Evaluation of Modulation-MFCC Features and DNN Classification for Acoustic Event Detection

Janek Ebbers[1], Alexandru Nelus[2], Rainer Martin[2], Reinhold Häb-Umbach[1]

[1] *Fachgebiet Nachrichtentechnik, Universität Paderborn, {ebbers, haeb}@nt.uni-paderborn.de*

[2] *Institut für Kommunikationsakustik, Ruhr Universität Bochum, {alexandru.nelus, rainer.martin}@rub.de*

Abstract

Acoustic event detection, i.e., the task of assigning a human interpretable label to a segment of audio, has only recently attracted increased interest in the research community. Driven by the DCASE challenges and the availability of large-scale audio datasets, the state-of-the-art has progressed rapidly with deep-learning-based classifiers dominating the field. Because several potential use cases favor a realization on distributed sensor nodes, e.g. ambient assisted living applications, habitat monitoring or surveillance, we are concerned with two issues here. Firstly the classification performance of such systems and secondly the computing resources required to achieve a certain performance considering node level feature extraction. In this contribution we look at the balance between the two criteria by employing traditional techniques and different deep learning architectures, including convolutional and recurrent models in the context of real life everyday audio recordings in realistic, however challenging, multisource conditions.

Introduction

Recent industry trends point towards an increase in popularity of the Internet of Things (IoT) approach using networked devices such as smartphones, tablets and wearables, personal assistants (Amazon Echo, Google Home, Apple Homepod, etc.), and even autonomous vehicles. Additionally the advent of 5G connectivity promises to further increase IoT's popularity. From the audio sensing perspective, the benefits of these networks of acoustic sensors can lead to high potential applications in the realm of ambient assisted living, habitat monitoring, surveillance, smart-cities, and smart-driving.

The idea of audio feature extraction at node level can expand the capabilities of IoT oriented acoustic sensor networks by transferring a part of the computational load from the server side to the end-user, it can help improve bandwidth economy by reducing the data transmission's bitrate and for the cases where data is stored in the Cloud, it can help raise privacy levels by replacing the high-resolution data with a lower-resolution representation. However, the algorithms used for audio feature extraction at the end-user node must provide a sensible utility versus expenditure balance.

In this context we measure utility from the perspective of the system's performance in acoustic event detection and audio tagging tasks, using the metrics later described in the Experiments section. The term expenditure refers to the computational resources used in fulfilling a specific task, as typical IoT devices come with physical size, execution time and battery life constraints, demanding an optimized usage of the available platform. We have measured the expenditure by computing the number of multiply-accumulate operations per second.

In this paper we compare a manually designed, resource economical audio feature extraction approach based on the modulation spectrum of mel-frequency cepstral coefficients (MFCCs) against a more complex and resource demanding deep neural network (DNN) approach, analyzing them from the utility versus expenditure perspective. Both feature extractors start off with the same input, namely mel-band energy features, they then extract their own lower resolution audio features which are finally passed to an identical neural network based detection and tagging module. For the detection task the event and its onset and offset times must be estimated, while the tagging task asks for the detection of the active labels of each sound file only, not requiring start/end time detection. The two tasks chosen for this comparison are part of the DCASE 2017 challenge [1], both aiming at the street level audio environment, however with different levels of complexity. In the following sections we present the two feature extraction approaches, describe the classification modules, detail the metrics and experimental scenarios, and in the end show the advantages and disadvantages of the proposed methods in the two differently scaled audio environments.

Related Work

The DCASE challenges [1, 2] and the availability of large-scale audio datasets [3] have allowed the state-of-the-art in acoustic event detection to progress rapidly with deep-learning-based classifiers dominating the field as also demonstrated by some of the DCASE winning approaches [4, 5]. In the context of acoustic sensor networks, the idea of performing node level feature extraction has been previously promoted by works like [6], where the authors investigate reducing computational costs in state-of-the-art deep learning architectures for acoustic event detection. In contrast to the above, our approach does not try to optimize end node DNN expenditure, but is more focused on comparing it to the expenditure of less complex hard-wired feature extractors.

Previous works have successfully employed modulation MFCC (Mod-MFCC) features [7, 8, 9] for music genre and general audio classes discrimination, and for speaker identification, respectively. The available aggregation dimensions of these feature sets allow for task versatility and feature stream dimensionality control.

Modulation-MFCCs

Borrowing the notation from [9] we hereby describe the computation of Mod-MFCC features by first introducing an intermediary step, which is the computation of log mel-band energy (LMBE) features. These features will be later used in the Convolutional Neural Network section. Starting with the short-time Fourier transform (STFT) representation $X_{\mathrm{stft}}(\kappa, b)$ with window length L_1 and step H_1 of the signal $x(t)$, where κ and b denote the frequency bin and time frame index, respectively, the squared-magnitude spectrum is mapped onto the Mel scale [10], resulting in the Mel-spectrum $X_{\mathrm{mel}}(k', b)$, where $k' = 0, 1, \ldots, K' - 1$ is the index of the Mel scale frequency bin. The LMBE features are then obtained by taking the logarithm of the absolute Mel-spectrum:

$$X_{\mathrm{lmbe}}(k', b) = \log |X_{\mathrm{mel}}(k', b)|. \tag{1}$$

The MFCCs $X_{\mathrm{mfcc}}(\eta, b)$ with the cepstral coefficient index $\eta = 0, 1, \ldots, K'' - 1$ are computed by taking the discrete cosine transform of the LMBE representation and selecting the first K'' coefficients. We then apply a sliding window discrete Fourier transform (DFT) in order to get the short-time MFCC modulation spectrum

$$\hat{X}_{\mathrm{mfcc}}(\nu, \eta, \iota) = \left| \sum_{\ell_2=0}^{L_2-1} X_{\mathrm{mfcc}}(\eta, \iota H_2 + \ell_2) e^{-j\frac{2\pi \ell_2 \nu}{L_2}} \right|, \tag{2}$$

where, starting at sub-frame index $b = \iota H_2$, the sliding window considers L_2 consecutive frames. From this we extract the absolute value. The modulation frequency bin index is specified by $\nu = 0, 1, \ldots, L_2/2$ and ι and H_2 denote the temporal modulation window index and shift, respectively, with $\iota = 0, 1, \ldots, I - 1$ [7, 11]. The values of the modulation spectrum are then averaged by means of moving average (3) where ϕ denotes the averaging window index and L_3 and H_3 denote the averaging window length and shift respectively.

$$\tilde{X}_{\mathrm{mfcc}}(\nu, \eta, \phi) = \frac{1}{L_3} \sum_{\ell_3=0}^{L_3-1} \hat{X}_{\mathrm{mfcc}}(\nu, \eta, \phi H_3 + \ell_3) \tag{3}$$

Due to the previous steps we obtain a reduction of the temporal resolution by the factor of $R = H_2 H_3$. To summarize the modulation spectrum, cepstral modulation ratios (CMR) $\rho_{\nu_1|\nu_2}(\eta, \phi)$ are computed,

$$\rho_{\nu_1|\nu_2}(\eta, \phi) = \frac{\sum_{\nu=\nu_1}^{\nu_2} \tilde{X}_{\mathrm{mfcc}}(\nu, \eta, \phi)}{(\nu_2 - \nu_1 + 1)\tilde{X}_{\mathrm{mfcc}}(0, \eta, \phi)}. \tag{4}$$

In addition to CMRs, we further compute $\bar{X}_{\mathrm{mfcc}}(\eta)$ for the feature vector, where

$$\bar{X}_{\mathrm{mfcc}}(\eta, \phi) = \frac{2}{L_2 + 2} \sum_{\nu'=0}^{L_2/2} \tilde{X}_{\mathrm{mfcc}}(\nu', \eta, \phi) \tag{5}$$

is the modulation spectrum averaged over all modulation frequencies ν for each MFCC bin η. These are rewritten in vector notation and stacked together into one feature vector

$$\mathbf{M}_T = (\bar{\mathbf{X}}^{\intercal}_{\mathrm{mfcc}}, \boldsymbol{\rho}^{\intercal}_{1|1}, \boldsymbol{\rho}^{\intercal}_{2|8})^{\intercal}. \tag{6}$$

For the following, we set $L_1 = 1764$, $H_1 = 882$ $K' = 64$, $K'' = 32$, $H_2 = 8$, $L_2 = 16$, $H_3 = \{1, 2\}$, $L_3 = \{1, 4\}$.

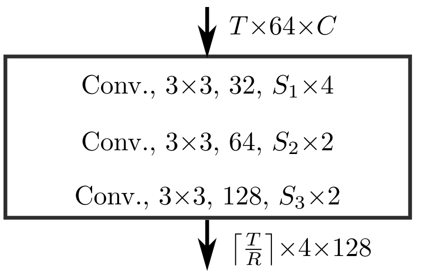
$T \times 64 \times C$
Conv., 3×3, 32, $S_1 \times 4$
Conv., 3×3, 64, $S_2 \times 2$
Conv., 3×3, 128, $S_3 \times 2$
$\lceil \frac{T}{R} \rceil \times 4 \times 128$

Figure 1: The CNN extractor consisting of three layers. Each layer is represented as Conv., <filter size>, <number of filters>, <pool size>. Between the convolution and pooling stages batch normalization and ReLU activation functions are applied.

Convolutional Neural Network

Convolutional neural networks (CNNs) are state-of-the-art not only in a wide range of computer vision tasks but also in speech and audio processing including acoustic event detection [12, 13]. Having data containing local structures such as images and spectrograms, CNNs are trained to extract features suitable for the task of interest. Each layer of a CNN performs convolutions of the input feature maps with learnable filters. Additional pooling stages are widely used to make extraction invariant to small translations.

The structure of the CNN-architecture used in this work is shown in Fig. 1. The input to the CNN is given by the LMBE spectrogram with 64 LMBEs per time frame, which are globally normalized to zero mean and unit variance. Depending on whether we have monaural or binaural audio data the input consists of $C = 1$ or $C = 2$ feature maps, respectively. Three layers of convolutions are used, each followed by batch normalization [14], ReLU activation function and max pooling. While with each layer the number of features in the feature maps is reduced, the number of filters is increased. Stacking the output feature maps results in an output feature size of $F = 4 \times 128 = 512$. To reduce the computational effort in the subsequent classification, the temporal resolution is reduced by performing pooling on the time axis as well. Depending on the pool sizes S_i the temporal resolution is, overall, reduced by the factor of R. The CNN is trained jointly with the classification DNN described in the following section.

Classification

To perform event recognition, we aim to predict whether an event is active or inactive in a certain time segment. We therefore perform K binary classifications in each time segment with K being the number of events of interest. Acoustic event detection requires high resolution classification to be able to determine on- and offsets of events. For audio tagging in contrast it is enough to perform one classification for each file and event.

In both cases, however, DNNs are used to provide frame-level logits $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T] = \mathrm{DNN}(\mathbf{X})$ with the same temporal resolution as the input features $\mathbf{X}$. We compare two different classifier networks here: 1) a simple multi layer perceptron (MLP) network consisting of two

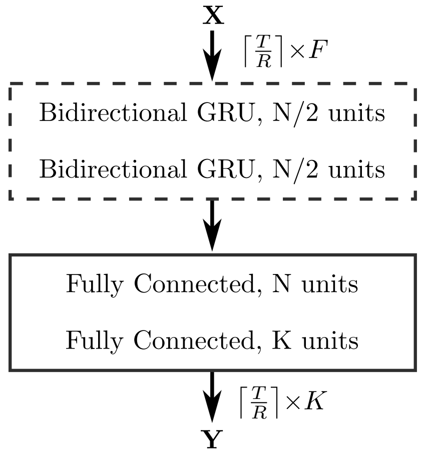
X
$\lceil \frac{T}{R} \rceil \times F$
Bidirectional GRU, N/2 units
Bidirectional GRU, N/2 units
Fully Connected, N units
Fully Connected, K units
$\lceil \frac{T}{R} \rceil \times K$
Y

Figure 2: DNN topologies of MLP and RNN classifiers.

fully connected layers and 2) a recurrent neural network (RNN) having two additional layers of bidirectional gated recurrent units in front of the MLP [15]. The classification networks are illustrated in Fig. 2. When performing tagging, frame-level logits are reduced to only a single vector of file-level logits in terms of averaging $\mathbf{Y} \leftarrow \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t$. Applying the sigmoid function results in class specific posteriors $\tilde{\mathbf{Z}} = \sigma(\mathbf{Y})$.

At training time the binary cross entropy

$$L(\tilde{\mathbf{Z}}, \mathbf{Z}) = \sum_{t=1}^{T}\sum_{k=1}^{K} z_{tk} \log \tilde{z}_{tk} - (1 - z_{tk}) \log(1 - \tilde{z}_{tk})$$

is used as loss function with $\mathbf{Z}$ denoting the binary target matrix. In this work neither data augmentation nor event balancing have been used. The systems are trained using the Adam optimizer [16] with a learning rate of 0.0001 and mini-batches of 16 examples each with a maximum length of 512 LMBE-frames.

At testing time binary predictions $\hat{z}_{tk} = [\tilde{z}_{tk} > c_k]$ are obtained by performing thresholding, where c_k is the decision threshold. For acoustic event detection the predictions in an audio file are additionally median filtered with a filter size of 0.5 s.

Experiments

In the following we evaluate the different feature extractions on sound event detection and audio tagging tasks. The tasks chosen are part of the DCASE 2017 challenge [1] and are aimed at the street level audio environment. We compare the approaches with respect to performance and computational expenses.

Performance is measured using metrics proposed in [17]. The main metric used here is the error rate measured as

$$ER = \frac{\sum_{m=1}^{M} S(m) + D(m) + I(m)}{N_{\text{ref}}}$$

where $S(m)$, $D(m)$ and $I(m)$ are the number of substitutions, deletions and insertions, respectively, made by the system in a segment m and N_{ref} is the total number of positive events in the reference annotation. For the detection task the evaluation is performed in 1 s segments, while for the tagging task m refers to the file index. As a second metric the F-measure is used given as the harmonic mean of precision P and recall R:

$$F = \frac{2PR}{P + R}$$

with

$$P = \frac{\sum_{m=1}^{M} TP(m)}{N_{\text{sys}}} \qquad R = \frac{\sum_{m=1}^{M} TP(m)}{N_{\text{ref}}}$$

with $TP(m)$ being the number of true-positives in a segment m and N_{sys} being the total number of positive events in the systems prediction. We choose that model from the course of the training and those decision thresholds c_k which minimize the ER in the validation. Similar to [6] the computational complexity is measured in terms of Multiply-Accumulation operations (MACs), with one MAC given by $a \leftarrow a + b \cdot c$.

Table 1: Acoustic Event Detection (N=64, C=2)

Feat.	**R**	**DNN**	**MAC [M/s]**	Dev		Eval	
				ER	**F**	**ER**	**F**
LMBE	1	MLP	10.0	61.4%	56.4%	83.7%	40.6%
LMBE	1	RNN	12.3	63.5%	56.5%	81.4%	42.0%
CNN	1	RNN	62.0	60.7%	58.7%	77.3%	50.9%
CNN	8	MLP	26.4	60.8%	59.0%	80.2%	50.2%
CNN	8	RNN	27.0	58.5%	59.6%	76.8%	50.6%
MOD	8	MLP	9.7	62.5%	57.4%	81.6%	49.6%
MOD	8	RNN	10.0	61.6%	56.8%	80.2%	45.5%

The first task considered is sound event detection in real life audio. Development and evaluation datasets consist of 92 min and 29 min of binaural audio, respectively, and annotations for six events are provided with on- and offsets. Due to the small size of the dataset, four folds are provided for cross-validation, splitting the development set into training and validation partitions, such that each example is used for validation exactly once. Predictions are made using chunks of 512 LMBE-frames. Validation and evaluation are performed jointly, where an event is marked active in a segment of the evaluation set if at least two of the trained models mark the event as active. The results are shown in Tab. 1. It can be seen that the recurrent neural network classifiers tend to perform better than the MLP classifiers. Comparing the different features to each other, the CNN feature extraction clearly outperforms LMBE features and Mod-MFCC features. However, it can also be seen that the CNN architectures have significantly increased computational complexity. Mod-MFCC features in contrast are able to outperform LMBE features while requiring less computational expenses than both of the other approaches.

The second task considered is audio tagging. Here a subset of the large-scaled Google AudioSet [3] is used as employed in task 4 of [1]. It consists of 144 hours of multisource recordings split into 51172, 488 and 1103 files with an average length of 9.83 seconds for training, validation and evaluation, respectively. For each of the files tags are provided for 17 events, stating whether an event is active or inactive in the file. Results for this task are shown in Tab. 2. Again we can observe that RNNs are performing better than MLPs. Benefiting from the

Table 2: Audio Tagging (N=256, C=1)

Feat.	**R**	**DNN**	**MAC** [M/s]	Dev		Eval	
				ER	**F**	**ER**	**F**
LMBE	1	MLP	5.8	84.7%	19.0%	76.3%	29.3%
LMBE	1	RNN	30.5	70.8%	39.5%	70.4%	44.8%
CNN	1	RNN	92.8	55.9%	54.3%	57.2%	55.2%
CNN	16	MLP	13.5	66.2%	46.1%	63.9%	51.7%
CNN	16	RNN	15.8	60.9%	49.8%	56.7%	55.1%
MOD	16	MLP	4.9	82.5%	28.0%	78.6%	33.7%
MOD	16	RNN	6.5	74.9%	34.9%	75.1%	36.1%

increased amount of data the learned CNN feature extraction is outperforming the other approaches with a big margin. Also the higher resolution LMBEs are outperforming Mod-MFCC features here. A possible explanation is that, due to the larger amount of data, the classifier is able to learn more detailed patterns which are discarded in the hand-crafted Mod-MFCC features.

Conclusions

Our investigations so far have confirmed that state-of-the-art deep learning approaches for acoustic event detection and tagging offer very good performance, especially when paired with a sufficiently large training database, but they also involve high computational expenditure. With this is mind we can observe that for smaller scaled tasks the hand-wired Mod-MFCC features offer decent performance at much lower costs. As outlook, the combination of Mod-MFCCs in the form of $\tilde{X}_{\text{mfcc}}(\nu, \eta, \phi)$ as input for the CNN architecture is of interest, as it promises to lower expenditure. This of course can be coupled with CNN expenditure reducing strategies as indicated in [6].

Acknowledgements

This work has been supported by DFG under contract nos Ha 3455/15-1 and Ma 1769/6-1 within the Research Unit FOR 2457 (acoustic sensor networks).

References

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “Dcase 2017 challenge setup: Tasks, datasets and baseline system,” in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[2] A. Mesaros, T. Heittola, and T. Virtanen, “Tut database for acoustic scene classification and sound event detection,” in *Signal Processing Conference (EUSIPCO), 2016 24th European*, pp. 1128–1132, IEEE, 2016.

[3] J. F. Gemmeke, D. P. 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 *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 776–780, IEEE, 2017.

[4] S. Adavanne and T. Virtanen, “A report on sound event detection with different binaural features,” *arXiv preprint arXiv:1710.02997*, 2017.

[5] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, “Large-scale weakly supervised audio classification using gated convolutional neural network,” *arXiv preprint arXiv:1710.00343*, 2017.

[6] M. Meyer, L. Cavigelli, and L. Thiele, “Efficient convolutional neural network for audio event detection,” *arXiv preprint arXiv:1709.09888*, 2017.

[7] R. Martin and A. Nagathil, “Cepstral modulation ratio regression (cmrare) parameters for audio signal analysis and classification,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 321–324, IEEE, 2009.

[8] A. Nelus, S. Gergen, J. Taghia, and R. Martin, “Towards opaque audio features for privacy in acoustic sensor networks,” in *Speech Communication; 12. ITG Symposium; Proceedings of*, pp. 1–5, VDE, 2016.

[9] A. Nelus, S. Gergen, and R. Martin, “Analysis of temporal aggregation and dimensionality reduction on feature sets for speaker identification in wireless acoustic sensor networks,” in *Multimedia Signal Processing (MMSP), 2017 IEEE 19th International Workshop on*, pp. 1–6, IEEE, 2017.

[10] S. Furui, *Digital speech processing: synthesis, and recognition.* CRC Press, 2000.

[11] S. Gergen, A. Nagathil, and R. Martin, “Classification of reverberant audio signals using clustered ad hoc distributed microphones,” *Signal Processing*, vol. 107, pp. 21–32, 2015.

[12] 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.

[13] N. Takahashi, M. Gygli, B. Pfister, and L. Van Gool, “Deep convolutional neural networks and data augmentation for acoustic event detection,” *arXiv preprint arXiv:1604.07160*, 2016.

[14] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[17] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.