# ANOMALOUS SOUND DETECTION WITH LOOK, LISTEN, AND LEARN EMBEDDINGS

Technical Report

*Kevin Wilkinghoff*

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

## ABSTRACT

The goal of anomalous sound detection is to unsupervisedly train a system to distinguish normal from anomalous sounds that substantially differ from the normal sounds used for training. In this paper, a system based on Look, Listen, and Learn embeddings, which participated in task 2 “Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring” of the DCASE challenge 2020, is presented. The experimental results show that the presented system significantly outperforms the baseline system of the challenge both in detecting outliers and in recognizing the correct machine type or exact machine id. Moreover, it is shown that an ensemble consisting of the presented system and the baseline system performs even better than both of its components.

***Index Terms*—** anomalous sound detection, machine listening, deep audio embeddings, outlier detection

## 1. INTRODUCTION

Anomalous sound detection has many applications. Examples are detecting accidents in audio streams of road surveillance systems [1, 2], detecting screams or breaking glass as indicators of terror attacks in subway stations [3] or detecting mechanical failure in factories [4]. However, gathering anomalous data for training automatic systems is difficult because these events rarely occur and are very diverse. Thus, a system is trained unsupervisedly using normal data only and its task is to detect anomalous data that substantially differs from the training data. This task is known as outlier detection [5]. Among the models that are used for detecting anomalous sounds are one-class SVMs [6], convolutional neural networks as for example WaveNet [3] and many types of autoencoders [7] as autoencoders with a specific objective function [8, 9], autoencoders in combination with a bidirectional long short-term memory (BLSTM) [2] or denoising autoencoders with a BLSTM [10]. In [7], it has also been shown that enhancing sound quality by dereverberation and denoising before applying an ensemble of deep autoencoders is beneficial.

In this paper, the goal is to investigate the use of Look, Listen, and Learn ($L^3$-Net) embeddings [11, 12] for anomalous sound detection. For this purpose, experiments are conducted within task 2, titled “Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring”, of the DCASE challenge 2020 [13]. The dataset of the task is divided into a development set, an additional training set and an evaluation set. The development set consists of audio recordings from $4$ different machines for each machine type and is divided into a training set with around $1000$ normal samples per machine and a test set with $100$ to $200$ normal and anomalous sounds. Note that this test set is not allowed to be used for training the final system submitted to the challenge. This means that only normal samples are allowed to be used for training. The additional training set consists of audio recordings from $3$ different machines for each machine type with around $1000$ additional normal audio samples. These machines are different from the ones of the development set. The evaluation set consists of around $400$ samples for each machine present in the additional training set and contains normal as well as anomalous samples. In total, the dataset contains six different machine types, namely “fan”, “pump”, “slide rail”, “valve” from MIMII [4] and “toy-car”, “toy-conveyor” from ToyADMOS [14]. Each audio file has a length of 10s with a sampling rate of 16kHz.

The contributions of this paper are the following. First and foremost, an anomalous sound detection system based on look, listen, and learn embeddings is presented. Second, the system is compared to the baseline system of task 2 of the DCASE 2020 challenge. It is shown that the system based on $L^3$-Net embeddings performs significantly better when detecting anomalous sounds and when predicting the machine type or exact machine id of recorded sounds. As a third contribution, an ensemble of both system is proposed, which performs even better than both subsystems.

## 2. SYSTEM DESCRIPTION

### 2.1. Baseline system

The baseline system is based on an autoencoder trained on stacked frames of log-Mel spectrograms. After training another autoencoder for each machine type, the reconstruction loss belonging to the correct machine type is utilized as an anomaly score: a low loss corresponds to a normal machine sound and a high loss to an anomalous one.

As stated before, the input of the autoencoder are stacked frames of log-Mel spectrograms. More concretely, an audio file is converted into a log-Mel spectrograms using a frame size of 64ms, a hop size of $50\%$ and $128$ Mel bins. Then, for each frame its $P$ preceding and $P$ following frames are concatenated into a single vector. In all experiments, $P$ is set to 2 and thus the input dimension is $128 \cdot (2P + 1) = 640$.

The autoencoder consists of $4$ encoding layers with a dimension of $128$, a code layer of dimension $8$, $4$ decoding layers with a dimension of $128$ and an output layer of dimension $640$. In each layer but the output layer, a rectified linear unit (ReLU) is used as a nonlinearity and batch normalization [15] is applied. The network is trained for $100$ epochs with a batch size of $512$ using Adam [16] and is implemented via Keras [17] and Tensorflow [18].

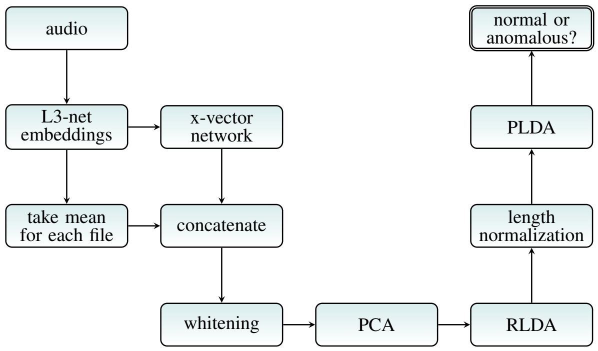


Figure 1: Processing chain of the L3-Net embeddings for obtaining decision scores.

### 2.2. Look, Listen, and Learn Embeddings

The basic idea of Look, Listen, and Learn ($L^3$-Net) embeddings [11, 12] is to detect audio-visual correspondence between a video frame and an audio clip of length 1s. The $L^3$-Net consists of two convolutional networks, an audio subnetwork and a video subnetwork, and a fusion subnetwork that concatenates the vector-sized outputs of the audio and video subnetwork (the embeddings) and predicts whether they belong together or not. The entire network can be trained unsupervisedly by using video frames and audio clips of the same video as positive samples and frames and clips of different videos as negative samples. Thus, the training data does not need to be labeled, which is a costly process. After training, audio embeddings can be obtained by only using the audio subnetwork. Throughout this paper, the term $L^3$-Net embeddings always refers to these audio embeddings. More details can be found in [11, 12].

To extract $L^3$-Net embeddings, the open-source implementation openL3 [19] pretrained on the music subset of AudioSet [20] has been used. The embeddings are extracted from log-Mel spectrograms with 256 Mel bins, which in turn are extracted from overlapping windows with a length of 1s and a hop size of 0.1s. For all experiments, an embedding size of 512 has been chosen and all embeddings have been normalized by subtracting the mean and dividing by the standard deviation of the embeddings belonging to the training split of the DCASE 2020 dataset.

### 2.3. X-vector based system

An x-vector model [21] is the state-of-the-art in speaker recognition. Its purpose is to extract speaker embeddings containing all relevant information about a speaker from audio data. To do this, Mel-frequency cepstral coefficients (MFCCs) [22] are computed and temporal convolutions are applied. Then, x-vectors are obtained by stacking the means and standard deviations of the convolutional output using so-called statistical pooling layers. More details can be found in [21].

In [23], it has been shown that an x-vector model can also be applied to open-set machine listening applications by using $L^3$-Net embeddings instead of MFCCs. Therefore, all $L^3$-Net embeddings belonging to a single audio file can be combined into a single x-vector that contains all relevant information. The x-vector network is trained for 100 epochs with a batch size of 32 using Adam [16] and is implemented via Keras [17] and Tensorflow [18]. When training the network, only manifold mixup [24] is used to augment the data. Basically, manifold mixup means to apply regular mixup [25] to the data representations of all layers and not just the input layer. To this end, mixup layers with mixing coefficients drawn from a uniform distribution have been used. Note that the original manifold mixup technique does only apply mixup at a single randomly chosen layer for each batch. Here, it is always applied at each mixup layer. The complete x-vector network structure can be found in Tab. 1.

Table 1: Architecture of the network for combining embeddings.

| Subnetwork | Layer | Output Shape |
|---|---|---|
| Preprocessing | Input | (T, 512) |
| | Mixup | (T, 512) |
| | Gaussian noise (standard deviation: 0.1) | (T, 512) |
| X-vector | 1D Convolution (kernel size=3, Leaky ReLU: 0.1) | (T, 256) |
| | Mixup | (T, 256) |
| | 1D Convolution (kernel size=3, Leaky ReLU: 0.1) | (T, 256) |
| | Mixup | (T, 256) |
| | 1D Convolution (kernel size=5, Leaky ReLU: 0.1) | (T, 256) |
| | Mixup | (T, 256) |
| | 1D Convolution (kernel size=1, Leaky ReLU: 0.1) | (T, 256) |
| | Mixup | (T, 256) |
| | 1D Convolution (kernel size=1, Leaky ReLU: 0.1) | (T, 512) |
| | Mixup | (T, 512) |
| | Mean | 512 |
| | Standard deviation | 512 |
| | Concatenation | 1024 |
| | Dense (Linear) | 256 |
| | Length normalization | 256 |
| Classifier | Gaussian noise (standard deviation: 0.1) | 256 |
| | Mixup | 256 |
| | Leaky ReLU: 0.1 | 256 |
| | Batch normalization | 256 |
| | Dropout (rate: 0.8) | 256 |
| | Mixup | 256 |
| | Dense (Leaky ReLU: 0.1) | 256 |
| | Batch normalization | 256 |
| | Dropout (rate: 0.5) | 256 |
| | Mixup | 256 |
| | Dense (Leaky ReLU: 0.1) | 128 |
| | Batch normalization | 128 |
| | Mixup | 128 |
| | Dense (Softmax) | #Classes |

Since x-vector networks are trained discriminatively and thus intra-class information may be lost, any resulting x-vector is concatenated with the mean of the embeddings this x-vector is derived from. It has been shown to significantly improve the outlier detection performance (see [23]). All x-vectors are further processed with a whitening operation, principal component analysis (PCA) as implemented in [26] and regularized linear discriminant analysis (RLDA) as used in [27] while not reducing the dimension. After length normalization, two-covariance probabilistic linear discriminant analysis (PLDA) [28, 29] as implemented in [30] is used to obtain decision scores. PLDA has the advantage that its output is a log-likelihood ratio comparing the likelihood of two x-vectors belonging to the same class to the likelihood that they do not belong to the same class. This is especially useful when detecting outliers because a fixed threshold can be used to mark machine sounds as anomalous whenever the log-likelihood ratio is below that threshold. It should be emphasized, that the x-vector network as well as the RLDA and PLDA models are trained to discriminate between the exact machine ids instead of the machine types. This led to significantly better performance and is another difference to the baseline system where only a single model is trained for all machine ids belonging to the same machine type. The whole processing chain of the $L^3$-Net embeddings is depicted in Figure 1. For more details, see [23].

Using an x-vector based system instead of autoencoders has many benefits: First and foremost, only one model instead of an additional model for each class needs to be trained. Furthermore, an x-vector based model is trained discriminatively and thus is designed to classify among the classes. Although it is still possible to classify with autoencoders by choosing the class that corresponds to the smallest loss, the performance is much worse (see Subsection 3.1). A third benefit is that the input data of the models is much smaller because for autoencoders multiple temporal sections of the spectrogram are stacked and thus the data size is artificially increased. Hence, evaluating the x-vector model is much faster. A possible downside of using an x-vector based model is that one needs to retrain the entire system when adding another class instead of training an additional autoencoder.

### 2.4. Ensembling strategy

Both models, the baseline model and the x-vector based model, are completely different and are even based on different features. Hence, it seems reasonable that both are making at least some independent errors and thus combining both into an ensemble can increase the performance significantly. An ensembling strategy using logistic regression as used in [31] is not possible because scores obtained with the test split of the development set are not allowed to be used for training. Instead, all relevant information of the subsystems are concatenated into a single vector before applying PCA, RLDA and PLDA as described in Subsection 2.3. More concretely, this concatenated vector is of the following form:

$$\begin{pmatrix} X := \mathrm{XV}\big((e_1, ..., e_K)\big) \\ \mu_{\text{emb}} := \frac{1}{K}\sum_{k=1}^{K} e_k \\ \mu_{\text{err}} := \frac{1}{T}\sum_{t=1}^{T}\big(\psi_t - \mathrm{AE}(\psi_t)\big)^2 \end{pmatrix} \in \mathbb{R}^{V+S+D(2P+1)} \qquad (1)$$

where $(e_k)_{k=1,...,K} \subset \mathbb{R}^S$ denote the embeddings belonging to one audio file, $(\psi_t)_{t=1,...,T} \subset \mathbb{R}^{D(2P+1)}$ denote all stacked $(2P+1)$ consecutive frames of a log-Mel spectrogram computed from that audio file, XV denotes the x-vector network and AE an autoencoder belonging to the correct machine type.

## 3. EXPERIMENTAL RESULTS

### 3.1. System performances on the development set

The results obtained on the development set are depicted in Fig. 2. One can immediately see that the x-vector based system significantly outperforms the baseline system. This is especially true for the machine type “Valve” where the AUC improves from $0.6515$ to $0.9565$. There is only one machine type, namely “ToyConveyor”, for which the baseline system yields better results than the x-vector based system.

Furthermore, the ensemble performs better than both of its components, even for the machine type “ToyConveyor” where the x-vector based system performed worse than the baseline system. This shows that both models, the baseline system and the x-vector based system, make at least some independent errors. Again, there is only one exception: The AUC for the machine type “Valve” decreases from $0.9565$ and $0.9841$ to $0.9155$ and $0.9675$, respectively. This is most likely caused by the relatively poor performance of the baseline system for this particular class.

Another observation to be made is, that using additional training data when training the ensemble improves the overall performance only for some machine types while slightly degrading the performance for other machine types (e.g. “ToyConveyor”). This may seem counterintuitive at first, but the machines contained in the additional training dataset do not match those contained in the training and test split of the development set. Note, that the machine types of the machines do match, only the machine ids are different. The systems trained with more data are expected to achieve better performances when encountering sounds of unknown machines of the six machines types. But for the specific machines of the development dataset adding the additional training data can be seen as adding noisy data and thus leads to worse results in some cases.

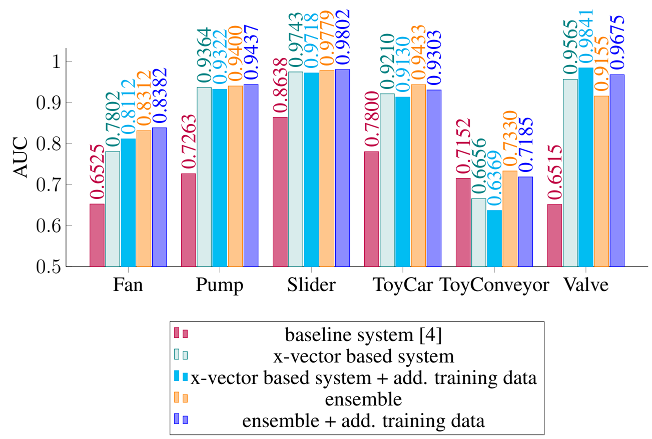


Figure 2: AUCs obtained on the development set with baseline system, x-vector based system and proposed ensemble.

### 3.2. Comparison of closed-set classification performance

A good closed-set classification performance is not necessary for detecting anomalous sounds. Still, perfect classification results are useful when operating with the system in practical applications because the user does not need to select which machine (type) is being recorded. This greatly simplifies handling the software or maintenance device. Moreover, it is also possible to obtain the machine type or even exact machine id from the recording without any additional costs. This is especially useful for non-experts who need further information about a machine or experts who need additional information about a specific machine as for example its production year or the date of the last maintenance check.

The closed-set classification accuracies when detecting the machine types can be found in Fig. 3. To evaluate the baseline system, the class corresponding to the autoencoder with the smallest loss has been chosen. As expected, the x-vector based system performs significantly better than the baseline system. The reason is that the x-vector based system is trained discriminatively whereas the baseline system is not. Furthermore, both systems have a higher classification accuracy with normal machine sounds than with anomalous sounds. More concretely, the x-vector based-system has a nearly perfect accuracy for normal sounds and an accuracy of about $90\%$ for anomalous sounds. Here, the reason is that recordings from fully functioning machines sound alike whereas different mechanical failures can alter the sounds in many different ways making it more difficult to recognize the correct machine type. Including training data of additional machines slightly improves the performance in case the machines are not present in the test set and significantly improves the performace when they are present. While the overall results look promising, one needs to keep in mind that there are only six different machine types present in these experiments. In realistic applications, more machine types and thus lower classification accuracies are to be expected.

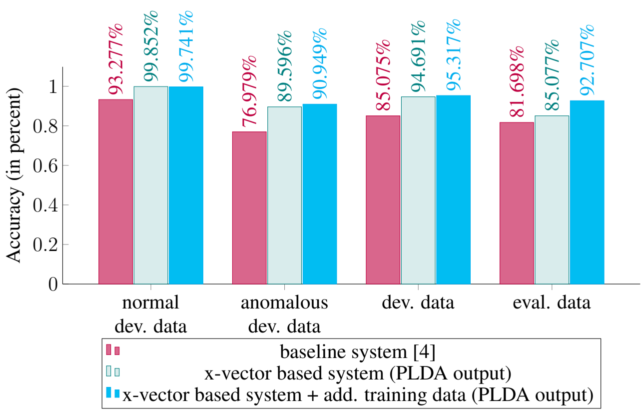


Figure 3: Closed-set classification accuracy by machine type.

The closed-set accuracies for detecting the exact machine ids are depicted in Fig. 4. When comparing the results to the ones obtained with the machine types, it is immediatly visible that the performace is worse because the task is inherently more difficult. Again, sounds belonging to normal machines are still recognized close to perfectly whereas the performance degrades even more when anomalous sounds are encountered for the same reason as stated above. Here, using additional training data not belonging to the machines present in the dataset slightly degrades the performance. This seams reasonable since knowing additional machines does not help to distinguish the machines one is interested in.

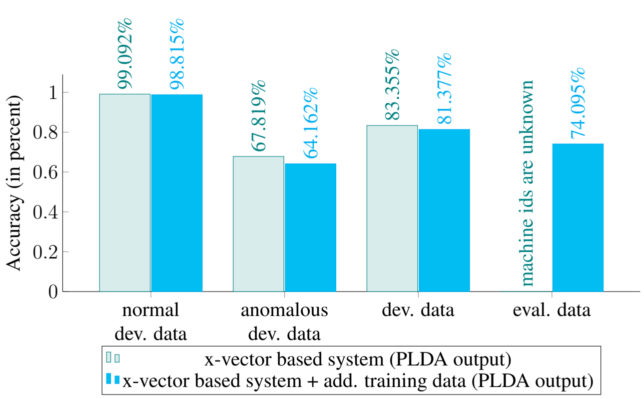


Figure 4: Closed-set classification accuracy by machine id.

### 3.3. Submitted systems

The four systems submitted to the challenge are presented in Table 2. With this particular choice, the performance of the ensemble can be compared to the performance obtained with the x-vector based system. Note that the baseline system will be evaluated by the organizers of the challenge and thus the resulting performance can be used for comparing the baseline system to the other systems. Furthermore, one can compare the influence of using additional training data (here the training split of the development set) belonging to machines that are not present in the evaluation set.

Table 2: Submitted systems.

| | system | trained with |
|---|---|---|
| 1) | x-vector based system | add. training set |
| 2) | x-vector based system | training split of dev. set + add. training set |
| 3) | ensemble | add. training set |
| 4) | ensemble | training split of dev. set + add. training set |

## 4. CONCLUSIONS AND FUTURE WORK

In this paper, an x-vector based system using $L^3$-Net embeddings for anomalous sound detection has been presented and evaluated in task 2 of the DCASE challenge 2020. It has been shown that the system significantly outperforms the baseline system when detecting anomalous sounds as well as when detecting the machine type or exact machine id a sound belongs to. Furthermore, an ensemble of the x-vector based system and the baseline system has been presented, which performs even better than both of its components.

In the future, it is planned to try other loss functions for the x-vector network that do not enforce a discriminative behaviour on the x-vectors. When detecting anomalous data, a discriminative structure is not needed and might even mask valuable information leading to worse performance [32]. Thus, another loss function and replacing RLDA with a non-discriminative technique as within-class covariance normalization (WCCN) [33, 34] may lead to better performance. Further improvements in terms of performance could be gained by training another autoencoder for each machine id instead of for each machine type. In addition to that, a more sophisticated autoencoder architecture than the baseline system as for example a convolutional autoencoder can possibly lead to improved results.

## 5. REFERENCES


[1] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, “Audio surveillance of roads: A system for detecting anomalous sounds,” *Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 279–288, 2015.

[2] Y. Li, X. Li, Y. Zhang, M. Liu, and W. Wang, “Anomalous sound detection using deep audio representation and a BLSTM network for audio surveillance of roads,” *IEEE Access*, vol. 6, pp. 58 043–58 055, 2018.

[3] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, “Anomalous sound event detection based on WaveNet,” in *26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2494–2498.

[4] 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 *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. New York University, 2019, pp. 209–213.

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

[6] F. Aurino, M. Folla, F. Gargiulo, V. Moscato, A. Picariello, and C. Sansone, “One-class SVM based approach for detecting anomalous audio events,” in *International Conference on Intelligent Networking and Collaborative Systems*. IEEE, 2014, pp. 145–151.

[7] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, “Anomaly detection based on an ensemble of dereverberation

and anomalous sound extraction," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 865–869.

[8] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing acoustic feature extractor for anomalous sound detection based on Neyman-Pearson lemma," in *25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 698–702.

[9] 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," *Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, 2018.

[10] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 1996–2000.

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

[12] ——, "Objects that sound," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 435–451.

[13] 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*, 2020. [Online]. Available: https://arxiv.org/abs/2006.05822

[14] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 313–317.

[15] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning (ICML)*, vol. 37, 2015, pp. 448–456.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[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] 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.

[20] 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 *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[22] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[23] K. Wilkinghoff, "On open-set classification with L3-Net embeddings for machine listening applications," in *28th European Signal Processing Conference (EUSIPCO)*, 2020.

[24] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *36th International Conference on Machine Learning (ICML)*, vol. 97. PMLR, 2019, pp. 6438–6447.

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

[26] 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.

[27] H. Zeinali, L. Burget, and J. Cernocky, "Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. Tampere University of Technology, 2018, pp. 202–206.

[28] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *11th International Conference on Computer Vision. ICCV*. IEEE, 2007, pp. 1–8.

[29] N. Brümmer and E. De Villiers, "The speaker partitioning problem." in *ODYSSEY - The Speaker and Language Recognition Workshop*, 2010, pp. 202–209.

[30] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Proc. S+SSPR*. Springer, 2014, pp. 464–475, software available at https://sites.google.com/site/fastplda/.

[31] K. Wilkinghoff and F. Kurth, "Open-set acoustic scene classification with deep convolutional autoencoders," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. New York University, 2019, pp. 258–262.

[32] K. Wilkinghoff, "On open-set speaker identification with i-vectors," in *The Speaker and Language Recognition Workshop (Odyssey)*. ISCA, 2020, pp. 408–414.

[33] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *9th International Conference on Spoken Language Processing (ICSLP)*. ISCA, 2006.

[34] A. O. Hatch and A. Stolcke, "Generalized linear kernels for one-versus-all classification: Application to speaker recognition," in *International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2006, pp. 585–588.