OPEN-SET ACOUSTIC SCENE CLASSIFICATION WITH DEEP CONVOLUTIONAL AUTOENCODERS

Technical Report

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. In evaluations conducted on the leaderboard dataset of the challenge, the proposed system significantly outperforms the baseline systems and improves the score by 35.4% from 0.46666 to 0.63166.

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 always open-set problems. The reason is that one can never 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 dataset consists of 46 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. 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 is to make use of 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, which makes the task even harder.

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

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

where X and Y are random variables denoting the data and class label, respectively, and K is a binary random variable indicating whether the data belongs to one of the known classes (see [10]).

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

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

To classify log-mel spectrograms, CNNs are the method of choice in the era of deep learning. 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 up to 60% in time 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 is higher, their corresponding scores have been used twice to give them more weight than the scores resulting from the other two features. We used the entire development set, training split and validation split, for training the CNNs as more data results in more knowledge and thus better performance.

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
		∑1,880,578

2.3. Outlier detection

It is well known, that the variability of the unknown class space cannot be captured sufficiently by using samples of unknown classes (see for example [10]). This is the reason why one should always prefer to train one-class classification models for the known classes. In conclusion, training data belonging to unknown classes is not needed to train the outlier detection system.

The particular structure we have chosen for the DCAEs can be found in Table 2. The basic task is to reduce the feature from a dimension of $(64, 442)$ to a dimension of $(16, 49)$ and reconstruct it as good as possible. To make it clear, we trained another DCAE for each of the ten known classes resulting in a total of ten models per feature. 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. This is also the reason why less epochs are sufficient for training. We still trained different models for all three features but this time only the training data split of the development set has been used because the validation data set is needed in the next step.

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 softmax output of the CNNs can be interpreted as a probability distribution,

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

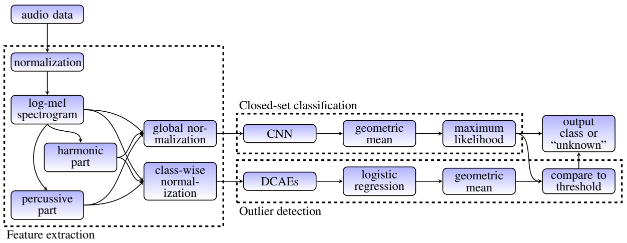

Figure 1: Structure of our proposed open-set acoustic scene classification system.

the loss of the DCAEs is just the mean squared error, which is not even bounded. 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, our method of choice is 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 directly for training a 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.

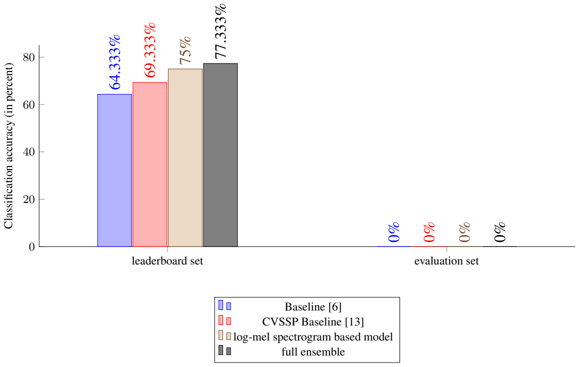

Figure 2: Comparison of closed-set classification accuracies obtained in task 1A.

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. 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

For all experiments carried out in this paper, we did not use any external data resources nor did we use any pretrained models, although this has been recommended by the organizers of the challenge to better capture the variability of the unknown class space.

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. Therefore, we compared the performance of our closed-set classification system to those obtained with other systems. For this purpose, we used the dataset provided for subtask A of task 1. Using the dataset of subtask C for this purpose is impossible because the score also includes the system's outlier detection performance. 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, performs significantly better than

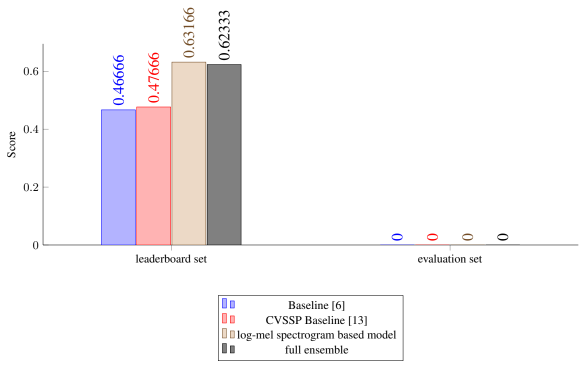

Figure 3: Comparison of open-set classification scores obtained in task 1C.

the single model. This justifies the final design of our closed-set classification system.

3.2. Open-set classification

The results of our open-set classification performance obtained with the dataset of task 1C can be found in Figure 3. It is immediately visible that our system massively outperforms the baseline system as well as the system presented in [13]. More concretely, the performance gain with respect to the score is 35.6% when comparing to the challenge's baseline system. Suprisingly, our ensemble performs slightly worse than our system based on only log-mel spectrograms. This could be caused by the relatively small number of leaderboard data. Since the improvements of our system over the baseline systems are much larger in this open-set setting than in task 1A, much of the success needs to be credited to using DCAEs for outlier detection. This shows that the overall structure of our open-set acoustic scene classification system is suitable for this 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 35.4% without using any external data resources, increasing the score from 0.46666 to 0.63166 on the leaderboard data.

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, possible future work may be to train a neural network with another loss function that is specifically targeted at one-class classification (e.g. [22]). The results can also be compared to those obtained with an OpenMax layer [23], 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 [24] or OpenL3 [25]. These embeddings could be used in the same way as i-vectors [26] or x-vectors [27] in open-set speaker recognition (see e.g. [10]). Note that both, i-vector and x-vector, have been successfully applied for closed-set acoustic scene classification [28, 29] in past editions of the DCASE challenge. Thus, utilizing embeddings seems promising. Lastly, improving our relatively simple closed-set classification model with more sophisticated data augmentation techniques and other methods 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," *IEEE/ACM 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 *2018 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," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 35, July 2013.

[4] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.

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

[6] 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 (DCASE)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[7] 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, 2016, pp. 1–5.

[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 *Proceedings of the International Conference on Data Mining*. SIAM, 2017, pp. 90–98.

[10] K. Wilkinghoff, "Training an open-set speaker recognition system without using non-blacklist speakers," *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. 1–5, 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 *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6.

[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 *Proceedings of the 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 andEvents (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," *arXiv preprint arXiv:1710.09412*, 2017.

[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] 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.

[23] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1563–1572.

[24] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 609–617.

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

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

[27] 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 *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[28] 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," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[29] H. Zeinali, L. Burget, and J. Cernocky, "Convolutional neural networks and x-vector embedding for dcase2018 acoustic scene classification challenge," *arXiv preprint arXiv:1810.04273*, 2018.