CNN-LTE: A CLASS OF 1-X POOLING CONVOLUTIONAL NEURAL NETWORKS ON LABEL TREE EMBEDDINGS FOR AUDIO SCENE CLASSIFICATION

Huy Phan*†, *Philipp Koch**, *Lars Hertel**, *Marco Maass**, *Radoslaw Mazur**, and *Alfred Mertins**

*Institute for Signal Processing, University of Lübeck, Germany
†Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Germany
{phan, koch, hertel, maass, mazur, mertins}@isip.uni-luebeck.de

ABSTRACT

We present in this work an approach for audio scene classification. Firstly, given the label set of the scenes, a label tree is automatically constructed where the labels are grouped into meta-classes. This category taxonomy is then used in the feature extraction step in which an audio scene instance is transformed into a label tree embedding image. Elements of the image indicate the likelihoods that the scene instances belong to different meta-classes. A class of simple 1-X (i.e. 1-max, 1-mean, and 1-mix) pooling convolutional neural networks, which are tailored for the task at hand, are finally learned on top of the image features for scene recognition. Experimental results on the DCASE 2013 and DCASE 2016 datasets demonstrate the efficiency of the proposed method.

Index Terms— audio scene classification, convolutional neural network, label tree embedding, pooling

1. INTRODUCTION

Audio scene classification (ASC) is an important but challenging task of computational auditory scene analysis [1, 2]. The scenes usually exhibit a complex composition of sounds, causing difficulties in obtaining good representations for them. In general, foreground sound events [3, 4, 5], background noise [6], and their combination [7] can be used as a footprint to represent a scene [3, 4, 5].

Many features have been proposed for the task. They can be roughly categorized in two groups: low-level and high-level features. The former includes Mel frequency cepstral coefficients (MFCCs) [8, 2] and Gammatone filterbank coefficients [9], Histogram of Oriented Gradients (HOG) [10, 7], and Gabor dictionary [11] to mention a few. The latter is usually built on top of low-level features via classification or clustering schemes, such as bag-of-features (BOF) models [12], restricted Boltzmann machines (RBM) [13], and non-negative matrix factorization (NMF) [14]. More recently, scene representation via its similarity to speech patterns has been reported to give good generalization [15]. After the feature extraction step, the classification is finally accomplished by some back-end classifiers, such as Hidden Markov Models (HMMs) [16], GMMs [17, 18], Support Vector Machines (SVMs) [8, 10], and Deep Neural Networks (DNNs) [19].

The work in [15] demonstrated that learned hierarchical representations that take into account the structure of scene data can be highly discriminative, as state-of-the-art performance can be obtained even with simple linear classifiers. More specifically, a class taxonomy is constructed by learning to group similar categories into meta-classes on a tree structure. An explicit embedding is then derived to map each audio segment into the semantic space that underlies the class hierarchy. However, the average pooling to form a global feature vector for a scene instance [15] results in loss of details about the scene, such as foreground events. To avoid this issue, in this work we skip the average pooling and represent a scene as a 2-dimensional LTE image. Afterward, we train different 1-X pooling convolutional neural networks (CNN), including *1-max*, *1-mean*, and *1-mix pooling CNNs*, on top of these images for classification. The rational behind this is that with the high-level LTE image features, we have quantized and reduced the complex mixture of sounds into meta-class likelihoods of the LTE images on which even a very simple CNN is able to yield good performance. The proposed CNNs are particularly designed for pattern learning and matching from the LTE images for classification. While the 1-max CNN is expected to uncover patterns corresponding to foreground events of the scenes, the 1-mean one tends to capture the average background patterns, and the 1-mix one is to combine both types of information into the same model.

2. LTE IMAGE FEATURES FOR AUDIO SCENES

2.1. The LTE image features

Given an audio scene dataset of C classes, for example the DCASE 2013 dataset [2], we firstly decompose the audio signals into multiple segments of length 500 ms with an overlap of 250 ms. This results in $T = 118$ segments for each 30-second snippet. Each segment is characterized by a low-level feature vector of size M, such as MFCCs, and labeled by the label of the original scene signal. Using this set of data, we then learn to construct a label tree which recursively groups similar categories into $(C-1) \times 2$ meta-classes two of which are associated with the left and right child nodes of one out of $(C-1)$ split nodes (cf. [15] for further details). Suppose that we have indexed the split nodes of the label tree as i where $1 \leq i \leq C-1$. Afterwards, an audio segment $\mathbf{x} \in \mathbb{R}^M$ is embedded into the space of meta-class likelihoods via the explicit mapping $\Psi : \mathbb{R}^M \to \mathbb{R}^{(C-1)\times 2}$. Formally, $\Psi(\mathbf{x}) = \left(\psi_1^L(\mathbf{x}), \psi_1^R(\mathbf{x}), \ldots, \psi_{C-1}^L(\mathbf{x}), \psi_{C-1}^R(\mathbf{x})\right)$ where $\psi_i^L(\mathbf{x})$ and $\psi_i^R(\mathbf{x})$ denote the likelihoods that $\mathbf{x}$ belongs to two meta-classes on the left and right child nodes of the split node index i, respectively. The likelihoods $\psi_i^L(\mathbf{x})$ and $\psi_i^R(\mathbf{x})$ can be obtained as the classification probability outputs of some binary classification models that are trained to discriminate the meta-classes on the left and right child nodes of the split node i. As in [15], we trained random forest classifiers [20] with 200 trees for this purpose.

This work was supported by the Graduate School for Computing in Medicine and Life Sciences funded by Germany's Excellence Initiative [DFG GSC 235/1].

Using the label tree embedding learning algorithm , we derived the following LTE representations with different low-level feature sets: (1) Gammatone cepstral coefficients [9, 21], (2) MFCCs [17], and (3) and log-frequency filter bank coefficients [22, 23]. We also study how the presence/absence of background noise affects the LTE representations. We reprocess the input signals using minimum statistics noise estimation and subtraction [24] whenever we need to remove background noise. As a result, six LTE images are obtained for a single scene instance, namely LTE0-Gam, LTE0-MFCC, LTE0-Log, LTE1-Gam, LTE1-MFCC, and LTE1-Log where "0" and "1" denote presence/absence of the background noise.

LTE0-Gam and LTE1-Gam. We use $M = 64$ Gammatone cepstral coefficients [21] in the frequency range of 20 Hz to half of the sampling frequency for each 500 ms audio segment. To accomplish this, an audio segment is further decomposed into 50 ms frames with 50% overlap. 60 Gammatone cepstral coefficients are then extracted for each small frame. In turn, the feature vector for the entire segment is computed by averaging the feature vectors of its constituent frames. Via the label tree embedding, each 30-second scene instance is eventually transformed into an $F \times T$ LTE image where $F = (C - 1) \times 2$.

LTE0-MFCC and LTE1-MFCC. Similarly, for these LTE features, we employed $M = 60$ MFCC features in replacement for Gammatone cepstral coefficients in LTE0-Gam and LTE1-Gam. MFCCs were calculated for each 50 ms frame with Hamming window and 40 mel bands. Beside the first 20 coefficients (including 0th order coefficients), 20 delta coefficients, and 20 acceleration coefficients were also calculated using a window length of nine frames.

LTE0-Log and LTE1-Log. We utilized the set of features in our previous works [23, 22] as low-level features in replacement for Gammatone cepstral coefficients in the LTE0-Gam and LTE1-Gam. They include 20 log-frequency filter bank coefficients, their first and second derivatives, zero-crossing rate, short-time energy, four subband energies, spectral centroid, and spectral bandwidth. The overall feature dimension is $M = 65$.

2.2. Potential issues of classification with global LTE features

In [15], average pooling over time is applied to the LTE images to produce global LTE feature vectors which are presented to SVM classifiers for classification. This recognition scheme achieves state-of-the-art performance on different audio scene datasets (c.f. [15]), thanks to the discriminative powers of LTE features. However, we argue that this classification scheme is actually not optimal.

Excluding the background noise, an acoustic scene usually involves various kinds of foreground sounds which are sparsely and irregularly distributed. As a result, it can be interpreted as foreground events on the bed of background noise. Although foreground events [13, 3, 4, 5] and background noise [6] have been used as a footprint to represent a scene, they should be considered separately [7]. Unfortunately, with the average pooling, we tend to blend the sparse foreground events into the dominating background noise. To overcome this issue, we alternatively propose to classify directly on the LTE images using 1-X pooling CNNs in Section 3 where 1-X pooling stands for 1-max, 1-mean, and 1-mix pooling operators.

3. CLASSIFICATION WITH 1-X POOLING CNNS ON MULTI-CHANNEL LTE IMAGES

In contrast to typical deep CNN architectures, the proposed network architecture is relatively simple. It consists of one convolutional layer, one 1-X pooling layer, and one softmax layer. An illustration for 1-max pooling CNN is given in Fig. 1. However, this architecture is expected to fit well to the task at hand since it tends to extract useful patterns from the LTE images for classification. While the first network is expected to uncover patterns corresponding to foreground events of the scenes, the second one tends to capture the average background, and the third one is to combine both types of information into the same model.

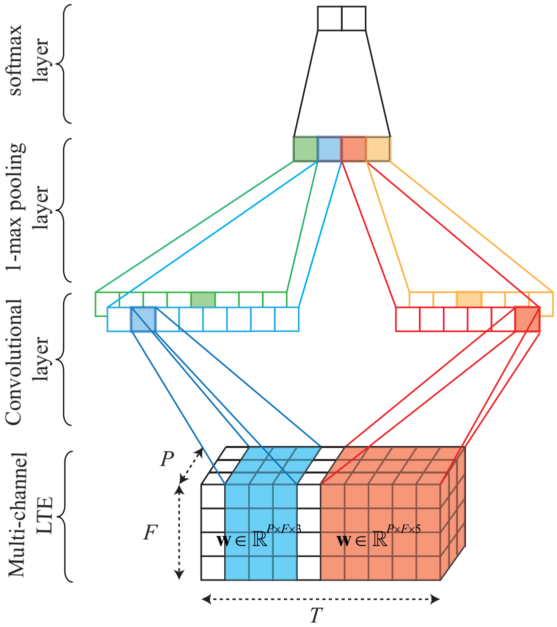
softmax layer
1-max pooling layer
Convolutional layer
Multi-channel LTE
P
F
T
$\mathbf{w} \in \mathbb{R}^{P \times F \times 3}$
$\mathbf{w} \in \mathbb{R}^{P \times F \times 5}$

Fig. 1. Illustration of 1-max pooling CNN architecture on a P-channel LTE image ($P = 6$ in this work). The network consists of two filter sets with two different widths $w = \{3, 5\}$ at the convolutional layer. There are two individual filters on each filter set.

3.1. Multi-channel LTE images as data augmentation

The inputs to the networks are the whole LTE images. Furthermore, our experiments reveal that different low-level features (e.g. Gammatone cepstral coefficients, MFCCs, and log-frequency filter banks) used to derive LTE images are good for different scene categories. In addition, background noise is also shown useful for some event classes. Therefore, it is reasonable to let the CNNs look at multiple LTE images at the same time to learn cross-channel features. To accomplish this, we stack the individual LTE images to produce the multi-channel LTE image of size $P \times F \times T$ for the scene instance when $P = 6$ is the number of single LTE images. This can also be considered as a data augmentation method to regularize the networks.

3.2. Convolutional layer

We aim to use the convolutional layer to extract discriminative features within the whole signals that are useful for the classification task at hand. Suppose that an LTE image presented to the network is given in the form of a 3-dimensional matrix $\mathbf{S} \in \mathbb{R}^{P \times F \times T}$. We then perform convolution on it via 3-dimensional linear filters. For simplicity, we only consider convolution in time direction, i.e. fix two dimensions of a filter to be equal to P and F and vary the remaining dimension to cover different number of adjacent audio segments.

Let us denote a filter by the weight matrix $\mathbf{w} \in \mathbb{R}^{P\times F\times w}$ with the width of w audio segments. Therefore, the filter contains $P \times F \times w$ parameters that need to be learned. We further denote the temporal adjacent spectral slices (e.g. audio segments) from i to j by $\mathbf{S}[i:j]$. The convolution operation $*$ between $\mathbf{S}$ and $\mathbf{w}$ results in the output vector $\mathbf{O} = (o_1, \ldots, o_{T-w+1})$ where:

$$o_i = (\mathbf{S} * \mathbf{w})_i = \sum_{k,l,m} (\mathbf{S}[i:i+w-1] \odot \mathbf{w})_{k,l,m}. \quad (1)$$

Here, $\odot$ denotes the element-wise multiplication. We then apply an activation function h to each o_i to induce the feature map $\mathbf{A} = (a_1, \ldots, a_{T-w+1})$ for this filter:

$$a_i = h(o_i + b), \quad (2)$$

where $b \in \mathbb{R}$ is a bias term. Among the common activation functions, we chose *Rectified Linear Units* (ReLU) due to their computational efficiency [25]:

$$h(x) = \max(0, x). \quad (3)$$

To allow the network to extract complementary features and enrich the representation, we learn Q different filters simultaneously. Moreover, foreground events in a scene may have different durations. We learn filters with different sizes simultaneously in order to capture them more efficiently. More specifically, we learn R different sets of Q filters, each of which has different width w to form $Q \times R$ filters in total.

3.3. 1-X pooling layer

The feature maps produced by the convolution layer are forwarded to the pooling layer. We propose three different pooling operations that are especially designed for scene recognition. In addition, these pooling strategies offer a unique advantage. That is, although the dimensionality of the feature maps varies depending on the length of audio events and the width of the filters, the pooled feature vectors have the same size [26, 27, 28]. Therefore, the signals can be of any arbitrary size instead of being fixed to 30-second long as the common setting for the task.

1-max pooling. This pooling operation on a feature map aims to reduce a feature set to a single most dominant feature [29]. Coupled with the 1-max pooling function, each filter in the convolutional layer is optimized to detect a specific event that is allowed to occur at any time in a scene signal. Pooling on $Q \times R$ feature maps results in $Q \times R$ features that will be joined to form a feature vector that is then presented to the final softmax layer.

1-mean pooling. A feature map is averaged to result in a single mean feature. Due to averaging, this feature is supposed to capture the average background of the signal. $Q \times R$ features are produced from $Q \times R$ feature maps.

1-mix pooling. This operation performs both 1-max and 1-mean pooling at the same time with the hope to capture both foreground events and the average background. The final feature vector contains $2 \times Q \times R$ features, one half consists of 1-max features and the other half of 1-mean features.

3.4. Softmax layer

The fixed-size feature vector after the pooling layer is subsequently presented to the standard softmax layer to compute the predicted probability over the class labels. The network is trained by minimizing the cross-entropy error. This is equivalent to minimizing the KL-divergence between the prediction distribution $\hat{y}$ and the target distribution y. With the binary one-hot coding scheme and the network parameter Θ, the error for N training samples is given by

$$E(\Theta) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i(\Theta)) + \frac{\lambda}{2}||\Theta||^2. \quad (4)$$

The hyper-parameter λ governs the trade-off between the error term and the ℓ_2-norm regularization term. Furthermore, we also employ dropout [30] at this layer by randomly setting values in the weight vector to zero with a predefined probability. The optimization is performed using the *Adam* gradient descent algorithm [31].

4. EXPERIMENTS

4.1. Datasets

We employed the following two datasets in our experiments:

DCASE2013 dataset. This dataset was used in the DCASE2013 challenge [3, 2]. It consists of ten scene categories. The audio signals were recorded in different locations in London at different time points with a sampling frequency of 44.1 kHz. The dataset has two subsets: public and private subsets, each contains 100 30-second-long scene instances with ten examples for each class. The former was released during the challenge for participants to tune their classification systems. The latter was used to evaluate the submissions and also made public after the challenge. The submitted systems were evaluated with five-fold stratified cross validation on the private subset [2]. We follow the cross validation setting here, however, to alleviate possible overfitting due to the small size of this dataset, at each time, we combined the public set and the training folds of the private set to make the training data.

DCASE2016 dataset. The setup is based on the development setting as described in Task 1 of the DCASE 2016 challenge [32, 17]. The signals were recorded with a sampling frequency of 44.1 kHz. The development data consists of 30-second audio signals of 15 scene classes divided into 4-fold cross-validation. The average classification accuracy over all folds is reported. Especially, to handle the errors in some of the recordings (as informed by the challenge), we simply removed erroneous segments from the signals. This results in an LTE image with $T < 118$ segments which was then circularly padded to make 118 segments.

4.2. Parameters

The proposed 1-X pooling CNNs involve different hyperparameters which are specified in Table 1. The filter width w was set to $\{3, 5, 7\}$ segments which are equivalent to 1, 1.5, and 2 seconds duration. For the DCASE2013 dataset, the networks were trained for 200 epochs with a minibatch size of 30. In particular, due to the relatively small number of examples of this dataset, we repeated the network training five times and report the average performance for this dataset. For the DCASE2016 dataset, the networks were trained for 500 epochs with a minibatch size of 50. In fact, the training history shows that the training converged very fast, and the networks do not experience overfitting after convergence.

4.3. Baselines

We employed the classification scheme with the global LTE features in [15] for performance comparison. The baselines (i.e. LTE0-Gam, LTE0-MFCC, LTE0-Log, LTE1-Gam, LTE1-MFCC, and LTE1-Log) were trained using one-vs-one χ^2 kernel SVMs. In addition,

Table 1. Hyper-parameters of the proposed CNN networks.

Hyper-parameter	Value
Filter width w	$\{3, 5, 7\}$
Number of filter for each size	500
Learning rate for the Adam optimizer	0.0001
Dropout rate	0.5
Regularization parameter λ	0.001

we also used a fusion system, denoted by LTE-Fusion, that combines different global LTE feature vectors as an additional baseline. LTE-Fusion is expected to take advantage of representation power from different perspectives (i.e. different low-level features). The fusion is accomplished using the extended Gaussian-χ^2 kernel [33] given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\Big(-\sum_k \frac{1}{\bar{D}^k} D\big(\Psi^k(\mathbf{x}_i), \Psi^k(\mathbf{x}_j)\big)\Big) \tag{5}$$

where $D\big(\Psi^k(\mathbf{x}_i), \Psi^k(\mathbf{x}_j)\big)$ is the χ^2 distance between the global LTE feature vectors of the embedded scene instances $\Psi^k(\mathbf{x}_i)$ and $\Psi^k(\mathbf{x}_j)$ with respect to the k-th channel where

$$\begin{aligned} k \in \{&\text{LTE0-Gam}, \text{LTE0-MFCC}, \text{LTE0-Log}, \\ &\text{LTE1-Gam}, \text{LTE1-MFCC}, \text{LTE1-Log}\}. \end{aligned} \tag{6}$$

$\bar{D}^k$ is the mean χ^2 distance of the embedded scene instances in training data for the k-th channel.

For the sake of comparison, we also collate performance of our systems to other reported results on the datasets. It should be noticed that although there exist other works on the DCASE2013 dataset after the DCASE2013 challenge, we only mention those with performance equivalent or higher than that of the best submission in the challenge. For the case of DCASE2016, the baseline system provided by the challenge [17] is used for this purpose.

4.4. Experimental results

The overall classification performance obtained by different systems is shown in Tables 2 and 3 for the DCASE2013 and DCASE2016 datasets, respectively. As can be seen for individual LTE features, the three employed low-level feature sets perform differently. While LTE0-Log and LTE1-Log outrun others on the DCASE2013 dataset, a comparable performance obtained by different LTE features was seen on the DCASE2016 one. These results imply that, for the audio scene classification task, it is important to adopt appropriate low-level features for high-level feature learning and to adapt them for different datasets. As expected, integrating them in the fusion system LTE-Fusion leads to significant gains in classification accuracy. Absolute gains of up to 2.0% and 3.5% against the best constituent (LTE1-Log) are obtained for the DCASE2013 and DCASE2016 datasets, respectively. It is also worth mentioning that the LTE-Fusion already outperforms the best previously reported results on both datasets (i.e. AMS+LDA [34] for DCASE2013 and DCASE2016 baseline for DCASE2016 [17]) by 4.4% and 4.5%, respectively.

Table 2. Classification accuracy (%) on the DCASE2013 dataset.

Systems	Accuracy
1-Max CNN-LTE	**88.8**
1-Mean CNN-LTE	**88.4**
1-Mix CNN-LTE	**89.4**
LTE0-Gam	73.0
LTE0-MFCC	75.0
LTE0-Log	83.0
LTE1-Gam	80.0
LTE1-MFCC	80.0
LTE1-Log	84.0
LTE-Fusion	86.0
RNH [2]	76.0
MV [2]	77.0
Human [3]	75.0
HOG [10]	76.0
AMS+LDA [34]	85.0

Table 3. Classification accuracy (%) on the DCASE2016 dataset.

Systems	Accuracy
1-Max CNN-LTE	**80.3**
1-Mean CNN-LTE	**79.8**
1-Mix CNN-LTE	**81.2**
LTE0-Gam	72.6
LTE0-MFCC	69.9
LTE0-Log	72.8
LTE1-Gam	70.7
LTE1-MFCC	72.2
LTE1-Log	73.5
LTE-Fusion	77.0
DCASE2016 baseline [17]	72.5

It can be clearly seen that the classification accuracy is significantly improved with the proposed 1-X pooling CNNs. Compared to the best baseline (i.e. LTE-Fusion), classification with the CNNs leads to absolute accuracy improvements of 2.8%, 2.4%, and 3.4% for the DCASE2013 dataset with 1-max, 1-mean, and 1-mix pooling schemes, respectively. The gains for the DCASE2016 reach 3.3%, 2.8%, and 4.2%, respectively. Between the pooling schemes, 1-max appears to be more efficient that 1-mean. These results comply with our knowledge on the acoustic scenes that both foreground sounds and background noise can serve as footprints, but the former is more representationally capable. Combining them with the 1-mix pooling scheme in the 1-mix CNN is expected to lead to an even better classification system.

5. CONCLUSIONS

In conclusion, we present an efficient approach to address the audio scene classification problem. Our systems rely on label tree embedding image features which are automatically learned to encode the structure of the data. Different 1-X (i.e. 1-max, 1-mean, and 1-mix) pooling CNNs are then proposed to optimize on top of these high-level features for classification. The proposed CNN architecture is simple but tailored for the task. Our experimental results on the DCASE2013 and DCASE2016 datasets show that while classification with individual LTE features themselves and their fusion obtains very good performance, the accuracies are significantly improved with 1-X pooling CNNs trained on multi-channel stacked LTE images. Absolute improvement of 3.4% and 4.2% against the fusion baseline are achievable with the 1-mix pooling CNNs on the DCASE2013 and DCASE2016 datasets.

6. REFERENCES

[1] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, Wiley-IEEE Press, 2006.

[2] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[3] D. Barchiesi, D. Giannoulis, D. Stowell, and M. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[4] T. Heittola, A. Mesaros, A. J. Eronen, and T. Virtanen, "Audio context recognition using audio event histogram," in *Proc. EUSIPCO*, 2010, pp. 1272–1276.

[5] R. Cai, L. Lu, and A. Hanjalic, "Co-clustering for auditory scene categorization," *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 596–606, 2008.

[6] S. Deng, J. Han, C. Zhang, T. Zheng, and G. Zheng, "Robust minimum statistics project coefficients feature for acoustic environment recognition," in *Proc. ICASSP*, 2014, pp. 8232–8236.

[7] J. Ye, T. Kobayashi, M. Murakawa, and T. Higuchi, "Acoustic scene classification based on sound textures and events," in *Proc. ACM Multimedia*, 2015, pp. 1291–1294.

[8] G. Roma, W. Nogueira, and P. Herrera, "Recurrence quantification analysis features for environmental sound recognition," in *Proc. WASPAA*, 2013, pp. 1 – 4.

[9] X. Valero and F. Alías, "Gammatone cepstral coefficients: biologically inspired features fro non-speech audio classification," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 1684–1689, 2012.

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

[11] R. Mogi and H. Kasaii, "Noise-robust environmental sound classification method based on combination of ICA and MP features," *Artificial Intelligence Research*, vol. 2, no. 1, pp. 107–121, 2013.

[12] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the Acoustical Society of America*, vol. 122, pp. 881–891, 2007.

[13] K. Lee, Z. Hyung, and J. Nam, "Acoustic scene classification using sparse feature learning and event based pooling," in *Proc. WASPAA*, 2013, pp. 1–4.

[14] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6445–6449.

[15] H. Phan, L. Hertel, M. Maass, P. Koch, and A. Mertins, "Label tree embeddings for acoustic scene classification," in *Proc. ACM Multimedia 2016*, Amsterdam, The Netherlands, October 2016.

[16] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

[17] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. EUSIPCO*, 2016.

[18] S. Chu, S. Narayanan, and C.-C.J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[19] Y. Petetin, C. Laroche, and A. Mayoue, "Deep neural networks for audio scene recognition," in *Proc. EUSIPCO*, 2015, pp. 125–129.

[20] L. Breiman, "Random forest," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[21] D. P. W. Ellis, "Gammatone-like spectrograms," 2009.

[22] H. Phan, L. Hertel, M. Maass, R. Mazur, and A. Mertins, "Representing nonspeech audio signals through speech classification models," in *Proc. Interspeech*, 2015, pp. 3441–3445.

[23] H. Phan, M. Maaß, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, 2015.

[24] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[25] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 315–323.

[26] H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," in *Proc. Interspeech*, 2016.

[27] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[28] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. SIGIR*, 2015, pp. 959–962.

[29] Y. L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. ICML*, 2010, pp. 111–118.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014.

[31] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015, pp. 1–13.

[32] ," `http://www.cs.tut.fi/sgn/arg/dcase2016/`.

[33] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc CVPR*, 2008, pp. 1–8.

[34] S. Ağcaer, A. Schlesinger, F.-M. Hoffmann, and R. Martin, "Optimization of amplitude modulation features for low-resource acoustic scene classification," in *Proc. EUSIPCO*, 2015, pp. 2556–2560.