DISTILLING THE KNOWLEDGE OF TRANSFORMERS AND CNNS WITH CP-MOBILE

Florian Schmid[1], *Tobias Morocutti*[2], *Shahed Masoudian*[1], *Khaled Koutini*[2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU), [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
{florian.schmid, tobias.morocutti, shahed.masoudian}@jku.at

ABSTRACT

Designing lightweight models that require limited computational resources and can operate on edge devices is a major trajectory in deep learning research. In the context of Acoustic Scene Classification (ASC), the DCASE community hosts an annual challenge on low-complexity ASC, contributing to the research on Knowledge Distillation (KD), Model Pruning, Quantization and efficient neural network design. In this work, we propose a system that contributes to the latter by introducing CP-Mobile, a lightweight CNN architecture constructed of residual inverted bottleneck blocks and Global Response Normalization. Furthermore, we improve Knowledge Distillation by showing that ensembling CNNs and Audio Spectrogram Transformers form strong teacher ensembles. Our proposed system improves the results on the *TAU Urban Acoustic Scenes 2022 Mobile development dataset* by around 5 percentage points in accuracy compared to the top-ranked submission for Task 1 of the DCASE 22 challenge and achieves the top rank in the DCASE 23 challenge[1].

Index Terms— CP-Mobile, Receptive Field Regularization, Patchout FaSt Spectrogram Transformer (PaSST), CP-ResNet, Knowledge Distillation, Device Impulse Response augmentation, Freq-MixStyle

1. INTRODUCTION

The task of Acoustic Scene Classification (ASC) is to assign a scene label to an audio clip. The *Low-Complexity Acoustic Scene Classification* task of the DCASE 23 challenge [1] is based on the *TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22)* [2], consisting of 1-second audio clips, each belonging to one of 10 different acoustic scenes. Audio clips are recorded by three real devices and six simulated devices, including three simulated devices that are not included in the train split, making device generalization an important and challenging task. The challenge further introduces limits on the model size (128 kB) and the computational complexity in terms of multiply-accumulate operations (30 million MACs). Systems are ranked according to class-wise averaged accuracy, consumed MACs for the inference of a 1-second audio clip, and the model size, encouraging participants to design models with a good performance-complexity trade-off.

ASC Architectures: Convolutional Neural Networks (CNNs) are well-established models to tackle low-complexity ASC and dominated the leaderboard in previous editions of the challenge [1–3]. Common practice is to regularize the receptive field of CNNs [4, 5], which has been shown to improve the generalization performance. Particularly successful implementations of receptive-field regularized CNNs (RFR-CNNs) include BC-ResNet [6, 7] and CP-ResNet [8–10]. Recently, Audio Spectrogram Transformers achieved competitive results on multiple downstream tasks in the audio domain, including the Patchout FaSt Spectrogram Transformer (PaSST) [11] achieving state-of-the-art results on the *TAU Urban Acoustic Scenes 2020 Mobile development dataset (TAU20)* [2].

Efficient Model Design: A substantial amount of prior work exists on making conventional CNNs more efficient by factorizing convolution operations. In this regard, MobileNets [12, 13] and EfficientNets [14], introduced in the vision domain, have been successfully adapted to the audio domain [15, 16]. MobileNets and EfficientNets are based on inverted bottleneck blocks and inspire CP-Mobile, introduced in Section 3.

Model Compression Techniques: Besides designing efficient architectures, model compression techniques such as Parameter Pruning [17, 18], Quantization [19, 20] and Knowledge Distillation (KD) [21, 22] are popular to reduce a system's complexity further. Quantization to 8-bit precision was forced by the DCASE 22 challenge [1] rules, Parameter Pruning has been successfully applied to ASC systems [6, 9, 23], and KD has been the most successful technique in previous editions of the challenge with the top 3 teams using KD in the DCASE 22 and 23 challenges [1].

Device Generalization Methods: Many different approaches have been applied to counter the distribution shift caused by the unseen devices at test time. In this regard, Domain Adaptation has been used to force device-invariant representations extracted by the model [8, 24]. Other approaches tried to train device translators [6], change the sampling frequency of devices [7], or remove device-specific information by normalization [25]. An augmentation technique called Freq-MixStyle (FMS) [25, 26] lead to the best performance on unseen devices in the DCASE 22 challenge, which recently has been paired with device impulse response (DIR) augmentation to boost the performance further [27].

In this work, we propose a new ASC system, outperforming the top-ranked system in the DCASE 22 challenge by 5% accuracy on the TAU22 development dataset and achieving the top rank in the DCASE 23 challenge. The main contribution of our ASC system is twofold: 1) we achieve a new state-of-the-art teacher model performance by ensembling Audio Spectrogram Transformers and CNNs trained with different FMS and DIR settings, and 2) we introduce CP-Mobile, an efficient, factorized CNN that can distill the knowledge of the large teacher ensemble under low-complexity limits. We introduce the teacher ensemble in Section 2, CP-Mobile in Section 3 and connect them in the KD setup described in Section 4. The results are presented in Section 5, including a detailed ablation study assessing the components of our system.

[1] Source Code: `https://github.com/fschmid56/cpjku_dcase23`

2. TEACHER ENSEMBLE: PASST AND CP-RESNET

Audio spectrogram transformer models such as PaSST [11] are purely self-attention-based models, making them excellent at capturing the global context of an audio clip. PaSST has been shown to be a good teacher model for low-complexity CNNs [10,16,26]. CP-ResNet (CPR) [4], however, is a RFR-CNN that gradually builds local features covering a spatially restricted size before applying a global pooling operation.

Experiments in [26] and [16] show that high-performing ensembles can be achieved by ensembling PaSST models trained with varying FMS [25,26] and model configurations. To further increase the diversity of predictions in the ensemble, we experiment with including models trained with DIR augmentation [27] and CPR models. We follow the model configurations and training protocols used in [27] and use a CPR with 128 base channels, resulting in a model with approximately 4M parameters. We finetune the AudioSet [28] pre-trained PaSST, consisting of 85M parameters, on the TAU22 dataset, using a structured patchout of 6 on the frequency dimension. In addition to the training protocol of [27], we augment TAU22 by adding shifted crops of the reassembled 10-second audio clips, as proposed in [29]. PaSST and CPR models are trained in 4 different configurations: 1) using no device generalization method, 2) using DIR, 3) using FMS and 4) using the combination of DIR and FMS. Hyperparameters for DIR and FMS are chosen according to [27] and set to $\alpha = 0.4$, $p_{FMS} = 0.4$ and $p_{DIR} = 0.6$ for PaSST and to $\alpha = 0.4$, $p_{FMS} = 0.8$ and $p_{DIR} = 0.4$ for CPR. The results for the teacher ensembles are presented in Section 5.1.

3. STUDENT MODEL: CP-MOBILE

In this section, we introduce CP-Mobile (CPM), a novel efficient architecture for ASC. The goal is to maintain beneficial properties from CPR [4,5], such as the regularized receptive field, while reducing the complexity and factorizing convolution operations, such as in MobileNets [12,13] or EfficientNets [14]. Given that the teacher ensemble consists of multiple millions of parameters, an important point is to increase the student model's capacity to be able to distill as much knowledge as possible from the teacher ensemble to the student, even in a low-complexity setting.

Input	Operator	Stride
256 x 64 x 1	Conv2D@3x3, BN, ReLU	2 x 2
128 x 32 x BC/4	Conv2D@3x3, BN, ReLU	2 x 2
64 x 16 x BC	CPM Block S	1 x 1
64 x 16 x BC	CPM Block D	2 x 2
32 x 8 x BC	CPM Block S	1 x 1
32 x 8 x BC	CPM Block T	2 x 1
16 x 8 x BC*CM	CPM Block S	1 x 1
16 x 8 x BC*CM	CPM Block T	1 x 1
16 x 8 x BC*CM^2	Conv2D@1x1, BN	
16 x 8 x CLS	Avg. Pool	

Table 1: CP-Mobile Architecture: *Input*: frequency bands x time frames x channels; *Conv2D@KxK*: Conv2D with kernel size KxK; *BC*: Base Channels; *CM*: Channels Multiplier; *CPM Block S/D/T*: Standard/Downsampling/Transition; *CLS*: Number of Classes

First, we factorize all 3x3 convolution operations in CPR into a pointwise expansion convolution, a depthwise convolution and a pointwise projection convolution and obtain residual inverted bottleneck blocks (referred to as *CPM blocks* in the following). We replace the max pool operations with strided convolutions to downsample the spatial dimensions. All shortcut paths that require an additional pointwise convolution are removed and the strided input convolution is split into two separate strided convolutions to reduce the computational burden when operating on the high-dimensional input spectrograms. We experiment with Relaxed Instance Frequency-wise Normalization [25], SubSpectral Normalization [30] and Global Response Normalization (GRN) [31] integrated into different positions in the CPM blocks. While substantial improvements for multiple normalization and position combinations can be achieved, using GRN after adding the shortcut and before the final ReLU activation leads to the highest performance gain.

Table 3 shows the architecture of CPM. CPM's complexity scales in four dimensions: number of blocks (depth), number of base channels (BC), network width modified using the channels multiplier (CM) and expansion rate of inverted bottlenecks (EXP). The depth of the network and the strides determine the receptive field of the model. The overall spatial downsampling factor and the position of the strided convolutions are inspired by the max pooling layer positions in the low-complexity CP-ResNet in [10]. Experimenting with CPM models of varying depths, we find that using 6 CPM blocks creates a suitable receptive field size.

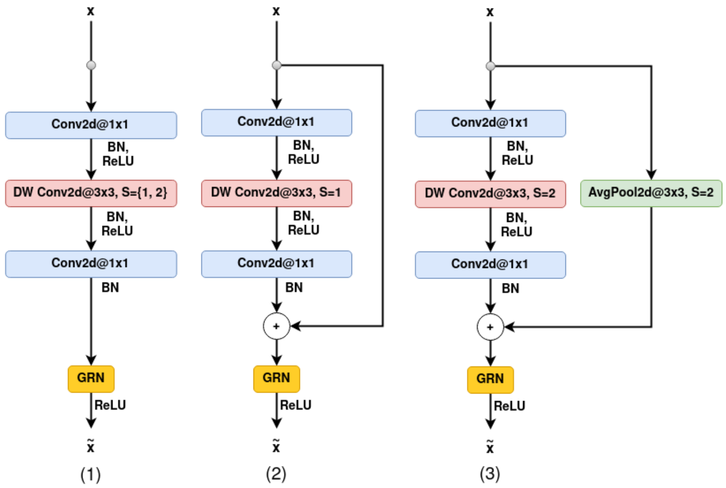

Figure 1: CPM blocks: (1) Transition Block (input channels ≠ output channels), (2) Standard Block, (3) Spatial Downsampling Block (S denotes stride)

Figure 1 depicts the structure of a CPM block consisting of two pointwise and a depthwise convolution. The depthwise convolution operates on the expanded channel representation, which has the size of the number of block input channels times the scaling factor EXP. We differentiate between Transition, Standard and Spatial Downsampling blocks (CPM blocks T, S, D). CPM block T increases the channel dimension, uses no residual connection and can be used with a strided depthwise convolution. CPM blocks S and D have matching input and output channel dimensions and use a residual connection. CPM block D uses average pooling with a kernel size of 3 and a stride of 2 on the shortcut path to match the spatial dimensions of the block output. GRN [31] is applied before the final ReLU activation. GRN calculates a normalization value $\mathcal{N}_i$ for each channel, where $||X_i||$ is the L2-norm of channel i:

$$\mathcal{N}_i = \frac{||X_i||}{\sum_c^C ||X_c||/C} \tag{1}$$

The normalization values $\mathcal{N}_i$ are used to calibrate the channel responses, including two trainable parameters γ and β and a residual connection: $\hat{X}_i = \gamma * \mathcal{N}_i * X_i + \beta + X_i$. GRN was introduced in [31] to increase the feature diversity across channels. The main consideration for using GRN in CPM is to avoid feature redundancies in models with restricted capacity.

4. KNOWLEDGE DISTILLATION SETUP

Similar to [16], CPM is trained on the one-hot encoded labels and the pre-computed predictions of the teacher ensemble described in Section 5.1. Compared to the hard labels, the teacher soft labels describe blurred decision boundaries and establish important similarity structures between classes. The loss, consisting of a combination of hard label loss L_l and distillation loss L_{kd}, is given in Equation 2. λ is a weight that trades off label and distillation loss, z_S and z_T are student and teacher logits, y denotes the hard labels and τ is a temperature to control the sharpness of the probability distributions created by the softmax activation δ. L_l is the Cross-Entropy loss and Kullback-Leibler divergence is used as distillation loss L_{kd}.

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

4.1. Experimental Setup

Preprocessing: For training the student models, the raw audio is downsampled to 32 kHz and Mel spectrograms with 256 frequency bins are computed. Short-Time Fourier Transformation is applied with a window size of 96 ms and a hop size of 16 ms. Increasing the window size from 64 to 96 ms and applying a 4096-point FFT leads to a slight improvement compared to [10], as shown in Table 4 (*large FFT window*).

Training: CPM student models are trained for 75 epochs on the TAU22 dataset with the shifted crops dataset augmentation described in [29]. We use a batch size of 256, Adam optimizer [32] and a learning rate scheduler that increases the learning rate to its peak value until epoch 7 and linearly decreases it from epoch 25 to 67 to 0.5% of the peak value. The peak learning rate varies for models of different sizes and is listed in Table 3. For device generalization, we use FMS [25, 26] and DIR augmentation [27] and set the hyperparameters $\alpha = 0.4$, $p_{FMS} = 0.4$ and $p_{DIR} = 0.6$. For KD [21], setting $\tau = 2$ and using a high weight on the distillation loss with $\lambda = 0.02$ turned out beneficial.

5. RESULTS

Below, we give the results of the teacher ensembles, analyze the performance of CPM models scaled to different complexity levels and offer a detailed ablation study of our system's main components.

5.1. Teacher Ensemble Results

Table 2 lists CPR and PaSST models trained with different DIR and FMS configurations and the accuracies achieved by individual models and the respective ensembles. Rows starting with *Configs* specify the combination of PaSST and CPR models or models trained with different FMS and DIR settings. The models in the *Configs* ensembles are chosen randomly from the pool of available models, such that each config, indicated by the superscript, is equally represented. All ensembles are created by averaging the logits of the individual models and # specifies the number of models in the ensemble.

Besides the known fact [27] that device generalization via FMS and DIR improves the accuracy substantially compared to the baselines (^{1} and ^{5}), two important observations can be made:

1) Training with different device generalization methods leads to models with varying device expertise, increasing the ensemble's diversity. Therefore, ensembles consisting of models trained with different settings for FMS and DIR outperform ensembles consisting of models trained with the same configuration. This effect is more dominant for CPR, where the setting *Configs:* ^{2,3,4} improves by 0.74% accuracy over the *CPR*^{4} configuration, even though the individual models that make up the ensemble score on average 1.24% lower in accuracy compared to the *CPR*^{4} setting.

2) The ensemble's diversity is further extended to different views on the data. CPR focuses on building local features while PaSST models the global context. Independent of the device generalization method, ensembling PaSST and CPR leads to a substantial performance improvement with the ensembles *Configs:* ^{1,5} and *Configs:* ^{4,8} outperforming the individual models that make up the ensemble by around 5% accuracy.

Model Config				Ensemble	
Model	**FMS**	**DIR**	**Acc.**	**#**	**Acc.**
CPR^1	✗	✗	$56.40_{\pm 0.18}$	3	57.47
CPR^2	✗	✓	$58.96_{\pm 0.21}$	3	60.06
CPR^3	✓	✗	$62.27_{\pm 0.22}$	3	63.22
CPR^4	✓	✓	$62.56_{\pm 0.33}$	3	63.74
Configs: ^{2,3,4}			$61.32_{\pm 1.67}$	3	64.48
PaSST^5	✗	✗	$59.48_{\pm 0.64}$	3	60.99
PaSST^6	✗	✓	$61.55_{\pm 0.05}$	3	62.51
PaSST^7	✓	✗	$61.08_{\pm 0.38}$	3	62.06
PaSST^8	✓	✓	$62.19_{\pm 0.15}$	3	63.28
Configs: ^{6,7,8}			$61.82_{\pm 0.40}$	3	63.37
Configs: ^{1,5}			57.48	2	62.52
Configs: ^{4,8}			62.40	2	67.30
Configs: ^{2,3,4,6,7,8}			$61.49_{\pm 1.30}$	12	**68.16**

Table 2: Results of the teacher models CPR and PaSST and the respective ensembles on TAU22 [2]. The *Model Config* section indicates the configuration and the average accuracy and standard deviation of individual models. The *Ensemble* section lists the ensemble size (#) and the accuracy achieved by the ensemble.

For building the final teacher ensemble, we exploit both observations. *Configs:* ^{2,3,4,6,7,8} is constructed of 6 CPR and 6 PaSST models, each including 2 models using DIR, 2 using FMS and 2 using DIR and FMS. Constructing even larger ensembles does not improve the accuracy considerably. This final ensemble achieves an accuracy of 68.16%, leading to an improvement of approximately 5.3% accuracy compared to the PaSST-only teacher ensemble used in the top-ranked submission for the DCASE 22 challenge (62.82%) [10]. We generate the predictions for the TAU22 development set and the added shifted crops [29], average the logits of the

12 models and reuse them to train our CPM students.

5.2. Student Models at Different Scales

Table 3 shows CPM models with different model scaling hyperparameters **BC**, **CM** and **EXP**, as introduced in Section 3. We find that the number of base channels **BC** should be adapted to the required complexity level, e.g., models below 10k parameters achieve the best performance with **BC=8**, while models with **BC=32** work best for models above 50k parameters. While small accuracy improvements can be achieved when scaling up **CM** and **EXP**, the performance quickly saturates for values larger than 2. To achieve the best performance, the learning rate needs to be increased for smaller models.

All accuracies presented in Table 3 are based on models quantized to 8-bit precision. The Quantization Aware Training [20] applied to CPM models is detailed in [29]. Our smallest model outperforms the DCASE baseline system [1] by almost 10% accuracy while requiring only around 12% of the model size and 5% of MACs. The largest CPM model presented achieves an accuracy of 63.21%, improving the accuracy by around 4% compared to the top-ranked system [10] of the DCASE 22 challenge [1] while being more than two times smaller in terms of model size and requiring around 50% of the number of MACs.

Model	**BC**,**CM**,**EXP**,**LR**	**Size (B)**	**MMACs**	**Acc.**
CPM	8,2.1,1.7,0.003	5,722	1.58	$52.61_{\pm 1.25}$
CPM	16,1.5,1.75,0.003	12,310	4.35	$58.42_{\pm 0.51}$
CPM	24,1.5,1.9,0.002	30,106	9.64	$61.77_{\pm 0.54}$
CPM	32,1.7,1.9,0.001	54,182	16.80	$63.21_{\pm 0.44}$
DCASE BL. [1]		46,512	29.23	$42.9_{\pm 0.77}$

Table 3: **BC**, **CM** and **EXP** are model scaling hyperparameters introduced in Section 3 and **LR** denotes the learning rate. **Model Size** is given in Bytes after quantization and **MMACs** specifies million multiply-accumulate operations required for the inference of a 1-second audio clip. The presented accuracies are reported in terms of the mean and standard deviation of 3 independent runs.

5.3. Ablation Study

Table 4 presents an ablation study of our system using a CPM with scaling factors BC=32, CM=2.3 and EXP=3, resulting in a model with 127k parameters and 29 million MACs. Removing one component at a time, the results reveal that KD, and even more, the new CPM architecture, are the dominating performance factors. In the following, the results are analyzed in detail.

CPM: The setting ”- CP-Mobile” indicates the use of the low-complexity CP-ResNet used in the top-ranked submission for DCASE 22 [10] integrated into our setup. CPM outperforms CPR by 4.54% in accuracy while the two models are of comparable complexity, demonstrating the increased model capacity of CPM to distill knowledge from the teacher ensemble. GRN is an integral part of the CPM blocks, improving accuracy by 1.53% and the residual connections are also an important factor accounting for an increase of 1% in accuracy.

KD: KD is an important component of our system, increasing the accuracy by 3.41%. However, using no KD, CPM still performs only 0.31% worse in accuracy than the best single teacher model (CPR[4]) while having only 3.2% of its parameters, underlining the efficiency of CPM. Excluding the PaSST or CPR models from the teacher ensemble leads to a drop in accuracy of 0.81% and 1.22%, respectively, showing that the student benefits from the performance gain of ensembling Transformers and CNNs but can not fully exploit the large improvement of the teacher ensemble.

Device Generalization: The results underline that the combination of DIR and FMS to tackle device generalization works best and using no device generalization method leads to a severe performance drop (-1.87% accuracy). In particular, the ability to generalize to unseen devices suffers with the unseen accuracy dropping by 4.18% in terms of accuracy when neither DIR, nor FMS is used.

Augmentation and Preprocessing: Using a larger FFT window size compared to the setup used in [10] and applying the shifted crop dataset augmentation introduced in [29] improves the system's performance slightly.

System	**Accuracy**	**Acc. Diff**	**Unseen Acc.**
Our Proposed System	$\mathbf{65.66}_{\pm 0.88}$	Ref. Val.	$61.68_{\pm 1.15}$
- CP-Mobile	$61.12_{\pm 0.44}$	-4.54	$57.45_{\pm 0.63}$
- GRN	$64.13_{\pm 0.58}$	-1.53	$60.51_{\pm 0.88}$
- Residual Connections	$64.65_{\pm 0.23}$	-1.01	$61.07_{\pm 0.38}$
- KD	$62.25_{\pm 0.41}$	-3.41	$56.72_{\pm 0.23}$
- PaSST teachers	$64.85_{\pm 0.21}$	-0.81	$60.70_{\pm 0.51}$
- CP-ResNet teachers	$64.44_{\pm 0.37}$	-1.22	$61.19_{\pm 0.68}$
- DIR	$64.74_{\pm 0.33}$	-0.92	$59.99_{\pm 0.23}$
- FMS	$65.15_{\pm 0.36}$	-0.51	$60.05_{\pm 0.59}$
- DIR, FMS	$63.79_{\pm 0.39}$	-1.87	$57.50_{\pm 0.64}$
- large FFT window	$65.29_{\pm 0.04}$	-0.37	$61.68_{\pm 0.34}$
- Shifted Crops	$65.28_{\pm 0.11}$	-0.38	$61.73_{\pm 0.07}$

Table 4: Ablation Study of our proposed setup using CPM (127k params, 29 million MACs) and removing one component at a time. **Acc. Diff.** specifies the difference in accuracy compared to the full system and **Unseen Acc.** is the accuracy on devices unseen during training. All accuracies are reported in terms of the mean and standard deviation of 3 independent runs.

6. CONCLUSION

In this work, we propose a system that advances the state of the art in low-complexity Acoustic Scene Classification with two main contributions: Firstly, we improve Knowledge Distillation by forming teacher ensembles consisting of CNNs and Transformers trained with Freq-MixStyle and Device Impulse Response augmentation. Secondly, we introduce an efficient CNN architecture, CP-Mobile, with residual inverted bottleneck blocks and Global Response Normalization. CP-Mobile can be scaled down to a size of 5.7 kB while still beating the DCASE baseline system by almost 10 % in accuracy. Finally, we assess the importance of our system's components in a detailed ablation study and confirm the high impact of CP-Mobile and Knowledge Distillation. The proposed system outperforms the top-ranked approach for the DCASE 22 challenge by more than 5% in terms of accuracy on the TAU22 development set.

7. ACKNOWLEDGMENT

The LIT AI Lab is supported by the Federal State of Upper Austria. GW's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

8. 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," in *DCASE Workshop*, 2022.

[2] 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.

[3] I. Martin, 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.

[4] 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.

[5] 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*. IEEE, 2019.

[6] 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.

[7] 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.

[8] 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.

[9] K. Koutini, S. Jan, and G. Widmer, "CPJKU Submission to DCASE21: Cross-Device Audio Scene Classification with Wide Sparse Frequency-Damped CNNs," DCASE2021 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," DCASE2022 Challenge, Tech. Rep., 2022.

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

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

[13] A. Howard, R. Pang, H. Adam, Q. V. Le, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, and Y. Zhu, "Searching for mobilenetv3," in *ICCV*. IEEE, 2019.

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

[15] Y. Gong, Y. Chung, and J. R. Glass, "PSLA: improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2021.

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

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

[18] C. Liu, Z. Zhang, and D. Wang, "Pruning deep neural networks by optimal brain damage," in *INTERSPEECH*. ISCA, 2014.

[19] 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.

[20] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. G. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *CVPR*. IEEE, 2018.

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

[22] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *NeurIPS*, 2014.

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

[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 Workshop*, 2019.

[25] 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*. ISCA, 2022.

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

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

[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*. IEEE, 2017.

[29] F. Schmid, T. Morocutti, S. Masoudian, K. Koutini, and G. Widmer, "CP-JKU submission to dcase23: Efficient acoustic scene classification with cp-mobile," DCASE2023 Challenge, Tech. Rep., 2023.

[30] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," in *ICASSP*. IEEE, 2021.

[31] 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," *CoRR*, 2023.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.