LOW-COMPLEXITY MODELS FOR ACOUSTIC SCENE CLASSIFICATION BASED ON RECEPTIVE FIELD REGULARIZATION AND FREQUENCY DAMPING

Khaled Koutini[1], *Florian Henkel*[1], *Hamid Eghbal-zadeh*[1,2], *Gerhard Widmer*[1,2]

[1]Institute of Computational Perception (CP-JKU) & [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
khaled.koutini@jku.at

ABSTRACT

Deep Neural Networks are known to be very demanding in terms of computing and memory requirements. Due to the ever increasing use of embedded systems and mobile devices with a limited resource budget, designing low-complexity models without sacrificing too much of their predictive performance gained great importance. In this work, we investigate and compare several well-known methods to reduce the number of parameters in neural networks. We further put these into the context of a recent study on the effect of the Receptive Field (RF) on a model's performance, and empirically show that we can achieve high-performing low-complexity models by applying specific restrictions on the RFs, in combination with parameter reduction methods. Additionally, we propose a filter-damping technique for regularizing the RF of models, without altering their architecture and changing their parameter counts. We will show that incorporating this technique improves the performance in various low-complexity settings such as pruning and decomposed convolution. Using our proposed filter damping, we achieved the 1st rank at the DCASE-2020 Challenge in the task of Low-Complexity Acoustic Scene Classification.[1]

Index Terms— low-complexity, acoustic scene classification, receptive-field regularization, pruning, network decomposition

1. INTRODUCTION

The recent advances in machine learning have been mainly due to the unprecedented successes of deep neural networks with millions or even billions of trainable parameters that can learn from a large amount of data and solve complex problems. Although in deep learning, the main attention has been on reaching the highest performance, as these models continue to expand their applications from research into industry, the memory efficiency, energy consumption, and latency of these models become more and more important.

To address these problems, a new line of research has been established to design low-complexity neural networks that are capable of reaching the performance of the large models, while having many orders of magnitude fewer parameters. In this area, three main approaches have been followed in the literature. A first approach is *knowledge distillation* [1], which trains a smaller network known as the student, by using the output of a bigger network (the teacher) as the training signal. Second are methods that focus on proposing efficient neural *architectures* that by design have fewer parameters [2, 3]. For example, strategies such as depth-wise separable convolutions [4] or careful tuning of the width and depth of the networks [3] aim at producing efficient networks with lower complexity. Another approach to designing low-complexity architectures is to use *decomposition layers*; given the fact that neural network parameters are mostly represented with high-dimensional weight tensors, several approaches have been developed to decompose these into smaller tensors to reduce the computational load, without sacrificing the model's performance. In [5–7], the authors propose different methods to decompose a single convolutional layer into multiple smaller ones, resulting in more compact models with computational speedup. Similarly, [8–10] investigate the use of 1 × 1 convolutions to reduce the number of channels before applying more expensive operations with larger filter sizes. A third class of approaches aims at starting with a large high-performance model, to then remove a large part of the parameters by network pruning [11–14]. *Parameter Pruning* is the process of compressing a neural network by zeroing out some of its parameters. Some methods focus on removing the weights of pre-trained models [12, 15, 16], while more recent approaches incorporate iterative training, by pruning and resetting only the non-pruned weights [13], or pruning without the use of any training data [17]. We will focus our analysis on the width and depth restriction methods (Section 3.1), on decomposed CNNs (Section 3.2), and on parameter pruning (Section 3.3), which have gained more popularity and interest among the scientific community, and leave aside the Knowledge Distillation approaches due to their high complexity and slow nature.

Recently CNNs have been successfully used for end-to-end Acoustic Scene Classification (ASC) [18–21], outperforming previous approaches and setting new state-of-the-art. Recent studies on ASC with CNNs have revealed that regularizing the RF of CNNs can significantly improve their generalisation [20–24]. Further, authors in [18] provide a systematic way for controlling the receptive-field of CNNs by adapting the architectural design of the networks. Although such regularizations improve generalisation, they affect the architectural design and the number of parameters used in a model. Hence, finding methods that can achieve high performance by following the RF-regularization principles, while having minimal complexity becomes a challenging and important task.

In this paper, we aim at connecting the concept of RF-regularization, with the low-complexity CNNs, and investigate the relationship between the RF size, architecture complexity, and generalisation performance of CNNs for the task of ASC. To this end, we analyse the performance of different low-complexity ASC methods, under various RFs, and show how RF affects the performance in low-complexity settings. We empirically evaluate different approaches to low-complexity ASC, and analyse their performance under various maximum receptive fields to reveal the connection be-

[1] Code available at: `https://github.com/kkoutini/cpjku_dcase20`

tween generalisation in a low-complexity setup, and the RF of the models. Further, we propose a novel RF-regularization technique called "Damping" which regularizes the RF of any CNN, without a need to alter the topological design of the network. We show that our Damping RF regularization achieves the best performance with both pruning and decomposition architectures, and hence is a suitable approach for improving generalisation of models in low-complexity settings.

2. ARCHITECTURES

Previous work has shown the success of RF-regularized CNNs in various acoustic tasks [22, 23]. Therefore, we base our work on the RF-regularized ResNet architecture introduced in [18]. Furthermore, we introduce a new technique for further restricting the effective receptive field of the network and provide empirical evidence on its success in ASC.

2.1. Baseline ResNet Architecture: *CP-ResNet*

The details of the RF-regularized ResNet architecture are explained in [20], where the authors introduce the hyperparameter ρ in the architecture design, such that the RF of the architecture can be changed by varying ρ. Since previous work [18, 22, 24] showed the optimal range of the RF for different ASC datasets to be approximately between 75 and 150 (for the input spectograms explained in Section 4), we restrict our experiments to ρ values in range 3-12. Furthermore, we remove the tailing 1×1 convolutional layers by removing the last 5 residual blocks,[2] in order to reduce the number of parameters in the baseline. As shown in architecture RN1 in [18], these tailing layers have a minor effect on the performance of the model. We refer to this RF-regularized ResNet architecture as *CP-ResNet* throughout this paper.

2.2. Frequency Damping: *Damped CP-ResNet*

Previous work [18] has shown that restricting the RF of deep CNNs, especially over the frequency dimension, results in better generalization on different ASC datasets. While [18] introduces a method for systematically tuning the RF of CNNs, the proposed approach requires changes to the architecture, which as a result changes the number of parameters. To address these drawbacks, we propose a novel method to restrict the *Effective Receptive Field (ERF)* of CNNs – the part of the RF that has the most influence on the output activation, as detailed in [18, 25] – by *damping* the convolutional filter weights over the frequency dimensions. Each convolutional neuron has a limited receptive field of its layer input. Damping works in such a way that the further the input is from the center of a neuron's RF, the less influence it will have on that neuron.

In practice, we damp the filters of a convolutional layer by applying an element-wise multiplication between the convolution filter weights and a non-trainable constant matrix $C \in \mathbb{R}^{T \times F}$ (damping matrix). The damping matrix matches the spatial shape of the filters. It decays linearly away from the center, so that the outermost elements of the filter over the frequency dimension have a smaller influence on the activation.

The resulting network is called *damped CNN*, where every convolution operation $O_n = W_n * Z_{n-1} + B_n$ is replaced by $O_n = (W_n \odot C_n) * Z_{n-1} + B_n$, $*$ is the convolution operator, $\odot$ is the element-wise multiplication operator, Z_{n-1} is the output of the previous layer, W_n is the filter trainable weight, and B_n is the bias. The matrix has a value of 1 in the center, and decays linearly to reach a value λ; we used $\lambda = 0.1$ in our experiments. This approach has shown empirical improvement over *CP-ResNet* in different ASC datasets. We refer to this architecture as *Damp* throughout this paper.

3. MODEL COMPLEXITY REDUCTION APPROACHES

In this section, we investigate 3 different approaches to reduce the number of parameters of CNNs. We follow the principle of reducing the number of parameters, while keeping the final receptive field of the network constant, which allows the comparison with the baseline models in each receptive field setting.

3.1. Width and Depth restriction

Reducing the width (number of channels) and depth (number of layers) of CNNs is a simple technique to reduce the network size. Tan and Le [3] showed that after an optimal width of the network is reached, increasing the width further will result in only a minimal performance gain at best. The width of the network has quadratic influence on the number of weights, while the depth has a lower effect as the number of parameters grows linearly with the number of layers. Based on this fact, and the goal of our experiments that is comparing different RF setups, we focus our efforts on different network widths.

As explained in Section 2.1, we remove the tailing residual blocks with 1×1 convolutions since we are only studying networks with ρ values in range $3 - 12$. Therefore, we decrease the depth and the number of parameters from the baseline network in [18, 20], with only a minimal performance impact. However, removing these tailing layers with 1×1 convolutions result in a significant reduction of the number of parameters in the baseline (reported in [24] with $\rho = 7$) from $3956K$ to $1715K$.

We change the width of our CNNs (Section 2) by changing the number of channels in initial layers from 128 (in the baseline architecture) to 64 and 32. This results in reducing the baseline parameter count from $1715K$ to $431K$ and $109K$ respectively.

3.2. Decomposed Convolutions

Inspired by the use of singular-value-decomposition (SVD) for convolutional neural networks [7], we propose to directly train decomposed convolutional layers instead of decomposing and then fine-tuning a pretrained model, as done in [7] . Given a regular convolutional layer with dimensionality

$$C_{in} \times C_{out} \times k \times k, \tag{1}$$

with C_{in} and C_{out} being the number of input and output filters respectively, and k being the kernel size. Such a layer can be decomposed into three convolutional layers using a compression factor Z:

$$\begin{gathered} C_{in} \times (C_{out}/Z) \times 1 \times 1 \\ (C_{out}/Z) \times (C_{out}/Z) \times k \times k \\ (C_{out}/Z) \times C_{out} \times 1 \times 1 \end{gathered} \tag{2}$$

For example, a $128 \times 128 \times 3 \times 3$ convolution has 147456 parameters (neglecting the bias). Using a compression factor $Z = 4$, we get three convolutions $128 \times 32 \times 1 \times 1$, $32 \times 32 \times 3 \times 3$ and $32 \times 128 \times 1 \times 1$, resulting in 17408 parameters. Similar approaches to

[2] The residual blocks from 8 to 12 as explained in Table 1 of [20]

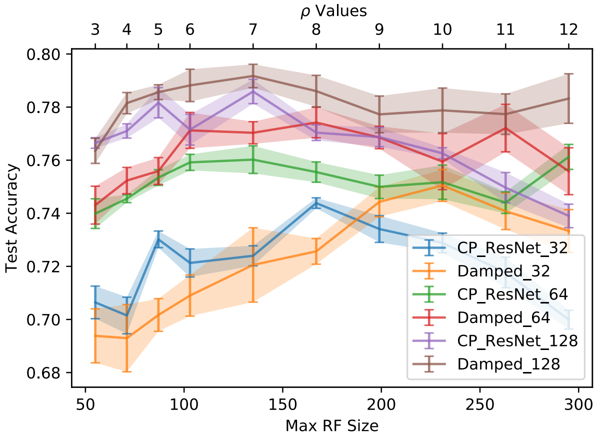

Figure 1: Testing accuracy of CP-ResNet on DCASE'18 with/without filter damping on the frequency dimension using different CNN width setups as explained in Section 3.1.

decompose convolutions for the purpose of parameter reduction are explored in [8–10]. Our proposed decomposition shares the same structure as the "bottleneck" building block described in [10], but is different from them as we do not use non-linearities and batch-normalization within the decomposed block, which proved to yield better performance.

3.3. Parameter Pruning

Pruning is well studied in neural network literature and several methods and adaptation have been proposed [12–17]. We use magnitude pruning [12, 14] with iteratively increasing the pruning ratio until reaching the desired number of parameters.

Authors in [17] show that pruning approaches perform very similarly, if the compression-ratio is up to 1%. Since in our low-complexity setting, we are not targeting compression-rates lower than 1%, we choose to use the more simple magnitude-pruning approach. We ramp up the number of pruned parameters with an exponential decay from 0 to the final desired number in 100 epochs, to get models with 250K, 300k, 400k, and 500k parameters. This allows us to remove more weights at the beginning of the training, and fewer weights in later stages as the model converges.

4. EXPERIMENTAL SETUP

We evaluate our proposed approaches introduced in Section 3 on two acoustic scene classification (ASC) datasets.

DCASE'20 Low Complexity ASC [26]: The dataset contains 40 hours of recordings from 12 European cities in 10 different acoustic scenes that are summarized into three categories, *indoor*, *outdoor* and *transportation*. We follow the training/test split provided by the task organizers where recordings from one city are not seen during training.

DCASE'18 ASC [27]: The dataset comprises recordings from six European cities with similar acoustic scenes as the DCASE'20 dataset. We choose this dataset because it has 10 classes which makes it a more challenging task compared to DCASE'20, therefore we can better differentiate the performance between different approaches with the 500KB limit (and more) that was used in the DCASE'20 challenge. This dataset has a relatively smaller size resulting in shorter training time. Therefore, we use DCASE'18 for our detailed analysis.

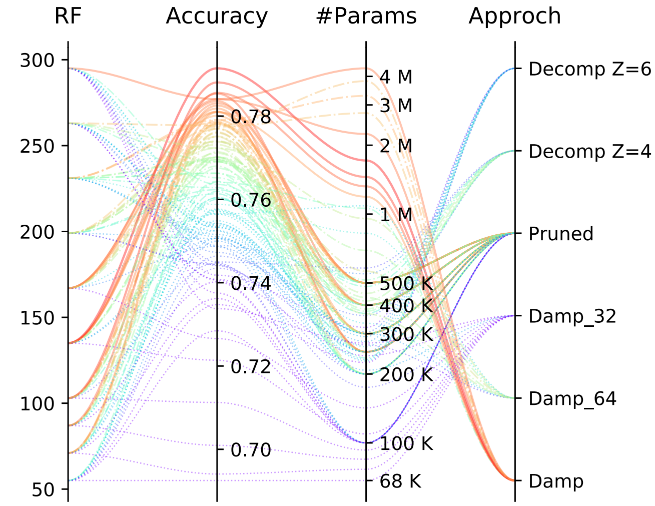

Figure 2: Comparison of different complexity reduction methods in different receptive field setups on DCASE'18. All methods apply frequency-damping. The line color represents the accuracy.

For feature extraction, we compute perceptually weighted Mel-spectrograms from the provided 10 second audio snippets which we down-sample to 22.05kHz similar to [18]. Each input channel of the stereo audio is processed independently, normalized using the training set statistics and provided as a two-channel-spectrogram input to the CNN. We use the same experiments and training setup as explained in [20].

5. RESULTS

Table 1 summarizes the results of our best performing models on the DCASE'20 and DCASE'18 datasets. On both datasets we observe an improvement for the damped ResNet architecture, compared to the baseline. Furthermore, we observe that a simple width reduction, i. e., reducing the number of base channels and thus decreasing the overall number of parameters, is not as effective as pruning the weights of a bigger network. Following the pruning strategy explained in Section 3.3, we can achieve models with a significantly lower number of non-zero parameters, with the same performance on DCASE'20 or only minor performance degradation on DCASE'18.

5.1. Frequency Damping Comparison

To further investigate the effect of *damping*, we compare the performance of several baseline CP-ResNets with and without *damping*, using different maximum RF setups on DCASE'18. Fig. 1 shows that *damping* not only improves the overall performance, but is also more robust to the choice of the maximum RF. We also observe that the damped ResNet with 128 base channels (Damped) yields the best overall performance using $\rho = 7$ to regularize the RF. The model with 64 base channels (Damped_64) decreases the performance, while significantly reducing the number of parameters

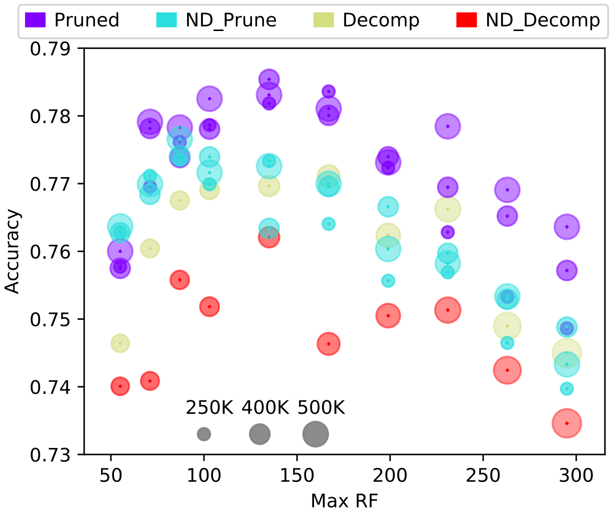

Figure 3: Comparing different complexity reduction methods in different receptive field setups with and without damping on DCASE'18. All methods use 128 base channels. Decomp uses a compression factor $Z = 4$. ND refers to no damping method being applied to the original Network. The size of the dots relates to the number of model parameters.

	DCASE'20		DCASE'18	
Model	**Non-Zero**	**Acc.**	**Non-Zero**	**Acc.**
CP-ResNet	1061.6K	.9726	1715.6K	.7852
Damp.	1061.6K	.9745	1715.6K	.7915
Damp.-Restricted	268.6K	.9722	431.8K	.7701
Damp.-Pruned	250K	.9745	400K	.7854
Damp.-Decomp.	361K	.9700	417K	.7696

Table 1: Comparison of the performance of the best models of the discussed approaches on the DCASE'20 and DCASE'18 datasets. We report the accuracy (Acc.) and the number of non-zero parameters. *Damp.* refers to a damped ResNet as introduced in Section 2.2, *Damp.-Restriced* is a width restricted ResNet with 64 base channels (cf. Section 3.1), *Damp.-Pruned* is a pruned ResNet (cf. Section 3.3) and *Damp.-Decomp.* is a ResNet where all convolutions are replaced with decomposed convolution and a compression factor $Z = 4$ (cf. Section 3.2). The table shows the mean of the accuracy over the last 10 epochs for each model. All DCASE'20 and DCASE'18 models use $\rho = 4$ and $\rho = 7$, respectively.

(shown in Figure 2), and 32 base channels seems to result in an insufficient amount of parameters for this task. As can be seen in Fig. 3, Damped CP-ResNets outperforms non-damped CP-ResNets with the same number of parameters, both in pruning and decomposition cases.

5.2. Parameter Pruning vs. Network Decomposition

Comparing all parameter reduction methods in Figs. 2 and 4, we see that pruning is the closest model to the frequency-damped baseline (Damp) in terms of performance, and achieves the highest accuracy with limited parameters. Fig. 4 further shows that the pruned model achieves an accuracy close to the baseline, while significantly reducing the number of non-zero parameters. Using decomposed convolutions, the performance degradation is significantly higher than for pruning. Similarly, simply reducing the width of the network results in a large loss in accuracy.

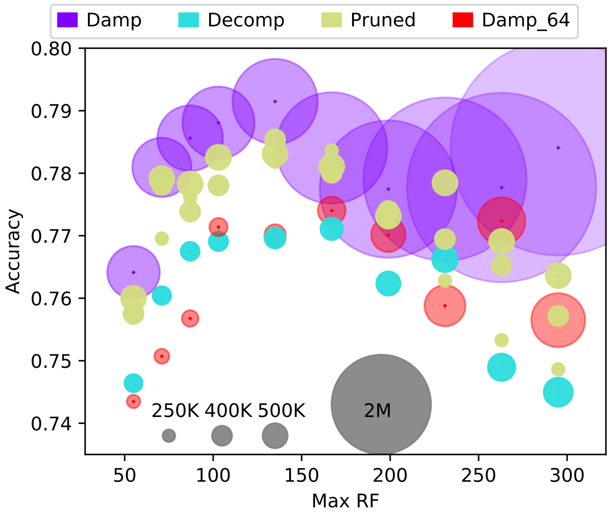

Figure 4: Comparing different complexity reduction approaches on the DCASE'18 dataset. All networks are damped. The size of the dots relates to the number of model parameters.

6. CONCLUSION

In this paper, we analysed various low-complexity CNN-based approaches for ASC, and studied the relationship between the RF and the performance in each approach. We showed that pruning achieves better performance compared to decomposition and width reduction methods. We proposed a filter damping technique that can be used in low-complexity settings to regularize the RF of models, without altering the architectures. We showed that using filter damping improves the performance on all architectures and datasets we evaluated, hence is a simple and effective technique for improving generalisation of models. Our results on two datasets for ASC suggests that the proposed filter damping can achieve state-of-the-art performance in low-complexity ASC. Using this approach, we achieved the 1st rank at the DCASE-2020 Challenge *Low-Complexity Acoustic Scene Classification* (Task1.b).

7. ACKNOWLEDGMENT

This work has been supported by (1) the COMET-K2 Center of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria, and (2) the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement number 670035, project ”Con Espressione”).

8. REFERENCES

[1] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[3] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Proc. of the 36th Int. Conf. on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017.

[5] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, “Speeding-up Convolutional Neural Networks using fine-tuned CP-Decomposition,” in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2015.

[6] M. Jaderberg, A. Vedaldi, and A. Zisserman, “Speeding up convolutional neural networks with low rank expansions,” in *British Machine Vision Conf. (BMVC)*, 2014.

[7] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, “Compression of deep convolutional neural networks for fast and low power mobile applications,” *arXiv preprint arXiv:1511.06530*, 2016.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[11] Y. LeCun, J. S. Denker, and S. A. Solla, “Optimal brain damage,” in *Advances in Neural Information Processing Systems 2*, 1990, pp. 598–605.

[12] S. Janowsky, “Pruning versus clipping in neural networks.” *Physical review. A, Atomic, molecular, and optical physics*, vol. 39, no. 12, pp. 6600–6603, 1989.

[13] J. Frankle and M. Carbin, “The lottery ticket hypothesis: Finding sparse, trainable neural networks,” in *7th Int. Conf. on Learning Representations (ICLR)*, 2019.

[14] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1135–1143.

[15] Y. LeCun, J. S. Denker, and S. A. Solla, “Optimal brain damage,” in *Advances in Neural Information Processing Systems 2*, 1989, pp. 598–605.

[16] M. Mozer and P. Smolensky, “Skeletonization: A technique for trimming the fat from a network via relevance assessment,” in *Advances in Neural Information Processing Systems 1*, 1988, pp. 107–115.

[17] H. Tanaka, D. Kunin, D. L. K. Yamins, and S. Ganguli, “Pruning neural networks without any data by iteratively conserving synaptic flow,” *CoRR*, vol. abs/2006.05467, 2020.

[18] 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 *Proc. of the European Signal Processing Conf. (EUSIPCO)*, 2019.

[19] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, “Integrating the data augmentation scheme with various classifiers for acoustic scene modeling,” DCASE2019 Challenge, Tech. Rep., 2019.

[20] K. Koutini, H. Eghbal-zadeh, and G. Widmer, “Receptive-field-regularized CNN variants for acoustic scene classification,” in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.

[21] S. Suh, S. Park, Y. Jeong, and T. Lee, “Designing Acoustic Scene Classification Models with CNN Variants,” DCASE2020 Challenge, Tech. Rep., 2020.

[22] K. Koutini, H. Eghbal-zadeh, and G. Widmer, “CP-JKU submissions to DCASE’19: Acoustic scene classification and audio tagging with receptive-field-regularized CNNs,” DCASE2019 Challenge, Tech. Rep., 2019.

[23] K. Koutini, S. Chowdhury, V. Haunschmid, H. Eghbal-Zadeh, and G. Widmer, “Emotion and theme recognition in music with Frequency-Aware RF-Regularized CNNs,” in *MediaEval Benchmark Workshop*, 2019.

[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] W. Luo, Y. Li, R. Urtasun, and R. Zemel, “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 29*, 2016, pp. 4898–4906.

[26] T. Heittola, A. Mesaros, and T. Virtanen, “Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions,” in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020.

[27] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proc. of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, 2018, pp. 9–13.