THE DETAILS THAT MATTER: FREQUENCY RESOLUTION OF SPECTROGRAMS IN ACOUSTIC SCENE CLASSIFICATION

Karol J. Piczak

Institute of Computer Science
Warsaw University of Technology

ABSTRACT

This study describes a convolutional neural network model submitted to the acoustic scene classification task of the DCASE 2017 challenge. The performance of this model is evaluated with different frequency resolutions of the input spectrogram showing that a higher number of mel bands improves accuracy with negligible impact on the learning time. Additionally, apart from the convolutional model focusing solely on the ambient characteristics of the audio scene, a proposed extension with pretrained event detectors shows potential for further exploration.

Index Terms— acoustic scene classification, spectrogram, frequency resolution, convolutional neural network, DCASE 2017

1. INTRODUCTION

The area of environmental sound classification has recently experienced a significant increase in the quantity of performed studies. One of the main driving factors in 2016 was the organization of the first DCASE workshop [1], complemented by an open challenge focusing on the detection and classification of acoustic scenes and events. This unique opportunity enabled researchers to exchange ideas and evaluate various approaches on a common set of tasks and datasets, a valuable initiative which continues in 2017 with a second installment of the workshop [2].

Looking at previous submissions to this challenge, a clear picture emerges on how diverse the methods employed to tackle these tasks can be. In 2013, when the very first DCASE challenge [3] was organized, although most approaches used a support vector machine (*SVM*) classifier, the input frames spanned a vast range of features:

- mel-frequency cepstral coefficients (*MFCC*) [4, 5, 6],
- mel spectrograms processed through a sparse RBM extractor [7],
- statistics from a cochleogram based on a tone-fit algorithm [8],
- responses of modulation tuned filters (2D Gabors) [9],
- visual features (*HOG*) computed on a constant-Q transform [10].

At the same time, other teams evaluated the usefulness of hidden Markov models (*HMM*) [11], an i-vector approach combined with MFCCs [12], bagging of decision trees with MFCCs and wavelets [13] and a random forest classifier working on an embedding through dissimilarity representation [14].

In contrast, the DCASE 2016 challenge saw an emergence of deep learning techniques with numerous systems shifting to deep neural networks (*DNN*) [15, 16, 17, 18], convolutional neural networks [19, 20, 21, 22, 23, 24], recurrent models [25, 26, 27, 28, 29] and their fusions with other approaches like the i-vector [30]. Although MFCCs were still widely encountered as input features in sound event detection, more low-level representations such as mel band energy and various forms of spectrograms were much more common in the acoustic scene classification task.

When developing models relying on spectrograms as their input, one of the decisions that has to be made is the resolution of the data generated in the preprocessing step. What should it be? One obvious response is: “the higher, the better”. As visualized by Figure 1, choosing a more fine-grained representation means less information is being lost at the very beginning and this should hopefully allow for more nuanced differentiation between similar training examples in later stages. However, there are three countervailing issues that we have to take into consideration here.

First of all, although increasing the time and frequency resolution of the employed representation may be desirable, the uncertainty principle imposes a theoretical limit on how these two can be combined. It is always a trade-off. Wide windows give good frequency resolution, but their temporal resolution is affected for the worse. Narrow windows behave in the opposite way.

One can counter this claim by stating that, theoretical limits notwithstanding, in most cases it is still possible to maintain a temporal resolution sufficient for an audio classification task while using wider windows. Even then, however, a practical aspect of resource constraints remains. Will the impact on memory and storage requirements introduced by a higher resolution be acceptable in a given application? Is a longer computation time, both in the preprocessing and learning phase, really worth it? Especially in scenarios combining real-time processing with deployment on low-power devices these issues can become crucial.

Finally, dimensionality reduction of the input data is a proven way to facilitate learning. Looking from this perspective, a single audio frame of 10 milliseconds, sampled at 44.1 kHz, contains 441 datapoints in its raw form. On the contrary, MFCCs can succinctly describe it with only a dozen of coefficients. With longer frames the discrepancy will be even more pronounced. Therefore, a valid concern arises whether a high-resolution spectrogram with hundreds of frequency bands will not become an overkill that effectively impedes efficient learning.

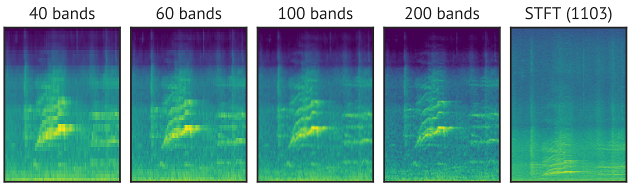


Figure 1: A visual comparison of 3-second-long fragments of spectrograms with different frequency resolutions (first four use a mel scale, the last one is a plain STFT).

The source code for this study can be found at:
https://github.com/karoldvl/paper-2017-DCASE

Evaluating related works in this area, it seems indeed that the prevailing tendency is to limit the number of computed frequency bands to less than 100. Although greater values can be occasionally encountered (100 in [28], 128 in [31], 150 in [32] and even 1025 in [27]), 60 bands [20, 23] and 40 bands [17, 26, 29, 33] are the dominant option. This would imply either that the gains potentially achievable from a higher resolution are counterbalanced by other negative factors, or that the issue is deemed, so far at least, only tangential to the actual problem of model construction and has not received much attention of itself.

This specific research question is the main motivation behind this study. A thorough analysis of all the issues voiced in the introduction is not possible in a scope of a short paper, so it will be limited to an evaluation of a single submission to the acoustic scene classification task of the DCASE 2017 challenge [2]. Nevertheless, it will hopefully signal whether this problem could be worth investigating further in a more generalized manner.

2. EXPERIMENT SETUP

2.1. Task and dataset

The goal of the acoustic scene classification task proposed in the DCASE 2017 challenge is to determine the context of a given recording by choosing one appropriate label from a set of 15 predetermined acoustic scenes. For each scene, there are 312 audio segments in the development dataset with each segment having a length of 10 seconds and a sampling rate of 44.1 kHz. The challenge organizers prearranged the development dataset into 4 folds for comparable cross-validation in such a manner that segments originating from one physical location are contained in the same fold. The final scoring of submitted systems is based on the fraction of correctly classified segments from the evaluation dataset. Further information about the recording and annotation procedure can be found in the paper describing the dataset [34].

2.2. Data preprocessing

The first step of the proposed solution consists in converting all the provided recordings into spectrograms with *librosa v0.5.1* [35]. Mel spectrograms are created with an FFT window length of 50 ms (2205 samples), hop length of 20 ms (882 samples), and a number of bands that is either 40, 60, 100 or 200, in all cases covering a frequency range of up to 22050 Hz. Additionally, a plain STFT spectrogram (1103 bands) is created with the same window and hop length for comparison. Finally, the spectrograms are converted to a decibel scale and standardized by subtracting the mean and dividing by the standard deviation computed on a random batch of 1000 examples. In this manner, the resulting dimension of a 10-second-long segment representation is b rows and 500 columns, where b is the number of generated frequency bands.

During training, slight data augmentation is introduced by a uniformly distributed offset of the start time of up to 1 second. Moreover, in each case a randomly sized tail of the generated example is replaced with a different segment belonging to the same class, creating some additional variety in the training batches.

2.3. Model architecture

Most acoustic scenes can be conceptually described as an ensemble of two distinct elements. The ambience layer consists of a nondescript theme recurring in the background with little to no change (e.g. sound of a noisy street). Every now and then a more specific event of a short-lived nature occurs (e.g. a book page being flipped in a library). In many situations the background information alone is quite sufficient for establishing an actual context with little ambiguity. However, browsing through the provided dataset and trying to deduce how human perception copes with such a task, it seems that in some cases very subtle clues (as the aforementioned page flipping) are the key elements that drastically shift the expectations between similar contexts (e.g. *home* and *library*).

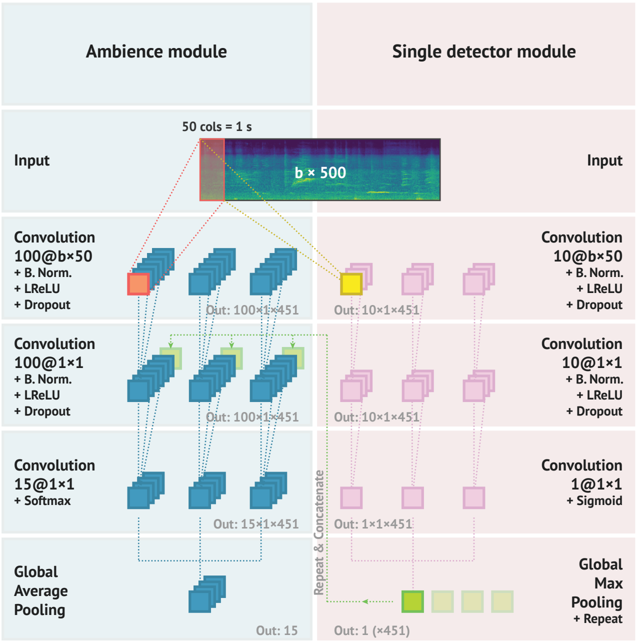


Figure 2: A schematic of the model with its ambience part and a possible extension with a detector module.

Based on this observation, a natural question is whether a machine learning model incorporating such an assumption would be advantageous. On the other hand, taking into consideration the high accuracy of the baseline solution and the results of Mafra et al. [36] where the authors indicate that good results can be obtained by representing each recording with only a single averaged frame, it is thus very likely that a good architecture should not be overly complicated in this case.

Therefore, the system described in this work has a very simple design, coming in two flavors depicted in Figure 2. The first variant is a three block convolutional network focusing on processing the ambience content. Its first layer takes the whole input spectrogram ($b \times 500$) and applies a convolution with a stride of 1, filter size of $b \times 50$ (i.e. over fragments of 1 second) and the number of filters set at 100. The response is batch normalized and processed through a LeakyReLU activation ($\alpha = 0.3$) combined with dropout ($p_{\text{drop}} = 0.25$). The second processing block is identical, except for the filter size which in this case is reduced to 1×1, meaning that there is no spatial convolution but only aggregation across feature maps. The final layer consists of 15 convolutional filters of 1×1 and a softmax activation that is computed separately for each step. For training purposes, output probabilities are averaged with a global pooling layer. However, during the prediction step no pooling is performed, but instead these values are binarized with a threshold of 0.5 and only then averaged over the whole time span, which is equivalent to a majority vote.

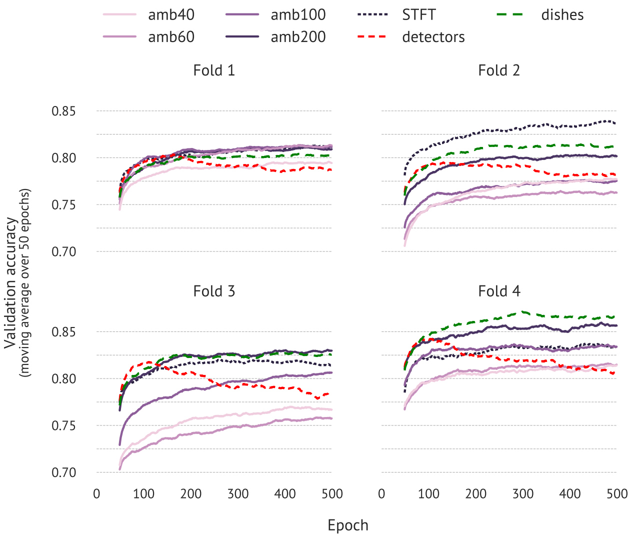


Figure 3: Comparison of validation accuracy for the evaluated systems achieved on the development dataset. Results are presented as a moving average over 50 epochs for better clarity.

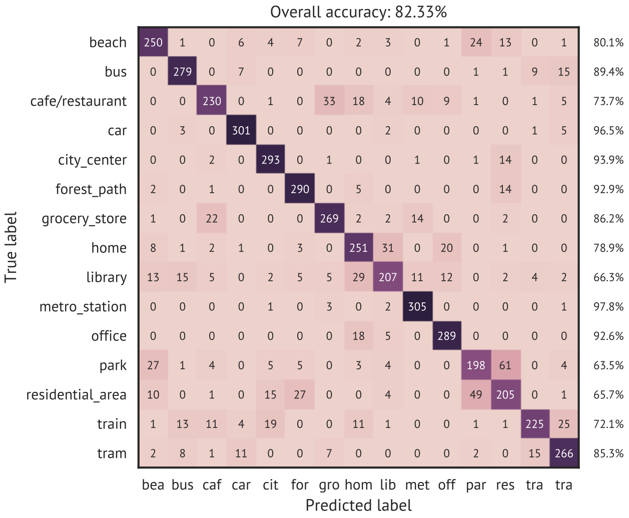
The second variant extends this model with a module that we will further call a *detector*. An architecture of a detector is exactly the same as the already described ambience part with the difference that convolutional blocks use only 10 filters and the last layer consists of a single convolutional unit with sigmoid activation that is max-pooled over the whole time span. The rationale behind this is that the output of such a network should signal whether a given event (template match) has occurred anywhere in the whole recording. The whole variant then combines the ambient module with a predefined number of detectors by concatenating their output to the input of the last convolutional layer (same global event detection value is repeated for each step of the ambient model).

Two remarks about the implications of such an architecture. First of all, while we are using a “convolutional” designation for this model, were it not for some subtle differences coming from the use of normalization layers and joint training, it could be validly understood as a simple multi-layer perceptron that is being applied to consecutive frames of the input, an approach very similar to the baseline implementation.

Moreover, by filling the first layer with filters of a very large size, spanning the whole frequency range, we can limit the impact of higher resolutions to this layer only. This means that the increase in computation time is not that severe. The prospects here would be much worse with networks stacking multiple layers of small-sized filters, where such changes propagate in the output dimensions of deeper layers, a drawback which should not be overlooked.

2.4. Training procedure

Before training, all model weights are initialized with a *He uniform* [37] procedure. Training is performed for 500 epochs with an *Adam* optimizer (learning rate of 0.001, batch size of 32) and a categorical cross-entropy loss function. 400 segments are carved out from the training fold as an additional holdout batch. The best performer on the holdout batch is retained as the final model, whereas validation results are calculated on a completely separate fold as provided by the organizers. Separate models are trained for each combination of training and validation folds. A hierarchical learning method similar to the one reported in [17] was tentatively evaluated, however the difference achieved with the employed architecture was not noticeable enough to warrant further investigation.


Overall accuracy: 82.33%

True label \ Predicted label	bea	bus	caf	car	cit	for	gro	hom	lib	met	off	par	res	tra	tra	
beach	250	1	0	6	4	7	0	2	3	0	1	24	13	0	1	80.1%
bus	0	279	0	7	0	0	0	0	0	0	0	1	1	9	15	89.4%
cafe/restaurant	0	0	230	0	1	0	33	18	4	10	9	1	0	1	5	73.7%
car	0	3	0	301	0	0	0	0	2	0	0	0	0	1	5	96.5%
city_center	0	0	2	0	293	0	1	0	0	1	0	1	14	0	0	93.9%
forest_path	2	0	1	0	0	290	0	5	0	0	0	0	14	0	0	92.9%
grocery_store	1	0	22	0	0	0	269	2	2	14	0	0	2	0	0	86.2%
home	8	1	2	1	0	3	0	251	31	0	20	0	1	0	0	78.9%
library	13	15	5	0	2	5	5	29	207	11	12	0	2	4	2	66.3%
metro_station	0	0	0	0	1	0	3	0	2	305	0	0	0	0	1	97.8%
office	0	0	0	0	0	0	0	18	5	0	289	0	0	0	0	92.6%
park	27	1	4	0	5	5	0	3	4	0	0	198	61	0	4	63.5%
residential_area	10	0	1	0	15	27	0	0	4	0	0	49	205	0	1	65.7%
train	1	13	11	4	19	0	0	11	1	0	0	1	1	225	25	72.1%
tram	2	8	1	11	0	0	7	0	0	0	0	2	0	15	266	85.3%

Figure 4: Confusion matrix of the submitted *amb200* model (ambience only, 200 mel bands) combined over all folds of the development set. The rightmost column presents class-wise accuracies.

3. RESULTS

The main system presented in this work, codenamed *amb*, consisted solely of the ambience processing module (left part of Figure 2). Five variants of this model were evaluated, four using mel spectrograms with 40, 60, 100 and 200 frequency bands respectively and one working on STFT spectrograms with 1103 bands (denoted as *STFT* later on). Additionally, a model combining the *amb200* variant with 15 independent detector modules was created (*detectors*). The results of these models are depicted in Figure 3 and presented in a numerical way in Table 1, while Figure 4 more specifically details class-wise performance of the *amb200* model.

The analysis of these results indicates that a higher number of mel frequency bands quite uniformly improves the achieved validation accuracy. There is almost a 4 percentage point difference between *amb40* and *amb200* variants, showing that, in this setup at least, higher resolution models have a greater predictive capacity. The *STFT* variant is on average comparable to *amb200*, it is however underperforming in fold 4 and strongly outperforming in fold 2. Taking into consideration the processing overhead (approximate epoch processing time on a GTX 980 Ti card was 23 s for *amb40* up to 25 s for *amb200* and 52 s for *STFT*) the *amb200* variant is a clear winner.

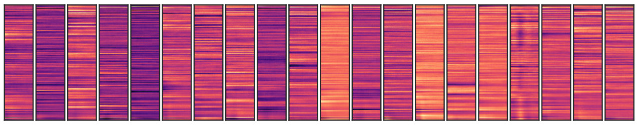


Figure 5: First 20 filters learned by the initial convolutional layer of the *amb200* model (ambience only, 200 mel bands).

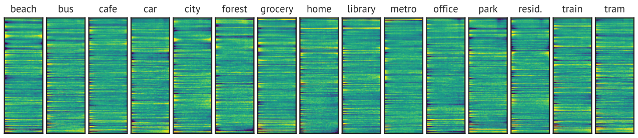


Figure 6: Synthetic examples of input patterns resulting in maximum output activation for a given class ($p_{\text{class}}(\boldsymbol{X}) = 1.0$). Slight contrast squashing was applied for presentation purposes.

On the other hand, a disappointing behavior of the *detectors* model combines poor validation accuracy with very high training time. It is quite evident that for this particular dataset the capacity of such an architecture is too high and after 100 epochs of training significant overfitting occurs. Therefore, an additional model *dishes* is proposed. What is peculiar about it, it extends the *amb200* model with only a single detector. Moreover, this detector module is separately pretrained on additional hand-annotations specifically created for this purpose indicating the occurrence of specific events in the *cafe/restaurant* scene that could be described as sounds involving cups, plates, kitchenware etc. Unfortunately, due to time constraints and the effort involved in creating a more complete annotation of the dataset, it was not possible to evaluate a model with a broader range of pretrained detectors. However, taking into consideration that initial results reported here hint at a possible improvement, it could be an interesting further avenue for research.

Finally, trying to understand a bit better on what is going on behind the scenes, we can see that the ambience model learns to be a strong frequency discriminator as seen both by the convolutional filters visualized in Figure 5 and examples of patterns that induce the strongest activation for a specific class (Figure 6). This would explain why a high frequency resolution of the input data might be so important for this task. However, especially looking at Figure 6, the perceptual differences are minuscule apart from some intricate patterns of narrow frequency bands, so a real question is on what is being learned. Is it actually the semantic differentiator between different types of scenes? In some cases, like interiors of vehicles, most probably yes. However, for classes such as *home* or *library* it is quite possible that these specific frequency patterns concentrate on what would be perceived by a human listener as recording artifacts. Further study would be required, but there is a high risk that the resulting model would be prone to adversarial attacks.

Table 1: Results of the proposed systems.

System	Development					Final
	Fold 1	Fold 2	Fold 3	Fold 4	1—4	
amb40	79.4 (0.5)	77.7 (0.8)	76.7 (1.0)	81.4 (1.0)	78.8	—
amb60	81.3 (0.6)	76.3 (1.0)	75.8 (0.9)	81.5 (1.0)	78.7	62.0
amb100	81.1 (0.6)	77.5 (0.9)	80.6 (0.7)	83.4 (1.3)	80.7	67.7
amb200	80.9 (0.8)	80.2 (0.8)	83.0 (0.9)	85.6 (1.3)	82.4	70.6
STFT	81.1 (0.9)	83.6 (0.8)	81.4 (0.9)	83.4 (1.3)	82.4	—
detectors	78.7 (0.9)	78.1 (1.1)	78.6 (1.3)	80.8 (1.4)	79.1	—
dishes	80.3 (0.9)	81.4 (0.7)	82.6 (0.6)	86.6 (1.0)	82.7	69.6

Mean (standard deviation) of validation accuracies across 50 final epochs of training on the development set and official evaluation results for submitted models. Values in percentages.

4. CONCLUSION

This paper described a submission to the acoustic scene classification task of the DCASE 2017 challenge based on a convolutional neural network model specifically limited to focusing on the ambient characteristic of auditory scenes by average pooling responses for consecutive fragments of the recording. Experiments completed in this study showed that a very important determinant of the final performance in this task is the frequency resolution of the input representation being used, most probably due to the fact that the network is learning a form of a frequency discriminating function. Therefore, increasing the number of mel bands up to 200, well above what is most commonly encountered in related works, proved to be most effective. At the same time, using plain STFT spectrograms with even higher number of bands did not provide additional gains, while considerably increasing the computation time.

It is hard to tell in the scope of this work whether these results could be generalized for other contexts (e.g. event detection), where apart from the frequency content, changes in time also play a crucial role. Concurrently, while the exact scope of the increase in the processing time when employing different types of models is not clear, a valid concern is whether the gains achieved will compensate for the longer training time, especially when using deep convolutional architectures with very small filters. This would have to be evaluated on a case-by-case basis. Nevertheless, the aim of this study is to underline that this hyperparameter should also be taken into consideration, even if only to squeeze some additional performance out of the very final model.

5. REFERENCES

[1] T. Virtanen, *et al.*, *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. Tampere University of Technology. Department of Signal Processing, 2016.

[2] A. Mesaros, *et al.*, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," in *Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, Munich, 2017.

[3] D. Stowell, *et al.*, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[4] J. T. Geiger, B. Schuller, and G. Rigoll, "Recognising acoustic scenes with large-scale audio feature extraction and SVM," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[5] W. Nogueira, G. Roma, and P. Herrera, "Sound scene identification based on MFCC, binaural features and a support vector machine classifier," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[6] G. Roma, *et al.*, "Recurrence quantification analysis features for auditory scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[7] J. Nam, Z. Hyung, and K. Lee, "Acoustic scene classification using sparse feature learning and selective max-pooling by event detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[8] J. D. Krijnders and G. ten Holt, "A tone-fit feature representation for scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[9] K. Patil and M. Elhilali, “Multiresolution auditory representations for scene classification,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[10] A. Rakotomamonjy and G. Gasso, “Histogram of gradients of time-frequency representations for audio scene classification,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 142–153, 2015.

[11] M. Chum, *et al.*, “IEEE AASP scene classification challenge using hidden Markov models and frame based classification,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[12] B. Elizalde, *et al.*, “An i-vector based approach for audio scene detection,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[13] D. Li, J. Tam, and D. Toub, “Auditory scene classification using machine learning techniques,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[14] E. Olivetti, “The wonders of the normalized compression dissimilarity representation,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[15] S. Mun, *et al.*, “Deep neural network bottleneck feature for acoustic scene classification,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[16] G. Takahashi, *et al.*, “Acoustic scene classification using deep neural network and frame-concatenated acoustic feature,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[17] Y. Xu, *et al.*, “Hierarchical learning for DNN-based acoustic scene classification,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[18] I. Choi, *et al.*, “DNN-based sound event detection with exemplar-based approach for noise reduction,” in *Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, 2016.

[19] Y. Han and K. Lee, “Convolutional neural network with multiple-width frequency-delta data augmentation for acoustic scene classification,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[20] M. Valenti, *et al.*, “DCASE 2016 acoustic scene classification using convolutional neural networks,” in *Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, 2016.

[21] T. Lidy and A. Schindler, “CQT-based convolutional neural networks for audio scene classification and domestic audio tagging,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[22] H. Phan, *et al.*, “CNN-LTE: a class of 1-X pooling convolutional neural networks on label tree embeddings for audio scene recognition,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[23] D. Battaglino, *et al.*, “Acoustic scene classification using convolutional neural networks,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[24] E. Çakir, T. Heittola, and T. Virtanen, “Domestic audio tagging with convolutional neural networks,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[25] T. H. Vu and J.-C. Wang, “Acoustic scene and event recognition using recurrent neural networks,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[26] S. H. Bae, I. Choi, and N. S. Kim, “Acoustic scene classification using parallel combination of LSTM and CNN,” in *Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, 2016.

[27] M. Zöhrer and F. Pernkopf, “Gated recurrent networks applied to acoustic scene classification and acoustic event detection,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[28] T. Hayashi, *et al.*, “Bidirectional LSTM-HMM hybrid system for polyphonic sound event detection,” in *Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, 2016.

[29] S. Adavanne, *et al.*, “Sound event detection in multichannel audio using spatial and harmonic features,” in *Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, 2016.

[30] H. Eghbal-Zadeh, *et al.*, “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*, 2016.

[31] 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, 2017.

[32] P. Giannoulis, *et al.*, “Improved dictionary selection and detection schemes in sparse-CNMF-based overlapping acoustic event detection,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[33] I. Sobieraj and M. Plumbley, “Coupled sparse NMF vs. random forest classification for real life acoustic event detection,” in *Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, 2016.

[34] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in *24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, 2016.

[35] B. McFee *et al.*, “librosa 0.5.0,” Feb. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.293021

[36] G. Mafra, *et al.*, “Acoustic scene classification: An evaluation of an extremely compact feature representation,” *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2016.

[37] K. He, *et al.*, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *IEEE International Conference on Computer Vision (ICCV 2015)*, Washington, DC, 2015.