

# ANOMALOUS SOUND DETECTION AS A SIMPLE BINARY CLASSIFICATION PROBLEM WITH CAREFUL SELECTION OF PROXY OUTLIER EXAMPLES

Paul Primus<sup>1</sup>, Verena Haunschmid<sup>1</sup>, Patrick Praher<sup>3</sup>, and Gerhard Widmer<sup>1,2</sup>

<sup>1</sup>Institute of Computational Perception

<sup>2</sup>LIT Artificial Intelligence Lab

Johannes Kepler University, Austria

<sup>3</sup> Software Competence Center Hagenberg GmbH, Austria

## ABSTRACT

Unsupervised anomalous sound detection is concerned with identifying sounds that deviate from what is defined as “normal”, without explicitly specifying the types of anomalies. A significant obstacle is the diversity and rareness of outliers, which typically prevent us from collecting a representative set of anomalous sounds. As a consequence, most anomaly detection methods use unsupervised rather than supervised machine learning methods. Nevertheless, we will show that anomalous sound detection can be effectively framed as a supervised classification problem if the set of anomalous samples is *carefully substituted* with what we call *proxy outliers*. Candidates for proxy outliers are available in abundance as they potentially include all recordings that are neither normal nor abnormal sounds. We experiment with the machine condition monitoring data set of the 2020’s DCASE Challenge and find proxy outliers with matching recording conditions and high similarity to the target sounds particularly informative. If no data with similar sounds and matching recording conditions is available, data sets with a larger diversity in these two dimensions are preferable. Our models based on supervised training with proxy outliers achieved rank three in Task 2 of the DCASE2020 Challenge.

**Index Terms**— Unsupervised Anomaly Detection, Proxy Outliers, Outlier Exposure, Anomalous Sound Detection, Machine Condition Monitoring, DCASE2020

## 1. INTRODUCTION

Automatic detection of anomalies in audio signals is an active field of research in machine learning. This class of problems has various applications in diverse domains, such as novelty-detection in music [1], audio surveillance of public spaces [2], or acoustic Machine Condition Monitoring (MCM) for predictive maintenance [3].

In this work, we elaborate on our findings for Task 2 of 2020’s IEEE DCASE Challenge [4, 5], which is concerned with unsupervised MCM. MCM systems aim to detect sounds that deviate from what is considered “normal” for a specific machine or a class of machines. Since anomalies (i.e., outliers) typically are rare and diverse, defining and collecting all possible variants in a sufficient quantity to train a classifier is hardly feasible. Moreover, collecting anomalous sounds in MCM often means damaging or destroying the machine, resulting in undesired costs. One workaround is to learn models without explicitly specifying and collecting anomalies, which is commonly referred to as unsupervised Anomaly Detection (AD). However, in this more general scenario, no guidance

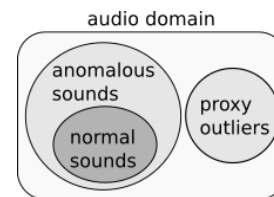


Figure 1: Venn diagram of the data categories. Normal and anomalous sounds are emitted by the target machine in normal and anomalous state, respectively. Proxy outliers are carefully selected from unrelated data in the audio domain.

through anomalous data is available, which makes generalization to future outliers hard.

Most studies in the field disregard data that is unrelated to the task. However we will show in this work that unrelated data, *if chosen carefully*, can be used as a substitute for unavailable anomalous data (Fig. 1). This consequently allows us to frame the unsupervised AD problem as a regular classification problem. We call samples from the unrelated data that will be used for AD *Proxy Outliers* (PO). Our approach is supported by the recommendation for outlier analysis by Aggarwal [6] who suggests to “always use supervision where possible”. In the experimental section, we will investigate under which conditions unrelated sounds are informative for anomaly detection and finally compare our results to unsupervised baselines based on reconstruction error and density estimation.

By merely training a ResNets to distinguish between normal machine sounds and carefully selected proxy outliers, we achieved the 3rd team-rank in Task 2 of the 2020’s DCASE Challenge (Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring). This is especially remarkable as we did not apply data augmentation, transfer learning, model ensembling, or sophisticated postprocessing. Our implementation is available on GitHub<sup>1</sup>.

## 2. RELATED WORK

According to Chandola et al. [7], anomaly detection (AD) is the task of identifying patterns in the data that differ from what is regarded as normal and, in contrast to noise, are of interest for various downstream tasks. Following Aggarwal’s taxonomy [6], we distinguish between the supervised and unsupervised *AD settings* based on whether the characters of anomalies are well defined or

<sup>1</sup>[https://github.com/OptimusPrimus/dcase2020\\_workshop](https://github.com/OptimusPrimus/dcase2020_workshop)

unclear. In *supervised AD*, both normal and anomalous data are defined, available, and labeled for training; the learning task is to fit a classifier that generalizes to future data. Unfortunately, due to the variety of anomalies, it is often hard, in practice, to define and collect anomalies. Even if representative sampling is possible, the class imbalance due to the rare nature of anomalies makes learning particularly challenging. On the other hand, *unsupervised AD* does not make any prior assumptions about the nature of anomalies and is therefore applicable in broader scenarios such as machine condition monitoring. The learning objective turns into creating a model of the “normal”, which is then used to assign anomaly scores to new samples at inference time (e.g., based on log-likelihoods).

Pimentel et al. [8] categorize AD methods into probabilistic, reconstruction-based, distance-based, domain-based, and information-theoretic approaches. The baseline methods we will apply in our work fall into the first two categories and are both based on neural networks. *Reconstruction-based methods* are typically based on Autoencoders or Variational Autoencoder [9] trained to minimize the reconstruction error on the normal training set (e.g., [3, 10]). At test time, samples are encoded and decoded using parameters tailored for normal samples, which usually yields a higher reconstruction error for novel patterns in anomalous data. Note that this method also yields small reconstruction errors for outlier sounds if the Autoencoder generalizes to the anomalous data. Early *probabilistic models* for AD were based on parametric and non-parametric density estimation methods like Gaussian mixture models and kernel density estimation [8]. Recent work in this category uses Normalizing Flows [11] to learn a tractable likelihood approximation of the training data. However, Kirichenko et al. [12] show that the latent representations of normalizing flows are mostly based on local pixel correlations, leading to undesired large likelihood values for semantically unrelated samples.

Recent work in unsupervised AD leverages large amounts of unrelated image data, which is commonly referred to as *outlier exposure* [13, 14]. Most related to our work are the findings of Ruff et al. [14], who report that a classifier trained on normal data and only 64 random unrelated images can outperform the current state of the art in deep AD. In their experiments, they attribute the unrelated dataset’s informativeness to the diversity of the content and the rich multi-scale structure of images.

### 3. METHOD

To overcome the scarcity problem of anomalous sounds, we propose to substitute real outliers with *Proxy Outliers* (PO), i.e., *carefully selected* recordings that are neither normal nor abnormal sounds (Fig. 1). Note that, compared to anomalous sounds, POs are cheap and easy to collect if not already available in abundance. In the experimental section, we test a variety of candidate PO sets and investigate under which conditions such arbitrary recordings can be used as POs. To show that our approach can be applied in an unsupervised AD setting (i.e., without explicitly defining anomalies), we make no assumptions on the nature of anomalies and presume only normal data and various candidates for POs are given.

To determine what kind of POs can be utilized for acoustic MCM, we take advantage of machine sounds contained in the combined version of the MIMII [15] and ToyAdmos [16] data sets released for Task 2 of the DCASE2020 Challenge [4]. This set includes sounds of 41 machine instances, categorized into six different machine types: fan, pump, slider, toy car, toy conveyor, and valve. We train deep anomaly detectors per machine instance,

using the normal sounds of each instance as target data set and several combinations of the remaining machines instances’ training sets as POs. Note that the test sets of MIMII and ToyAdmos contain real anomalous machine sounds, and are therefore exclusively used to evaluate our approach. To further increase the variety of PO candidates, we also experiment with the TAU Urban Acoustic Scenes 2020 Mobile (TAU-UAS) data set [17] and the balanced version of AudioSet [18], which contains sounds of humans, animals, and even music.

#### 3.1. Normal Sound Data Sets

The ToyAdmos and the MIMII data set contain recordings of several machines in a normal and an anomalous operation state, which are grouped into two and four different machine types, respectively. For each machine type, recordings of seven machine instances are available, except for toy conveyor where only six instances are given. Following the DCASE Challenge, we split the data of each machine instance into a training and a testing set, such that the first one contains only normal sounds, and the second one contains both normal and anomalous sounds. To train an anomaly detector for a specific target machine instance, we use the training set of this particular machine instance as normal data. Note that for three instances of each machine type, the test data labels are not given, since these are used to rank the DCASE Challenge submissions. Therefore, we use the training data of these instances only as part of the proxy outlier data. For further details on the exact recording procedure and the distribution of train and test data per machine instance, we would like to refer the reader to the original data set papers [16, 15] and the DCASE Challenge description [4].

#### 3.2. Proxy Outlier Set Candidates

Having defined the normal data set for each machine instance, we will now describe how the corresponding six candidate PO sets are created from the remaining machines’ normal sets, TAU-UAS, and AudioSet.

First, we create three combinations of the remaining machines’ normal data so that recording conditions are preserved. Note that ToyAdmos and MIMII were recorded using separate hardware and following a different procedure, and therefore must not be used simultaneously for training if identical recording conditions are desired. For a particular target machine instance in ToyAdmos (MIMII), we consequently combine only the remaining machines’ normal data in ToyAdmos (MIMII). The three PO variants (Fig. 2) are obtained by combining

- (a) the remaining training sets of the *same* machine type,
- (b) the remaining training sets of *different* machine types, or
- (c) *all* remaining training sets.

Furthermore, we experiment with three PO candidate sets with mismatching recordings conditions, specifically with

- (d) the training sets of ToyAdmos if the target instance is in MIMII and vice versa,
- (e) the TAU-UAS data set, and
- (f) the balanced version of AudioSet.

Note that AudioSet and TAU-UAS are both rich in recording conditions and content, while ToyAdmos and MIMII only contain machine sounds and follow a rigorous recording procedure.

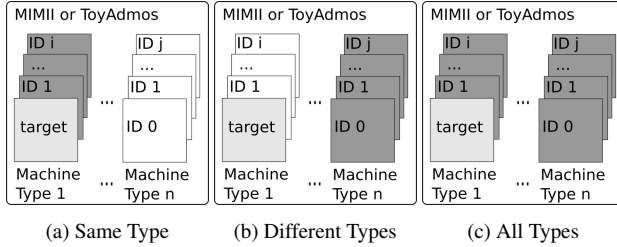


Figure 2: Variants of proxy outlier sets with equal recording conditions. The target set contains the normal sounds (light grey). Proxy outliers sets are selected from the remaining machines’ training sets (dark grey).

## 4. EXPERIMENTAL SETUP

The following section gives a more detailed account of the model architecture, the training procedure, the baseline systems which are based on reconstruction error and density estimation, and the evaluation method.

### 4.1. Network Architecture

We choose the model architecture introduced by Koutini et al. [19], a receptive-field-regularized, fully convolutional, residual network (ResNet) [20], which has been successfully adopted for various audio-related classification tasks [21, 22]. The model consists of three stages with four residual blocks each. The first stage is preceded by a convolutional layer with 64 filters of kernel size  $5 \times 5$ . We apply global average pooling after the last layer, which allows us to use inputs of varying sizes. In the first stage, a max-pooling layer with kernel size  $2 \times 2$  follows after the first, second, and fourth residual block. Each residual block consists of two convolutional layers, where we use 64, 128, and 256 filters in the first, second, and third stage, respectively. In the first stage, we use filters of size  $3 \times 3$ , except for the second convolution of the first and fourth block where we use  $1 \times 1$  filters. In the other two stages, we use filters of size  $1 \times 1$ . A batch normalization layer [23] follows each convolutional layer and we use ReLU activations.

### 4.2. Preprocessing, Training & Inference

Following the DCASE 2020 Challenge task 2 baseline system [3], we re-sample the audio signals to 16000Hz and compute a mono-channel Short Time Fourier Transform using 1024-sample windows and a hop-size of 512 samples. The re-sampled audio waveform is normalized to unit variance. We weight the resulting power spectrogram with a mel-scaled filterbank of 128 filters and apply the logarithm to dampen large outliers. Training of the model is performed on random snippets of 256 frames length to minimize the binary cross-entropy loss for 100 epochs using the Adam update rule [24] with  $\beta_{11} = 0.9$  and  $\beta_{22} = 0.99$  and a batch size of 64. Batches are stratified to contain 32 positive and 32 negative samples. During one epoch the network sees all normal samples; proxy outliers are drawn randomly. We set the initial learning rate to  $10^{-4}$  and decay it each epoch by a factor of 0.99. The anomaly score for each test example is obtained by collecting all 256-frame windows of the input, computing the logit score for each of them, and then mean aggregating all the logit scores to a single value.

### 4.3. Baselines

The section below describes the baseline methods based on reconstruction error and density estimation.

#### 4.3.1. Reconstruction Error Based

Following the DCASE2020 Challenge baseline [3], we train a symmetric autoencoder with eight fully connected layers to minimize the reconstruction error on spectrogram snippets of 5 frames length on the normal data set. All layers have 128 units, except for the bottleneck layer in the middle, which only has eight units. The test samples’ reconstruction error, averaged over the whole sample, is used as anomaly score. Compared to the official baseline system, we found this method to perform slightly better if trained per machine instance, and, for this reason, use it as a new baseline. We apply the preprocessing steps and training procedure described above, but change the batch size to 512 and the learning rate weigh decay to one to match the training procedure of the DCASE2020 Challenge baseline system.

#### 4.3.2. Density-Estimation Based

In addition to the autoencoder baseline we train a Masked Autoregressive Flow (MAF) [25] model as described in [26]. This approach is inspired by [11] who used different types of normalizing flows (including MAFs) for novelty detection in industrial time series data. Due to their fast data likelihood estimation MAFs are well suited for anomaly detection. The model learns the distribution of the training data which allows using the negative log-likelihood of an unseen sample as its anomaly score. The anomaly score for each sample at test time is computed as the average over all non-overlapping 4 frame snippets. The reported model has 2048 hidden units per layer, 1 hidden layer per invertible block, and consists of 4 such blocks. The model is trained in an unsupervised fashion (only access to normal samples) but conditioned on the machine ID (resulting in a 41-dimensional one-hot encoded vector  $y$ ). To be able to condition on the machine ID one model was trained on all machines available for training at once. We apply preprocessing steps as described above and train the model as described in [26].

### 4.4. Evaluation

As standard in AD, we use the Area Under the Receiver Operating Characteristics Curve (AUC), which simultaneously takes true positive rate and false positive rate into account without fixing a threshold to distinguish between normal and anomalous samples. Let  $N_+$  and  $N_-$  be the set of anomalous and normal samples, respectively, and  $\mathcal{A}$  an anomaly detector. The AUC score is then defined as follows:

$$AUC = \frac{1}{|N_+| \cdot |N_-|} \sum_{x_- \in N_-} \sum_{x_+ \in N_+} \mathbb{1}_{\mathcal{A}(x_+) - \mathcal{A}(x_-) > 0}$$

The AUC can be interpreted as the probability that the anomaly detector  $\mathcal{A}$  ranks two randomly selected positive and negative samples correctly.

## 5. RESULTS & DISCUSSION

We train three anomaly detectors for each baseline method and each variant of the PO data set. To give an overview of the results, we

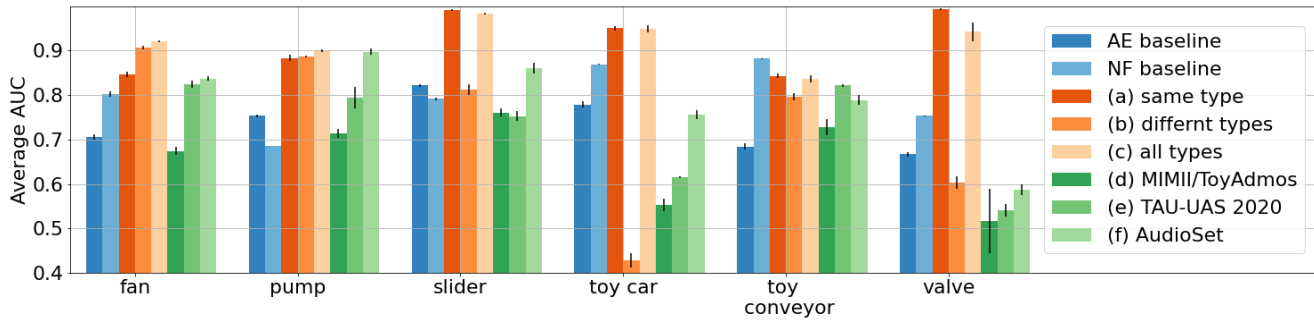


Figure 3: AUC averaged over machine instances per machine type. Blue bars represent the baseline systems. Orange bars are proxy outlier (PO) sets with similar recording conditions, green bars PO sets with mismatching recording conditions. Black error bars represent  $\pm 1$  standard deviation of three runs.

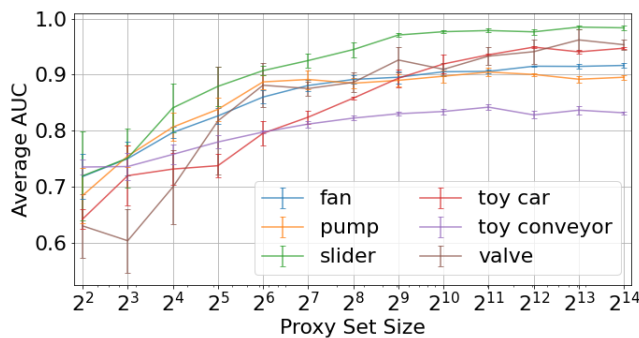


Figure 4: AUC averaged per machine type with increasing number of proxy outlier samples. Proxy outliers are randomly selected from all remaining machine instances with matching recording conditions (c). Error bars represent  $\pm 1$  standard deviation of three runs.

average the AUC scores per machine type and show the results in Figure 3.

We observe a significant improvement over the baselines (blue bars) if the PO set is composed of samples with matching recording conditions (orange bars), and the set includes machines of the same type as the target machine (dark and light orange). Sounds of the same machine type may be more useful as PO because they are closer to the actual decision boundary.

What stands out in our results is that only using the training set of different machine types leads to a substantial drop in AUC for some machine types, especially for slider, toy car, and valve. This could be explained by the fact that the sounds of different machines are too dissimilar to the target sounds and do not contain diverse content. As a consequence, the classifier might be able to distinguish between normal sounds and POs based on simple statistics and does not learn discriminative features for AD.

For the PO sets with mismatching recording conditions (green bars), we observe a positive correlation between the improvement over the baseline and the diversity of recording conditions and content. If AudioSet is used as PO set, we notice a significant increase in AUC over the baselines for fan, pump, slider, and toy conveyor and a considerable decrease for toy car and valve (light green). A

similar but weaker pattern arises if the PO set is composed of samples from the TAU-UAS dataset (medium green): We observe an improvement for fan, pump, and toy conveyor and no increase or even a deterioration for slider, toy car, and valve. Using MIMII for machine instances in ToyAdmos and vice versa yields a considerable drop in AUC for all machine types except for toy conveyor.

Next, we investigate the role of the PO set size by randomly sampling subsets from all training sets of the remaining machine instances with matching recording conditions. We start with four samples and gradually increase the PO set size to the maximum of 16392 samples (Fig. 4). On average, our method outperforms the autoencoder and NF baseline (except for toy conveyor) with only 128 and 512 POs, respectively.

Note that our observations are consistent with the findings by Ruff et al. [14], who found that the performance of their outlier detector increased with the diversity of the POs, and only a few POs are necessary to outperform unsupervised baselines. Besides, we identify similarity to the target sounds and matching recording conditions as critical success factors.

## 6. CONCLUSION

In this study, we show that carefully selected proxies for anomalous sounds can be leveraged to frame the task of anomalous sound detection as a simple classification problem. In our machine condition monitoring experiments, we find that matching recording conditions, similarity to the target sounds, and content diversity make POs exceptionally informative. We also show that relatively few PO samples are needed to outperform unsupervised baselines based on reconstruction error and density estimation.

## 7. ACKNOWLEDGMENTS

We thank Khaled Koutini for making his implementation of the receptive-field-regularized ResNet available.

Parts of this research have been funded within the project AutoDetect (FFG project no. 862019) and by the BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG.

## 8. REFERENCES

- [1] Y. Lu, C. Wu, A. Lerch, and C. Lu, “Automatic outlier detection in music genre datasets,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 101–107.
- [2] H. Lim and J. Park, “Rare sound event detection using 1d convolutional recurrent neural networks,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, September 2017, pp. 80–84.
- [3] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, “Unsupervised detection of anomalous sound based on deep learning and the neyman-pearson lemma,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 1, pp. 212–224, 2019.
- [4] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, “Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring,” in *arXiv e-prints: 2006.05822*, June 2020, pp. 1–4.
- [5] P. Primus, “Reframing unsupervised machine condition monitoring as a supervised classification task with outlier-exposed classifiers,” DCASE2020 Challenge, Tech. Rep., July 2020.
- [6] C. C. Aggarwal, *Outlier Analysis*. Springer, 2013, ch. Supervised Outlier Detection.
- [7] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.
- [8] M. A. F. Pimentel, D. A. Clifton, L. A. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Process.*, vol. 99, pp. 215–249, 2014.
- [9] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [10] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. W. Schuller, “A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. IEEE, 2015, pp. 1996–2000.
- [11] M. Schmidt and M. Simic, “Normalizing flows for novelty detection in industrial time series data,” *CoRR*, vol. abs/1906.06904, 2019.
- [12] P. Kirichenko, P. Izmailov, and A. G. Wilson, “Why normalizing flows fail to detect out-of-distribution data,” *CoRR*, vol. abs/2006.08545, 2020.
- [13] D. Hendrycks, M. Mazeika, and T. G. Dietterich, “Deep anomaly detection with outlier exposure,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [14] L. Ruff, R. A. Vandermeulen, B. J. Franks, K. Müller, and M. Kloft, “Rethinking assumptions in deep anomaly detection,” *CoRR*, vol. abs/2006.00339, 2020.
- [15] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, “MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, November 2019, pp. 209–213.
- [16] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, “ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection,” in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, November 2019, pp. 308–312.
- [17] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13.
- [18] 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 776–780.
- [19] K. Koutini, H. Eghbal-zadeh, and G. Widmer, “Receptive-field-regularized CNN variants for acoustic scene classification,” *CoRR*, vol. abs/1909.02859, 2019.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.
- [21] 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 *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*, 2019, pp. 1–5.
- [22] K. Koutini, S. Chowdhury, V. Haunschmid, H. Eghbal-zadeh, and G. Widmer, “Emotion and theme recognition in music with frequency-aware rf-regularized CNNs,” *CoRR*, vol. abs/1911.05833, 2019.
- [23] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 448–456.
- [24] 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.
- [25] G. Papamakarios, I. Murray, and T. Pavlakou, “Masked autoregressive flow for density estimation,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 2338–2347.
- [26] V. Haunschmid and P. Praher, “Anomalous sound detection with masked autoregressive flows and machine type dependent postprocessing,” DCASE2020 Challenge, Tech. Rep., July 2020.