CP-JKU SUBMISSIONS TO DCASE'20: LOW-COMPLEXITY CROSS-DEVICE ACOUSTIC SCENE CLASSIFICATION WITH RF-REGULARIZED CNNS

Technical Report

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

This technical report describes the CP-JKU team's submission for Task 1 – Subtask A (Acoustic Scene Classification with Multiple Devices) and Subtask B (Low-Complexity Acoustic Scene Classification) of the DCASE-2020 challenge [1]. For Subtask 1.A, we provide our *Receptive Field (RF) regularized CNN* model as a baseline, and additionally explore the use of two different domain adaptation objectives in the form of the *Maximum Mean Discrepancy (MMD)* and the *Sliced Wasserstein Distance (SWD)*. For Subtask 1.B, we investigate different parameter reduction methods such as *Pruning*, while maintaining the receptive field of the networks. Additionally, we incorporate a decomposed convolutional layer that reduces the number of non-zero parameters in our models while only slightly decreasing the accuracy, compared to the full-parameter baseline.[1]

Index Terms— acoustic scene classification, receptive-field regularization, domain adaptation, pruning, network decomposition

1. INTRODUCTION

Receptive-Field regularized Convolutional Neural Networks (RFR-CNNs) have proven to be very effective in different acoustic tasks such as Acoustic Scene Classification (ASC) [2, 3, 4], device-invariant ASC [5, 6, 4], open-set ASC [5, 7], Audio-tagging with noisy labels and minimal supervision [8, 4], and emotion and theme recognition in music [9]. Because of these successes, for our submissions to this year's DCASE challenge, we incorporate RFR-CNN architectures. In addition, we introduce a new method that allows us to not only limit the RF of the models, but also control the ***Effective*** *Receptive Field* [10, 2] in deep CNNs. We adapt the networks for the challenge tasks [1] using Domain Adaptation for Task 1A, specifically, Sliced Wasserstein Distance (SWD) [11], and Maximum Mean Discrepancy (MMD) [12]. For Task 1B, where a limited model size is required, we use weight pruning [13, 14], layer decomposition [15] and width/depth reduction of the basis network.

2. EXPERIMENTAL SETUP

We refer the reader to the CP-JKU team's DCASE 2019 technical report [4] as we use an identical setup. No external data was used.

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

3. ARCHITECTURES

3.1. Baseline ResNet Architecture: *CP-Res*

We base our experiments on the ResNet architecture explained in [3]. Our initial experiments showed that the optimal RF size for task 1A corresponds to $\rho = 6$ and $\rho = 7$. For task 1B, preliminary experiments indicated that best performance is achieved with $\rho = 3$ and $\rho = 4$. We call the original ResNet proposed in [2, 3] *CP-Res* throughout the report.

3.2. Frequency Damping: *CP-Res Damp*

The results of [2] showed that restricting the RF of the CNNs, especially over the frequency dimension, results in a better generalization in various ASC datasets. We improved the performance of the proposed architectures by further restricting the effective receptive field of the convolutional layer [10, 2]. Each convolutional neuron has a limited receptive field of its layer input. We reduce the influence of a neuron input, the further away the input is from the center of the neuron's receptive field. The resulting networks are called *damped* CNNs. In practice, we damp the filters of a convolutional layer by element-wise multiplying the weights with a non-trainable constant matrix $C \in \mathbb{R}^{T \times F}$ (damping matrix). The damping matrix matches the spatial shape of the filters and works by decaying the effect, on the output, of outermost elements of the filter over the frequency dimension. In the resulting network, every convolution operation $O_n = W_n * Z_{n-1} + B_n$ is replaced by $O_n = (W_n \odot C_n) * Z_{n-1} + B_n$, where $*$ is the convolution operator and $\odot$ 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 submissions. We refer to this architecture as *CP-Res Damp* in this report.

4. TASK 1A: ACOUSTIC SCENE CLASSIFICATION WITH MULTIPLE DEVICES

4.1. Domain Adaptation

As the distributions of sound from different devices differ, models trained on one device often have problems generalising to unseen devices [16, 6]. To reduce the mismatch between the source devices (given in the training set) and target devices (in the unseen test set), we incorporate two Domain Adaptation (DA) objectives,

namely, Sliced Wasserstein Distance (SWD) [11], and Maximum Mean Discrepancy (MMD) [12].

In order to reduce the distribution mismatch in our models, we first create two batches of samples with mixed devices, and then use the encoding of the high-level embeddings of the models to reduce mismatch by applying the DA loss. As the model continues to train, the reduction in the DA losses forces the model to create device-invariant representations. This approach is a simple, yet effective method that neither requires any paired data nor label information from various devices. A comparison of our methods can be found in Table 1.

4.2. Results

Table 1 compares the result of our models with and without domain adaptation on the development set. The frequency damping introduced in Section 3.2 improves the accuracy of our baseline ResNet by approximately 1%. Using a DA objective slightly improves our results in case of SWD, while MMD yields on average a worse accuracy than the reference baseline.

Model	DA	Accuracy
CP-Res [2, 3]	✗	0.6981
CP-Res Damp.	✗	0.7107
CP-Res Damp.	MMD	0.7080
CP-Res Damp.	SWD	0.7180

Table 1: Comparison of our domain adaptation (DA) approaches to two baselines on the DCASE'20 Development Test-Split. *Res Damp.* refers to a ResNet with frequency-damping as introduced in Section 3.2. We additionally test this architecture in combination with two DA objectives MMD and SWD. All the networks are built using $\rho = 6$ [3]. The table shows the mean of the accuracy of the last 10 epochs for each model.

4.3. Submissions Summary

All the submitted systems are trained on the whole development set. We average the prediction of the last 5 epochs because there is no remaining validation set.

- **Submission 1: CP_ResNet Frequency-Damped** We train a single network with $\rho = 7$ and frequency damping, and submit the mean of its predictions in the last 5 epochs of training.
- **Submission 2: CP_ResNet F-damped SWD** We train a single network with $\rho = 6$ and frequency damping with an additional Sliced Wasserstein Distance (SWD) as a domain adaptation loss, and submit the mean of its predictions in the last 5 epochs of training as well as the predictions of the 5 last Stochastic Weight Averaging (SWA) models (as explained in [4]).
- **Submission 3: CP_ResNet DA and non-DA ensemble** We average the predictions of 14 models ($\rho = 6$ or $\rho = 7$; damping or basis network; without DA, with SWD or with MMD). For each model we also average the last 5 SWA and non-SWA models.
- **Submission 4: CP_ResNet DA ensemble** Similar to submission 3, but we only average the predictions of the models trained with domain adaptation loss.

5. TASK 1B: LOW-COMPLEXITY ACOUSTIC SCENE CLASSIFICATION

The goal of Task 1B is to learn low-complexity models, i. e., models with a restricted number of parameters, to classify three different acoustic scenes. The model size is restricted to 500 KB of non-zero parameters which translates to 128K and 256K parameters with full and half floating point precision, respectively.

We train the models using full precision floating point data type for the parameters and the gradients. We cast the model weights to half precision floating point for inference and testing. Our experiments on the development set showed no (or insignificant) performance drop after casting the parameters to half precision.

5.1. Approaches

In the following, we investigate different methods to maintain a high classification accuracy while keeping the number of parameters within the allowed limit. In all the approaches, we follow the principle of reducing the number of parameters while keeping the final receptive field of the network constant.

5.1.1. Width and Depth restriction

The basis ResNet [2, 3] has many 1×1 convolutional layers that significantly increase the number of parameters without affecting the final receptive field of the network. Since the task is simplified to a 3 class classification problem, we experimented with removing several number of this layers without a significant performance drop. Furthermore, reducing the width of the network has a large impact on the number of parameters without a significant performance drop. Since the width of the network i. e., the number of channels in the convolutional layers has a quadratic relationship with the number of parameters in the convolutional weights.

5.1.2. Parameter Pruning

For each convolutional layer we learn a pruning mask by replacing each convolution $O_n = W_n * Z_{n-1}$ by $O_n = (W_n \odot g(M_n)) * Z_{n-1}$ Where $*$ 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. M_n is the learnable pruning mask weights and has the same shape as W_n. $g(x)$ is the gating function. We define $g(x)$ to work in the forward pass as a binary gate:

$$g(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \tag{1}$$

In the backward pass g passes the gradient to the pruning mask M_n as a Sigmoid activation: $\frac{d}{dx} g(x) = \frac{e^{-x}}{(1+e^{-x})^2}$. This learning approach allows the original weights of the network to be learned as if $g(M_n)$ is a constant mask, while the pruning weights are learned as a scale of the original weights. Therefore, we can sort the pruning weights based on their value in order to reach a required pruning ratio.

The overall adaptive-pruning learning algorithm works by linearly increasing the pruning ratio p from 0 to a specific ratio to achieve the required number of parameters. We start training by setting $M_n = 1$ for all the layers. At the end of each training epoch we sort the learned pruning mask weights M (either per layer or globally) and set a portion (corresponding to p) of the smallest weights

to -1 effectively pruning the corresponding weight. We reset the remaining elements of M to 1 for the next epoch, and repeat after the epoch for the new value of p.

5.1.3. Decomposed Convolutions

Inspired by the use of singular-value-decomposition (SVD) for convolutional neural networks [15], we propose to directly train decomposed convolutional layers. Given a regular convolutional layer with dimensionality

$$C_{in} \times C_{out} \times k \times k, \tag{2}$$

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

$$\begin{gathered} C_{in} \times (C_{out}/Z) \times 1 \times 1 \\ (C_{out}/Z) \times (C_{out}/Z) \times k \times k \\ (C_{out}/Z) \times C_{out} \times 1 \times 1 \end{gathered} \tag{3}$$

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. In this way we construct a model with less than 20000 parameters which achieves more than 95.8% accuracy on the development set.

5.2. Results

Table 2 compares the result of our models with different number of parameters on the development set. Our baseline ResNet with frequency-damping achieves 97.6% accuracy on the test data, however with more than 3 million parameter it exceeds the parameter limitation. Our pruned network (*CP-Res Damp.-GP*) is able to achieve a similar accuracy of 97% with approximately 13 times less parameters, thus staying within the parameter limit. Our smallest models relying on decomposed convolutions still achieve an accuracy close to 96% with only 1% of the parameters of the baseline model.

Model	# NZ/Total	Size (KB)	Acc.
CP-Res [2, 3]	3415K/3431K	6671.5	0.9685
CP-Res Damp.	3415K/3431K	6671.5	0.9761
CP-Res Damp.-R	224K/227K	437.8	0.9709
CP-Res Damp.-GP	247K/345K	483.5	0.9737
CP-Res Dec.	17.5K/18.3K	34.2	0.9583
CP-Res Damp.-Dec.	17.5K/18.3K	34.2	0.9595

Table 2: Comparison of our parameter reduction methods to two baseline architectures on the DCASE'20 Development Test-Split. We report the accuracy (Acc.) over all classes, the number of non-zero (NZ) parameters (without batch normalization), the total number of parameters as well as the size of the model in KB. Note that we use half-precision (Float16) for the calculation of the model size. *Res Damp.* refers to a ResNet with frequency-damping as introduced in Section 3.2, *Res Damp.-R* is a frequency-damped ResNet with width and depth restriction (cf. Section 5.1.1, *Res Damp.-GP* is a frequency-damped ResNet in combination with gate pruning (cf. Section 5.1.2) and *Res Dec.* is a ResNet where all convolutions are replaced with decomposed convolution and a compression factor $Z = 4$ (cf. Section 5.1.3). The table shows the mean of the accuracy of the last 10 epochs for each model.

5.3. Submissions Summary

All the submitted system are trained on the whole development set, the weights are converted to half precision floating point, the learned pruning mask (if exist) is applied to the weights and pruning weights are set to zero. We then report the final number of non-zero parameters and the model size.

- **Submission 1: CP_ResNet Decomposed**
  We decompose the convolutional layers of the network ($\rho = 4$) as explained in Section 5.1.3. the resulting network has 17520 parameters and a total size of 34.21875 KB, the performance of the network when trained on the development set is reported as "Res Dec." in Table 2.
- **Submission 2: CP_ResNet RF-Damp Gate Prune**
  We apply the adaptive pruning on a frequency damped Resnet (Section 3.2) ($\rho = 4$) as explained in Section 5.1.2. We prune only the weights of 1×1 convolutional layers. The network has in total 345990 parameters including 77824 parameters of the 1×1 convolutional and the same number of parameters for the pruning mask weights. After training and adaptive pruning the network the total number of non-zero parameters adds up to 247562 with a total size of 483.520 KB
- **Submission 3: CP_ResNet RF-Damp small width/depth**
  We reduce the width and depth of the basis network so that it fits within the size limit. The resulting network has 242592 parameters and a size of 473.813 KB
- **Submission 4: CP_ResNet ensemble of smaller models**
  In this submission we average 3 smaller models so that their total size fits in the size limit with 249386 parameters and 487.082 KB . The three models are as follows: a) A damped ResNet with decomposed weights and adaptive pruning with a total number of non-zero parameters of 87168. b) A restricted ResNet with 100288 parameters. c) A ResNet with adaptive pruning with a total number of non-zero parameters of 61930.

6. CONCLUSION

In this technical report, we detailed our approaches to tackle Task 1A and 1B of the DCASE-2020 challenge. We showed that by adding a further limitation on the effective receptive field in the form of frequency-damping, we improve the accuracy of our RF-regularized baseline ResNet. Additionally, we investigated several approaches to reduce the number of parameters in our models. Our results suggest that we can design networks with only 1% of the parameters of a reference model without a severe performance degradation.

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] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, submitted. [Online]. Available: https://arxiv.org/abs/2005.14623

[2] 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 *Proceedings of the European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain, 2019.

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

[4] 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., June 2019.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in dcase 2019 challenge: Closed and open set classification and data mismatch setups," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 164–168.

[6] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 204–208.

[7] B. Lehner and K. Koutini, "Acoustic scene classification with reject option based on resnets," DCASE2019 Challenge, Tech. Rep., June 2019.

[8] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, and X. Serra, "Audio tagging with noisy labels and minimal supervision," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 69–73.

[9] 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*. ceur-ws.org, 12 2019.

[10] 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.

[11] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, "Generalized sliced wasserstein distances," in *Advances in Neural Information Processing Systems*, 2019, pp. 261–272.

[12] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012. [Online]. Available: http://jmlr.org/papers/v13/gretton12a.html

[13] 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.

[14] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[15] 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 International Conference on Learning Representations (ICLR) (arXiv:1412.6553)*, San Diego, USA, 2015.

[16] H. Eghbal-zadeh, M. Dorfer, and G. Widmer, "Deep within-class covariance analysis for robust audio representation learning," *Interpretability and Robustness in Audio, Speech and Language Workshop, NeurIPS*, 2018.