

EXPLOITING PARALLEL AUDIO RECORDINGS TO ENFORCE DEVICE INVARIANCE IN CNN-BASED ACOUSTIC SCENE CLASSIFICATION

Paul Primus¹, Hamid Eghbal-zadeh^{1,2}, David Eitelsebner¹

Khaled Koutini¹, Andreas Arzt¹, Gerhard Widmer^{1,2}

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

ABSTRACT

Distribution mismatches between the data seen at training and at application time remain a major challenge in all application areas of machine learning. We study this problem in the context of machine listening (Task 1b of the DCASE 2019 Challenge). We propose a novel approach to learn domain-invariant classifiers in an end-to-end fashion by enforcing equal hidden layer representations for domain-parallel samples, i.e. time-aligned recordings from different recording devices. No classification labels are needed for our domain adaptation (DA) method, which makes the data collection process cheaper. We show that our method improves the target domain accuracy for both a toy dataset and an urban acoustic scenes dataset. We further compare our method to Maximum Mean Discrepancy-based DA and find it more robust to the choice of DA parameters. Our submission, based on this method, to DCASE 2019 Task 1b gave us the 4th place in the team ranking.

Index Terms— Domain Adaptation, Recording Device Mismatch, Parallel Representations, Acoustic Scene Classification

1. INTRODUCTION

Convolutional Neural Networks (CNNs) have become state of the art tools for audio related machine learning tasks, such as acoustic scene classification, audio tagging and sound event localization. While CNNs are known to generalize well if the recording conditions for training and unseen data remain the same, the generalization of this class of models degrades when there is a distribution dissimilarity between the training and the testing data [1].

In the following work we elaborate our findings for subtask 1b of 2019's IEEE DCASE Challenge, which is concerned with a domain mismatch problem. The task is to create an acoustic scene classification system for ten different acoustic classes. A set of labelled audio snippets recorded with a high-quality microphone (known as Device A) is provided for training. Additionally, for a small subset of samples from device A, parallel recordings from two lower quality microphones (devices B and C) are given. Evaluation of methods is done based on the overall accuracy on unseen samples from devices B and C. The acoustic scene, the city, and the device labels are provided for samples of the development set only. The main challenge of task 1b is to develop a model that, although trained mostly on samples from device A, is able to generalize well to samples from devices B and C. Since this problem is related to the field of *Domain Adaptation (DA)*, we refer to the

distribution of device A samples as the *source domain*, and the distribution of samples of B and C devices as the *target domain*. In this work we explain how a state-of-the-art CNN model which by itself achieves high accuracy can be further improved by using a simple DA technique designed for problems where parallel representations are given.

2. RELATED WORK

Domain Adaptation (DA) is a popular field of research in transfer learning with multiple areas of application, e.g. bird audio detection [2]. Kouw et al. [3] distinguish between three types of data shifts which lead to a domain mismatch: prior, covariate and concept shift. In this work we focus on domain mismatches which are caused by covariate shifts (i.e., changes in feature distributions).

According to Shen et al. [4] solutions to domain adaptation can be categorized into three types: (i) Instance-based methods: reweight or subsample the source dataset to match the target distribution more closely [5]. (ii) Parameter-based methods: transfer knowledge through shared or regularized parameters of source and target domain learners [6], or by weighted ensembling of multiple source learners [7]. (iii) Feature-based methods: transform the samples such that they are invariant of the domain. Weiss et al. [8] further distinguish between symmetric and asymmetric methods. Asymmetric methods transform features of one domain to match another domain [9] symmetric feature-based methods embed samples into a common latent space where source and target feature distributions are close [10]. Symmetric feature-based methods can be easily incorporated into deep neural networks and therefore have been studied to a larger extent. The general idea is to minimize the divergence between source and target domain distributions for specific hidden layer representations with the help of some metric of distribution difference. For example, the deep domain confusion method [10] and deep adaptation network [11] use Maximum Mean Discrepancy (MMD) [12] as a non-parametric integral probability metric. Other symmetric feature-based approaches exist that use adversarial objectives to minimize domain differences [13, 4]. These methods learn domain-invariant features by playing a minimax game between the domain critic and the feature extractor where the critic's task is to discriminate between the source and the target domain samples and the feature extractor learns domain-invariant and class-discriminative features. However, training the critic introduces more complexity, and may cause additional problems such as instability and mode collapse.

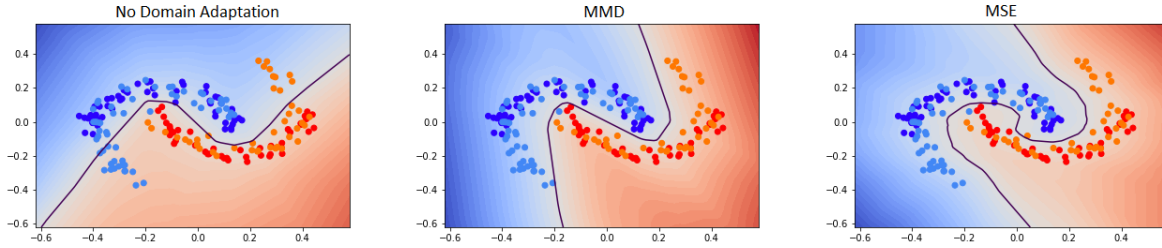


Figure 1: Two Moons dataset: best classifiers found by grid search over λ and n (Tab. 1). The source domain is represented by dark blue and red data points, the shifted target domain by light blue and orange points. The black line shows the decision boundary of a classifier trained without DA (left), with MMD-DA (middle) and with MSE-DA (right). Red and blue shaded areas represent decision areas of the classifiers.

3. DOMAIN-INVARIANT LEARNING

We propose a symmetric feature-based loss function to encourage the network to learn device-invariant representations for parallel samples from the source X^s , and the target domain X^t . This loss exploits the fact that parallel samples contain the same information relevant for classification and differ only due to a covariate shift, e.g. time-aligned spectrograms (x^s, x^t) contain the same information about the acoustic scenes and differ only due to device characteristics. Let $\phi_l(x^s)$ and $\phi_l(x^t)$ be d -dimensional hidden layer activations of layer l for paired samples x^s and x^t from the source and the target domains, respectively. A domain-invariant mapping $\phi_l(\cdot)$ projects both samples to the same activations without losing the class-discriminative power. To achieve this, we propose to jointly minimize classification loss \mathcal{L}_{CL} and the Mean Squared Error (MSE) over paired sample activations, where the latter one is defined as

$$\mathcal{L}_{l,MSE} = \frac{1}{n \cdot d} \sum_{i=1}^n \|\phi_l(x_i^s) - \phi_l(x_i^t)\|_2^2 \quad (1)$$

for some fixed network layer l (this is a hyper-parameter). As we will show in Section 4 the DA mini-batch size n is critical, and our results suggest that bigger n yields better results. The final optimization objective we use for training is a combination of classification loss \mathcal{L}_{CL} and DA loss $\mathcal{L}_{l,MSE}$:

$$\mathcal{L} = \mathcal{L}_{CL} + \lambda \mathcal{L}_{l,MSE} \quad (2)$$

Here, λ controls the balance between the DA loss and the classification loss during training. Note that for $\mathcal{L}_{l,MSE}$ no class label information is required and the labeled samples from all domains can be used for the supervised classification loss \mathcal{L}_{CL} .

4. EXPERIMENTS

In the following we evaluate the performance of our approach on the two moons dataset as well as on real-world acoustic data: the DCASE 2019 Task 1b dataset on acoustic scene classification [14]. We compare our proposed DA objective to the multi-kernel MMD-based approach used by Eghbal-zadeh et al. [15] for DCASE 2019 Subtask 1b. In all experiments, parallel samples are used without any class-label information. For both datasets, we find that when paired samples are given, MSE achieves higher accuracy on the target set compared to MMD.

$n \setminus \lambda$	MSE				MMD			
	0.1	1	5	10	0.1	1	5	10
8	.999	.999	.999	.999	.805	.862	.760	.749
32	.999	.999	.999	.999	.817	.771	.995	.990
128	.999	.999	.999	.999	.801	.859	.754	.739
256	.999	.999	.999	.999	.804	.861	.997	.744

Table 1: Domain Adaptation (DA) results on the two moons dataset: Accuracy on the target domain for models trained with different choices of DA loss, λ (columns) and n (rows). Baseline without DA is at 0.814.

4.1. Experimental Setup

We compare our approach to a baseline that uses the same CNN architecture and classification loss, but does not incorporate a DA loss. As another baseline, we use multi-kernel MMD-based DA [11], a non-parametric symmetric feature-based approach. MMD represents distances between two distributions as distances between mean embeddings of features in reproducing kernel Hilbert space \mathcal{H}_k :

$$d_k^2(X^s, X^t) = \|\mathbb{E}_{X^s} [k(\phi_l(x^s), \cdot)] - \mathbb{E}_{X^t} [k(\phi_l(x^t), \cdot)]\|_{\mathcal{H}_k}^2$$

The kernel k associated with the feature mapping for our experiments is a combination of four equally weighted RBF kernels with $\sigma \in \{0.2, 0.5, 0.9, 1.3\}$. We use the empirical version of this metric as DA loss, for which we randomly sample batches of size n from X^s and X^t . Therefore batches do not necessarily contain parallel representations of samples. Compared to our approach, MMD-based DA matches the distribution between the hidden representations of the source and the target domains, and not between the parallel representations. For both DA methods best results were obtained when applying the DA to the output layer. A plausible explanation is that using higher layer activations gives the network more flexibility for learning domain invariant representations.

4.2. Two Moons

Two moons (see Fig. 1) is a toy dataset often used in the context of transfer learning. It consists of two interleaved class distributions, where each is shaped like a half circle. We use this synthetic dataset to demonstrate our domain adaptation technique under controlled conditions.

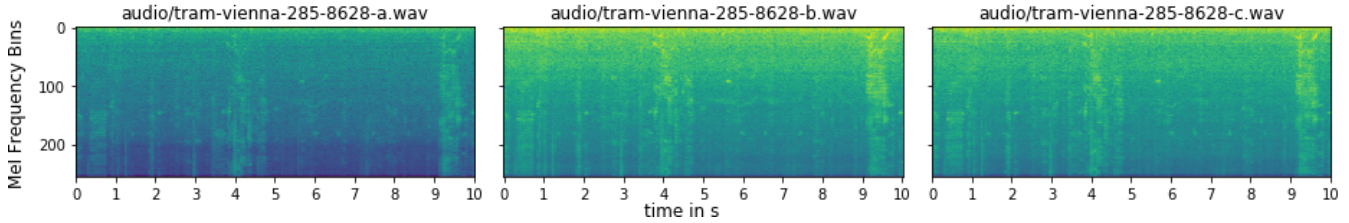


Figure 2: From left to right: Time-aligned recordings from devices A (Soundman OKM II Klassik/Studio A3 Microphone & Zoom F8 Recorder), B (Samsung Galaxy S7) and C (iPhone SE). Spectrograms show microphone-specifics, e.g. samples from devices B and C have more noise in lower Mel bins, compared to those from device A, and samples from device A seem to have fewer energy in all frequency bins.

4.2.1. Dataset & Architecture & Training

We utilize *sklearn* to generate two class-balanced two moons datasets with Gaussian distributed noise ($\mu = 0$ and $\sigma = 0.1$) and 10.000 samples each. Features are normalized to fall into the range of $[-0.5, 0.5]$. Domain-parallel representations are obtained by applying an artificial covariate shift to one of the two datasets. For our initial experiments we use two transformations: a stretching along the y-dimension by a factor of 1.5, and a rotation by -45 degrees (Fig. 1). We assume no label information is available for the parallel dataset. All experiments use a common model architecture, which is a fully connected network with one hidden layer of size 32 and ReLU activations. The output layer consists of one unit with a sigmoid activation function. The weights are initialized with He normal initialization [16]. We train for 250 epochs with mini-batches of size 32, binary cross-entropy loss, ADAM [17] update rule and constant learning rate of 0.001 to minimize Eq. 2.

4.2.2. Results

Without DA the model scores 81.4% accuracy on the target validation dataset (Fig. 1 left). To find a good parameter setting for both domain adaptation techniques we perform grid search over the DA weight $\lambda \in \{0.1, 1, 5, 10\}$ and the DA batch size $n \in \{8, 32, 128, 256\}$. Results are summarized in Table 1. At its best, MMD improves the accuracy on the target dataset to 99.7% (Fig. 1 middle). Regardless of the parameter combination the model with MSE-DA reaches 99.9% accuracy on both source and target domain validation sets. (Fig. 1 right). For all parameter configurations MSE-DA yields better results than MMD-DA.

4.3. Urban Acoustic Scene Dataset

The previous section has demonstrated that MSE-DA can be effectively used when domain-parallel representations are given. It is now necessary to evaluate our prior findings on a real world dataset, in our case the DCASE 2019 Task 1b dataset [14]. As explained in the introduction, our objective is to create a recording device invariant classifier by training it on a larger set of source domain samples and a few time-aligned recordings from the target domain. Fig. 2 shows three time-aligned recordings, for which we can observe the device-specific characteristics. Section 1 describes the DACSE 2019 task 1b in more details. An implementation of the following experiments is available on GitHub¹.

¹https://github.com/OptimusPrimus/dcase2019_task1b/tree/Workshop

4.3.1. Dataset

The dataset contains 12.290 non-parallel device A samples and 3.240 parallel recorded samples (1080 per device). We use the validation setup suggested by the organizers, i.e. 9185 device A, 540 device B, and 540 device C samples for training and 4185 device A, 540 device B, and 540 device C for validation. Preprocessing is done similar to [18]: We resample the audio signals to 22050Hz and compute a mono-channel Short Time Fourier Transform using 2048-sample windows and a hop size of 512 samples. We apply a dB conversion to the individual frequency bands of the power spectrogram and a mel-scaled filterbank for frequencies between 40 and 11025Hz, yielding 431-frame spectrograms with 256 frequency bins. The samples are normalized during training by subtracting the source training set mean and dividing by the source training set standard deviation.

4.3.2. Network Architectures

We use the model architecture introduced by Koutini et al. [19], a receptive-field-regularized, fully convolutional, residual network (ResNet) with five residual blocks (Tab. 2). The receptive field of this architecture is tuned to achieve the best performance in audio-related tasks using spectrograms, as discussed in [19].

Type	ResNet			Residual Block (RB)	
	#K	KS 1	KS 2	Type	KS
Conv+BN	128	5		Conv+BN	KS 1
RB	128	3	1	Conv+BN	KS 2
Max Pool	-	2	-	Add Input	
RB	128	3	3		
Max Pool	-	2	-		
RB	128	3	3		
RB	256	3	3		
Max Pool	-	2	-		
RB	512	3	1		
Conv+BN	10	3	-		
GAP	-	-	-		

Table 2: Model Architecture by [19] for experiments with the acoustic scenes dataset. #K and KS are the number of kernels and kernel size, respectively. Residual Blocks (RB) consist of two Convolutional (Conv) layers with #K kernels, each followed by a Batch Normalization (BN) layer. GAP is a Global Average Pooling Layer.

$n \setminus \lambda$	MSE			MMD		
	0.1	1	10	0.1	1	10
1	.494	.525	.488	-	-	-
8	.537	.592	.556	.467	.434	.412
16	.571	.592	.561	.456	.492	.233

Table 3: Domain Adaptation (DA) results on the acoustic scenes dataset: Accuracy on devices B and C for models trained with different choices of DA loss, λ (columns) and n (rows). Baseline model without DA scores .353 accuracy on the provided split.

4.3.3. Training

Although scene labels are available for all samples, we minimize \mathcal{L}_{CL} over the 8.645 non-parallel device A samples only. The 1.620 time-aligned samples are used to learn domain-invariant features by minimizing pairwise DA loss \mathcal{L}_L between the three devices. For each update-step, we draw a batch from the non-parallel samples and a batch from the parallel samples to compute \mathcal{L}_{CL} and \mathcal{L}_L , respectively. We then minimize the sum of these two losses (Eq. 2). Models are trained for 120 epochs with non-parallel mini-batches of size 32, categorical cross-entropy loss, and ADAM [17] update rule to minimize Eq. 2. The initial learning rate is set to 10^{-3} and decreased by a factor 0.5 if the mean accuracy for devices B and C does not increase for 10 epochs. If the learning rate is decreased, we also reset the model parameters to the best model in terms of mean accuracy of device B and C up to the last epoch. We further use MixUp augmentation [20] with parameters of the beta-distribution set to $\alpha = \beta = 0.2$ for classification as well as DA samples.

4.3.4. Results

The baseline model without domain adaptation scores 35.3% BC-accuracy. We perform grid search over parameters $\lambda \in \{0.1, 1, 10\}$ and $n \in \{1, 8, 16\}$ to find a good combination for both MMD- and MSE-DA. The best model validation accuracy on device B and C (BC-accuracy) over all 120 epochs for each experiment is reported in Table 3. MMD-DA improves the BC-accuracy compared to the baseline without DA for all except one experiment. At its best MMD-DA achieves an BC-accuracy of 49.2%, which is an improvement by 13.9 *p.p.* compared to the model trained without DA. Pairwise representation matching improves BC-accuracy even further: The best MSE-DA model scores 59.2% which is 23.9*p.p.* above the baseline without DA.

4.4. DCASE Challenge 2019 Subtask 1b

In the following section we describe the adjustments made to our challenge submission to be more competitive. Our technical report describes the submitted systems in more detail [21].

4.4.1. Dataset & Cross-Validation & Training

We split all audio segments into four folds, to have more domain parallel samples available for training. Furthermore, we minimize the classification loss over all available samples, including those from devices B and C. We increase the number of training and patience epochs to 250 and 15, respectively. For each fold, the model that scores the highest device BC-accuracy is selected for prediction on evaluation data. As we train every model on 4 folds, our

	Tr.\Te.	4-CV	K. Priv.	K. Pub.	Eval.
Ensemble	-	-	.770	.766	.742
MSE-DA	.644	.697	.762	.758	.734
No-DA	.612	.669	.705	.737	.713

Table 4: DCASE 2019 Task 1b results for different validation sets, from left to right: Device B and C validation accuracy (%) on the provided (Tr.\Te.) and custom split (4-CV), Kaggle private (K. Priv) and public leaderbord (K. Pub.), and the evaluation set (Eval.).

final submission models are ensembles of the outputs of the 4 folds. For submission 1 and 2 we average the softmax predictions of each fold’s best scoring model and select the class with the highest score. Submission 4 combines two independently trained models, again by averaging each of their 4 folds softmax outputs.

4.4.2. Results

The results of our challenge submission measured in BC-accuracy on unseen samples are reported in Table 4. The convolutional ResNet without DA achieved a BC-accuracy of 71.3% on the evaluation set, training on the suggested split achieved a BC-accuracy of 61.2%. The model used in submission three trained with MSE-DA loss gained an additional 2.1*p.p.* on the evaluation set over the base model, resulting in an accuracy of 73.4%. A larger gain can be seen for the proposed split, as with 64.35 the model performed 3.15*p.p.* better than our base model. Our ensemble of eight predictors achieves 74.2% BC-accuracy on the evaluation set which is our best result. The challenge submission by [15] which utilizes MMD-DA to learn device-invariant classifiers scores 74.5% on the final validation set, 0.3 *p.p.* better than ours. The MM-DA used in [15] incorporates across-device mixup augmentation, is applied on a different architecture, integrates ensemble models, and uses a larger batch size, which explains the performance differences.

5. CONCLUSION & FUTURE WORK

In this report, we have shown how an already well-performing ResNet-like model [19] can be further improved for DCASE 2019 task 1b by using a simple DA technique. Our DA loss is designed to enforce equal hidden layer representations for different devices by exploiting time-aligned recordings. In our experiment we find that pointwise matching of representations yields better results, compared to minimizing the MMD between the hidden feature distributions without utilizing parallel representations. Notably, the MSE-DA increased the performance by 3.15 *p.p.* on the validation set of the proposed split, and by 2.1 *p.p.* on the final validation set. Furthermore, acquiring data for our method is cheap as it does not require labels for domain-parallel samples. In future work, we would like to investigate if data from unrelated acoustic scenes, i.e. scenes not relevant for classification, can be used to create device-invariant classifiers, as this would decrease cost even further.

6. ACKNOWLEDGMENT

This work has been supported by the COMET-K2 Center of the Linz Center of Mechatronics (LCM), funded by the Austrian federal government and the Federal State of Upper Austria.

7. REFERENCES

- [1] H. Eghbal-zadeh, M. Dorfer, and G. Widmer, “Deep within-class covariance analysis for robust deep audio representation learning,” in *Neural Information Processing Systems, Interpretability and Robustness in Audio, Speech, and Language Workshop*, 2018.
- [2] F. Berger, W. Freillinger, P. Primus, and W. Reisinger, “Bird audio detection - dcase 2018,” DCASE2018 Challenge, Tech. Rep., September 2018.
- [3] W. M. Kouw, “An introduction to domain adaptation and transfer learning,” *CoRR*, vol. abs/1812.11806, 2018. [Online]. Available: <http://arxiv.org/abs/1812.11806>
- [4] J. Shen, Y. Qu, W. Zhang, and Y. Yu, “Wasserstein distance guided representation learning for domain adaptation,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 4058–4065. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17155>
- [5] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, “Correcting sample selection bias by unlabeled data,” in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, 2006, pp. 601–608. [Online]. Available: <http://papers.nips.cc/paper/3075-correcting-sample-selection-bias-by-unlabeled-data>
- [6] A. Rozantsev, M. Salzmann, and P. Fua, “Beyond sharing weights for deep domain adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 801–814, 2019. [Online]. Available: <https://doi.org/10.1109/TPAMI.2018.2814042>
- [7] L. Duan, D. Xu, and S.-F. Chang, “Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1338–1345.
- [8] K. R. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *J. Big Data*, vol. 3, p. 9, 2016. [Online]. Available: <https://doi.org/10.1186/s40537-016-0043-6>
- [9] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko, “Asymmetric and category invariant feature transformations for domain adaptation,” *International journal of computer vision*, vol. 109, no. 1-2, pp. 28–41, 2014.
- [10] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *CoRR*, vol. abs/1412.3474, 2014. [Online]. Available: <http://arxiv.org/abs/1412.3474>
- [11] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 97–105. [Online]. Available: <http://proceedings.mlr.press/v37/long15.html>
- [12] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, “A kernel two-sample test,” *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2188410>
- [13] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.
- [14] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” *CoRR*, vol. abs/1807.09840, 2018. [Online]. Available: <http://arxiv.org/abs/1807.09840>
- [15] H. Eghbal-zadeh, K. Koutini, and G. Widmer, “Acoustic scene classification and audio tagging with receptive-field-regularized CNNs,” DCASE2019 Challenge, Tech. Rep., June 2019.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 1026–1034. [Online]. Available: <https://doi.org/10.1109/ICCV.2015.123>
- [17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [18] M. Dorfer, B. Lehner, H. Eghbal-zadeh, H. Christop, P. Fabian, and W. Gerhard, “Acoustic scene classification with fully convolutional neural networks and I-vectors,” DCASE2018 Challenge, Tech. Rep., September 2018.
- [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 *Proceedings of the European Signal Processing Conference (EU-SIPCO)*, A Coruña, Spain, 2019.
- [20] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *CoRR*, vol. abs/1710.09412, 2017. [Online]. Available: <http://arxiv.org/abs/1710.09412>
- [21] P. Primus and D. Eitelsebner, “Acoustic scene classification with mismatched recording devices,” DCASE2019 Challenge, Tech. Rep., June 2019.