

Deep Learning Based Open Set Acoustic Scene Classification

Zuzanna Kwiatkowska, Benjamin Kalinowski, Michał Kośmider, Krzysztof Rykaczewski

Samsung R&D Institute
Warsaw, Poland

{z.kwiatkowsk, b.kalinowski, m.kosmider, k.rykaczewsk}@samsung.com

Abstract

In this work, we compare the performance of three selected techniques in open set acoustic scenes classification (ASC). We test thresholding of the softmax output of a deep network classifier, which is the most popular technique nowadays employed in ASC. Further we compare the results with the Openmax classifier which is derived from the computer vision field. As the third model, we use the Adapted Class-Conditioned Autoencoder (Adapted C2AE) which is our variation of another computer vision related technique called C2AE. Adapted C2AE encompasses a more fair comparison of the given experiments and simplifies the original inference procedure, making it more applicable in the real-life scenarios. We also analyse two training scenarios: without additional knowledge of unknown classes and another where a limited subset of examples from the unknown classes is available. We find that the C2AE based method outperforms the thresholding and Openmax, obtaining 85.5% Area Under the Receiver Operating Characteristic curve (AUROC) and 66% of open set accuracy on data used in Detection and Classification of Acoustic Scenes and Events Challenge 2019 Task 1C.

Index Terms: acoustic scenes, open set classification, deep learning, conditional autoencoders

1. Introduction

A significant amount of research is still needed to reliably recognise sound in real-life scenarios, where multiple sounds are present, often simultaneously, and are distorted by the environment. Deep learning methods have been successfully used in addressing many of such audio processing problems, because they are suitable for solving complex and nonlinear learning tasks. Those methods are important from the academic point of view, but also find applications in the real-world use cases, considering a rapidly growing market of audio-related AI features [1]. One of the problems considered in audio AI is the problem of acoustic scene classification (ASC).

ASC is a task of recognizing the environment of the recording device using only sound [2]. The challenges of such tasks depend on label granularity. If we consider a setting with broad scene classes, such as *indoor-outdoor*, the problem would be to obtain coherent class' features representations due to the high in-class data variability. When a finer granularity is considered, we usually focus on a selected subset of classes, but the models should still be able to react appropriately when the recording device is placed in an unknown environment. This creates a necessity to introduce a mechanism able to detect unknown examples and a need for improving methods concerning open set ASC. By an *unknown*, we understand an example that does not belong to any of the classes that are recognizable by the given closed-set classifier.

In an *open set* classification problem, we aim to classify within a known set of labels (so called *closed set*), but also to detect when a given sample is an unknown [3]. Although the problem was, so far, more researched in the computer vision (CV) field, it recently became popular in audio-related tasks. One of the reasons behind it was introducing the open set ASC as one of the problems in Task 1C of Detection and Classification of Acoustic Scenes and Events Challenge (DCASE 2019) [4]. However, the methods proposed there have been very limited, and most utilized very similar techniques (see Section 2). Therefore, the main goal of this paper is to analyse the behaviour of open set classification methods that were designed for CV problems when applied to ASC.

2. Related Work

Many methods described in this section utilize the output of a closed set classifier as a part of their open set inference framework. In both ASC and CV, convolutional neural networks gained popularity due to their performance [5, 6, 7]. In our experiments, we also used them as the open set methods' core [8].

In terms of the open set classification in CV, the first paper formalizing the problem was the work of Scheirer et. al. [3], where a simple SVM-based method was proposed and tested on the ImageNet [9]. A direct extension of this approach called W-SVM [10] additionally consisted of the Extreme Value Theory (EVT) outlier fitting as a method for calibrating scores obtained from a closed set classifier. Since then, the EVT became widely adopted [11, 12]. However, those methods are specific to SVM and are not easily applicable to deep learning.

More recently Openmax [13] was proposed, which also relies on EVT. It modifies softmax to accommodate the unknown class. It is also specifically designed for deep learning. The idea of this technique was to construct a feature representation of each class and measure how much new examples diverge from these representations. When the divergence was substantial, a probability that a new example was from a predicted class was penalized, in favour of the class representing the unknowns. An idea to extend Openmax with a feature representation produced by an autoencoder was introduced in the Classification-Reconstruction Learning [14].

In Class-Conditioned Autoencoder (C2AE) [15], the autoencoder was trained to reconstruct examples correctly if they were conditioned with a correct label, and incorrectly otherwise. During the inference, the reconstruction errors from all known labels were used to detect if the considered example diverges from the training data. More methods concerning open set classification in CV can be found in [16].

In ASC, methods used so far were mostly limited to thresholds [17, 18, 19, 20], meaning the value $\varepsilon \in (0, 1)$ was chosen and the example was predicted as unknown if the predicted class probability was lower than ε . Although in the DCASE

2019 Challenge they performed relatively well, the main concern was the neural networks’ robustness problem, where an incorrect classification may have a relatively high confidence. Similarly to the CV field, there were also attempts to use both SVM- [21] and autoencoder- [22] based methods, but, so far, with worse results than thresholds.

Taking into account the similarity between the closed set deep learning methods used in CV and audio processing, the question arises if using the open set algorithms developed in CV would produce an improvement in ASC over methods used so far.

3. Selected Methods

For experimental purposes, we decided to choose three open set classification techniques: the thresholds, Openmax and the Adapted C2AE (our variant of C2AE). The thresholds were chosen because, so far, they were used most often in ASC tasks and because it is the most straightforward and an easily reproducible technique. The Openmax algorithm was chosen as an example of the EVT fitting. We also wanted to evaluate a method based on autoencoders, due to their popularity in anomaly detection, and therefore decided to choose C2AE which is relatively new.

All of these techniques utilize the output of a closed set classifier. To allow for a more fair comparison of the selected methods, and not of the underlying classifiers, we first trained a common base model.

3.1. Closed Set Classifier Base

In our experiment, by the classifier we understand a deep neural network with a softmax output. We selected two approaches to training the classifier:

- (C1) the classifier is trained only on the target classes,
- (C2) the classifier is trained on the target classes and an additional class with some examples of unknowns.

In the (C1) approach, we need to separate the known subspace from the unknown one without any prior knowledge of the unknown subspace. In this case, the challenge is to classify correctly the unknown examples that are very similar to the target ones (for example if we had *metro* vs. *rest* classification, we could have difficulties with rejecting *tram* as an unknown class).

The (C2) approach can be beneficial, as we also model some part of the unknown examples subspace. We may choose unknown examples that are very similar to our target classes in terms of feature representations, which may allow more precise separation. However, in many real-life use cases, we do not have the data with unknown examples. Moreover, such unknown class may have a high in-class examples variability, which could lead to poorer classifier performance on this class.

3.2. Thresholds

Using thresholds on the softmax output of a closed set classifier is the most straightforward approach to enabling the usage of such models in solving an open set problem. However, it is based on an assumption that the correctness of prediction is correlated with the softmax confidence output, which may not always be the case (an example of such experiment can be seen in [23]).

Let us denote by \hat{y} a softmax output of a closed set classifier for a given example and by ε an assumed decision threshold.

Then we classify an example as

$$\begin{cases} \text{unknown,} & \text{if } \max_{1 \leq i \leq K} (\hat{y}_i) < \varepsilon, \\ \arg \max_{1 \leq i \leq K} (\hat{y}_i), & \text{otherwise,} \end{cases} \quad (1)$$

for $\varepsilon \in (1/a, 1)$, where K is the number of classes. Note that in (C2), the unknown class can be also predicted in the second case, because it is one of the classes that the classifier is trained on. Two out of DCASE 2019 Task 1C winning solutions were utilizing (C2) with thresholds method.

3.3. Openmax

The main idea of Openmax is to use EVT to extend softmax with an additional value representing the unknowns. Its meaning can be interpreted using notions of an outlier and a divergence. A divergence is defined as a weighed sum of Euclidean distance and cosine similarity. Outliers of a given class are defined as points diverging from the average logits (values just before the softmax activation) for examples of this class. Given this, the additional value represents the probability that the predicted logits are outliers from all of the classes.

To estimate the probability that the prediction is an outlier, K Weibull distributions are fitted to represent the maximal possible divergences from the average logits for each of the known classes. The probability that the maximum is smaller than the divergence for the considered example is used as the probability of being an outlier. The distributions are fitted solely using correctly classified examples for each of the classes from the training set. The training and inference procedures are shown in Algorithm 1 and 2 in [13].

3.4. Adapted C2AE

The C2AE model is based on the idea of class conditioning in the latent space of an autoencoder. The autoencoder is trained to reconstruct an example correctly when it is conditioned with its ground truth label, and incorrectly otherwise. Originally, the incorrect reconstruction means reconstructing the example into one from a different class, e.g. a *metro* into *office*.

The original C2AE training procedure begins with the encoder and classifier being trained jointly for the task of a closed set classification on some K classes. Then, the encoder (frozen) and decoder are trained for the task of reconstruction. When the training is finished, the reconstruction errors from correct and incorrect conditioning are collected for the training set examples. Based on both types of errors, the threshold is found to differentiate between good and bad reconstruction. The algorithm of finding such threshold is described in Section 3.2.3 of the original paper [15]. During the inference, a prediction is obtained from a classifier and the K reconstruction errors are obtained by conditioning a new example with each of the known classes. If the minimum of reconstruction errors is lower than a previously computed threshold, the prediction is adopted. Otherwise, it is rejected and the example is predicted as an unknown.

In order to utilize our base classifier in C2AE, we decided to introduce the Adapted C2AE. In terms of training, we separately trained a classifier and the entire autoencoder, so a classifier does not need to be crafted specifically for this task (i.e. include an encoder). Based on the results in the original paper, we also decided to choose the threshold manually instead of automatically, to simplify the implementation and because

the gain from choosing it automatically was not substantial (see Figure 4 in [15]).

In terms of the inference, we decided to condition the autoencoder with the output of a classifier and not all labels. This change was not significant, if the example considered in the inference was either an unknown or indeed from the predicted class. However, if the example was not from the predicted class, but still from some other known class, the minimum of the reconstruction errors would be small and the original C2AE would misclassify the example as an incorrect known class. Our inference would raise the unknown class in such case. From the practical point of view, the classification would still be incorrect in both cases, and therefore we decided to change the conditioning as it resulted in a faster inference procedure.

We also redefined what the incorrect reconstruction means by reconstructing the example into silence (zeros), instead of an example from another class, when conditioned with a wrong label. By choosing a constant, we aimed to stabilize the training.

4. Experiment Setup

4.1. Data and Processing

In the experiments, we used the *TAU Urban Acoustic Scenes 2019 Open set development dataset* [4] (*TAU Open Set*) that was also used in DCASE 2019 Task 1C. The organizers did not provide the labels for the leaderboard and the evaluation datasets after the contest ended. Therefore, for our experiments, we use the official training and testing subsets of the development dataset. We additionally separated 10% of examples for tuning purposes, maintaining the distribution of classes. The dataset consisted of 10 known classes (used in (C1) and (C2) training) and 1 unknown class (used only in (C2)). We did not augment the dataset.

Each example was a mono recording with 48 kHz sampling rate and of 10 seconds' length. We processed each example into a logarithmically scaled mel-spectrogram with 256 bins using Short Time Fourier Transform with a window of size 2048 samples and 512 samples overlap. Moreover, for each bin, we calculate its mean and standard deviation across all examples, and we standardize each bin with those values. This results in the output of size 862×256 , the former being a time dimension, and the latter a frequency dimension.

4.2. Deep Learning Models

In the experiment, the two neural networks were trained: a classifier as a base model, and an autoencoder for Adapted C2AE inference. For models training and inference implementations we used TensorFlow [24]. We implemented Openmax based on the code provided in [25].

4.2.1. Classifier

We decided to use a small model (70k parameters) with a straightforward training procedure to allow for an easier reproduction of the obtained results. The model has already proved successful in ASC [8, 26]. The classifier architecture is described in Table 1. The classifier outputs a vector of length $K = 10$ for (C1) and $K = 11$ for (C2).

The model utilizes the Batch Normalization technique [27]. Both (C1) and (C2) were trained using GPU for 100 epochs with a mini batch of size 32, using categorical cross entropy loss. As the final model we used the best one in terms of loss

Table 1: *The architecture of the classifier.*

| Layer | Outputs | Kernel | Stride |
|-----------------------|---------|--------|--------|
| <i>Conv2D+ReLU+BN</i> | 16 | 3 | 1 |
| <i>Conv2D+ReLU+BN</i> | 32 | 3 | 2 |
| <i>Conv2D+ReLU+BN</i> | 32 | 3 | 1 |
| <i>Conv2D+ReLU+BN</i> | 64 | 3 | 2 |
| <i>Conv2D+ReLU+BN</i> | 64 | 3 | 1 |
| <i>AveragePooling</i> | 64 | - | - |
| <i>Dense+Softmax</i> | K | - | - |

on the validation dataset. Data was shuffled before each epoch. We used the *Adam* optimizer [28] with $\alpha = 0.001$.

4.2.2. Autoencoder

In the autoencoder architecture, we decided to use convolutional layers, which proved useful in the tasks related to acoustic scenes, as well as dense layers, to guarantee the data compression in the latent space. As a conditioning layer, we used the FiLM layer [29], as proposed in the original C2AE paper [15], which is the following linear transformation on the latent space

$$o = \mathcal{H}_\alpha(y) \odot z + \mathcal{H}_\beta(y) \quad (2)$$

where \mathcal{H}_α and \mathcal{H}_β are dense layers with the output size matching that of the latent space, z is the latent space representation of a given example and y is the one-hot encoded label of that example (1 is used for positive and -1 for negative class). The \odot operation is the Hadamard product. A full architecture is shown in Table 2.

Table 2: *The architecture of the autoencoder.*

| Layer | Outputs |
|------------------------------|---------|
| <i>Conv2D+ReLU*</i> | 16 |
| <i>Conv2D+ReLU*</i> | 8 |
| <i>Conv2D+ReLU*</i> | 4 |
| <i>Dense</i> | 512 |
| <i>Dense</i> | 128 |
| <i>FiLM Layer</i> | 128 |
| <i>Dense</i> | 128 |
| <i>Dense</i> | 512 |
| <i>Dense</i> | 1224 |
| <i>Conv2DTranspose+ReLU*</i> | 4 |
| <i>Conv2DTranspose+ReLU*</i> | 8 |
| <i>Conv2DTranspose+ReLU*</i> | 16 |

* With kernel and stride 3.

Similarly to the classifier, the autoencoder was trained using a single GPU for 100 epochs with a mini batch of size 32. For each example in one batch, both reconstructions (conditioned correctly and incorrectly) were performed. As a loss for those two tasks we used a weighed sum of Mean Square Errors with weight 0.8 for the correct reconstruction and 0.2 for the incorrect one. As the final model we used the best one in terms of the loss on the validation dataset. The dataset was shuffled before each epoch, and we used *Adam* optimizer with $\alpha = 0.001$. The autoencoder had approximately 1.4M parameters. In the autoencoder fitting procedure, the final threshold was found manually and set to 0.3. The reconstruction errors were calculated using Mean Absolute Error.

5. Results

5.1. Validation Metrics

As a first validation metric, we used the metric from DCASE 2019 Task 1C challenge which was a weighed sum of known and unknown accuracy, denoted ACC_K and ACC_U , respectively. In order to calculate the score, first, the accuracy for each class, including unknown, was computed. Then, as ACC_K , we took an average of per-class accuracies of the known classes and as ACC_U we took the accuracy for the unknown class. The final score was computed per the formula

$$ACC = 0.5 \cdot ACC_K + 0.5 \cdot ACC_U. \quad (3)$$

For calculation we used the toolbox provided by the DCASE 2019 organizers [30]. However, we are aware of the disadvantages of this metric, such as an arbitrary choice of weights and susceptibility to fluctuations, and therefore we decided to use Area Under ROC (AUROC) metric as well.

5.2. Discussion

Table 3: DCASE 2019 score on TAU Open Set test set.

| Model | ACC_K | ACC_U | ACC |
|-------------------------------|---------|---------|-------|
| (C1) closed set | 71.5% | - | - |
| (C1) with $\varepsilon = 0.5$ | 63.8% | 35.9% | 49.9% |
| (C1) with $\varepsilon = 0.6$ | 55.3% | 56.5% | 55.9% |
| (C1) with $\varepsilon = 0.7$ | 45.8% | 76.2% | 61.0% |
| (C1) with Openmax | 46.7% | 60.9% | 53.8% |
| (C1) with Adapted C2AE | 60.2% | 70.4% | 65.3% |
| (C2) with $\varepsilon = 0.5$ | 65.9% | 33.3% | 49.6% |
| (C2) with $\varepsilon = 0.6$ | 57.3% | 49.3% | 53.3% |
| (C2) with $\varepsilon = 0.7$ | 48.8% | 66.4% | 57.6% |
| (C2) with Openmax | 38.9% | 74.8% | 56.8% |
| (C2) with Adapted C2AE | 59.2% | 72.8% | 66.0% |

The ACC results are shown in Table 3. The most surprising is that the (C2) methods, that utilize examples of unknowns, do not provide improvements over their (C1) counterparts. All approaches experience reduced accuracy on the known classes. This is expected due to the possibility of mistaking examples for unknowns. Openmax, despite its complexity, has lower accuracy than thresholding for unknowns at the same accuracy for known classes. The best results were reached using the Adapted C2AE, with a relatively small deterioration in known accuracy in relation to the gain obtained in the unknown classification.

Moreover, in Figure 1, we present a comparison of known and unknown distributions' separation. Note that the x axis is different in both subplots, because the threshold and separation is based on the predicted class probabilities in the softmax thresholding method, while in C2AE it is based on the reconstruction errors. It can be observed, that the intersection of knowns and unknowns in C2AE is smaller than the one based on probability, which may imply that the separation based on reconstruction error is more stable and produces better differentiation between known and unknown examples.

Because there was no substantial gain in the accuracy between (C2) and (C1), we calculated the AUROC scores only for (C1) training variant. AUROC was computed for unknown-vs-known classification by changing the thresholds. The results are shown in Table 4.

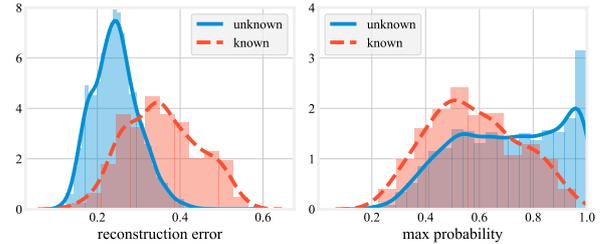


Figure 1: The comparison of known and unknown distributions separation for C2AE (left) and thresholding on softmax (right) on TAU Open Set test set.

Table 4: AUROC metric for (C1) on TAU Open Set test set.

| Model | Thresholds | Openmax | Adapted C2AE |
|-------|------------|---------|--------------|
| AUROC | 68.7% | 71.1% | 85.5% |

From AUROC we can conclude that the Openmax works better than a simple thresholding technique. The initial advantage of thresholds over the Openmax in terms of ACC could be the result of ACC favouring better score on ACC_U , which can be artificially increased by manipulating the threshold ε value. From the qualitative point of view, implementing and understanding the Openmax and its EVT grounds requires more time and expert knowledge than a straightforward thresholding method, and due to the negligible performance difference between them, thresholds could be a simpler, yet still effective choice. However, overall, the performance of Adapted C2AE is still much better than of the previous two, which confirms the results obtained through ACC.

There are multiple ways how the experiments proposed in this paper could be extended. In terms of test data, it could be beneficial to run similar experiments on a completely different dataset that partially includes the same labels, such as LITIS Rouen [31]. This would require applying additional techniques for classification with mismatched recording devices, because different devices could lower the classifier performance and distort the obtained results. Moreover, the (C2) classifier could be extended to use multiple unknown classes instead of just one, aggregating examples with similar category, for example *unknown transport*, *unknown indoor*, etc. Both classifiers could also be extended to a multi-label instead of a single-label classification. Finally, in this paper we evaluated discriminative methods, because we wanted to compare them in a common training framework, but it could be beneficial to extend this framework to generative methods as well (e.g. the G-Openmax [32]).

6. Summary

In this paper, we presented the results for open set ASC using three techniques. The first one was applying thresholds to the softmax output of a neural network classifier, which is the classic approach used so far in ASC. The second one, called Openmax, was a popular method derived from the CV field. The third one, we proposed the Adapted C2AE, which is a modified version of C2AE, where we applied changes allowing a more fair comparison with the other methods. Adapted C2AE outperformed both thresholding and Openmax, which shows that the autoencoder based solutions may be promising in the open set ASC.

7. References

- [1] Audio Analytic, “The 2019 Hearables survey. AI attitudes and expectations,” 2019.
- [2] S. Mcadams, “Recognition of sound sources and events,” in *Thinking in Sound. The Cognitive Psychology of Human Audition*. Oxford University Press, 4 1993, pp. 146–198. [Online]. Available: <https://doi.org/10.1093/acprof:oso/9780198522577.003.0006>
- [3] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boulton, “Toward open set recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.
- [4] 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)*, 2018, pp. 9–13.
- [5] J. Abeßer, “A review of deep learning based methods for acoustic scene classification,” *Applied Sciences*, vol. 10, no. 6, p. 2020, 3 2020. [Online]. Available: <https://doi.org/10.3390/app10062020>
- [6] S. Gharib, H. Derrar, D. Niizumi, T. Senttula, J. Tommola, T. Heittola, T. Virtanen, and H. Huttunen, “ACOUSTIC SCENE CLASSIFICATION: A COMPETITION REVIEW,” in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, Sep. 2018. [Online]. Available: <https://doi.org/10.1109/mlsp.2018.8517000>
- [7] A. Dhillon and G. K. Verma, “Convolutional neural network: a review of models, methodologies and applications to object detection,” *Progress in Artificial Intelligence*, 2019.
- [8] M. Kośmider, “Calibrating neural networks for secondary recording devices,” Samsung R&D Institute Poland, Tech. Rep., 2019.
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [10] W. Scheirer, L. Jain, and T. Boulton, “Probability models for open set recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, pp. 2317–2324, 11 2014.
- [11] L. P. Jain, W. J. Scheirer, and T. E. Boulton, “Multi-class Open Set Recognition Using Probability of Inclusion,” in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 393–409.
- [12] H. Zhang and V. M. Patel, “Sparse Representation-Based Open Set Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1690–1696, Aug. 2017, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [13] A. Bendale and T. E. Boulton, “Towards open set deep networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1563–1572.
- [14] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, “Classification-reconstruction learning for open-set recognition,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2019. [Online]. Available: <https://doi.org/10.1109/cvpr.2019.00414>
- [15] P. Oza and V. M. Patel, “C2AE: Class conditioned auto-encoder for open-set recognition,” 2019.
- [16] C. Geng, S.-J. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [17] H. Zhu, C. Ren, J. Wang, L. Yang, S. Li, and L. Wang, “Dcase 2019 challenge task1 technical report,” Samsung Research China-Beijing, Beijing University of Posts and Telecommunications, Tech. Rep., 2019.
- [18] A. Rakowski and M. Kośmider, “Frequency-aware CNN for open set acoustic scene classification,” Samsung R&D Poland, Tech. Rep., 2019.
- [19] B. Lehner, K. Koutini, C. Schwarzlmuller, T. Gallien, and G. Widmer, “Acoustic scene classification with reject option based on resnets,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019.
- [20] M. D. McDonnell and W. Gao, “Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths,” University of South Australia, Tech. Rep., 2019.
- [21] D. Battaglino, L. Lepauloux, and N. Evans, “The open-set problem in acoustic scene classification,” in *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, Sep. 2016. [Online]. Available: <https://doi.org/10.1109/iwaenc.2016.7602939>
- [22] K. Wilkinghoff and F. Kurth, “Open-set acoustic scene classification with deep convolutional autoencoders,” Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Tech. Rep., 2019.
- [23] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations*, 2015.
- [24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [25] A. Bendale and T. Boulton, “OSDN,” 2016. [Online]. Available: <https://github.com/abhijitbendale/OSDN>
- [26] T. Nguyen, F. Pernkopf, and M. Kośmider, “Acoustic scene classification for mismatched recording devices using heated-up softmax and spectrum correction,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5 2020, pp. 126–130. [Online]. Available: <https://doi.org/10.1109/icassp40776.2020.9053582>
- [27] 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 - Volume 37*, 2015, p. 448–456.
- [28] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 2014.
- [29] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, “Film: Visual reasoning with a general conditioning layer,” in *AAAI*, 2018.
- [30] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.
- [31] A. Rakotomamonjy and G. Gasso, “Histogram of gradients of time-frequency representations for audio scene detection,” 2015.
- [32] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, “Generative open-max for multi-class open set classification,” 2017.