DATA-EFFICIENT LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION IN THE DCASE 2024 CHALLENGE

Florian Schmid[1]*, Paul Primus*[1]*, Toni Heittola*[3]*,*
Annamaria Mesaros[3]*, Irene Martín-Morató*[3]*, Khaled Koutini*[2]*, Gerhard Widmer*[1,2]

[1]Institute of Computational Perception, Johannes Kepler University Linz, Austria
[2]LIT Artificial Intelligence Lab, Linz, Austria, [3]Computing Sciences, Tampere University, Finland
{florian.schmid, paul.primus, gerhard.widmer}@jku.at
{toni.heittola, annamaria.mesaros, irene.martinmorato}@tuni.fi

ABSTRACT

This article describes the *Data-Efficient Low-Complexity Acoustic Scene Classification* Task in the DCASE 2024 Challenge and the corresponding baseline system. The task setup is a continuation of previous editions (2022 and 2023), which focused on recording device mismatches and low-complexity constraints. This year's edition introduces an additional real-world problem: participants must develop data-efficient systems for five scenarios, which progressively limit the available training data. The provided baseline system is based on an efficient, factorized CNN architecture constructed from inverted residual blocks and uses Freq-MixStyle to tackle the device mismatch problem. The task received 37 submissions from 17 teams, with the large majority of systems outperforming the baseline. The top-ranked system's accuracy ranges from 54.3% on the smallest to 61.8% on the largest subset, corresponding to relative improvements of approximately 23% and 9% over the baseline system on the evaluation set.

Index Terms— DCASE Challenge, Acoustic Scene Classification, data-efficiency, low-complexity, multiple devices

1. INTRODUCTION

Acoustic Scene Classification (ASC) aims at detecting the environmental context in which audio was captured, based on a short excerpt [1]. The environmental context is given as a set of pre-defined acoustic scene classes such as *Metro station*, *Urban park*, or *Public square*. Since its inception, the ASC task has been an integral part of the DCASE Challenge. Each year's edition focused on one or multiple challenging machine-learning aspects in addition to the supervised classification task itself. These aspects include open-set classification [2], constraints on the model's size and computational complexity [3–5], and generalization across different recording devices [3,6]. These additional problems target the real-world applicability of ASC systems; for instance, the methods should be robust to diverse recording devices and sufficiently lightweight to be deployable on embedded devices. In the 2024 edition[1] of the ASC task, an additional challenging real-world aspect is addressed: the limited availability of training data. This setting intends to spark research on data-efficient learning methods capable of achieving high classification performance given only a small number of labeled acoustic scene examples for training.

---

[1]Task Description Page: https://dcase.community/challenge2024/task-data-efficient-low-complexity-acoustic-scene-classification
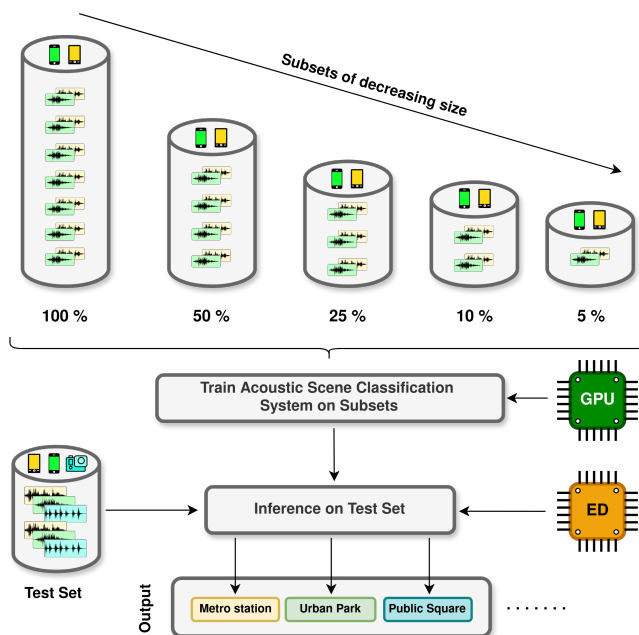

Figure 1: Overview of Data-Efficient Low-Complexity Acoustic Scene Classification. Submitted systems must be trained on five datasets of varying sizes, they must generalize to unseen recording devices, and they are required to be lightweight enough for inference on an embedded device (ED).

Figure 1 shows an overview of the task setup. The ASC systems must be trained on subsets of a fixed training set that progressively limit the number of training samples, where the smallest subset only contains 5% of the audio snippets in the full training set (see Section 3.2). The training procedure is not limited in terms of complexity and may be executed on high-end GPU hardware. However, aligned with real-world requirements, the system must be lightweight for inference such that it can be deployed on embedded devices (see Section 3.3). Additionally, the developed ASC system must be robust to unseen recording devices. To test this ability, the test set includes audio clips recorded by new devices that are not available in the training sets (see Section 3.2).

The rest of this paper is structured as follows: Section 2 briefly discusses the role of low-complexity constraints and the device generalization problem in previous editions of the task. Section 3 intro-

duces the setup for Data-Efficient Low-Complexity Acoustic Scene Classification in the DCASE 2024 Challenge; the baseline system is presented in Section 4. The outcome of the challenge is analyzed in Section 5 and the paper is concludes in Section 6.

2. PREVIOUS EDITIONS

The low-complexity aspect has already been investigated in previous DCASE challenges and has undergone several refinements. In the 2020 [3] and 2021 [4] editions, systems were limited with respect to model size, allowing 500 kB and 128 kB for non-zero parameters, respectively. In the 2022 edition [5], the complexity constraint additionally included computational complexity, allowing a maximum of 30 MMACs (million multiply-accumulate operations), modeled after Cortex-M4 devices. The maximum number of parameters was 128K, with the variable type fixed to INT8. The 2023 edition took this one step further and included model size and computational complexity as part of the ranking metric, requiring participants to tune the system's performance–complexity trade-off. In response to the low-complexity requirements, training techniques such as Sparsification [7], Pruning [8], Quantization [9], or Knowledge Distillation [10] have been extensively studied, and efficient factorized CNN architectures [11–13] have been designed.

Besides low-complexity techniques, substantial research has been conducted on the device mismatch problem. Efforts to improve device generalization involved suppressing device information via normalization [9] and domain adaptation [14], balancing the devices by changing the sampling distribution [15] and augmenting audio segments with device translators [9], Freq-MixStyle [10, 16], and device impulse response augmentation [17].

3. TASK SETUP

While low complexity and generalization across different recording devices are well-studied topics, the specific aspect of interest in the 2024 edition is the limited availability of acoustic scene data for training. Specifically, participants were encouraged to develop data-efficient systems and study techniques that can alleviate the data scarcity problem, such as using extensive audio augmentation methods, transferring knowledge from general-purpose audio datasets, or incorporating well-suited inductive biases.

3.1. Dataset

The task builds on top of the *TAU Urban Acoustic Scenes 2022 Mobile* dataset [3, 6], which was used in the 2022 and 2023 editions of the task [5]. The dataset provides one-second audio snippets with a sampling rate of 44.1 kHz in single-channel, 24-bit format and consists of recordings from ten distinct acoustic scenes.

The audio was recorded in multiple European cities with four recording devices in parallel. The primary device, referred to as device *A*, is a high-quality binaural device, while *B*, *C* and *D* are commonly available consumer devices. Additionally, 10 simulated devices (*S1-S10*) are created using audio from device A and a set of impulse responses from mobile devices. For details on the dataset creation and the exact distribution of devices, please refer to [3].

The data is split into a development and an evaluation set. The development set, consisting of 64 hours of audio, contains 3 real devices (A, B, C) and 6 simulated devices (S1–S6). The evaluation set comprises five unseen devices (D and S7-S10) and two unseen cities, in addition to devices and cities overlapping with the development set. The evaluation set is used to rank submissions and

therefore comes without corresponding scene labels. Device and city information is not provided for recordings in the evaluation set.

3.2. Data-Efficient Evaluation

The development set used for the 2024 challenge is the same one as used in the previous two years and described above. It comes with the same pre-defined split into a development-train and a development-test partition. The development-train set contains six devices (A, B, C, S1-S3), leaving three unseen devices (S4-S6) for the development-test set to measure the device generalization performance.

For the evaluation of data efficiency, this year's setup introduces five pre-defined subsets that progressively limit the available training data and contain 100%, 50%, 25%, 10%, and 5% of the recordings in the development-train set. The distribution of acoustic scenes, cities, and recording devices is kept similar across all subsets. The smaller subsets are fully included in the larger ones, corresponding to the idea of progressively collecting more data.

Participants are allowed to submit up to three different systems that may be specialized for the different training set sizes. Each system must be trained on all five subsets, and the performances on the development-test set must be reported. A system is considered to be the same if its architecture and design choices (such as building blocks, features, data augmentation techniques, decision-making, etc.) remain the same. However, basic hyperparameters like the number of update steps, learning rate, batch size, or regularization strength may vary for training on the different subsets.

All systems must be trained only on the respective subset and the explicitly allowed external resources. The allowed external resources include general-purpose audio datasets, such as AudioSet [18] or FSD50K [19], but no datasets specific to acoustic scenes.

The leaderboard ranking score is computed as follows. First, class-wise macro-averaged accuracies for all $P = 5$ development-train subsets and all N submissions are computed. The accuracy of the n-th submission on the $p\%$ subset is denoted as $ACC_{n,p}$. The scores are then aggregated by choosing the best-performing system for each subset and averaging the resulting accuracies.

$$\text{score} := \frac{1}{P} \sum_{p \in \{5,10,25,50,100\}} \max_{n \in \{1,\dots,N\}} ACC_{n,p} \qquad (1)$$

The outlined setup encourages research into the following scientific questions: how does the performance of systems vary with the number of available labeled training samples? how can systems be adapted to better cope with the limited availability of labeled training data? how can general-purpose audio datasets be exploited to mitigate the need for larger amounts of acoustic scenes?

3.3. System Complexity Requirements

The system complexity is limited in terms of model size and MMACs. The maximum memory allowance for model parameters is 128 kB, with no requirement regarding the numerical representation. That is, participants can trade off the number of parameters and the numerical representation. For example, the memory limit translates to 128K parameters when using 8-bit quantization, or 32K parameters when using 32-bit precision. The computational complexity is limited to 30 MMACs for the inference on a one-second audio segment. These complexity limits are modeled after Cortex-M4 devices (e.g., STM32L496@80MHz or Arduino Nano 33@64MHz).

System Label	Score	Team Rank	Size	MACs	Architecture	Complexity	Dev. Gen.	External
Han_SJTUTHU_task1_2	58.2	**1**	126kB	29M	SSCP-Mobile	fp16, KD, prun.	FMS	PaSST
Shao_NEPUMSE_task1_1	57.2	**3**	107kB	16M	IRMamba	int8, KD	FMS, DIR	PaSST
MALACH24_JKU_task1_1	57.0	**2**	122kB	29M	CP-Mobile	fp16, KD	FMS, DIR	AudioSet
Yeo_NTU_task1_2	56.1	**5**	122kB	29M	CP-Mobile	fp16, KD	FMS, DIR	PaSST
Cai_XJTLU_task1_3	56.0	**4**	126kB	29M	TF-SepNet	int8, KD	FMS, DIR	AudioSet
Park_KT_task1_2	55.4	**6**	126kB	26M	GhostRes2Net	fp16, KD	FMS, DIR	PaSST, EAT
OO_NTUPRDCSG_task1_1	54.8	**7**	116kB	29M	MofleNet	int8	FMS, DIR	-
Werning_UPBNT_task1_1	54.4	**8**	122kB	29M	CP-Mobile	fp16, KD	FMS	AudioSet
Truchan_LUH_task1_1	53.1	**9**	94kB	29M	Isotropic	fp16	FMS, DIR	-
Yan_NPU_task1_1	52.9	**10**	124kB	29M	MAR-CNN	fp32	FMS	-
Baseline	50.7		122kB	29M	CP-Mobile	fp16	FMS	-

Table 1: This table lists the top-ten teams' best systems according to their evaluation set performance. **Team Rank** indicates the team's overall rank, which is based on multiple submitted systems, and **Score** is the average accuracy across all splits of the respective system listed in the table. int8, fp16, and fp32 refer to the numerical precision of model parameters for inference, corresponding to 8, 16, and 32 bits, respectively. KD, FMS, and DIR are abbreviations for Knowledge Distillation, Freq.-MixStyle, and Device Impulse Response augmentation, respectively, and the column **External** indicates external resources used.

4. BASELINE SYSTEM

The baseline system is a simplified version of the top-ranked system submitted to the 2023 edition [20]. It is based on a receptive-field-regularized, factorized CNN design. Audio input is resampled to 32 kHz and converted to mel spectrograms using a 4096-point FFT with a window size of 96 ms and a hop size of approximately 16 ms, followed by a mel transformation with a filterbank of 256 mel bins. The system is trained for 150 epochs using the AdamW optimizer and a batch size of 256. Freq-MixStyle [10, 16] is applied to tackle the device mismatch problem, and time rolling of the waveform and frequency masking are used to augment the training data. The baseline system requires 29.4 MMACs for the inference on a one-second audio clip. The memory required for the model parameters amounts to 122.3 kB, resulting from the 61,148 parameters used in 16-bit precision (fp16).

The baseline's accuracy on the development-test split ranges from 42.40% for the smallest training subset (5%) to 56.99% accuracy for the full set (100%). The performance increases monotonically as the number of audio segments available for training increases. The code and a detailed description of the baseline system are available online[2].

5. CHALLENGE RESULTS

The task received 37 submissions from 17 teams and is therefore the second most popular task in the 2024 edition of the DCASE challenge. The slight decrease in popularity compared to the previous year's edition is likely due to the more complex setup. 16 out of 17 teams outperformed the baseline system and for most of the teams, the performance on the development-test split aligns well with the performance on the evaluation set. The challenge website contains detailed results and descriptions on all submitted systems[3].

Table 1 presents the best systems submitted by the ten top-ranked teams and lists details in terms of architectures, complexity handling, device generalization, and usage of external resources.

Score denotes the average accuracy across all five training set splits on the evaluation set. Note that a team's rank depends on all three allowed submissions, rather than only on the system achieving the highest score (which is why the *Team Rank* column of Table 1 is not perfectly sorted).

5.1. Architectures

In response to the low-complexity constraints and following the trend observed in the previous edition of this task [5], the large majority of systems are based on factorized CNN architectures. Most prominently, factorization is realized via inverted residual blocks, as used in the baseline architecture. Table 1 shows that four out of the ten best systems are based on modified versions of the CP-Mobile architecture [20]. The top-ranked system [21] further reduces CP-Mobile's complexity by factorizing the spatial convolutions with a 3x3 kernel into two separate convolutions with 1x3 and 3x1 kernels. Team *Shao_NEPUMSE* [22] enhances an inverted residual block-based architecture with parallel Mamba blocks [23], a derivative of state space models. Teams *Cai_XJTLU* [24] and *Park_KT* [25] use modified versions TF-SepNet [26] and BCRes2Net [9], respectively, both of which achieved high ranks in previous editions of this task and decouple spatial convolutions over frequency and time dimensions. Team *OO_NTUPRDCSG* [27] introduces MofleNet by enhancing the CP-Mobile architecture with channel shuffle operations; Team *Truchan_LUH* [28] uses an isotropic convolutional architecture following a patch embedding layer; and Team *Yan_NPU* [29] presents MAR-CNN, an asymmetric multi-branch convolutional architecture.

5.2. System Complexity

Knowledge Distillation (KD) can be identified as the most prominent technique to tackle the low-complexity constraints, with the six top-ranked teams using KD. The most popular teacher model is the audio spectrogram transformer PaSST [30]. Among other models that proved to be successful teachers are CP-ResNet [31] (Teams *MALACH24_JKU* [32] and *Shao_NEPUMSE* [22]), BEATs [33] (Team *Cai_XJTLU* [24]), EAT [34] (Team *Park_KT* [25]) and DyMN [35] (Team *Bai_JLESS* [36]). Regarding numerical representation of parameters, both 8-bit and 16-bit precision solutions are among the top-ranked systems. To convert parameters to 8-bit pre-
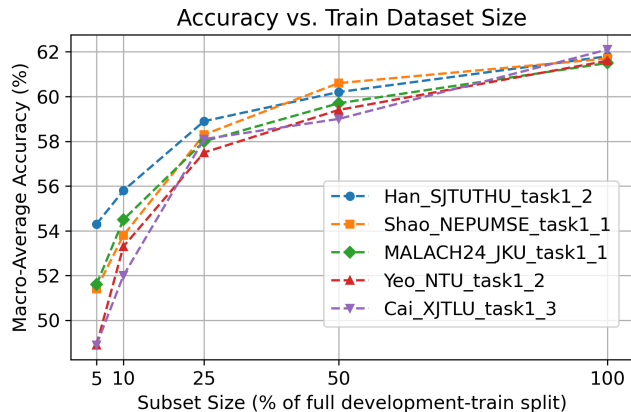
Figure 2: Performance of the best systems from the five top-ranked teams on the evaluation set for training on the five subsets (5%, 10%, 25%, 50%, 100%) of the development-train split.

cision, Teams *Shao_NEPUMSE* [22] and *OO_NTUPRDCSG* [27] use Quantization-Aware-Training, while Team *Cai_XJTLU* [24] shows that also Post-Training Static Quantization can lead to good results. In addition to KD, the top-ranked system by Team *Han_SJTUTHU* [21] uses pruning. They construct a large version of SSCP-Mobile by increasing the number of channels, apply pruning to meet complexity constraints, and then fine-tune the pruned model using KD.

5.3. Device Generalization

The majority of teams tackle the device mismatch with dedicated data augmentation techniques. In this regard, Freq-MixStyle [10, 16], which is also integrated into the baseline system, is used by all of the ten top-ranked teams. Additionally, seven out of the ten top-ranked systems use device impulse response augmentation, implemented using convolution with 66 freely available impulse responses from MicIRP[4]. An interesting alternative is presented by Team *Truchan_LUH* [28] using an adversarial device classifier that forces the feature extractor to learn device-invariant representations.

5.4. Limited Training Data

Figure 2 shows that the top systems submitted to the challenge perform similarly when trained on the 100% train split. However, the smaller the size of the training set, the larger the performance differences, underscoring the large impact of creating data-efficient systems. In fact, the top-ranked system does not achieve the highest accuracy for the 50% and 100% training splits, but it surpasses other systems on the 5%, 10%, and 25% subsets. In particular, on the smallest training set, it outperforms all other teams' systems by more than 2 percentage points in terms of accuracy.

In the following, we describe approaches by participants to counteract the performance dropoff for small training sets.

General-Purpose Audio Datasets: Very commonly, participants make use of large general-purpose audio datasets, in particular, AudioSet [18], to alleviate the data scarcity problem. This is achieved in three different ways: (1) by fine-tuning a large, pre-trained model on ASC and using it as a teacher model in a KD setup; (2) by directly pre-training a low-complexity model on AudioSet; and (3) by extracting audio clips from AudioSet as ad-

ditional training data. The effectiveness of (1) is underlined by the fact that most of the top-ranked teams use an AudioSet pre-trained transformer model as a teacher in a KD setup. For example, Team *Cai_XJTLU* [24] achieves an accuracy of 55.7% on the development-test set when fine-tuning multiple BEATs [33] models on the 5% training subset, which is higher than the Baseline system's accuracy using 10 times as much training data. Regarding (2), the team with the second-best performance on the 5% and 10% subsets, Team *MALACH24_JKU* [32], pre-trains CP-Mobile on AudioSet and reports a large performance gain for fine-tuning on smaller training subsets. Concerning (3), Team *Werning_UPBNT* [37] trains a dataset domain classifier to extract audio clips from AudioSet that are similar to the samples in the respective training sets and uses these as additional samples for KD. Additionally, Team *Surkov_IMTO* [38] selects AudioSet clips from specific event classes such as *Bus* or *Train* and uses them as additional unlabeled samples in a mean-teacher approach.

Extensive Data Augmentation: Besides Freq-MixStyle and DIR augmentation, extensive data augmentation is applied to improve generalization performance on the small training sets. In this regard, Team *MALACH24_JKU* [32] uses FilterAugment [39], Team *Shao_NEPUMSE* [22]) experiments with audio playback, Team *Chen_SCUT* [40] uses Spectrum Modulation. SpecAugment, time rolling, and Mixup are widely used throughout submissions.

Model Size and Architecture: Team *Yeo_NTU* [41] investigated the relationship between model size and performance on small training splits and found that models of reduced complexity generalize better for small training splits. Team *Park_KT* [25] enhanced their network with Snake activation functions and showed that the introduced inductive bias on periodicity leads to a large performance gain on smaller training sets.

6. CONCLUSION

This paper has presented an analysis of Task 1 in the DCASE 2024 challenge, which focused on the real-world deployment of ASC systems with low-complexity constraints, device mismatch, and training data scarcity being the main hurdles to overcome. The task remained the second most popular in the DCASE 2024 challenge, underscoring the high interest in the task despite the increasingly challenging setup. Multiple strategies have been proposed to tackle the limited availability of training data; most highly-performing systems transferred knowledge from a large general-purpose audio dataset to the ASC task, either in the form of pre-trained models or by extracting additional ASC-related audio clips for training. Data augmentation remained a highly important aspect, not only to address device generalization but also to improve generalization capabilities, with only a small training set available. Other solutions to the data scarcity problem involve adapting the model's complexity or building inductive biases into the model architecture. Summarizing the output of the task, several promising techniques have been proposed that can boost performance on downstream tasks when only a small training set is available.

7. ACKNOWLEDGMENT

---

[4]http://micirp.blogspot.com/

8. REFERENCES

[1] E. Benetos, D. Stowell, and M. D. Plumbley, "Approaches to complex sound scene analysis," in *Cham: Springer International Publishing*, 2018.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: Closed and open set classification and data mismatch setups," in *DCASE Workshop*, 2019.

[3] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in DCASE 2020 challenge: Generalization across devices and low complexity solutions," in *DCASE Workshop*, 2020.

[4] I. Martín-Morató, T. Heittola, A. Mesaros, and T. Virtanen, "Low-complexity acoustic scene classification for multi-device audio: Analysis of DCASE 2021 challenge systems," in *DCASE Workshop*, 2021.

[5] 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 *DCASE Workshop*, 2022.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *DCASE Workshop*, 2018.

[7] C.-H. H. Yang, H. Hu, S. M. Siniscalchi, Q. Wang, W. Yuyang, X. Xia, Y. Zhao, Y. Wu, Y. Wang, J. Du, and C.-H. Lee, "A lottery ticket hypothesis framework for low-complexity device-robust neural acoustic scene classification," DCASE Challenge, Tech. Rep., 2021.

[8] K. Koutini, J. Schlüter, and G. Widmer, "CPJKU submission to DCASE21: Cross-device audio scene classification with wide sparse frequency-damped CNNs," DCASE Challenge, Tech. Rep., 2021.

[9] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE Challenge, Tech. Rep., 2021.

[10] 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 Challenge, Tech. Rep., 2022.

[11] J. Tan and Y. Li, "Low-complexity acoustic scene classification using blueprint separable convolution and knowledge distillation," DCASE Challenge, Tech. Rep., 2023.

[12] Y. Cai, M. Lin, C. Zhu, S. Li, and X. Shao, "DCASE2023 task1 submission: Device simulation and time-frequency separable convolution for acoustic scene classification," DCASE Challenge, Tech. Rep., 2023.

[13] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to DCASE23: Efficient acoustic scene classification with cp-mobile," DCASE Challenge, Tech. Rep., 2023.

[14] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU submissions to DCASE'20: Low-complexity cross-device acoustic scene classification with RF-regularized CNNs," DCASE Challenge, Tech. Rep., 2020.

[15] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Hyu submission for the DCASE 2022: Efficient fine-tuning method using device-aware data-random-drop for device-imbalanced acoustic scene classification," DCASE Challenge, Tech. Rep., 2022.

[16] 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.

[17] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *EUSIPCO*, 2023.

[18] 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.

[19] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2022.

[20] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Distilling the knowledge of transformers and CNNs with CP-mobile," in *DCASE Workshop*, 2023.

[21] H. Bing, H. Wen, C. Zhengyang, J. Anbai, C. Xie, F. Pingyi, L. Cheng, L. Zhiqiang, L. Jia, Z. Wei-Qiang, and Q. Yanmin, "Data-efficient acoustic scene classification via ensemble teachers distillation and pruning," DCASE Challenge, Tech. Rep., 2024.

[22] Y.-F. Shao, P. Jiang, and W. Li, "Low-complexity acoustic scene classification with limited training data," DCASE Challenge, Tech. Rep., 2024.

[23] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *CoRR*, vol. abs/2312.00752, 2023.

[24] Y. Cai, M. Lin, S. Li, and X. Shao, "DCASE2024 task1 submission: Data-efficient acoustic scene classification with self-supervised teachers," DCASE Challenge, Tech. Rep., 2024.

[25] J. Park, T. Kim, D. Rho, J. Kim, and G. Lee, "Kt submission: Periodic activation and knowledge distillation for data-efficient low-complexity acoustic scene classification," DCASE Challenge, Tech. Rep., 2024.

[26] Y. Cai, P. Zhang, and S. Li, "TF-SepNet: An efficient 1d kernel design in CNNs for low-complexity acoustic scene classification," in *ICASSP*, 2024.

[27] Y. Oo and N. Srikanth, "Low complexity acoustic scene classification with moflenet," DCASE Challenge, Tech. Rep., 2024.

[28] H. Truchan, T. H. Ngo, and Z. Ahmadi, "Ascdomain: Domain invariant device-adversarial isotropic convolutional neural architecture," DCASE Challenge, Tech. Rep., 2024.

[29] C. Yan, Y. Yu, and X. Xiong, "Submission for DCASE 2024 task1: An asymmetric residual deep neural network for low-complexity acoustic scene classification," DCASE Challenge, Tech. Rep., 2024.

[30] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech*, 2022.

[31] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

[32] N. David, R. Aida, and S. Patrick, "Data-efficient acoustic scene classification with pre-trained CP-Mobile," DCASE Challenge, Tech. Rep., 2024.

[33] 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 *ICML*, 2023.

[34] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: self-supervised pre-training with efficient audio transformer," *CoRR*, 2024.

[35] F. Schmid, K. Koutini, and G. Widmer, "Dynamic convolutional neural networks as efficient pre-trained audio models," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2024.

[36] J. Bai, M. Wang, E.-L. Tan, J. J. S. Yeo, J. W. Yeow, S. Peksi, D. Shi, W.-S. Gan, and J. Chen, "Hierarchical acoustic scene classification with knowledge distillation and pre-trained dynamic networks," DCASE Challenge, Tech. Rep., 2024.

[37] A. Werning and R. Haeb-Umbach, "Upb-nt submission to DCASE24: Dataset pruning for targeted knowledge distillation," DCASE Challenge, Tech. Rep., 2024.

[38] M. Surkov, "Efficient acoustic scene classification using mean-teacher and knowledge distillation," DCASE Challenge, Tech. Rep., 2024.

[39] H. Nam, S. Kim, and Y. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP*, 2022.

[40] G. Chen and Y. Li, "Data-efficient low-complexity acoustic scene classification using parallel attention broad-cast-residual network," DCASE Challenge, Tech. Rep., 2024.

[41] S. Yeo, E.-L. Tan, J. Bai, S. Peksi, and W.-S. Gan, "Data efficient acoustic scene classification using sing teacher-informed confusing class instruction," DCASE Challenge, Tech. Rep., 2024.