CP-JKU SUBMISSION TO DCASE23: EFFICIENT ACOUSTIC SCENE CLASSIFICATION WITH CP-MOBILE

Technical Report

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

In this technical report, we describe the CP-JKU team's submission for Task 1 *Low-Complexity Acoustic Scene Classification* of the DCASE 23 challenge. We introduce a novel architecture, *CP-Mobile*, with regularized receptive field and residual inverted bottleneck blocks. We use Knowledge Distillation to teach *CP-Mobile* from an ensemble of multiple *Patchout faSt Spectrogram Transformer (PaSST)* and *CP-ResNet* models. To enhance cross-device generalization performance, Freq-MixStyle and Device Impulse Response (DIR) augmentation are applied while training teachers and students. *CP-Mobile* is fine-tuned using Quantization Aware Training and then quantized to perform computations in 8-bit precision. The improved teacher ensemble, the efficient student architecture and DIR augmentation improve 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[1].

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

1. INTRODUCTION

In Task 1 of the DCASE 23 Challenge [1], *Low-Complexity Acoustic Scene Classification (ASC)*, participants are required to design a system that accurately predicts scene labels for 1-second audio clips. Well-known challenges from previous editions of this task [2, 3] are the recording device mismatch between train and test sets and the model complexity limits. In addition to the hard complexity limits in terms of model size (128 kB) and multiply-accumulate operations (30 Million MACs), in this year's edition, ASC systems are ranked not only based on class-wise accuracy but also based on model size and MACs. This introduces a new important objective: optimizing systems towards a good performance-complexity trade-off.

Convolutional Neural Networks (CNN) are well-established models to tackle low-complexity ASC and dominated the leaderboard in previous editions of the challenge [3–7]. Common practice is to regularize the receptive field of CNNs [8, 9], which has been shown to improve the generalization performance. Recently, Audio Spectrogram Transformers achieved competitive results on multiple downstream tasks in the audio domain, including the Patchout FaSt Spectrogram Transformer (PaSST) [10] achieving state-of-the-art results on the *TAU Urban Acoustic Scenes 2020 Mobile development dataset (TAU20)* [2]. Transformers are complex models and do not scale to the complexity constraints imposed by the challenge. However, it has been shown that PaSST models are excellent teachers for low-complexity CNNs [4, 11, 12], leading to the top-ranked submission in Task 1 of the DCASE 22 challenge, and a new state-of-the-art performance on AudioSet [13]. The top-ranked submission for DCASE 22 Task 1 [4] involved a low-complexity version of the *CP-ResNet* [8, 9] trained from a PaSST ensemble using Knowledge Distillation, and Freq-MixStyle [11, 14] to tackle generalization to unseen devices.

This technical report describes substantial improvements over the system outlined above. Firstly, we find that ensembling transformers and CNNs forms very strong teachers, outperforming PaSST-only teacher ensembles by a wide margin. Secondly, to improve device generalization further, we apply Device Impulse Response augmentation [15] in addition to Freq-MixStyle. Thirdly, the main contribution is a new CNN design based on residual inverted bottleneck blocks, including Global Response Normalization [16], which we call *CP-Mobile*.

2. DATA PREPROCESSING AND AUGMENTATION

2.1. Preprocessing

For CP-Mobile, we use audio at a sampling rate of 32 kHz to compute Mel spectrograms with 256 frequency bins. Short Time Fourier Transformation (STFT) 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 the setting in [4].

Regarding the teacher models, we match the PaSST [10] AudioSet [13] pre-training settings using a window size of 25 ms and a hop size of 10 ms and create Mel spectrograms with 128 bins. For CP-ResNet [8], we downsample the audio to 22.05 kHz, use a hop size of approximately 9 ms, a window size of 23 ms and 256 Mel bins.

For all models, we randomly roll the waveform over time with a maximum shift of 125 ms. For CP-Mobile and PaSST, we additionally use frequency masking with a maximum size of 48 Mel bins and apply pitch shifting by randomly changing the maximum frequency of the Mel filter bank [17].

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

2.2. Shifted Crops

The *TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22)* [2] has the same content as the TAU20 dataset, except that the 10-second audio clips of TAU20 have been split into 1-second audio fragments, which makes the prediction task considerably harder. However, the 1-second audio fragments of the TAU22 dataset are annotated with sequence numbers, allowing for reassembling the full 10-second audio clip. To increase the diversity in the training dataset, we apply a circular shift of 0.5 seconds, split the 10-second clip into 1-second pieces again and add the shifted audio clips to the original TAU22 dataset. Compared to random 1-second crops at training time as used in [4], we can precompute the teacher predictions on the standard and the shifted versions of TAU22 and perform offline Knowledge Distillation (KD) [18], creating a much more efficient training routine.

2.3. Freq-MixStyle and Device Impulse Response Augmentation

Freq-MixStyle (FMS) [11, 14] is a frequency-wise version of the original MixStyle [19] that operates on the channel dimension. FMS normalizes the frequency bands in a spectrogram and then denormalizes them with mixed frequency statistics of two spectrograms. FMS is applied to a batch with a certain probability specified by the hyperparameter p_{FMS} and the mixing coefficient is drawn from a Beta distribution parameterized by a hyperparameter α.

As introduced in [15], we use the 66 freely available device impulse responses (DIRs) from MicIRP[2] to augment the waveforms. The recording devices in MicIRP have a very characteristic frequency response, making them a good source for simulating a wide range of different recording devices. DIR augmentation has one hyperparameter p_{DIR} that specifies the probability of a waveform being convolved with a DIR.

To train the student model CP-Mobile, we use a combination of FMS and DIR augmentation and choose the hyperparameters according to [15] as $\alpha = 0.3$, $p_{FMS} = 0.4$ and $p_{DIR} = 0.6$. As specified in Section 3, we use teachers trained with different FMS and DIR augmentation settings and ensemble them.

3. TEACHER MODELS: PASST AND CP-RESNET

Audio spectrogram transformer models such as PaSST [10] are purely self-attention-based models, making them excellent at capturing the global context of an audio clip. PaSST has shown to be a good teacher model for low-complexity CNNs [4, 11, 12]. CP-ResNet [8] is a receptive-field regularized CNN (RFR-CNN) that gradually builds local features covering a spatially restricted size.

Experiments in [11] and [12] show that high-performing ensembles can be achieved by ensembling PaSST models trained with varying configurations. To further increase the diversity of predictions in the ensemble, we experiment with including CP-ResNet models. In total, we train six different PaSST and six different CP-ResNet models, including two models using only DIR, two models using only FMS and two models using a combination of both. This results in ensembles that include different views on the data (PaSST – global context, CP-ResNet – local features) and different device experts. For PaSST and CP-ResNet models we use the model configurations and the training protocol described in [15].

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

Ensemble	**# Models**	**Accuracy**
CP-ResNet (DIR)	2	59.57
CP-ResNet (FMS)	2	63.65
CP-ResNet (DIR + FMS)	2	63.62
CP-ResNet (All)	6	64.85
PaSST (DIR)	2	62.30
PaSST (FMS)	2	62.11
PaSST (DIR + FMS)	2	63.11
PaSST (All)	6	63.63
CP-ResNet, PaSST (DIR + FMS)	4	67.84
CP-ResNet, PaSST (All)	12	**68.31**

Table 1: Results of training the teacher models CP-ResNet and PaSST on TAU22 [2] and ensembling the logits. DIR refers to device impulse response augmentation, $p_{DIR} = 0.4$ is used for CP-ResNet and $p_{DIR} = 0.6$ is used for PaSST. FMS refers to Freq-MixStyle, $p_{FMS} = 0.4$ is set for PaSST and $p_{FMS} = 0.8$ is set for CP-ResNet. $\alpha = 0.4$ is the same for both model types. **Accuracy** is calculated based on the averaged logits of **# Models**.

The results presented in Table 1 support our claims. For both CP-ResNet and PaSST models, ensembling DIR, FMS, and DIR + FMS trained models significantly boosts the accuracy, outperforming the PaSST ensemble used in the top-ranked submission of last year (62.6% accuracy) [4]. A major improvement is achieved by ensembling CP-ResNet and PaSST logits, lifting the ensemble performance to 68.31% accuracy. We generate the predictions for the TAU22 development set and the added shifted crops described in Section 2.2, ensemble the logits of the 12 models and reuse them to train all our CP-Mobile students introduced in Section 4.

4. STUDENT MODEL: CP-MOBILE

Our baseline architecture is the low-complexity CP-ResNet model described in [4]. Different versions of the CP-ResNet performed well in previous editions of the challenge [1, 2, 5, 17]. We redesign the model to increase its representation capability and efficiency and call the final model CP-Mobile (CPM). The following steps evolve a CP-ResNet into a CPM:

- The most expensive operations in CP-ResNet are 3x3 convolutions. We replace each 3x3 convolution with a residual inverted bottleneck block (CPM block) similar to the design of MobileNets [20, 21] and EfficientNets [22, 23].
- We experiment with Relaxed Instance Frequency-wise Normalization [14], SubSpectral Normalization [24] and Global Response Normalization (GRN) [16] integrated into different positions in the CPM blocks. While we found substantial improvements for multiple normalization and position combinations, using GRN after the final ReLU activation leads to the highest performance gain.
- Instead of max pooling layers, we use strided convolutions to downsample the spatial dimensions.
- In the introduced CPM blocks, we only use shortcuts if the number of input and output channels are matching, thus we save the complexity for shortcut paths that require a pointwise upsampling convolution.

- We find that the input convolution operating on high spatial dimensions consumes a high amount of MACs. To reduce this amount, we split the input convolution into two separate convolutions and use a stride of 2 for both.

Table 2 shows the overall architecture of CPM. CPM's complexity scales in four dimensions: number of blocks (depth), base channels (BC), network width (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 original max pooling layer positions in the low-complexity CP-ResNet [4]. At the early stages of model design, we experimentally fixed the depth to 7 CPM blocks and with it the model's receptive field.

Table 2: CP-Mobile Architecture

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*CM²	CONV2D@1X1, BN	
16 x 8 x CLS	AVG. POOL	

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

Figure 1 depicts the structure of a CPM block. A standard convolution layer is factorized into a pointwise expansion convolution, a depthwise convolutional and a pointwise projection 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 is used to increase the channel dimension, uses no residual connection and can be used with a strided depthwise convolution. CPM blocks S and D both 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 as the shortcut path to match the spatial dimensions of the block output. GRN [16] 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:

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

The normalization values $\mathcal{N}_i$ are used to calibrate the channel responses, including two trainable parameters γ and β and a residual connection: $\hat{X}_i = \gamma * \mathcal{N}_i * X_i + \beta + X_i$. In the original paper [16], GRN is used to increase the feature diversity across channels. Our main consideration for using GRN in CPM is to avoid feature redundancies in models with restricted capacity.

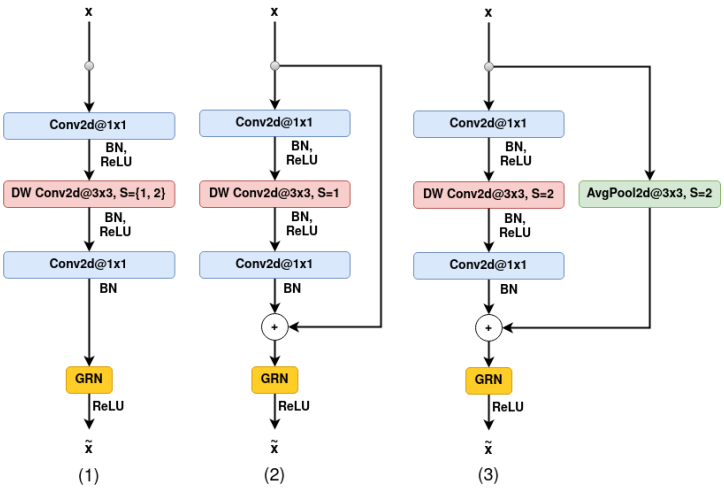

Figure 1: CPM blocks: (1) Transition Block (input channels ≠ output channels), (2) Standard Block, (3) Spatial Downsampling Block (S denotes stride)

5. KNOWLEDGE DISTILLATION FRAMEWORK

Similar to [4,11,12], CPM is trained on both the one-hot encoded labels and the pre-computed predictions of the ensemble *CP-Resnet, PaSST (All)* described in Table 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 depicted in Equation 2. λ is a weight that trades off label and distillation loss, z_S and z_T are student and teacher logits 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}.

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

5.1. Experimental Setup

We train our models for 75 epochs on the combined standard and shifted crops datasets. We use a batch size of 256, Adam optimizer [25] 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 KD [18], we use $\tau = 2$ and found that setting a high weight on the distillation loss with $\lambda = 0.02$ works best.

6. QUANTIZATION AWARE TRAINING

After completing the training routine outlined in 5.1, we fine-tune our models for 24 epochs using Quantization Aware Training (QAT) [26]. In this fine-tuning phase, we use a peak learning rate of

ID	BC	CM	EXP	LR	Pruned	Model Size (B)	MMACs	Acc.	Unseen Acc.	Quant. Acc.
S1	8	2.1	1.7	0.003	✗	5,722	1.58	54.66	51.83	52.61
S2	16	1.5	1.75	0.003	✗	12,310	4.35	59.50	56.76	58.42
S3	24	1.5	1.9	0.002	✗	30,106	9.64	62.34	58.44	61.77
S4	32	1.7	1.9	0.001	✓	54,182	16.80	65.16	61.26	64.08

Table 3: Model configurations submitted to the challenge. **BC**, **CM**, **EXP** are model scaling hyperparameters introduced in Section 4. The learning rate **LR** needs to be increased for smaller models to achieve high performance. **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. **Unseen Acc.** denotes the accuracy on recording devices unseen during training. All results are averages over 3 independent runs and the last 4 epochs of training.

ID	Training (kWh)	Inference (kWh)
S1	1.889	0.033
S2	1.96	0.035
S3	2.125	0.037
S4	3.56	0.036

Table 4: Consumed energy of our four submissions (**S1-S4**) during training and inference. The baseline reference value is 0.236 kWh.

$5e-5$ and linearly decrease it to 10% until epoch 16. All Conv2d + BN + ReLU combinations are fused into a single layer and we use PyTorch's [27] 'fbgemm' quantization config. In the forward pass, we perform all computations in int8 except for the computations performed in the GRN layer.

We note that CPM is more difficult to quantize than CP-ResNet, for which Post-Training Static Quantization is sufficient to approximately retain floating point performance. We also observed that losses in accuracy are more severe for smaller models.

7. ENERGY CONSUMPTION

To measure the energy consumption during training and inference, we use a system with a 16-core 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz CPU, 32 GB of RAM and an NVIDIA GeForce RTX 3090 GPU. For training, we sum up the consumed energy for the standard training phase, a possible pruning phase as described in Section 8 and the quantization-aware fine-tuning phase. We set the batch size to 1 for measuring the energy consumption during the inference of all files in the evaluation set. We only use the CPU since quantized model inference is not supported on GPU in PyTorch. We computed the reference value for the provided baseline on the same system, resulting in 0.236 kWh of consumed energy. We report the energy values for our submitted models (**S1-S4** as described in detail in Table 3) in Table 4.

8. SUBMISSIONS AND RESULTS

As the ranking of systems in the challenge is not only based on accuracy but also model size and MACs, we tried to find the models with the best performance-complexity trade-off. In Table 3, we present CPMs in four different configurations that are our final submissions (**S1-S4**) to the challenge. The models differ in terms of the scaling dimensions BC, CM, and EXP as introduced in Section 4, and the learning rates need to be increased for smaller models.

Firstly, we fixed the BC dimension by scaling models with different BC values along the CM and EXP dimensions. We found, for instance, that models with BC=8 have the best performance-complexity trade-off for models below 10k parameters. Similarly, experiments showed that BC=32 dominates the range of models above 50k parameters and in between the sweet spots of BC=16 and BC=24 are located. For all values of BC, we found that the performance gain of increasing CM and EXP quickly saturates. These diminishing returns allow us to use models of low complexity that still achieve high performance. Compared to a system that attains the complexity limits (65.66% accuracy), **S4** only sacrifices 0.5% accuracy (before quantization) while having around 50% of MACs and less than 50% of parameters. **S2** is comparable to the performance of the top-ranked system of the DCASE 22 challenge [4] while having about 10 times fewer parameters and more than 6 times fewer MACs.

For the final submission, we train the selected models (**S1-S4**) on all audio clips in the development set.

Pruning: We also experimented with pruning in **S4**. In particular, we apply structured pruning to a model that attains the complexity limits. We prune each CPM block based on the L2-norm of filters in the depth-wise convolution layer and remove the corresponding filters from the expansion and projection convolutions. Additionally, we use pruning to reduce the number of output channels of a block and remove the corresponding filters from the first convolution of the following block. After training, we prune all weights in a single step and re-train the resulting network using the same schedule. This procedure results, on average, in an improvement of around 1.2% accuracy compared to training the small model from scratch.

9. CONCLUSION

In this technical report, we describe the CP-JKU submission to Task 1 of the DCASE 23 challenge. We show that our system outperforms the top-ranked system of the DCASE 22 challenge by 5% in terms of accuracy and we can match its performance with only 10% of parameters and 6 times less MACs. The main improvement is attributable to an efficient, novel architecture, CP-Mobile, constructed of residual inverted bottleneck blocks and global response normalization. Additionally, our results substantially improve by the finding that ensembling Audio transformers and CNNs form strong teacher ensembles. We use an external device impulse response dataset to improve our system's robustness to unseen devices and we increase the FFT window size and add shifted crops to the original TAU23 dataset to further boost the performance.

10. ACKNOWLEDGMENT

The LIT AI Lab is financed by the Federal State of Upper Austria.

11. 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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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 *European Signal Processing Conference, EUSIPCO*. IEEE, 2019.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2022.

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

[12] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[13] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[14] 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 *Annual Conference of the International Speech Communication Association (Interspeech)*. ISCA, 2022.

[15] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *Submitted to EUSIPCO*, 2023. [Online]. Available: https://arxiv.org/abs/2305.07499

[16] 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*, vol. abs/2301.00808, 2023.

[17] 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.

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

[19] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2021.

[20] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE Computer Society, 2018.

[21] 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 *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019.

[22] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2019.

[23] ——, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021.

[24] S. Chang, H. Park, J. Cho, H. Park, S. Yun, and K. Hwang, "Subspectral normalization for neural audio data processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[26] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE Computer Society, 2018.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.