CREATING A GOOD TEACHER FOR KNOWLEDGE DISTILLATION IN ACOUSTIC SCENE CLASSIFICATION

Tobias Morocutti[2], *Florian Schmid*[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
{tobias.morocutti, florian.schmid, khaled.koutini}@jku.at

ABSTRACT

Knowledge Distillation (KD) is a widespread technique for compressing the knowledge of large models into more compact and efficient models. KD has proved to be highly effective in building well-performing low-complexity Acoustic Scene Classification (ASC) systems and was used in all the top-ranked submissions to this task of the annual DCASE challenge in the past three years. There is extensive research available on establishing the KD process, designing efficient student models, and forming well-performing teacher ensembles. However, less research has been conducted on investigating which teacher model attributes are beneficial for low-complexity students. In this work, we try to close this gap by studying the effects on the student's performance when using different teacher network architectures, varying the teacher model size, training them with different device generalization methods, and applying different ensembling strategies. The results show that teacher model sizes, device generalization methods, the ensembling strategy and the ensemble size are key factors for a well-performing student network.

Index Terms— Acoustic Scene Classification, Knowledge Distillation, CP-Mobile, Patchout FaSt Spectrogram Transformer (PaSST), CP-ResNet

1. INTRODUCTION

The objective of Acoustic Scene Classification (ASC) involves labeling an audio clip with a corresponding scene. The DCASE23 challenge's [1] Low-Complexity Acoustic Scene Classification task focuses on utilizing the *TAU Urban Acoustic Scenes 2022 Mobile development dataset (TAU22)* [2]. This dataset comprises one-second audio snippets from ten distinct acoustic scenes. In an attempt to make the models deployable on edge devices, a complexity limit on the models is enforced: models are constrained to have no more than 128,000 parameters and 30 million multiply-accumulate operations (MMACs) for the inference of a 1-second audio snippet. Among other model compression techniques such as Quantization [3] and Pruning [4], Knowledge Distillation (KD) [5–7] proved to be a particularly well-suited technique to improve the performance of a low-complexity model in ASC.

In a standard KD setting, a low-complexity model learns to mimic the teacher by minimizing a weighted sum of hard label loss and distillation loss. While the soft targets are usually obtained by one or multiple possibly complex teacher models, the distillation loss tries to match the student predictions with the computed soft targets based on the Kullback-Leibler divergence.

Jung et al. [8] demonstrate that soft targets in a teacher-student setup benefit the learning process since one-hot labels do not reflect the blurred decision boundaries between different acoustic scenes. Knowledge distillation has also been a very popular method in the DCASE challenge submissions. For example, Kim et al. [9] apply KD using a pretrained teacher. Further, [10] and [11] employ KD to train a low-complexity network on the predictions of a more complex one. Schmid et al. [12] use KD to train a low-complexity CNN on a teacher ensemble consisting of five PaSST [13] models.

To enhance generalization across recording devices, Kim et al. propose a modified version of MixStyle [14] called Freq-MixStyle [12, 15]. This method normalizes each frequency band and denormalizes it with mixed frequency statistics of two different samples.

Another method for improving the device generalization is Device Impulse Response Augmentation [16] which was introduced by Morocutti et al. It convolves audio signals with impulse responses of vintage microphones to increase the recording device variety in the training phase.

In this work, we study the effects of training a low-complexity network on the predictions of a single teacher or a teacher ensemble. We experiment with different network architectures, model sizes and device generalization methods to create the single teacher model that leads a student to perform best on the validation set. Additionally, we analyze the effect of combining teacher models with different network architectures, sizes, or device generalization methods.

2. NETWORK ARCHITECTURES

We experiment with three different teacher networks that were shown [17] to perform well as a teacher for the task of ASC. The architectures consist of two receptive-field regularized [18] convolutional neural networks (CNNs): CP-ResNet [19] and CP-Mobile [17], as well as a Transformer model: Patchout faSt Spectrogram Transformer (PaSST) [13].

2.1. CP-Mobile

CP-Mobile (CPM) [17] is an efficient architecture optimized for ASC. This architecture is designed to be less complex than CP-ResNet by factorizing convolution operations, such as in MobileNets [20, 21] and EfficientNets [22], while maintaining important properties that were shown to be important for ASC tasks, such as the regularized receptive field [18, 19].

In the following experiments, the student model has the CPM architecture with the following configuration: 32 base channels, an expansion rate of 3 and a channels multiplier of 2.3. The details of the CPM architecture are described in [17]. In short, these attributes control the scale of the network: the base channels represent the width of the first few blocks of the network; the channels multiplier determines the expansion in the number of channels as the network gets deeper (i.e. the number of channels in the last convolutional blocks is the number of channels of the previous blocks multiplied by channels multiplier); the expansion rate determines the number of channels in the depthwise convolution. The resulting model consists of almost 128K parameters and 29 million multiply-accumulate operations (MMACs).

We choose CPM as a student model since the architecture is designed for low-complexity ASC and has been shown to outperform CP-ResNet in previous work [17]. In addition, we experiment with using a scaled-up version of CPM as a teacher model for KD. To scale up the network, we increase the width via the base-channels hyperparameter.

2.2. CP-ResNet

CP-ResNet (CPR) [18, 19] is a receptive-field regularized CNN which has been shown to be very successful for ASC in previous editions of the DCASE ASC challenge [1, 2, 23, 24]. Therefore, we also use this network as a teacher model. We use the number of base channels to scale up the network in order to create teacher models with different sizes, similar to the procedure outlined for CPM.

2.3. PaSST

The Patchout faSt Spectrogram Transformer (PaSST) [13] is a complex, self-attention-based model, which is pre-trained on AudioSet [25] and consists of 85M parameters. The pre-trained model can be fine-tuned to achieve state-of-the-art performances on multiple downstream tasks, including ASC [13]. Additionally, PaSST models have proven to be excellent teachers for low-complexity CNNs [12, 26, 27]. Therefore, we also experiment with PaSST as a teacher model.

3. KNOWLEDGE DISTILLATION

We train our student model on the pre-computed predictions of the teacher model or teacher ensemble in addition to the one-hot encoded labels, similar to [27]. Training the student model on the soft labels of the teacher (ensemble) results in the student model learning blurred decision boundaries and establishing important similarity structures between classes. The loss is given in Equation 1 and consists of the hard label loss L_t and distillation loss L_{kd}. The label and distillation loss are weighted using the factor λ. The student and teacher logits are denoted by z_s and z_t, while y stands for the hard labels. τ is a temperature to control the sharpness of the probability distributions created by the softmax activation δ. L_l indicates the Cross-Entropy loss and the Kullback Leibler divergence is used as distillation loss L_{kd}.

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

As suggested in [5], we multiply the distillation loss by τ^2 since the magnitudes of the gradients produced by the soft targets scale as $1/\tau^2$. This ensures that the relative contributions of the hard and soft targets remain roughly unchanged if the temperature used for distillation is modified.

3.1. Experimental Setup

We train the teacher models as well as the student models on the TAU22 [2] dataset with the shifted crops dataset augmentation described in [17]. Regarding Knowledge Distillation, we use the values of 0.02 and 2 for λ and temperature τ, respectively.

For device generalization (DG) we experiment with Freq-MixStyle (FMS) [12, 15] and Device Impulse Response (DIR) augmentation [16] and the combination thereof. FMS is configured by two parameters: α_{fms} determines the shape of the Beta distribution used to randomly draw mixing coefficients, and p_{fms} specifies the probability of whether it is applied to a batch or not. Similar to FMS, DIR is guided by a probability p_{dir} that determines the augmentation strength by specifying the proportion of samples to augment.

The configurations used for FMS and DIR are adapted for each architecture. Results in [16] show that PaSST performs best with $\alpha_{fms} = 0.4$, $p_{fms} = 0.4$ and $p_{dir} = 0.6$ whereas CPR achieves the highest validation accuracy using $\alpha_{fms} = 0.3$, $p_{fms} = 0.8$ and $p_{dir} = 0.4$. While our experiments found that CPM teachers perform well using the same configuration as used for CPR, setting α_{fms}, p_{fms} and p_{dir} to 0.3, 0.4 and 0.6 when training the student network results in higher validation accuracy. More details about our experimental setup are reported in [17].

4. SINGLE MODEL TEACHER

In this section, we compare the performance of different teachers and evaluate the performance of students trained on the predictions of different teacher models using KD. We experiment with using a single CPM, CPR or PaSST model as the teacher and a low-complexity CPM as the student.

4.1. Scaling the Teacher

To investigate the effect of training the student on teachers of different complexity, we scale CPM and CPR by increasing the number of base channels, which modifies the width of the network. We test the effect of scaling the teacher only on CPM and CPR since we use a pre-trained PaSST model.

We trained CPM and CPR models in five different complexity configurations such that their number of parameters is approximately 128K, 450K, 1M, 4M and 8M. Since the number of parameters of CPM and CPR does not scale equally when increasing the base channels, we selected the number of base channels for each size and architecture individually. We used 32, 56, 88, 168 and 232 base channels for CPR and 32, 64, 96, 184 and 264 base channels for CPM.

All different configurations are evaluated over three runs and to ensure that our experiments are independent of each other, we train one student on each of the three teachers.

Additionally, we apply a combination of Freq-MixStyle and Device Impulse Response augmentation to all student as well as all teacher models. From now on, we will refer to the combination of DIR and FMS as DIRFMS.

Table 1 shows that for the teacher, CPM outperforms CPR in each complexity configuration. Additionally, even the smallest variant of CPM achieves a higher validation accuracy than PaSST, which has several orders of magnitude more parameters.

However, the students trained on CPM perform worse than the ones trained on CPR for each size of teacher. Furthermore, the students trained using PaSST as a teacher outperform the best students trained on a CPR variant by only 0.05%. While the teacher with 450K parameters works best for CPR, the variant with 128K parameters makes the best CPM teacher.

In short, the results show that the right scale of a CNN teacher can improve the performance of the students by more than 1%. Furthermore, smaller CNNs can be better teachers, even if the larger

		CPR		CPM		PaSST	
		T	S	T	S	T	S
Teacher size	**128K**	60.28	63.94	62.66	**63.70**	-	-
	450K	62.05	**64.60**	62.81	62.48	-	-
	1M	62.58	63.99	63.92	62.76	-	-
	4M	62.74	63.51	64.28	62.43	-	-
	8M	**63.28**	63.43	**64.62**	62.83	-	-
	85M	-	-	-	-	**62.20**	**64.65**

Table 1: Validation accuracy of different teacher networks, and a student model trained on these. T and S denote the performance of the teacher and student, respectively. While the teacher networks vary in architecture and size, the student model is always a CPM model with 128k parameters. All results are averages over three independent runs and the last 4 epochs of training.

teachers outperform the smaller ones. Finally, having a different architecture for teacher and student improves the performance of the student.

4.2. Effect of Device Generalization Methods

Table 2 presents the impact of the device generalization (DG) methods DIR, FMS and DIRFMS. For studying the effects of these methods, we use the teacher variations with 128K and 450K parameters for CPM and CPR, respectively, since these teacher models result in the best performing student models, as shown in Section 4.1.

	CPR		CPM		PaSST	
	T	S	T	S	T	S
Validation Accuracy						
DIRFMS	**62.05**	**64.60**	**62.66**	**63.70**	**62.20**	**64.65**
DIR	57.34	62.47	57.23	61.57	61.64	64.39
FMS	60.99	63.40	61.18	63.66	61.08	64.56
NO AUG	54.13	62.74	53.15	62.47	59.39	63.76
Unseen Accuracy						
DIRFMS	**56.95**	**60.43**	**57.92**	**59.20**	**58.73**	**61.03**
DIR	49.30	56.74	48.62	55.54	57.91	60.90
FMS	54.94	58.91	54.92	58.76	57.57	61.00
NO AUG	44.75	56.70	43.94	56.21	54.08	59.60

Table 2: Validation accuracy of teacher networks trained using different DG methods, and a student model trained on the corresponding teacher predictions. T and S denote the performance of the teacher and student, respectively. The CPM teacher has 128K parameters, the CPR teacher has 450K parameters. While the teacher network varies in architecture and used DG method, the student is always a CPM model with 128k parameters trained with DIRFMS. All results are averages over three independent runs and the last 4 epochs of training.

The results in Table 2 show that FMS, DIR and/or DIRFMS boost both the performance of the teacher models as well as the performance of the student models significantly. The results show that there is a clear effect of these methods on the validation accuracy. Moreover, this effect tends to be even higher on the unseen accuracy. Applying DIRFMS results in the best validation and unseen accuracy, outperforming DIR and FMS. We define *unseen accuracy* as the accuracy on the subset of the validation set that consists of samples of devices not present in the training set. Consistent with the findings in [16], FMS, DIR and DIRFMS have less effect on the performance of PaSST, compared to CPR or CPM.

5. ENSEMBLE TEACHER

Previous work [17] shows that training the student on the predictions of multiple teacher networks is a highly effective method to improve the accuracy of the student in the KD framework. This effect is even more significant when ensembling different architectures or models trained with different device generalization methods. In this section, we will experiment with different ensemble configurations and show their effect on the low-complexity student. We ensemble different models by averaging their logits.

	CPR		CPM		PaSST
size of teacher	128K	450K	128K	450K	85M
1 teacher	63.94	**64.60**	**63.70**	62.48	**64.65**
3 teacher	**64.53**	64.36	**63.97**	62.77	**64.81**

Table 3: Validation accuracy of student models trained on the predictions of either one or three teacher models which apply both Freq-MixStyle and Device Impulse Response augmentation (**DIRFMS**). The highest accuracy per architecture and per number of teacher is marked bold. For CPR and CPM, the teacher models consist of either 128K or 450K parameters. All results are averages over three independent runs and the last 4 epochs of training.

5.1. Ensembling Teachers with Identical Training Setup

This section presents experiments about ensembling different models that use the same training setup but different seeds. More precisely, we ensemble different models that share the same architecture, complexity and DG methods. The goal is to test if the averaged logits of multiple teacher models are better soft targets for training the student model.

Since the results in Table 2 indicate that DIRFMS has the most positive effect on the students for all teacher architectures, we evaluate the performance of students learning from a teacher ensemble trained with DIRFMS. Additionally, we choose to test the training of the student on the teacher ensembles with two different complexity configurations of the CPR and CPM teachers. Due to the fact that CPR performs best with 450K and CPM with 128K parameters, we select these two complexity levels to evaluate the teacher ensembling on both architectures.

As the results in Table 3 show, the CPR teacher with 128K parameters outperforms the variant with 450K parameters when using an ensemble of three teachers. Further, the variant with 128K parameters also works best for the CPM teacher, outperforming the 450K-parameters variant by 1.2%. When we train the students on the averaged logits of three PaSST models, the validation accuracy of the student increases slightly by 0.16%, compared to using only one PaSST teacher. However, PaSST outperforms the other architectures, with CPM performing worse than CPR.

Teacher Ensemble Variations							
Teacher Architecture							
CPR	✓	✗	✗	✓	✓	✗	✓
CPM	✗	✓	✗	✗	✓	✓	✓
PaSST	✗	✗	✓	✓	✓	✓	✗
Device Generalization Methods							
DIR + FMS	64.25	62.35	64.47	-	-	-	-
DIRFMS + DIR	64.21	63.45	64.63	-	-	-	-
DIRFMS	64.53	**63.97**	64.81	65.19	65.09	**65.15**	64.66
DIRFMS + FMS	**64.74**	63.76	64.89	**65.81**	65.12	64.67	**64.67**
DIRFMS + DIR + FMS	64.10	63.76	**65.16**	65.39	**65.18**	64.85	64.03

Table 4: The accuracy of the student model being trained on a teacher ensemble. The teacher ensembles differ in the combination of architectures and the combination of DG methods. A mark indicates that three models of the corresponding architecture are included in the ensemble. All results are averages over three independent runs and the last 4 epochs of training.

5.2. Ensembling Teachers Trained with Different DG Methods

In this section, we experiment with combining models with the same architecture but trained using different DG methods in order to create a better teacher ensemble. We choose 128K parameters as the teacher complexity for CPR and CPM, since this complexity performs best when combining multiple models, as shown in Table 3. We evaluate the effect of training the student on these teacher ensembles and compare the results with the performance of the students trained using the DIRFMS teacher ensemble described in Section 5.1. All evaluated teacher ensembles contain three models for each included DG method. This implies that the different ensembles stated in the left part of Table 4 contain between 3 and 9 models.

The results in Table 4 indicate that including teachers trained using DIRFMS in the ensemble is essential for every architecture, since the ensembles DIRFMS+FMS, DIRFMS and DIRFMS+DIR+FMS perform best for the CPR, CPM and PaSST architecture, respectively. Including the DIR teacher in the DIRFMS+FMS ensemble only increases the performance of students trained on the predictions of PaSST models. The best-evaluated ensemble of only one architecture is the PaSST DIRFMS+DIR+FMS ensemble, increasing the accuracy by 0.35% compared to the previously best PaSST DIRFMS ensemble.

5.3. Ensembling Teachers with Different Architectures

In this section, we experiment with ensembling different architectures motivated by the assumption that different architectures can learn different features and aspects of the training data and therefore ensembling them would result in a more robust model.

We test each combination of CPR, CPM and PaSST using the combinations of DG methods, which performed best on single architecture ensembles. It is worth noting that the teacher ensemble size depends on the number of used architectures and DG methods. It can therefore range from 6 (2 architectures x 1 DG method x 3 models) to 27 (3 architectures x 3 DG methods x 3 models).

The results in Table 4 clearly show that the teacher ensembles consisting of CPR and PaSST models result in the best-performing students. Adding CPM models to ensembles of CPR and PaSST models worsens the performance of the students for all evaluated DG configurations. More precisely, ensembling CPM and CPR does not lead to performance improvement, and neither does ensembling CPM and PaSST.

Regarding the DG methods, ensembling teacher models trained with DIRFMS and FMS results in the best student performance for the CPR and PaSST combination, creating the best-evaluated ensemble with 65.81% validation accuracy of the student.

6. CONCLUSION

In this work, we show that low-complexity CNNs like the CPM learn more important features from Transformers or relatively small CNNs compared to large CNNs when using Knowledge Distillation. Additionally, we show that applying Device Impulse Response (DIR) augmentation, Freq-Mixstyle (FMS) and especially the combination thereof (DIRFMS) to the teacher models significantly boosts the performance of the teachers and the students. The effect of these DG methods is even more noticeable on the unseen accuracy, compared to the total validation accuracy. Surprisingly, it turns out that the performance of the student does not necessarily improve with the scale of the teacher. For example, ensembling smaller teacher networks can be more beneficial than ensembling bigger ones. Furthermore, we show that the performance of the student improves when the teacher architecture is different than the student architecture. For example, when using PaSST or CPR to train CPM. In contrast, the low-complexity CPM student performs worse when it is trained on any higher complexity variation of the same architecture. Additionally, the predictions of PaSST and CPR complement each other, resulting in better student performance. Finally, using an ensemble of CPR and PaSST trained either using DIRFMS or FMS results in our best student, which has an accuracy of 65.81% with 128K parameters and 32 million MACCS, outperforming the much larger CPR, CPM and PaSST models.

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. 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.

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

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

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

[7] A. M. Tripathi and O. J. Pandey, "Divide and distill: New outlooks on knowledge distillation for environmental sound classification," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2023.

[8] H. Heo, J. Jung, H. Shim, and H. Yu, "Acoustic scene classification using teacher-student learning with soft-labels," in *Interspeech*. ISCA, 2019.

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

[10] 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.

[11] R. Anastácio, L. Ferreira, F. Mónica, and C. B. Luís, "Ai4edgept submission to DCASE 2022 low complexity acoustic scene classification task1," DCASE2022 Challenge, Tech. Rep., 2022.

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

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

[14] J. Fu, Y. Zhong, and F. Yang, "Adversarial domain generalization with mixstyle," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2022.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*. IEEE, 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 *ICCV*. IEEE, 2019.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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