, "Knowledge distillation: A good teacher is patient and consistent," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 10 915–10 924.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.