ACOUSTIC SCENE CLASSIFICATION BY COMBINING AUTOENCODER-BASED DIMENSIONALITY REDUCTION AND CONVOLUTIONAL NEURAL NETWORKS

Jakob Abeßer, Stylianos Ioannis Mimilakis, Robert Gräfe, Hanna Lukashevich

Fraunhofer IDMT, Ilmenau, Germany

ABSTRACT

Motivated by the recent success of deep learning techniques in various audio analysis tasks, this work presents a distributed sensor-server system for acoustic scene classification in urban environments based on deep convolutional neural networks (CNN). Stacked autoencoders are used to compress extracted spectrogram patches on the sensor side before being transmitted to and classified on the server side. In our experiments, we compare two state-of-the-art CNN architectures subject to their classification accuracy under the presence of environmental noise, the dimensionality reduction in the encoding stage, as well as a reduced number of filters in the convolution layers. Our results show that the best model configuration leads to a classification accuracy of 75% for 5 acoustic scenes. We furthermore discuss which confusions among particular classes can be ascribed to particular sound event types, which are present in multiple acoustic scene classes.

Index Terms— Acoustic Scene Classification, Convolutional Neural Networks, Stacked Denoising Autoencoder, Smart City Networks

1. INTRODUCTION

Particularly in urban environments, various acoustic scenes such as road traffic and railway transport, construction sites, open air concerts, or sport events often cause noise pollution and lead to resident complaints. Following the idea of a smart city network, a distributed system for intelligent acoustic analysis allows to objectively identify causes of noise pollution to the local city administration. A more effective processing of the incoming noise complaints allows to better plan future events in the residential area(s) of the city. As part of the *StadtLärm* [1] (City noise) research project, we first focus on identifying the acoustic scene that causes a potential noise exposure. In the given application scenario, the classification models furthermore need to be robust towards unwanted background noises such as wind and rain. Secondly, we aim to measure the exposure of citizens to noise in different parts of the city based on the German technical guidelines for noise reduction (*TA Lärm* [2]).

In this paper, we present a distributed system for automatic scene classification, which consists of two units: i) the acoustic sensor units with microphones placed around the city and ii) the central server application unit, where the audio scene analysis is performed. On the acoustic sensor side, non-negative time-frequency patches are extracted from a continuously audio input stream. Due to mobile communication bandwidth restrictions, we reduce the dimensionality of the aforementioned patches using a deep denoising auto-encoder (DAE) [3]. The encoded information is transmitted to the central server unit, where the patches are reconstructed using the decoder part of the DAE. The reconstructed time-frequency patches are then used for the acoustic scene classification. Due to project constraints, we focus on the five acoustic scene classes: i) music event, ii) sport event, iii) traffic, iv) roadworks, and v) public place. We compare two state-of-the-art systems based on Convolutional Neural Networks (CNN), recently proposed by Salamon and Bello [4] (**SB**) and Takahashi et al. [5] (**TAK**).

2. RELATED WORK

Several research projects investigated how to integrate intelligent audio analysis algorithms into smart city application scenarios. For instance, the LIFE+ project DYNAMAP focuses on noise measurement in road infrastructures [6], the EU FP7 EAR-IT project [7] investigated large-scale indoor and outdoor acoustic sensor networks, and the SONYC research project developed algorithms and devices for monitoring noise pollution in the urban environment of New York City[8, 4].

The application of Convolutional Neural Networks (CNN) led to state-of-the-art results in various image processing tasks. Consequently, CNNs were successfully adopted to audio recognition tasks such as speech recognition [9], music transcription [10], and environmental sound classification [11]. Most methods for acoustic scene classification apply CNNs to learn characteristic spectro-temporal patterns for different sound classes from the audio signal. Commonly used spectrogram representations are either perceptually or musically motivated. In [12] for example, Lidy and Schindler propose to apply the constant-Q transform as input to the network as it allows to analyze low and mid-low frequencies with a better time resolution compared to the commonly used mel-frequency spectrogram [4]. Similarly to image recognition tasks where the RGB channels of an image control the depth of the convolutional filters of the CNN, researchers in acoustic scene classification have proposed to incorporate additional features to that aim. More specifically, Piszak [11] has proposed to use the first-order derivative of the magnitude spectrogram as an additional depth dimension, while Takahashi et al. [5] proposed to use the first and the second order derivatives as input to the CNN.

The abovementioned methods vary regarding hyper-parameters such as the filter size, the stride of the pooling layers, the number of convolutional and fully-connected layers, as well as regularization techniques such as dropout or weight regularization. Takahashi et al. followed the idea of the VGG CNN architecture [13] and replaced larger convolution kernels (e.g. 5 x 5) by stacking pairs of layers with 3 x 3 kernels without intermediate pooling [5]. This approach leads to a reduction of the number of model parameters but at the same time to more expressive features due to the additional non-linearity. Lidy and Schindler proposed two parallel convolutional layers to separately capture relevant patterns in audio signals along frequency and time [12]. Other approaches, such as the two compared architectures **SB** and **TAK**, and the recently proposed stacked CNNs and recurrent neural networks (RNN) [14, 15] use

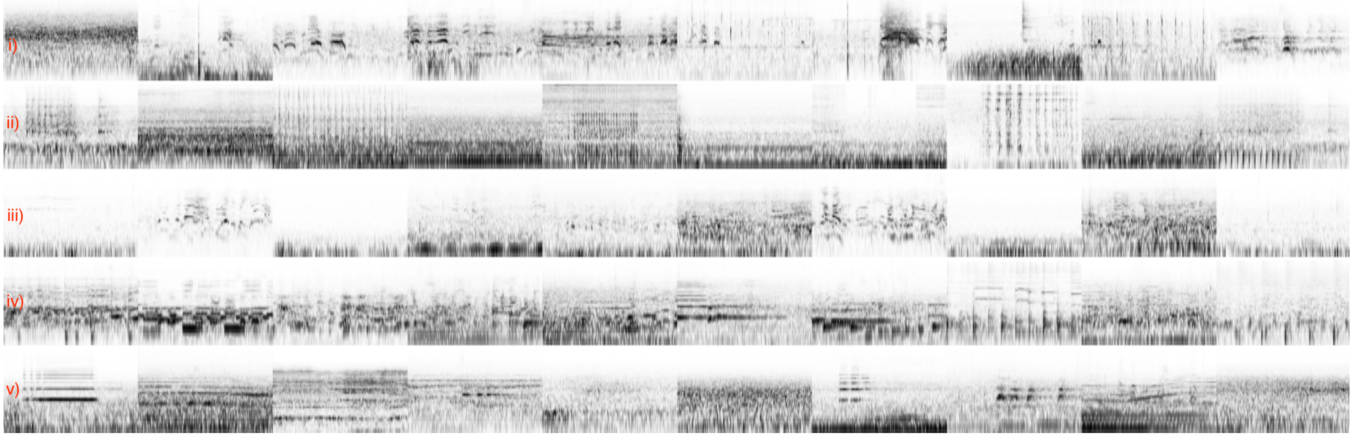

Figure 1: Examples of time-frequency patches of length three seconds, sorted row wise as follows: i) *sports event*, ii) *roadworks*, iii) *public place*, iv) *music event*, and v) *traffic*. The vertical and horizontal axes correspond to frequency and time, respectively.

successive layers instead. In [16], the authors combined deep denoising autoencoder architectures for feature learning in the context of acoustic event detection.

3. PROPOSED METHOD

3.1. System Overview

The proposed system architecture comprises of 25 distributed sensor units and one server unit as illustrated in Figure 3. On each sensor unit, the incoming audio stream is recorded at a sample rate of $f_s = 44.1\,\text{kHz}$ and processed with a short-time Fourier Transform (STFT) using a hop size of 882 (20 ms), a window size of 1024 (23.2 ms), and a zero-padding of factor 4. Then, the STFT magnitude spectrogram is logarithmically compressed and mapped to a logarithmically-spaced frequency axis of 49 bins (between 50 Hz and 15 kHz with a resolution of 6 bins per octave) using a triangular shaped filter-bank.

Spectrogram patches of size 49×50 (1 s duration) are reshaped to the dimensionality of 2450 and forwarded to the encoder part of the DAE described in Section 3.4. Then, the encoded patches are transmitted to the server unit where the decoder part of the DAE reconstructs the spectrogram patches and stores them in a buffer of size 3 seconds. Therefore, three consecutive patches are concatenated and forwarded to the classifier.

3.2. Dataset

Based on the given application scenario described in Section 1, we focus on the five acoustic scenes *sport event* (soccer games in stadium), *roadworks* (jackhammer, construction site), *public place* (conversations, walking), *music event* (busking, open air concerts), and *traffic* (car, train, tram). Therefore, we compiled a new dataset from the TUT Sound Events (real audio) 2016 development set [17], the Urban Sound Dataset [8], and the IEEE AASP public & private datasets [18], as well as various Youtube videos (particularly for soccer game recordings in the sport event class). Table 1 summarizes the number of files and total duration of files in hours for each class in the our dataset.

Class	Short name	# Files	Total Duration (h)
Sport event	SE	34	2.37
Roadworks	RW	35	1.29
Public place	PP	127	3.10
Music event	ME	72	3.67
Traffic	TR	97	1.56

Table 1: Compiled dataset—number of files and total duration in hours per class.

3.3. Data Augmentation

We apply a two-step data augmentation procedure to enrich our data set. Firstly, each audio file is processed using pitch shifting (± 1 semitone), time stretching (stretch factors of 0.93 and 1.07), and dynamic range compression using the *sox* library [19]. In addition to this “clean” version of the dataset, we created a second “noisy” version of the dataset by mixing each file with environmental background noise using a random signal to noise ratio (SNR) spanning between -14 and -10 dB. This was done in order to simulate the recording conditions in the targeted urban areas. For this purpose, we randomly select segments from five long-term recordings (total length of 135 min) of rain, thunderstorms, and wind (including microphone pop sounds), which were extracted from Youtube videos. Finally, for both datasets, we select 20 random excerpts of three second duration from each file. In total, the datasets each comprise of 43800 time frequency patches.

Figure 1 illustrates 10 randomly selected time-frequency patches for each of the acoustic scene classes. Sport event patches show both transient structures that result e. g. from hand claps as well as harmonic structures that are caused by screaming, speaking, and singing of fans and athletes. Patches from the roadworks class exhibit mostly repetitive structures from machine-like sounds such as drilling or jackhammer. Recordings from the public place class are more sparse with vehicle sounds (e. g. cars, motorbikes) in the background and often harmonic sounds (e. g. people talking, bird singing) in the foreground. The music event class shows clear harmonic structures that result from different musical instruments. Finally, in the traffic class, we observe noise-like structures

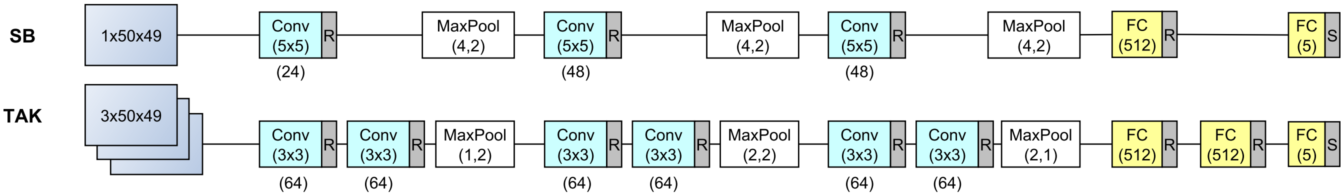

Figure 2: Flowchart of compared CNN architectures **SB** and **TAK**. Number of filters is given below the convolutional layers in brackets.

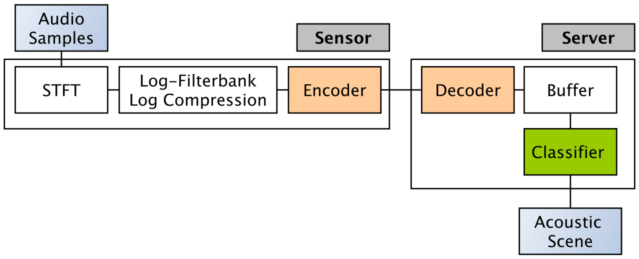

Figure 3: System architecture—distributed sensors and a central server applications.

from moving vehicles and harmonic structures from car honking (constant pitch) and sirens (continuously increasing and decreasing pitch).

3.4. Dimensionality Reduction via Denoising Autoencoders

Due to restrictions of the available mobile communication transmission bandwidth in the targeted area, the data packages from each sensor unit have to be reduced before being transmitted to the server unit. For that purpose, we train a deep neural network (DNN) that performs dimensionality reduction on the time-frequency patches, using a greedy layer-wise training process and the DAE as proposed by Vincent et al in [3]. Each layer is trained to denoise input features which are corrupted by masking noise. The masking noise is drawn from a zero mean and unit variance normal distribution and has a probability of 20 % to corrupt an input neuron. We iteratively train all layers for 30 epochs per layer using the *adadelta* algorithm [20]. The outcome of the above procedure is the trained DNN, denoted as DAE, consisting of an encoder and a decoder part. The encoder and the decoder parts incorporate 4 trained layers each. Through the encoding layers the dimensionality is linearly decreased until the last hidden layer produces the desired reduced dimensionality. On the other hand, the dimensionality in the decoder is increased accordingly such that its output matches the input data dimensionality. The encoder is encapsulated in the acoustic sensors unit, while the decoder is a part of the central server unit.

We compare four scenarios with different ratios of dimensionality reduction. The first scenario does not incorporate the DAE. This means that the non-negative time-frequency patches are transmitted to the server side directly. The second scenario assumes a dimensionality reduction by 25 % using the DAE denoted as DAE$^{0.75}$ in order to encode the time-frequency patches and transmit the encoded representation. Finally, the third and fourth scenarios employ the same idea but using they reduce the dimensionality by 50 % and 75 %, denoted as DAE$^{0.5}$ and DAE$^{0.25}$, respectively. In the future, we plan to test other image compression techniques such as JPEG or GIF or dictionary learning methods as alternatives for compressing the spectrogram representation.

3.5. Acoustic Scene Classification

As discussed in Section 1, we compare two model architectures **SB** and **TAK**, which are illustrated in Figure 2. Both models consist of multiple convolutional layers combined with maximum pooling layers, which learn suitable feature representations from the input time-frequency patches, and multiple feedforward neural networks for supervised classification. While **SB** has three layers consisting of convolutional filters of size 5×5, the **TAK** model has three layers of pairs of smaller convolutional filters of size 3×3. Concerning the max pooling, the **SB** employs larger downsampling over time than over frequency while the **TAK** model first performs pooling over frequency, then over time and frequency, and finally only over time. Another main difference is that while the **SB** model takes spectrogram patches as input, the **TAK** model also takes the first two time derivatives of the spectrogram as additional depth dimensions. In contrast to the original papers, we used a constant number of 64 filters per convolutional layer for the **TAK** model, and used 512 as the dimensionality of the fully connected layers in both models to have comparable parameter values. Apart from that, we adopt the hyper-parameter settings for both models from original papers.

For model training, we use 100 training epochs with early stopping, the *adam* algorithm [21] with a learning rate of 0.001, and a batch size of 200. All experiments were performed using the Keras python package [22]. For training and testing the **TAK** architecture, we concatenated the spectrogram patches in the dataset with their first-order and second-order derivatives as proposed in [5]. The final tensor X_{TAK} that contains the data is of the shape $X_{\text{TAK}} \in \mathbb{R}^{43800\times3\times150\times49}$. It is then split into training set (80 %), development set (10 %), and test set (10 %) based on unique source files. For the **SB** architecture, we only use the first depth dimension (magnitude spectrogram) leading to $X_{\text{SB}} \in \mathbb{R}^{43800\times1\times150\times49}$.

4. EVALUATION & RESULTS

4.1. Model Comparison

In the evaluation experiment, we compared several configurations of the **TAK** and **SB** models. Firstly, we investigate the influence of the number of filters in the convolutional layers (compare Figure 2). Here, we try the original number of filters (indicated by the fraction $\gamma = 1$), as well as 50 % ($\gamma = 0.5$) and 25 % ($\gamma = 0.25$) of the original number of filters. The corresponding models are indicated as TAK$^{\gamma}$ and SB$^{\gamma}$. Secondly, we investigate the models' performance on two datasets—with and without additional environmental noise (compare Section 3.3). Thirdly, we analyze the influence of the

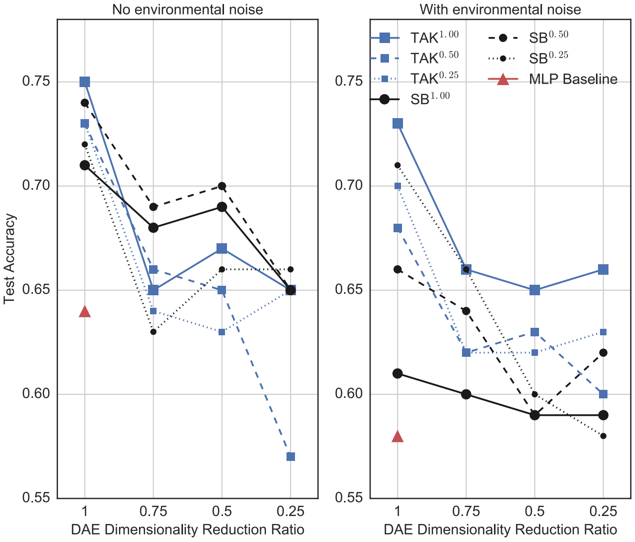

Figure 4: Evaluation results for **SB** and **TAK** models with and without environmental noise, different numbers of filters and different DAE dimensionality reduction ratios.

	ME	SE	TR	RW	PP
Music Event	**0.73**	0.02	0.02	0.09	0.14
Sport Event	0.12	**0.50**	0.07	0.00	0.31
Traffic	0.26	0.00	**0.71**	0.02	0.02
Roadworks	0.06	0.00	0.17	**0.74**	0.03
Public Place	0.01	0.22	0.28	0.02	**0.46**

Table 2: Confusion matrix for the model TAK$^{1.00}$ (no DAE dimensionality reduction, with environmental noise).

DAE dimensionality reduction step described in Section 3.4. We test the following dimensionality reduction ratios 1 (implying no dimensionality reduction), 0.75, 0.5, and 0.25, as described in Section 3.4. As a reference, we train and test the DCASE 2017 baseline system, which is based on a multilayer perceptron (MLP), using our given dataset [23]. Here, we do not apply the DAE-based compression. Figure 4 illustrates the test set accuracy values obtained with the different model configurations. For the dataset without environmental noise (left figure) and with additional environmental noise (right figure). Several observations can be made.

All CNN models (both TAK and SB) clearly outperform the baseline system for the case without DAE dimensionality reduction. If the DAE is used for data dimensionality reduction, we observe lower accuracy values for additional environmental noise, which is somewhat intuitive as the recognition task becomes harder. The **SB** model slightly outperforms the **TAK** model for the "clean" dataset without additional noise. In contrast, in case of additional noise, the **TAK** model with the full number of filters (TAK$^{1.00}$) shows the best performance thoughout all DAE dimensionality reduction ratios. Interestingly, in both noise settings, the **SB** model performs best with half the number of the originally proposed number filters [4] for our dataset (compare SB$^{0.50}$ vs. SB$^{1.00}$). In contrast, the **TAK** model shows the best performance for the full number of filters (TAK$^{1.00}$).

4.2. Class-wise Performance

In order to get further insights into the models' performance, we show as an example the confusion matrix obtained from the best-performing model TAK$^{1.00}$ without DAE dimensionality reduction, full number of filters, and with environmental noise in Table 2. It becomes apparent that the classes music event, traffic, and roadworks can be classified with good classification scores above 0.7 while the classes sport event and public place show significantly lower scores. As discussed already in Section 3.3, car honking, which is a prominent sound event in the traffic class, shows similar (horizontal) harmonic structures in the time-frequency patches as music instruments in the music event classes. This is confirmed by a confusion of 0.26 from traffic to music event patches. As both the public place and the sport event class include recordings of people speaking, we observe confusions of 0.22 and 0.31 between public place and sports event and vice versa. A third observation is the high confusion of 0.28 between public place and traffic, which is most likely due to passing vehicles in the background. Finally, the confusion of 0.17 from roadworks to traffic is also interesting, as the confusion from traffic to roadworks roadworks is only 0.02. This might be due to the fact that any roadwork scene is much likely to overlap with traffic, but not the other way around.

4.3. Reference Experiment - DCASE 2017 Task 1

We performed an additional baseline classification experiment using the development dataset from the task 1 of the DCASE 2017 challenge ("Acoustic scene classification"), which includes 4680 10 second long excerpts from 15 acoustic scene classes as well as a predefined partition for a 4-fold cross-validation [24]. We randomly sampled 10 one second long time frequency patches from each recording to enlarge the dataset. For the model SB$^{1.00}$, we obtain mean accuracy values of 0.91 (standard deviation 0.01) for the development set and 0.64 (0.02) for the test set. The TAK$^{1.00}$ shows slightly higher values of 0.93 (0.001) and 0.67 (0.02) for development and test set, respectively. It must be noted that we do not exploit the full length of the clips e.g. by late fusion techniques like model averaging but instead classify only short excerpts (1 s).

5. CONCLUSIONS

In this paper, we proposed a distributed system for acoustic scene classification in urban environments. Spectrogram patches, which are extracted on the sensor side, are compressed using a deep denoising autoencoder and transmitted to a central server unit, where they are forwarded to a CNN-based classification model. We compared two state-of-the-art network architectures for the task at hand and evaluate their performance depending on additional environmental background noise, the comression rate of the autoencoder, as well as the number of filters in the convolutional layers. Our results show that good classification scores can be achieved despite challenging class partitions with partially shared sound event types.

Acknowledgments

The *Stadtlärm* project is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi, ZF 4072003LF6). Stylianos Ioannis Mimilakis is funded by the European Union's H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement no 642685 MacSeNet.

6. REFERENCES

[1] "Stadtlärm Project Description," http://s.fhg.de/StadtLaerm (last visited: 08/07/2017), 2017.

[2] "German technical guidelines for noise reduction," http://www.verwaltungsvorschriften-im-internet.de/bsvwvbund_26081998_IG19980826.htm (last visited: 08/07/2017), 2017.

[3] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[4] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, March 2017.

[5] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep convolutional neural networks and data augmentation for acoustic event detection," in *Proceedings of the Interspeech Conference*, San Francisco, USA, September 8-12 2016, pp. 2982–2986.

[6] P. Bellucci, L. Peruzzi, and G. Zambon, "LIFE DYNAMAP project: The case study of Rome," *Applied Acoustics*, vol. 117, pp. 193–206, 2017.

[7] D. Hollosi, G. Nagy, R. Rodigast, S. Goetze, and P. Cousin, "Enhancing wireless sensor networks with acoustic sensing technology: Use cases, applications & experiments," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 2013, pp. 335–342.

[8] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, 2014, pp. 1041–1044.

[9] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. C. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," *CoRR*, vol. abs/1701.02720, 2017. [Online]. Available: http://arxiv.org/abs/1701.02720

[10] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, May 2016.

[11] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *Proceedings of the 25th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2015, pp. 1–6.

[12] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification and domestic audio tagging," DCASE2016 Challenge, Tech. Rep., September 2016.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR 2015)*, 2015, pp. 1–4.

[14] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017, pp. 771 – 775.

[15] E. Çakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," *CoRR*, 2017.

[16] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. B. Jackson, and M. D. Plumbley, "Unsupervised feature learning based on deep models for environmental audio tagging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, June 2017.

[17] G. Lafay, "IEEE DCASE 2016 Challenge - Task 2 - Train/Development Datasets," https://archive.org/details/dcase2016_task2_train_dev (last visited: 07/04/2017, 2016.

[18] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct 2015.

[19] "SoX library," http://sox.sourceforge.net/sox.html (last visited: 08/07/2017), 2017.

[20] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[23] "MLP based system, DCASE2017 baseline," https://tut-arg.github.io/DCASE2017-baseline-system/system_description.html (last visited: 08/01/2017, 2017.

[24] "DCASE 2017 - acoustic scene classiciation - task description," https://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-acoustic-scene-classification (last visited: 17/10/2017), 2017.