DATA-EFFICIENT ACOUSTIC SCENE CLASSIFICATION WITH PRE-TRAINING, BAYESIAN ENSEMBLE AVERAGING, AND EXTENSIVE AUGMENTATIONS

David Nadrchal, Aida Rostamza*, Patrick Schilcher**

Johannes Kepler University Linz, Austria
{k12213656, k12237081, k12222369}@students.jku.at

ABSTRACT

The task of Acoustic Scene Classification (ASC) is to categorize short audio recordings into predefined scene classes. The DCASE community hosts an annual competition on ASC with a special focus on real-world problems such as recording device mismatches, low-complexity constraints, and limited labelled data availability. Solutions like Knowledge Distillation (KD) and task-specific data augmentations have proven effective in tackling these challenges, as demonstrated by their successful application in top-ranked systems. This paper contributes to the research on the real-world applicability of ASC systems by analyzing the effect of AudioSet pre-training on downstream training sets of different sizes. We study the impact of extensive data augmentation techniques, including Freq-MixStyle, device impulse response augmentation, FilterAugment, frequency masking, and time rolling on different training set sizes. Furthermore, the effectiveness of Bayesian Ensemble Averaging over traditional mean ensembling in KD is investigated. The results demonstrate that the proposed methods improve the performance over the DCASE baseline system substantially, with a particularly large gain on the smallest training set, lifting the accuracy by more than 7 percentage points on the development-test split. [1]

***Index Terms*—** Acoustic Scene Classification, CP-Mobile, Knowledge Distillation, AudioSet pre-training, Bayesian Ensemble Averaging, Device Impulse Response augmentation, Freq-MixStyle, FilterAugment

1. INTRODUCTION

Acoustic Scene Classification (ASC) systems aim to categorize audio recordings into predefined scene classes. The Data-Efficient, Low-Complexity ASC task of the DCASE 2024 challenge [1] focuses on the real-world applicability of ASC systems by addressing three major problems, including recording device mismatch, low-complexity constraints, and limited training data availability. The task uses the TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22) [2], consisting of 1-second audio recordings from 10 different scenes. The audio is recorded with three real devices, and six additional devices are simulated, with three of the simulated devices being only available in the test split, highlighting the importance of device generalization. The low-complexity constraints limit the model size to 128 kB and the computational complexity to 30 million multiply-accumulate operations (MACs), ensuring applicability on edge devices. The 2024 edition of this challenge addresses the real-world scenario of limited labelled data, requiring systems to maintain high accuracy with restricted training data across five scenarios: 5%, 10%, 25%, 50%, and 100% of the audio clips in the full training dataset. Systems must be trained exclusively on these subsets and explicitly allowed external resources, such as AudioSet [3].

This paper contributes to the research on the practical application of ASC systems by studying the effect of pre-training the student model on AudioSet. We examine the influence of various data augmentation techniques, such as Freq-MixStyle [4], device impulse response augmentation (DIR) [5], FilterAugment [6], frequency masking, and time rolling (shifting audio and wrapping segments that exceed the end back to the start) on different training set sizes. Additionally, the study evaluates the effectiveness of Bayesian Ensemble Averaging (BEA) compared to traditional mean ensembling in the context of Knowledge Distillation (KD). The proposed systems achieved the second rank in Task 1 of the DCASE 2024 challenge.

We review related work in Section 2, followed by a description of teacher and student architectures in Section 3. We then present the data-efficient training pipeline in Section 4. In Sections 5 and 6, we present the experimental setup and the results, respectively, and the paper is concluded in Section 7.

2. RELATED WORK

ASC Architectures: Convolutional Neural Networks (CNNs) have consistently proven to be leading models for low-complexity ASC [7, 8]. Restricting the receptive field of CNNs, known as Receptive Field Regularization [9, 10], has been shown to notably improve the generalization performance, with successful implementations in BC-ResNet [11] and CP-ResNet [9]. Inspired by efficient CNN architectures from the vision domain [12, 13], CP-Mobile (CPM) is a low-complexity, receptive-field regularized CNN for ASC, constructed of efficient inverted residual blocks. CPM achieved the top rank in the 2023 edition and a slightly simplified version of this architecture, excluding GRN [14], is used as the baseline system in the 2024 edition of this challenge. Recently, Audio Spectrogram Transformers (AST), such as the Patchout faSt Spectrogram Transformer (PaSST) [15], have shown state-of-the-art performance on various downstream tasks in the audio domain, including ASC.

Low-Complexity Techniques: Besides developing efficient architectures, several model compression techniques have been used in the context of ASC to meet the complexity requirements. In this regard, Pruning [16], Quantization [17], and, most importantly, KD [18] have become popular for further reducing system complexity. KD can be pointed out as the single most important technique for reducing the complexity of ASC systems, as it has been

*These authors contributed equally to this work

[1] Source Code: https://github.com/SchilcherPatrick/DCASE24_Task1

consistently used in top-ranked systems submitted to the challenge [19, 20, 21, 22, 8].

Recording Device Generalization: To tackle the device mismatch and generalization problem, various techniques have been explored, including Domain Adaptation [23, 24], training device translators [25], adjusting device sampling frequency [26], and normalizing data [4]. Among these, Freq-MixStyle [4] and DIR [5] augmentation techniques have proven to be particularly effective in boosting performance on unseen devices.

Data augmentation: Data augmentation is a widely used technique in ASC to improve model generalization and prevent overfitting, especially when dealing with small datasets and device mismatches. Some commonly used techniques include Mixup [27], SpecAugment [28], Freq-MixStyle [4], and DIR augmentation [5].

3. ARCHITECTURES

In this section, we present the teacher and student model architectures used in the KD-based training pipeline presented in Section 4.

3.1. TEACHER MODELS: PaSST, CP-ResNet, CP-Mobile

The Patchout faSt Spectrogram Transformer (PaSST) [15] is a self-attention-based AST model that excels in capturing global audio context and has achieved state-of-the-art performance on various downstream tasks in the audio domain [15]. By introducing the patchout mechanism for improved generalization and computational efficiency, PaSST has been shown to be an excellent teacher for low-complexity ASC models [29, 5].

CP-ResNet [9], a receptive-field regularized CNN, has been a successful model in previous ASC tasks [29, 9]. This fully convolutional architecture incrementally builds local features over a spatially restricted area. Receptive-field regularization has been shown to be important for improved generalization in ASC [9, 10].

CP-Mobile [30] (the student model in our setup; as described in the following section) is also included in the teacher ensemble, as described in Section 5.4.

3.2. STUDENT MODEL: CP-Mobile

Inspired by CP-ResNet, CPM is a factorized CNN architecture designed for low-complexity ASC, enhancing both representation capability and efficiency. The core innovation of CPM is the CPM block [22], a computationally efficient alternative to the classical convolutional layer that implements an inverted bottleneck block [12]. Each CPM block includes three factorized convolutional layers integrated with batch normalization and ReLU activation, targeting efficiency, and high representation capability, making CPM ideal for inference on edge devices.

4. DATA-EFFICIENT TRAINING PIPELINE

In this section, we introduce our proposed data-efficient training pipeline. We experiment with training the low-complexity student model, CPM, in three stages: Firstly, we pre-train CPM on AudioSet (Section 4.1), secondly, we train CPM on the respective train split of the TAU dataset (Section 4.2), and finally, we fine-tune CPM on the respective TAU split using KD (Section 4.3).

4.1. AudioSet Pre-Training

We hypothesize that pre-training the student model on a large general-purpose audio dataset, such as AudioSet [3], can reduce the need for extensive labelled data on downstream tasks. AudioSet contains over 2 million human-labeled 10-second sound clips across 527 distinct sound categories. This dataset provides a comprehensive resource for training and evaluating audio recognition models. General knowledge about acoustic events may improve performance on downstream tasks with limited training sets.

Following the training routine in [31], we train the CPM on AudioSet using KD from a large transformer ensemble of nine PaSST [15] models. Despite the high task complexity, the low-complexity network achieves a reasonable mean average precision performance of 0.194.

4.2. Pre-Training on Acoustic Scenes

Before distilling the knowledge from the teacher ensemble into the low-complexity student model, we train the student on the allowed TAU train split. The hypothesis is that pre-training on both AudioSet and TAU would provide a robust initialization by leveraging the diversity of AudioSet and the specific characteristics of TAU. It may also be beneficial for the learning process of the student to gain knowledge on acoustic scene data, before being exposed to the predictions of larger teacher ensembles.

4.3. Knowledge Distillation Fine-Tuning

KD compresses knowledge from a large, high-performing teacher model into a more compact student model while maintaining robust performance. Following [5], we train the student using both the soft targets (probability distributions over classes) from the teacher and hard labels (standard one-hot encoded labels). The overall loss function is defined as:

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

Here, the hard label loss (L_l) is the cross-entropy loss, and the distillation loss (L_{kd}) is the Kullback-Leibler divergence between the teacher's and student's soft targets. z_S and z_T are the logits of the student and teacher models, respectively, and y represents the hard labels. The temperature parameter (τ) controls the distribution sharpness, and the factor τ^2 is a scaling factor for the distillation loss. The contributions of both losses are balanced using a weight λ. This dual training approach allows the student model to capture both explicit label information and the generalized knowledge represented in the teacher's soft targets.

In this KD fine-tuning phase, we indirectly make use of AudioSet [3] a second time, by using KD with an AudioSet pre-trained teacher model, namely, the transformer PaSST [15].

4.3.1. Bayesian Ensemble Averaging

Ensembling teacher models is a common strategy to improve KD. By integrating diverse insights from the teachers, typically done by averaging their logits [5], this technique enhances the robustness and generalization of the student model.

Bayesian Ensemble Averaging (BEA) [32] extends simple averaging of logits by using a probabilistic framework. Inspired by BEA, we implemented a simplified interpretation without explicit distributional assumptions for model outputs. We used the average prediction of the teacher models as the expected prediction (μ_{tl}) and the logit-wise variance (σ^2_{tl}) across teacher models for each sample independently as a proxy for uncertainty. The aggregated prediction

(E_{tl}) combines the mean with a scaled variance to adjust the uncertainty impact based on the number of models (n_{tl}), ensuring proper moderation.

$$E_{tl} = \mu_{tl} + \frac{\sigma_{tl}^2}{n_{tl}} \quad (2)$$

5. EXPERIMENTAL SETUP

5.1. Audio Preprocessing

For all models, we downsample audio to a 32 kHz sampling rate. For the student, we compute Mel spectrograms using 256 frequency bins. The Short Time Fourier Transformation (STFT) is applied with a window size of 96 ms and a hop size of 16 ms. For the PaSST [15] teacher model, we follow its original AudioSet pretraining configuration and for CP-ResNet, the preprocessing remains the same as that of the CPM student except for the hop size being 24 ms.

5.2. Optimization

The student model is pre-trained on TAU using the Adam optimizer for 150 epochs. The training parameters include a weight decay of 0.0001, a learning rate of 0.005, and a warm-up phase of 2000 steps for the scheduler.

For the pre-training experiments, we shortened the KD training to 75 epochs and decreased the learning rate to 0.0025 as the model converged faster, due to prior knowledge of the domain. In other experiments, we applied the same hyperparameters for KD fine-tuning as those used in the student's pre-training on TAU.

As for CP-ResNet, the hyperparameters remain largely the same, with key differences being a learning rate of 0.001 and a weight decay of 0.001.

For PaSST, the learning rate and weight decay values are set to 0.00001 and 0.001, respectively. We use a patch out of 6 on the frequency dimension. The KD ensemble experiments including PaSST were trained using a learning rate of 0.0025

5.3. Data Augmentation

For all models, we use frequency masking of up to 48 frequency bins, time rolling of up to 0.1 seconds, and linear FilterAugment augmentation from 3 to 6 Mel bands in the range of -6 to 6 dB. Other augmentation hyperparameters fine-tuned for the different models are detailed in Table 1.

5.4. Knowledge Distillation

By default, we use an ensemble of one CPM and four CP-ResNet teachers for KD, with their predictions aggregated by BEA. The CP-ResNet teachers receive the same input as the student. In contrast, the PaSST teacher operates on its own spectrograms, which are independently subjected to frequency masking. Notably, Freq-MixStyle and FilterAugment are not applied to PaSST inputs. However, the time-domain augmentations, time-rolling and DIR, remain consistent for the PaSST teachers, the student, and the CP-ResNets.

We use temperature parameter $\tau = 2$ and Kullback-Leibler divergence with a high weight of $\lambda = 0.02$ as our loss function.

Training	lr	DIR p	FMS p	FMS α
CPM	0.005	0.6	0.6	0.4
CP-ResNet	0.001	0.6,0.7,0.8	0.6, 0.7, 0.8	0.3
PaSST	0.00001	0.6	0.4	0.4

Table 1: Hyper-parameters settings for different models. For the student, we use the same hyperparameters both for pre-training on TAU22 and for KD. For CP-ResNet, some hyperparameters differed among the teachers (indicated by multiple values in one cell), creating a diverse ensemble. FMS abbreviates Freq.-MixStyle [4], p stands for the probability that the respective augmentation is applied. α stands for the mixing alpha [4].

6. RESULTS

In this ablation study, we systematically add or remove specific system components to assess their impact on overall performance. This approach helps us understand each component's contribution to the model's accuracy and efficiency. The results reported are averages over three independent experiments.

Effect of Pre-training the Student Model: We evaluated the influence of pre-training the student model before KD by training it in three scenarios: without any pre-training (*Student with no pre-training*), pre-trained on the respective TAU split (*Student pre-trained TAU*), and pre-trained on both AudioSet and TAU (*Student pre-trained on AudioSet and TAU*). The results in Figure 1 and Table 2 with over 3% accuracy improvement on the smallest data split and over 1% on average across all splits, approve our hypothesis(4.2) and indicate that pre-training substantially enhances the model's performance. In contrast, the model trained without pre-training exhibits the lowest accuracy, underscoring the importance of pre-training. Pre-training on both AudioSet and TAU achieves the highest accuracy for the smallest datasets, while its impact is less pronounced in larger datasets, highlighting the effectiveness of AudioSet pre-training in handling minimal data in ASC.

Impact of Teacher Aggregation Methods in Knowledge Distillation: We investigated the impact of teacher aggregation methods in KD, comparing BEA (*KD-BEA*) and mean averaging for CP-ResNet and CP-Mobile teachers (*KD-Mean*). The results in Figure 2 and Table 3 suggest that BEA improves upon mean averaging across all TAU subsets when distilling knowledge to the student model pre-trained on the AudioSet and TAU split subset.

Contribution of including PaSST in the Teacher Ensemble: We evaluated the impact of including PaSST in the teacher ensemble alongside CP-ResNet and CP-Mobile teachers by examining three aggregation scenarios. The first scenario applied BEA to all teachers (*KD-BEA with PaSST*). The second used mean averaging for all teachers (*KD-Mean with PaSST*). The third scenario applied BEA for CP-ResNets and CP-Mobile and used mean averaging for the output of the BEA and the PaSST teacher logits (*KD-Mixed with PaSST*). The results in Figure 2 and Table 3 indicate that while both (*KD-BEA with PaSST*) and (*KD-Mean with PaSST*) perform similarly across all training subsets, the combination of BEA and mean aggregation (*KD-Mixed with PaSST*) methods demonstrates an overall superior performance. However, incorporating PaSST in all scenarios did not lead us to any performance improvements, likely due to resource limitations and the low number of training epochs for PaSST.

Effect of Various Data Augmentation Techniques on Student Model Generalization: We investigated the effect of various data augmentation techniques. As described in Figure 3 and Table 4, using FilterAugment instead of frequency masking (*Using FilterAugment, No Frequency Masking*) resulted in decreased performance across all training subsets. However, incorporating both FilterAug and frequency masking (*Using FilterAugment*), although not outperforming the proposed model system, demonstrated higher performance than alternating between the two. Removing frequency masking (*No Frequency Masking*) entirely led to higher performance in all but one subset compared to the default system. Freq.-MixStyle (*No Frequency Mixstyle*) showed improved performance in smaller subsets, while removing DIR (*No DIR*) caused an overall decrease in performance.

7. CONCLUSION

This paper introduces a data-efficient ASC system and examines various design choices concerning training sets of different sizes. We show that pre-training substantially boosts student model performance in a KD fine-tuning stage and can reduce the need for larger labelled datasets in downstream tasks. Pre-training on comprehensive datasets like AudioSet transfers general knowledge about acoustic events, enhancing model performance on downstream tasks with small training sets. Our simplified BEA can surpass mean aggregation in teacher ensembling. We explore the PaSST transformer's effectiveness for small training sets and assess various data augmentation techniques on model generalization. Our system improves performance over the DCASE 2024 baseline, achieving a 7 percentage point accuracy increase on the development-test split, especially with the smallest training set.

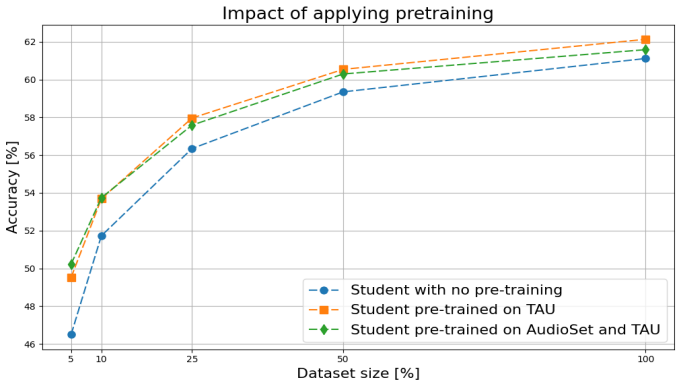

Figure 1: Effect of Pre-training the Student Model on AudioSet

Mode	5%	10%	25%	50%	100%	Avg.
-	46.49±.29	51.72±.12	56.34±.27	59.34±.48	61.11±.18	55.00
TAU	49.52±.15	53.68±.17	**57.96**±.31	**60.53**±.19	**62.13**±.27	**56.76**
AS, TAU	**50.22**±.10	**53.74**±.11	57.58±.20	60.29±.03	61.58±.11	56.68

Table 2: Impact of applying pre-training. AS and TAU indicate pre-training on AudioSet [3] and TAU, respectively, and **Avg.** denotes the average accuracy across all data splits.

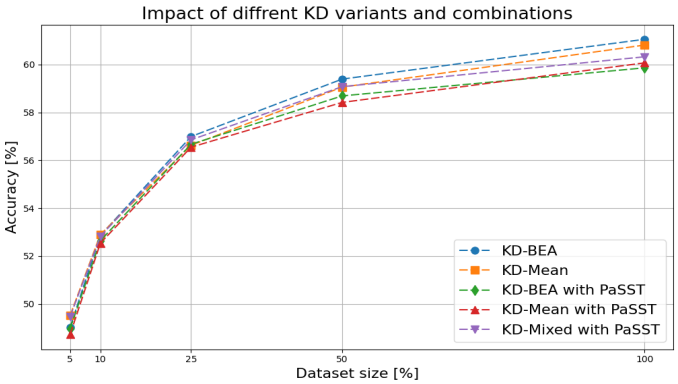

Figure 2: Impact of teacher aggregation methods

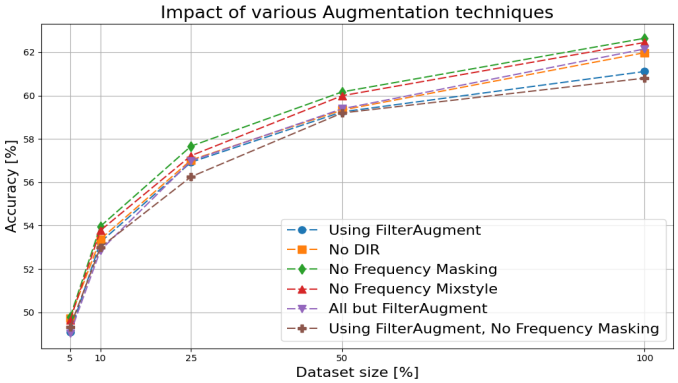

Figure 3: Effects of various data augmentation techniques

Mode	5%	10%	25%	50%	100%	Avg.
BEA	49.02±.48	52.85±.63	**56.99**±.36	**59.39**±.57	**61.05**±.78	**55.86**
Mean	49.12±.28	**52.89**±.45	56.62±.18	59.05±.39	60.81±.46	55.70
BEA+P	48.97±.28	52.64±.25	56.66±.14	58.68±.13	59.86±.35	55.36
Mean+P	48.72±.14	52.53±.03	56.54±.05	58.41±.13	60.06±.14	55.25
Mix+P	49.46±.28	52.82±.14	56.85±.21	59.07±.07	60.32±.17	55.71

Table 3: Impact of different KD variants and combinations. P indicates the inclusion of a PaSST model, and **Avg.** denotes the average accuracy across all data splits.

Method	5%	10%	25%	50%	100%	Avg.
All but FA	49.02±.48	52.85±.63	56.99±.36	59.39±.57	61.05±.78	55.86
+ FA	49.07±.24	53.23±.08	56.92±.07	59.25±.13	61.11±.10	55.92
- DIR	49.73±.41	53.37±.19	57.03±.12	59.33±.20	61.97±.37	56.29
- FM	**49.78**±.04	**53.96**±.10	**57.66**±.10	**60.17**±.16	**62.64**±.09	**56.84**
- FMS	49.64±.21	53.77±.22	57.23±.02	59.99±.10	62.45±.04	56.62
+ FA, - FM	49.32±.15	53.00±.13	56.25±.07	59.19±.06	60.80±.15	55.71

Table 4: Impact of different data augmentations. FA, FM, and FMS are abbreviate FilterAugment [6], frequency masking and Freq.-MixStyle [4], respectively, and **Avg.** denotes the average accuracy across all data splits.

8. REFERENCES

[1] DCASE Community, "Dcase 2024 challenge: Task - data-efficient low-complexity acoustic scene classification," https://dcase.community/challenge2024/task-data-efficient-low-complexity-acoustic-scene-classification, 2024, accessed: 2024-06-03.

[2] T. Heittola, A. Mesaros, and T. Virtanen, "TAU Urban Acoustic Scenes 2022 Mobile, Development dataset," 2022.

[3] 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*.

[4] 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.

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

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

[7] 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 2022*.

[8] 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 2021*.

[9] 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.

[10] 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 *EUSIPCO 2019*.

[11] 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," *CoRR*, 2022.

[12] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR 2018*.

[13] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML 2019*.

[14] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext V2: co-designing and scaling convnets with masked autoencoders," in *CVPR 2023*.

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

[16] 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*.

[17] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, 2017.

[18] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, 2015.

[19] 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," DCASE2024 Challenge, Tech. Rep., 2024.

[20] D. Nadrchal, A. Rostamza, and P. Schilcher, "Data-efficient acoustic scene classification with pre-trained cp-mobile," DCASE2024 Challenge, Tech. Rep., 2024.

[21] Y.-F. Shao, P. Jiang, and W. Li, "Low-complexity acoustic scene classification with limited training data," DCASE2024 Challenge, Tech. Rep., 2024.

[22] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "Cp-jku submission to dcase23: Efficient acoustic scene classification with cp-mobile," in *DCASE 2023*, 2023, pp. 161–165.

[23] 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," DCASE2020 Challenge, Tech. Rep., 2020.

[24] P. Primus, H. Eghbal-zadeh, D. Eitelsebner, K. Koutini, A. Arzt, and G. Widmer, "Exploiting parallel audio recordings to enforce device invariance in cnn-based acoustic scene classification," in *DCASE 2019*.

[25] 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," DCASE2021 Challenge, Tech. Rep., 2021.

[26] 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," DCASE2022 Challenge, Tech. Rep., 2022.

[27] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR 2018*.

[28] D. S. Park, W. Chan, Y. Zhang, 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.

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

[30] F. Schmid, P. Primus, T. Heittola, A. Mesaros, I. Martín-Morató, K. Koutini, and G. Widmer, "Data-efficient low-complexity acoustic scene classification in the dcase 2024 challenge," 2024.

[31] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP 2023*.

[32] J. Xu, S. Li, A. Deng, M. Xiong, J. Wu, J. Wu, S. Ding, and B. Hooi, "Probabilistic knowledge distillation of face ensembles," in *CVPR 2023*.