OPEN-SET ACOUSTIC SCENE CLASSIFICATION WITH DEEP CONVOLUTIONAL AUTOENCODERS

Kevin Wilkinghoff, Frank Kurth

Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Fraunhoferstraße 20, 53343 Wachtberg, Germany
kevin.wilkinghoff@fkie.fraunhofer.de, frank.kurth@fkie.fraunhofer.de

ABSTRACT

Acoustic scene classification is the task of determining the environment in which a given audio file has been recorded. If it is a priori not known whether all possible environments that may be encountered during test time are also known when training the system, the task is referred to as open-set classification. This paper contains a description of an open-set acoustic scene classification system submitted to task 1C of the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2019. Our system consists of a combination of convolutional neural networks for closed-set identification and deep convolutional autoencoders for outlier detection. On the evaluation dataset of the challenge, our proposed system significantly outperforms the baseline system and improves the score from 0.476 to 0.621. Moreover, our submitted system ranked 3rd among all teams in task 1C.

Index Terms— acoustic scene classification, deep convolutional autoencoder, open-set classification, outlier detection

1. INTRODUCTION

Acoustic scene classification is a subfield of machine listening, where systems need to determine the environment in which given audio files were recorded, and has always been an integral part of the DCASE challenge [1, 2]. Additionally, there is growing interest in open-set classification [3, 4] within the machine learning community since realistic scenarios and applications are almost always open-set problems. The reason is that one can only very rarely capture the entire space of classes when training a classification system. The only potential exception is a very artificial setup that ensures no encounters of data belonging to novel or unknown classes when running the system after training. But since change and evolution in general are inevitable this setup seems very unlikely, especially in real world applications. However, open-set classification is much more difficult than closed-set classification because one also needs to determine whether data belongs to one of the known classes or not (outlier detection [5]), which is an a priori assumption in closed-set classification. This difficulty is probably the reason why most research has been focused on closed-set classification.

To promote this research direction, in this year's edition of the DCASE challenge there is a subtask of the acoustic scene classification task entirely focusing on the open-set setting (task 1C) [6], which will also be the focus of this paper. The development dataset consists of 44 hours of 48kHz audio belonging to some unknown and ten known classes, namely airports, indoor shopping malls, metro stations, pedestrian streets, public squares, streets with medium level of traffic, traveling by a tram, traveling by a bus, traveling by an underground metro and urban parks. The evaluation dataset consists of 20 hours of audio. For all recordings the same recording device has been used (unlike to subtask 1B where four different devices have been used) and all have a length of 10 seconds. To evaluate the performance of the systems, the final score is computed as the weighted average accuracy of the known classes and unknown classes. For more information about the task, see [6].

To our best knowledge, previous work for open-set acoustic scene classification is extremely limited. Still, there are some papers entirely focusing on that task as for example [7] where the authors used one-class support vector machines for open-set classification. Another way to detect outliers and thus make open-set classification possible is to use deep convolutional autoencoders (DCAEs) [8, 9]. By training DCAEs with data belonging to the known classes, one can expect that the neural networks learn to reconstruct this data well but have difficulties when encountering data belonging to unknown classes. In turn, the reconstruction loss can be used as a heuristic to detect outliers.

The contributions of this work are the following. First and foremost, a system for open-set acoustic scene classification is presented[1]. More specifically, we propose to use CNNs for closed-set classification and DCAEs for rejecting unknown acoustic scenes via outlier detection. As a last contribution, an effective way to combine a closed-set classification system and outlier detection models into a single open-set system is presented. It is also worth mentioning, that we did not use any external data resources nor pretrained models for training our system. Although this makes the open-set classification task even more challenging, it also enables us to precisely compare the performance of our system with other submitted systems that did not use external data resources.

2. ACOUSTIC SCENE CLASSIFICATION SYSTEM

As already stated, this paper focuses on open-set acoustic scene classification. But in order to do open-set classification one also needs a well working closed-set classification chain. The reason is that the system needs to 1) determine whether given data belongs to one of the known classes (outlier detection) and if so, 2) predict the most likely of the known classes (closed-set classification). Mathematically, this corresponds to estimating

$$\begin{aligned} &P(Y = y_i, K = \text{true}|X = x) \\ =&P(Y = y_i|K = \text{true}, X = x)P(K = \text{true}|X = x) \end{aligned} \quad (1)$$

where X and Y are random variables denoting the data and one of the known class labels, respectively, and K is a binary random

---

[1]An open-source Python implementation of the presented system is available here: https://github.com/wilkinghoff/dcase2019

variable indicating that the data belongs to one of the known classes (see [10]). Thus, open-set classification (left hand side) can indeed be decomposed into the subtasks closed-set classification and outlier detection (right hand side).

We will now present our feature extraction procedure followed by descriptions of the closed-set classification and outlier detection systems. This section is then concluded by a description of how to combine both systems into a single open-set acoustic scene classification system.

2.1. Feature extraction

Almost all recently proposed acoustic scene classification systems as well as the baseline system utilize log-mel spectrograms as input features (see e.g. [2, 11, 12]). As this is the state-of-the-art, we also used log-mel spectrograms and closely followed [13] for the parameter settings with a few changes. More precisely, we also used a Hanning window size of 1024, a hop size of 500 and 64 mel bins but used the cutoff frequencies 50Hz and 16000Hz. Additionally, we normalized the audio files with respect to the maximum norm before extracting the features. The resulting features are of dimension 64×442.

Furthermore, we utilized median filtering for Harmonic-Percussive Source Separation [14] via Librosa [15] as many participants have done in past editions of the DCASE challenge (see e.g. [11, 16]). All mel-spectrograms were separated into harmonic and percussive parts before applying the logarithm resulting in a total number of three features per audio file: The log-mel spectrograms themselves and their harmonic and percussive parts.

Before inserting the features into a neural network, we standardized them in two different ways. For closed-set classification, we subtracted the mean and divided by the standard deviation of all training data, which belongs to any of the ten known classes. When detecting outliers, all features were standardized in the same way but only data belonging to a single known class was used to compute the mean and standard deviation. As we will train individual DCAEs for each class, the data is standardized with respect to that specific class beforehand.

2.2. Closed-set classification

In the era of deep learning, CNNs are the method of choice to classify log-mel spectrograms. Classifying acoustic scenes is not an exception. The CNN proposed in [13] is reported to perform better than the baseline system of the challenge. Thus, we used this CNN as a starting point but changed a few details leading to an even better performance while using less parameters. All CNNs have been implemented using Keras [17] with Tensorflow [18] and their structure can be found in Table 1. For each of the three features, namely log-mel spectrograms and their harmonic and percussive parts, another CNN is trained for 6000 epochs with a batch size of 32 by minimizing the categorical crossentropy. Mixup [19] and Cutout [20] have been used to augment the training data, which are known to be effective in terms of improving classification accuracy (see [12]). Additionally, random shifts in time up to 60% of the entire duration and up to 3 mel bins were used when augmenting data. To acquire a single score per class, the geometric mean of the output distributions obtained with the three CNNs is taken. But since the classification accuracy obtained with the log-mel spectrograms were higher on the validation set, their corresponding scores have been multiplied with a factor of two to give them more weight than

Table 1: CNN architecture for closed-set classification.

Layer	Output Shape	#Parameters
Input	(64, 442)	0
Convolution (kernel size: 3x3)	(64, 442, 64)	640
Batch Normalization	(64, 442, 64)	256
Non-linearity (ReLU)	(64, 442, 64)	0
Convolution (kernel size: 3x3)	(64, 442, 64)	36,928
Batch Normalization	(64, 442, 64)	256
Non-linearity (ReLU)	(64, 442, 64)	0
Average-Pooling (pool size: 2x3)	(32, 147, 64)	0
Convolution (kernel size: 3x3)	(32, 147, 128)	73,856
Batch Normalization	(32, 147, 128)	512
Non-linearity (ReLU)	(32, 147, 128)	0
Convolution (kernel size: 3x3)	(32, 147, 128)	147,584
Batch Normalization	(32, 147, 128)	512
Non-linearity (ReLU)	(32, 147, 128)	0
Average-Pooling (pool size: 2x3)	(16, 49, 128)	0
Convolution (kernel size: 3x3)	(16, 49, 196)	225,988
Batch Normalization	(16, 49, 196)	784
Non-linearity (ReLU)	(16, 49, 196)	0
Convolution (kernel size: 3x3)	(16, 49, 196)	345,940
Batch Normalization	(16, 49, 196)	784
Non-linearity (ReLU)	(16, 49, 196)	0
Average-Pooling (pool size: 2x3)	(8, 16, 196)	0
Convolution (kernel size: 3x3)	(8, 16, 256)	451,840
Batch Normalization	(8, 16, 256)	1,024
Non-linearity (ReLU)	(8, 16, 256)	0
Convolution (kernel size: 3x3)	(8, 16, 256)	590,080
Batch Normalization	(8, 16, 256)	1,024
Non-linearity (ReLU)	(8, 16, 256)	0
Global-Average-Pooling	256	0
Dense (Softmax)	10	2,570
		$\sum$ 1,880,578

the scores resulting from the other two features. Using this heuristic enabled us to use the entire development set, i.e. training and validation split, for training the CNNs and led to better performance because more data results in more knowledge. Note that one can usually achieve better results when carefully tuning model-specific weights but this would have required additional labeled data to obtain meaningful scores for training these weights.

2.3. Outlier detection

To detect outliers, we used one-class classification models, more concretely DCAEs. For each of the known classes, another DCAE was trained using only training data belonging to that particular class. By doing so, we avoided the direct usage of training data belonging to any unknown class. The reason for doing this is that the variability of the unknown class space cannot be captured sufficiently by using samples of unknown classes. However, in order for the outlier detection models to learn to distinguish between strong outliers and weak outliers, which are noisy samples still belonging to the known class a DCAE is trained for, samples belonging to unknown classes are still needed. A way to use these samples for training will be explained later in subsection 2.4.

The particular structure we have chosen for the DCAEs can be found in Table 2. The basic task is to reduce the feature space dimension from 64×442 to 16×110 and reconstruct the input features as accurately as possible. More precisely, we trained another DCAE for each of the ten known classes resulting in a total of ten models per feature type. Again, we implemented the DCAEs with Keras [17] and Tensorflow [18]. To train the DCAEs, we minimized the mean squared error for 1000 epochs using a batch size of 32. In contrast to the CNNs, no data augmentation techniques were applied while training, which is the reason why less epochs are sufficient. We still trained different models for all three features but this time only the training data split of the development set has

Table 2: DCAE architecture for outlier detection.

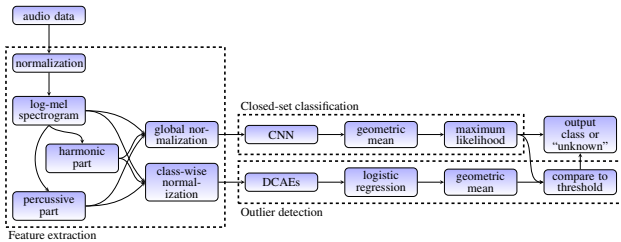| Layer	Output Shape	#Parameters
Input	(64, 442, 1)	0
Convolution (kernel size: 3x3)	(64, 442, 64)	640
Batch Normalization	(64, 442, 64)	256
Non-linearity (ReLU)	(64, 442, 64)	0
Convolution (kernel size: 3x3)	(64, 442, 64)	36,928
Batch Normalization	(64, 442, 64)	256
Non-linearity (ReLU)	(64, 442, 64)	0
Average-Pooling (pool size: 2x2)	(32, 221, 64)	0
Convolution (kernel size: 3x3)	(32, 221, 128)	73,856
Batch Normalization	(32, 221, 128)	512
Non-linearity (ReLU)	(32, 221, 128)	0
Convolution (kernel size: 3x3)	(32, 221, 128)	147,584
Batch Normalization	(32, 221, 128)	512
Non-linearity (ReLU)	(32, 221, 128)	0
Average-Pooling (pool size: 2x2)	(16, 110, 128)	0
Convolution (kernel size: 3x3)	(16, 110, 128)	147,584
Batch Normalization	(16, 110, 128)	512
Non-linearity (ReLU)	(16, 110, 128)	0
Convolution (kernel size: 3x3)	(16, 110, 128)	147,584
Batch Normalization	(16, 110, 128)	512
Non-linearity (ReLU)	(16, 110, 128)	0
Up-Sampling (size: 2x2)	(32, 220, 128)	0
Zero-Padding	(32, 221, 128)	0
Convolution (kernel size: 3x3)	(32, 221, 64)	73,792
Batch Normalization	(32, 221, 64)	256
Non-linearity (ReLU)	(32, 221, 64)	0
Convolution (kernel size: 3x3)	(32, 221, 64)	36,928
Batch Normalization	(32, 221, 64)	256
Non-linearity (ReLU)	(32, 221, 64)	0
Up-Sampling (size: 2x2)	(64, 442, 64)	0
Convolution (kernel size: 3x3)	(64, 442, 1)	577
Non-linearity (ReLU)	(64, 442, 1)	0
		$\sum$668,545

Figure 1: Structure of our proposed open-set acoustic scene classification system.

been used because the validation data set is needed in the next step. Note that using ReLU as an activation function in the last layer prevents the DCAE to perfectly reconstruct the data again since negative output values cannot be produced. But interesting events that are typical for an acoustic scene correspond to high energy in a mel-spectrogram and thus are still positive after normalization. An example are bird calls, which can only very rarely be heard in a metro but are one of the acoustic events one expects to hear in a park. Therefore, a DCAE trained on data belonging to the class "park" should be able to reconstruct bird calls but a DCAE associated with the class "park" should not, leading to a high reconstruction loss when encountering audio containing birds. In conclusion, the usage of ReLU can be seen as a form of regularization in this case and helped to improve the performance when detecting outliers.

2.4. Combined system

Since both subproblems, closed-set classification and outlier detection, have been tackled in some way, we can now determine the final output of the system. The only problem left is that, while the soft-

max output of the CNNs can be interpreted as a probability distribution, the loss of the DCAEs is just the mean squared error, which is not even bounded and scaled differently for each DCAE. Moreover, there is not only a single loss value per file but ten. Hence, it is highly non-trivial to find a suitable decision criterion when trying to detect outliers.

To solve this issue, we used logistic regression as implemented in Scikit-learn [21]. The idea is to treat the ten losses as ten dimensional features and train a binary classifier with them. For this purpose, we also made use of all audio files belonging to unknown classes. Although it is not a good idea to use these files or their spectral features directly for training a binary classifier, their losses should look much more close to each other (equally bad) than the outliers themselves. Hence, it may be a valid assumption to use them as valuable training data. In addition to that, the logistic regression model is very simple compared to all neural networks involved before. Thus, there is less room for the model to learn more than differentiating between losses corresponding to known classes and the "strange looking ones" belonging to unknown classes. In order to obtain meaningful positive examples of loss values belonging to known classes, we used the validation split of the development set. This is the only reason why the data files have not been used for training the DCAEs before.

To decide whether given data should be treated as an outlier, we used a threshold of 0.5 for all probabilities resulting from the logistic regression model. This means that for each encountered audio file, the class belonging to the maximum likelihood is chosen but if the score is smaller than 0.5, it is labeled as "unknown" instead. Choosing this particular threshold makes sense because the logistic regression model has been trained with balanced class weights to compensate for the different number of known and unknown training samples. In addition to that, we also labeled all audio files that had a maximum likelihood score less than 0.5 in the closed-set classification evaluation as "unknown". The underlying assumption is that most resulting scores are very high anyway and thus very small scores indicate that the model has difficulties in deciding which class the encountered data belongs to. This may indicate data belonging to unknown classes. See Figure 1 for an abstract overview of the entire system.

3. EXPERIMENTAL RESULTS

3.1. Closed-set classification

Closed-set classification is not the focus of this paper. Still, it is a vital part of any open-set classification system (see Equation 1). Therefore, we compared the performance of our closed-set classification system to those obtained with other systems. For this purpose, we used the datasets provided for subtask A of task 1. Using the datasets of subtask C for this purpose is impossible because the score also includes the system's outlier detection performance, which also affects the closed-set classification accuracy. The results can be found in Figure 2.

It can be seen that our closed-set classification accuracies are significantly higher than the ones obtained with the baseline system and with the system provided in [13]. Furthermore, our ensemble, which utilizes all three features types, performs significantly better than all models based on a single feature type. This justifies the final design of our closed-set classification system. We also included the winning system [22] to give an example of how much performance can be gained when improving the system.
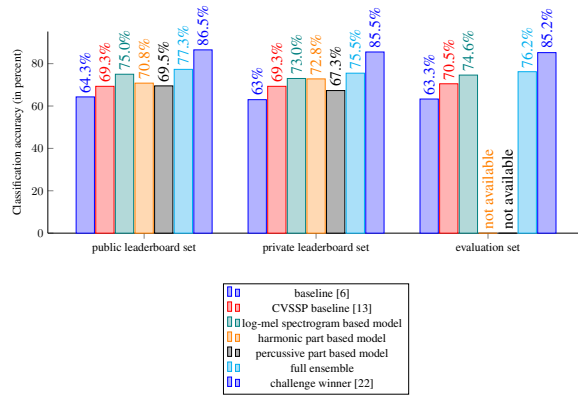
Figure 2: Comparison of closed-set classification accuracies obtained in task 1A.
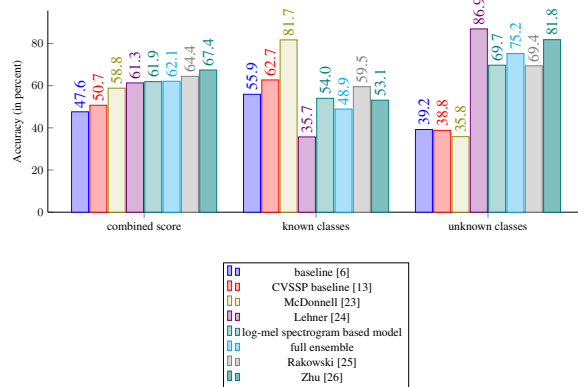

Figure 3: Comparison of all submitted systems' open-set classification scores obtained on the evaluation dataset of task 1C (final challenge results).

3.2. Open-set classification

The open-set classification performances on the evaluation dataset of task 1C obtained with all submitted systems including ours can be found in Figure 3. One can see that all systems outperform the baseline system as well as the system presented in [13] on the evaluation dataset. More concretely, the relative performance gain of our proposed system with respect to the score is 30.5% when comparing to the challenge's baseline system. Since these improvements are larger in this open-set setting than in task 1A, much of the success can be credited to using DCAEs for outlier detection. In contrast to the results obtained in the closed-set classification task, the accuracy of our ensemble significantly degraded for the known classes, which looks a bit strange at first sight. But since the accuracy significantly improved on the unknown classes, more test samples were predicted as outliers by the ensemble also resulting in more false rejections and a lower accuracy on the known classes.

When comparing all submitted systems, one can distinguish the three leftmost systems (baseline [6], CVSSP baseline [13] and McDonnell [23]), which have a relatively low accuracy on the unknown classes and thus are detecting only a few outliers, from the other systems. Because this also results in fewer false rejections, these three systems have a comparatively high accuracy on the known classes.

Compared to them, the other systems have a much higher accuracy on unknown classes and thus are detecting more outliers. However, since this also results in more false rejections, this degrades the performance on the known classes. It is worth pointing out that Lehner [24] has the highest accuracy on the unknown classes but the lowest accuracy on the known classes. In conclusion, this system detects too many outliers. The fact that our system ranks 3rd among all submitted systems shows that the overall structure of our open-set acoustic scene classification system is suitable for the task.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an open-set acoustic scene classification system that has been submitted to task 1C of the DCASE challenge 2019. It has been shown that a combination of CNNs for closed-set classification and DCAEs for outlier detection yields significant improvements over the baseline system. In fact, our system outperformed the baseline system by 30.5% without using any external data resources, increasing the score from 0.476 to 0.621 on the evaluation dataset. Using the presented system, our team ranked 3rd overall in task 1C of the challenge.

For future work, we plan to improve the structure of the DCAE. In addition, using the mean squared error of DCAEs for outlier detection is just a heuristic since the loss function to be optimized does not directly aim at rejecting unknown examples. Instead of using DCAEs, one may also train a neural network with another loss function that is specifically targeted at one-class classification (e.g. [27]). The results can also be compared to those obtained with an OpenMax layer [28], which can be understood as the open-set version of a softmax layer. Another path to be investigated is to make use of embeddings as for example the L3-Net embedding [29] or OpenL3 [30]. These embeddings could be used in the same way as i-vectors [31] or x-vectors [32] in open-set speaker identification (see e.g. [10]). Note that both, i-vector and x-vector, have been successfully applied for closed-set acoustic scene classification [33, 34] in past editions of the DCASE challenge. Thus, utilizing embeddings seems promising. Lastly, using external data for training the models or improving our relatively simple closed-set classification model with more sophisticated techniques as for example an attention mechanism [35] also improves the open-set performance.

5. REFERENCES

[1] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, Feb 2018.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification: An overview of DCASE 2017 challenge entries," in *16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, September 2018, pp. 411–415.

[3] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, July 2013.

[4] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.

[5] C. C. Aggarwal, *Outlier Analysis*, 2nd ed. Springer, 2017.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification in DCASE 2019 challenge: closed and open set classification and data mismatch setups," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2019.

[7] D. Battaglino, L. Lepauloux, and N. Evans, "The openset problem in acoustic scene classification," in *International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2016.

[8] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2002, pp. 170–180.

[9] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *International Conference on Data Mining*. SIAM, 2017, pp. 90–98.

[10] K. Wilkinghoff, "On open-set speaker recognition with i-vectors," *Preprint (submitted)*, 2019.

[11] Y. Han, J. Park, and K. Lee, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, pp. 56–50, 2017.

[12] S. Gharib, H. Derrar, D. Niizumi, T. Senttula, J. Tommola, T. Heittola, T. Virtanen, and H. Huttunen, "Acoustic scene classification: A competition review," in *28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 235–240.

[13] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.

[14] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *13th International Conference on Digital Audio Effects (DAFX)*, 2010.

[15] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *14th Python in Science Conference*, 2015, pp. 18–25.

[16] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge Report*, 2018.

[17] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[18] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[19] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR)*, 2018.

[20] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

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

[23] W. Gao and M. McDonnell, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," DCASE2019 Challenge, Tech. Rep., June 2019.

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

[25] A. Rakowski and M. Kośmider, "Frequency-aware CNN for open set acoustic scene classification," DCASE2019 Challenge, Tech. Rep., June 2019.

[26] H. Zhu, C. Ren, J. Wang, S. Li, L. Wang, and L. Yang, "DCASE 2019 challenge task1 technical report," DCASE2019 Challenge, Tech. Rep., June 2019.

[27] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning (ICML)*, 2018, pp. 4390–4399.

[28] A. Bendale and T. E. Boult, "Towards open set deep networks," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 1563–1572.

[29] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 609–617.

[30] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

[31] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[32] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[33] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," in *AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*. IEEE, 2016.

[34] H. Zeinali, L. Burget, and J. Cernocky, "Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge," pp. 202–206, 2018.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.