Tampereen teknillinen yliopisto - Tampere University of Technology

Tuomas Virtanen, Annamaria Mesaros, Toni Heittola, Mark Plumbley, Peter Foster, Emmanouil Benetos & Mathieu Lagrange (eds.)

Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)

TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Tampereen teknillinen yliopisto - Tampere University of Technology

Tuomas Virtanen, Annamaria Mesaros, Toni Heittola, Mark Plumbley, Peter Foster, Emmanouil Benetos & Mathieu Lagrange (eds.)

Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)

Tampere University of Technology. Department of Signal Processing
Tampere 2016

ISBN 978-952-15-3807-0

SOUND EVENT DETECTION IN MULTICHANNEL AUDIO USING SPATIAL AND HARMONIC FEATURES

Sharath Adavanne, Giambattista Parascandolo, Pasi Pertilä, Toni Heittola, Tuomas Virtanen

Department of Signal Processing, Tampere University of Technology

ABSTRACT

In this paper, we propose the use of spatial and harmonic features in combination with long short term memory (LSTM) recurrent neural network (RNN) for automatic sound event detection (SED) task. Real life sound recordings typically have many overlapping sound events, making it hard to recognize with just mono channel audio. Human listeners have been successfully recognizing the mixture of overlapping sound events using pitch cues and exploiting the stereo (multichannel) audio signal available at their ears to spatially localize these events. Traditionally SED systems have only been using mono channel audio, motivated by the human listener we propose to extend them to use multichannel audio. The proposed SED system is compared against the state of the art mono channel method on the development subset of TUT sound events detection 2016 database [1]. The usage of spatial and harmonic features are shown to improve the performance of SED.

Index Terms— Sound event detection, multichannel, time difference of arrival, pitch, recurrent neural networks, long short term memory

1. INTRODUCTION

A sound event is a segment of audio that a human listener can consistently label and distinguish in an acoustic environment. The applications of such automatic sound event detection (SED) are numerous; embedded systems with listening capability can become more aware of its environment [2][3]. Industrial and environmental surveillance systems and smart homes can start automatically detecting events of interest [4]. Automatic annotation of multimedia can enable better retrieval for content based query methods [5][6] .

The task of automatic SED is to recognize the sound events in a continuous audio signal. Sound event detection systems built so far can be broadly classified to monophonic and polyphonic. Monophonic systems are trained to recognize the most dominant of the sound events in the audio signal [7]. While polyphonic systems go beyond the most dominant sound event and recognize all the overlapping sound events in a segment [7][8][9][10]. We propose to tackle such polyphonic soundscape which replicates real life scenario in this paper.

Some SED systems have tackled polyphonic detection using mel-frequency cepstral coefficients (MFCC) and hidden Markov models (HMMs) as classifiers with consecutive passes of the Viterbi algorithm [7]. In [11], a non-negative matrix factorization was used as a pre-processing step, and the most prominent event in each of the stream was detected. However, it still had a hard constraint of estimating the number of overlapping events. This was overcome by using coupled NMF in [12]. Dennis et al [8] took an entirely different path from the traditional frame-based features by combining generalized Hough transform (GHT) with local spectral features.

More recently, the state of the art SED systems have used log mel-band energy features in DNN [9], and RNN-LSTM [10] networks trained for multi-label classification. Motivated by the good performance of RNN-LSTM over DNN as shown in [10], we continue to use the multi-label RNN-LSTM network.

The present state of the art polyphonic SED systems have been using a single channel of audio for sound event detection. Polyphonic events can potentially be tackled better if we had multichannel data. Just like humans use their two ears (two channels) to recognize and localize the sound events around them [13], we can also potentially train machines to learn sound events from multichannel of audio. Recently, Xiao et al [14] have successfully used spatial features from multichannel audio for far field automatic speech recognition (ASR) and shown considerable improvements over just using mono channel audio. This further motivates us to use spatial features for SED tasks. In this paper, we propose a spatial feature along with harmonic feature and prove its superiority over mono channel feature even with a small dataset of around 60 minutes.

The remaining of the paper is structured as follows. We describe in Section 2 the features used and the proposed approach. Section 3 presents a short introduction to RNNs and long short-term memory (LSTM) blocks. Section 4 presents the experimental set-up and results on a database of real life recordings. Finally, we present our conclusions in Section 5.

2. SOUND EVENT DETECTION

The sound event detection task involves identifying temporally the locations of sound event and assigning them to one among the known set of labels. Sound events in real life have no fixed pattern. Different contexts, for example, forest, city, and home have a different variety of sound events. They can be of different sparsity based on the context, and can occur in isolation or be completely overlapped with other sound events. While recognizing isolated sounds have been done with an appreciable accuracy [15], detecting the mixture of labels in an overlapped sound event is a challenging task, where still a considerable amount of improvements can be made. Figure 2 shows a snippet of sound event annotation, where three sound events - speech, car, and dog bark happen to occur. At time frame *t*, two events - speech and car are overlapping. An ideal SED system should be able to handle such overlapping events.

The human auditory system has been successfully exploiting the stereo (multichannel) audio information it receives at its ears to

The research leading to these results has received funding from the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND, and Google Faculty Research Award project “Acoustic Event Detection and Classification Using Deep Recurrent Neural Networks”. The authors also wish to acknowledge CSC-IT Center for Science, Finland, for computational resources.

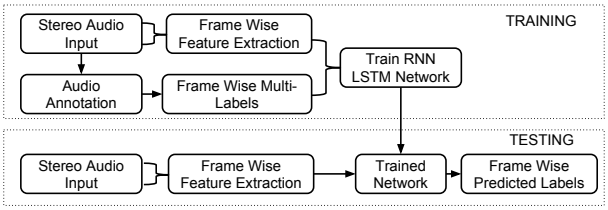


Figure 1: Framework of the training and testing procedure for the proposed system.

isolate, localize and classify the sound events. A similar set up is envisioned and implemented, where the sound event detection system gets a stereo input and suitable spatial features are implemented to localize and classify sound events.

The proposed sound event detection system, shown in Figure 1, works on real life multichannel audio recordings and aims at detecting and classifying isolated and overlapping sound events.

Three sets of features -log mel-band energies, pitch frequency, and its periodicity, and time difference of arrival (TDOA) in subbands, are extracted from the stereo audio. All features are extracted at a hop length of 20 ms to have consistency across features.

2.1. Log mel-band Energy

Log mel-band energies have been used for mono channel sound event detection extensively [9][10][16] and have proven to be good features. In the proposed system we continue to use log mel-band energies, and extract it for both the stereo channels. This is motivated from the idea that human auditory system exploits the interaural intensity difference (IID) for spatial localization of sound source [13]. Neural networks are capable of performing linear operations, which includes the difference. Therefore, when trained on the stereo log mel-band energy data, it will learn to obtain information similar to IID.

Each channel of the audio is divided into 40 ms frames with 50% overlap using hamming window. Log mel-band energies are then extracted for each of the frames (mel in Table 1). We use 40 mel-bands spread across the entire spectrum.

2.2. Harmonic features

The pitch is an important perceptual feature of sound. Human listeners have evolved to identify different sounds using the pitch cues, and can make efficient use of pitch to acoustically separate each of the mixture in an overlapping sound event [17]. Uzkent et al [18] have shown improvement in accuracy of non speech environmental sound detection used pitch range along with MFCC's. Here we propose using the absolute pitch and its periodicity as the features ($pitch$ in Table 1).

The librosa implementation of pitch tracking [19] on thresholded parabolically-interpolated STFT [20] was used to estimate the pitch and periodicity.

Since we are handling multi-label classification it is intuitive to identify as many dominant fundamental frequencies as possible and use them to identify the sound events. The periodicity feature gives the confidence measure for the extracted pitch value and helps the classifier to make better decisions based on pitch.

The overlapping sound events in the training data (Section 4.1) did not have more than three events overlapping at a time, hence we have limited ourselves to using the top three dominant pitch values per frame. So, for each of the channels, top three pitch values, and its respective periodicity values are extracted at every frame in 100-4000 Hz frequency range ($pitch3$ in Table 1).

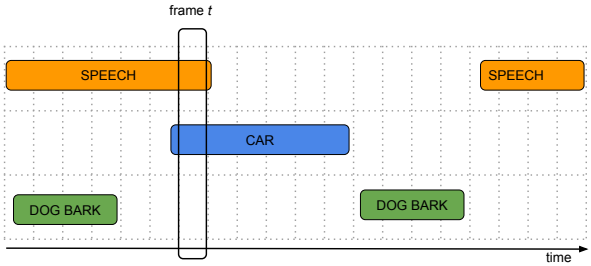


Figure 2: Sound events in a real life scenario can occur in isolation or overlapped. We see that at frame *t*, speech and car events are overlapping.

2.3. Time difference of arrival (TDOA) features

Overlapping sound events have forever troubled classification systems. This is mainly because the feature vector for the overlapped frame is a combination of different sound events. But, human listeners have been able to successfully identify each of the overlapping sound events by isolating and localizing the source spatially. This has been possible due to the interaural time delay (ITD) [13]

Each sound event has its own frequency band, some occur in low frequencies, some in high, and some occur all across the frequency band. If we can divide the frequency spectrum into different bands, and identify the spatial location of the sound source in each of these bands, then this is an extra dimension of the feature, which the classifier can learn to estimate the number of possible sources in each frame, and their orientation in the space. We implement this by dividing the spectral frame into five mel-bands and calculating the time difference of arrival (TDOA) at each of these bands.

For example, if a non-overlapping isolated sound event is spread across the entire frequency range, and we are calculating the TDOA in five mel-bands. We should have the same TDOA values for each of the bands. However, if we have two overlapping sounds S_1 and S_2, where S_1 is spread in the first two bands and S_2 is spread in the last two bands. The feature vector will have different TDOA values for each of the sounds, which the classifier can learn to isolate and identify them as separate sound events.

The TDOA can be estimated using the generalized cross-correlation with phase-based weighting (GCC-PHAT) [21]. Here, we extract the correlation for each mel-band separately:

$$R_b(\Delta_{12}, t) = \sum_{k=0}^{N-1} H_b(k) \frac{X_1(k,t) \cdot X_2^*(k,t)}{|X_1(k,t)||X_2(k,t)|} e^{i2\pi k \Delta_{12}/N}, \quad (1)$$

where N is the number of frequency bands, $X(k,t)$ is the FFT coefficient of the kth frequency band at time frame t and the subscript specifies the channel number, $H_b(k)$ is the magnitude response of the bth mel-band of total of B bands and Δ_{12} is the sample delay value between channels. The TDOA is extracted as the location of correlation peak magnitude for each mel-band and time frame.

$$\tau(b,t) = \underset{\Delta_{12}}{\text{argmax}} \{R_b(\Delta_{12}, t)\} \quad (2)$$

The maximum and minimum TDOA values are truncated between values $-2\tau_{\text{max}}, 2\tau_{\text{max}}$, where τ_{max} is the maximum sample delay between a sound wave traveling between microphones.

Feature Name	Length	Description
mel	40	Log mel-band energy extracted on a single channel of audio
$pitch$	2	Most dominant pitch value and periodicity extracted on a single channel
$pitch3$	6	Top three dominant pitch and periodicity values extracted on a single channel
$tdoa$	5	Median of multi-window TDOA's extracted from stereo audio
$tdoa3$	15	Concatenated multi-window TDOA's extracted from stereo audio

Table 1: Definitions of acoustic features proposed for sound event detection.

The sound events in the training set were seen to be varying from 50 ms to a few seconds. In order to accommodate such variable length sound events, TDOA was calculated in three different window lengths — 120, 240 and 480 ms, with a constant hop length of 20 ms. The TDOA values of these three windows were concatenated for each mel-band to form one set of TDOA features. So, TDOA values extracted in five mel-band, and for three window lengths, on concatenation gives 15 TDOA values per frame ($tdoa3$ in Table 1).

TDOA values in small windows are generally very noisy and unreliable. To overcome this, the median of the TDOA values from the above three different window lengths for each sub-band of the frame was used as the second set of TDOA features ($tdoa$ in Table 1). Post filtering across window lengths, the TDOA values in each mel-band were also median filtered temporally using a kernel of length three to remove outliers.

3. MULTI-LABEL RECURRENT NEURAL NETWORK BASED SOUND EVENT DETECTION

Deep neural networks have shown to perform well on complex pattern recognition tasks, such as speech recognition [22], image recognition [23] and machine translation [24]. A deep neural network typically computes a map from an input to an output space through several subsequent matrix multiplications and non-linear activation functions. The parameters of the model, i.e. its weights and biases, are iteratively adjusted using a form of optimization such as gradient descent.

When the network is a directed acyclic graph, i.e. information is only propagated forward, it is known as a feedforward neural network (FNN). When there are feedback connections the model is called a recurrent neural network (RNN). An RNN can incorporate information from previous timesteps in its hidden layers, thus providing context information for tasks based on sequential data, such as temporal context in audio tasks. Complex RNN architectures — such as long short-term memory (LSTM) [25] — have been proposed in recent years in order to attenuate the vanishing gradient problem [26]. LSTM is currently the most widely used form of RNN, and the one used in this work as well.

In SED, RNNs can be used to predict probabilities for each class to be active in a given frame at timestep t. The input to the network is a sequence of feature vectors $\mathbf{x}(\mathrm{t})$; the network computes hidden activations for each hidden layer, and at the output layer a vector of predictions for each class $\mathbf{y}(\mathrm{t})$. A sigmoid activation function is used at the output layer in order to allow several classes to be predicted as active simultaneously. By thresholding the predictions at the output layer it is possible to obtain a binary activity matrix.

3.1. Neural network configurations

For each recording, we obtain a sequence of feature vectors, which is normalized to zero mean and unit variance, and the scaling parameters are saved for normalizing the test feature vectors. The sequences are further split into non-overlapping sequences of length 25 frames. Each of these frames has a target binary vector, indicating which classes are present in the feature vector.

We use a multi-label RNN-LSTM with two hidden layers each having 32 LSTM units. The number of units in the input layer depends on the length of the feature being used. The output layer has one neuron for each class. The network is trained by back propagation through time (BPTT) [27] using binary cross-entropy as loss function, Adam optimizer [28] and block mixing [10] data augmentation. Early stopping is used to reduce over-fitting, the training is halted if the segment based error rate (ER) (see Section 4.2) on the validation set does not decrease for 100 epochs.

At test time we use scaling parameters estimated on training data to scale the feature vectors and present them in non-overlapping sequences of 25 frames, and threshold the outputs with a fixed threshold of 0.5, i.e., we mark an event is active if the posterior in the output layer of network is greater than 0.5 and otherwise inactive.

4. EVALUATION AND RESULTS

4.1. Dataset

We evaluate the proposed SED system on the development subset of TUT sound events detection 2016 database [1]. This database has stereo recordings which were collected using binaural Soundman OKM II Klassik/studio A3 electret in-ear microphones and Roland Edirol R09 wave recorder using 44.1 kHz sampling rate and 24-bit resolution. It contains two contexts - home and residential area. Home context has 10 recordings with 11 sound event classes and the residential area context has 12 recordings with 7 classes. The length of these recordings is between 3-5 minutes.

In the development subset provided, each of the context data is already partitioned into four folds of training and test data. The test data was collected such that each recording is used exactly once as the test, and the classes in it are always a subset of the classes in the training data. Also, 20% of the training data recordings in each fold were selected randomly to be used as validation data. The same validation data was used across all our evaluations.

4.2. Metrics

We perform the evaluation of our system in a similar fashion as [1] which uses the established metrics for sound event detection defined in [30]. The error rate (ER) and F-scores are calculated on one second long segments. The results from all the folds are combined to produce a single evaluation. This is done to avoid biases caused due to data imbalance between folds as discussed in [31].

4.3. Results

The baseline system for the dataset [1] uses 20 static (excluding the 0th coefficient), 20 delta and 20 acceleration MFCC coefficients

	Feature combination	Home		Residential area		Average	
		ER	F (%)	ER	F (%)	ER	F (%)
Baseline system using GMM classifier in DCASE 2016 [1][29]	$mfcc; delta; acc$	0.96	15.9	0.86	31.5	0.91	23.7
Mono channel feature With RNN-LSTM network	mel_1	**0.94**	**27.4**	0.88	38.3	0.91	32.9
Hybrid (mono and stereo) features with RNN-LSTM network	$mel_1; pitch_1$	0.97	25.4	0.85	43.4	0.91	34.4
	$mel_1; pitch3_1$	0.96	27.6	0.88	43.9	0.92	35.7
	$mel_1; tdoa$	1.02	19.4	0.89	40.2	0.96	29.8
	$mel_1; tdoa3$	0.98	25.9	0.87	40.5	0.92	33.2
Stereo features with RNN-LSTM network	mel_2	1.03	25.4	0.84	45.9	0.93	35.6
	$mel_2; pitch_2$	1.03	24.9	0.93	40.9	0.98	32.9
	$mel_2; pitch3_2$	0.97	26.6	0.88	41.7	0.92	34.2
	$mel_2; tdoa$	1.01	24.4	**0.82**	**46.4**	**0.91**	**35.4**
	$mel_2; tdoa3$	0.96	24.9	0.86	38.5	0.91	31.7
	$mel_2; tdoa3; pitch_2$	0.97	25.7	0.85	43.1	0.91	34.4
	$mel_2; tdoa3; pitch3_2$	0.99	26.5	0.91	35.2	0.95	30.9
	$mel_2; tdoa; pitch_2$	0.98	24.7	0.87	43.8	0.92	34.2
	$mel_2; tdoa; pitch3_2$	0.94	26.3	0.89	40.5	0.91	33.4

Table 2: Segment based error rate (ER) and F-score achieved for different feature combinations in home and residential area contexts for the development set. The features listed in Table 1 are used in different combinations with the proposed RNN-LSTM network. The subscripts '$_1$' and '$_2$' in the feature combinations column represent how many channels the features were extracted on. For example, feature combination $mel_2; tdoa; pitch_2$ means that the final feature vector has log mel-band energies, most dominant pitch and periodicity values extracted on both the stereo channels, and the time difference of arrival (TDOA) calculated between the stereo channels. The highlighted ER and F-score pair for each context is the best ER score achieved.

extracted on mono audio with 40 ms frames and 20 ms hop length. A Gaussian mixture model (GMM) consisting of 16 Gaussians is then trained for each of the positive and negative values of the class. This baseline system gives a context average ER of 0.91 and F-score of 23.%. An ideal system should have an ER of 0 and an F-score of 100%.

In Table 2 we compare the segment based ER and F-score for different combinations of proposed spatial and harmonic features. In all these evaluations, only the size of the input layer changes based on the feature set, with the rest of the configurations in the RNN-LSTM network remaining unchanged.

Mono channel audio was created by averaging the stereo channels in order to compare the performance of the proposed spatial and harmonic features for multichannel audio. One of the present state of the art SED system for mono channel is proposed in [10]. An RNN-LSTM network is trained in a similar fashion with log mel-band energy feature (Section 2.1) and evaluated. Across contexts, the F-score was seen to be better than the GMM baseline system with comparable ER. Here onwards we use this mono-channel log mel-band feature and RNN-LSTM network configuration result as a baseline for comparisons.

A set of hybrid combinations were tried as shown in Table 2. All combinations other than $mel_1; tdoa$ performed better than the baseline across contexts in F-score.

Finally, the full spectrum of proposed spatial and harmonic features were evaluated in different combinations with RNN-LSTM network. With a couple of exceptions - $mel_2; pitch_2$ and $mel_2; tdoa3; pitch3_2$, all the combinations of features performed equal to or better than the baseline in average F-scores, with marginally similar average ER as baseline. Given the dataset size of around 60 minutes, it is difficult to conclusively say that the binaural features are far superior to monaural features; but they surely look promising.

Binaural features - mel_2 and $mel_2; tdoa; pitch_2$ in Table 3 were submitted to the DCASE 2016 challenge [29], where they were evaluated as the top performing systems. Monaural feature mel_1 was submitted unofficially to compare the performance with binaural features. The hyper-parameters of the network were tuned before the submission, and hence the development set results in Table 3 are different from Table 2. Three hidden layers with 16 LSTM units each were used for mel_2, while mel_1 and $mel_2; tdoa; pitch_2$ were trained with two layers each having 16 LSTM units.

Feature combination	Evaluation dataset		Development dataset	
	ER	F (%)	ER	F (%)
mel_1	0.79	46.6	0.90	35.3
mel_2	0.80	47.8	0.88	34.7
$mel_2; tdoa; pitch_2$	0.88	37.9	0.87	34.8

Table 3: Comparison of segment based error rate (ER) and F-score for development and evaluation dataset. The evaluation dataset scores are the result of DCASE 2016 challenge [29].

5. CONCLUSION

In this paper, we proposed to use spatial and harmonic features for multi-label sound event detection along with RNN-LSTM networks. The evaluation was done on a limited dataset size of 60 mins, which included four cross validation data for two contexts — home and residential area. The proposed multi-channel features were seen to be performing substantially better than the baseline system using mono-channel features.

Future work will concentrate on finding novel data augmentation techniques. Augmenting spatial features is an unexplored space, and will be a challenge worth looking into. Concerning the model, further studies can be done on different configurations of RNN like extending them to bidirectional RNN's and coupling with convolutional neural networks.

6. REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *In 24rd European Signal Processing Conference 2016 (EUSIPCO 2016)*, 2016.

[2] S. Chu, S. Narayanan, and C. J. Kuo, "Environmental sound recognition with timefrequency audio features," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, 2009, p. 1142.

[3] S. Chu, S. Narayanan, C. C. J. Kuo, and M. J. Mataric, "Where am I? Scene recognition for mobile robots using audio features," in *IEEE Int. Conf. Multimedia and Expo (ICME)*, 2006, p. 885.

[4] A. Harma, M. F. McKinney, and J. Skowronek, "Automatic surveillance of the acoustic activity in our living environment," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[5] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, "Audio keywords generation for sports video analysis," in *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 4, no. 2, 2008, p. 11.

[6] D. Zhang and D. Ellis, "Detecting sound events in basketball video archive," in *Dept. Electronic Eng., Columbia Univ., New York*, 2001.

[7] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," in *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, 2013, p. 1.

[8] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," in *Pattern Recognition Letters*, vol. 34, no. 9, 2013, p. 1085.

[9] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi-label deep neural networks," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2015.

[10] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[11] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), Vancouver, Canada*, 2013,, p. 8677.

[12] O. Dikmen and A. Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013,, p. 1.

[13] J. W. Strutt, "On our perception of sound direction," in *Philosophical Magazine*, vol. 13, 1907,, p. 214.

[14] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and DongYu, "Deep beamforming networks for multi-channel speech recognition," in *ICASSP*, 2016,.

[15] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *European Signal Processing Conference (EUSIPCO 2014)*, 2014,.

[16] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[17] A. S. Bregman, "Auditory scene analysis: The perceptual organization of sound," in *MIT Press, Cambridge, MA*, 1990.

[18] B. Uzkent, B. D. Barkana, and H. Cevikalp, "Non-speech environmental sound classification using svms with a new set of features," in *International Journal of Innovative Computing, Information and Control*, 2012, p. 3511.

[19] B. McFee, M. McVicar, C. Raffel, D. Liang, O. Nieto, E. Battenberg, J. Moore, D. Ellis, R. YAMAMOTO, R. Bittner, D. Repetto, P. Viktorin, J. F. Santos, and A. Holovaty, "librosa: 0.4.1," Oct. 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.32193

[20] J. O. Smith, *Sinusoidal Peak Interpolation, in Spectral Audio Signal Processing*, accessed 23.06.2016, online book, 2011 edition. [Online]. Available: https://ccrma.stanford.edu/~jos/sasp/Sinusoidal_Peak_Interpolation.htm

[21] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug 1976.

[22] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[27] P. J. Werbos, "Backpropagation through time: what it does and how to do it," in *Proceedings of the IEEE,*, vol. 78 no. 10, 1990, p. 15501560.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *arXiv:1412.6980 [cs.LG]*, December, 2014.

[29] "Detection and classification of acoustic scenes and events," 2016. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-real-life-audio

[30] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," in *Applied Sciences*, vol. 6(6):162, 2016.

[31] G. Forman and M. Scholz, "Apples-to-apples in cross validation studies: Pitfalls in classifier performance measurement," in *SIGKDD Explor. Newsl.*, vol. 12, no. 1, Nov. 2010, p. 49.

ACOUSTIC SCENE CLASSIFICATION USING PARALLEL COMBINATION OF LSTM AND CNN

Soo Hyun Bae, Inkyu Choi and Nam Soo Kim

Seoul National University
Department of Electrical and Computer Engineering and INMC
Gwanak P.O.Box 34, Seoul 151-744, Korea
{shbae, ikchoi}@hi.snu.ac.kr, nkim@snu.ac.kr

ABSTRACT

Deep neural networks (DNNs) have recently achieved a great success in various learning task, and have also been used for classification of environmental sounds. While DNNs are showing their potential in the classification task, they cannot fully utilize the temporal information. In this paper, we propose a neural network architecture for the purpose of using sequential information. The proposed structure is composed of two separated lower networks and one upper network. We refer to these as LSTM layers, CNN layers and connected layers, respectively. The LSTM layers extract the sequential information from consecutive audio features. The CNN layers learn the spectro-temporal locality from spectrogram images. Finally, the connected layers summarize the outputs of two networks to take advantage of the complementary features of the LSTM and CNN by combining them. To compare the proposed method with other neural networks, we conducted a number of experiments on the TUT acoustic scenes 2016 dataset which consists of recordings from various acoustic scenes. By using the proposed combination structure, we achieved higher performance compared to the conventional DNN, CNN and LSTM architecture.

Index Terms— Deep learning, sequence learning, combination of LSTM and CNN, acoustic scene classification

1. INTRODUCTION

Acoustic scene classification aims to recognize the environmental sounds that occur for a period of time. Many approaches have been proposed for acoustic scene classification including feature representation, classification models, and post-processing. The support vector machine (SVM) was one of the most successful learning model in a number of scene classification tasks. As SVM is a binary classifier, some additional methods must be combined to apply them to the multi-class problems, such as the use of tree or clustering schemes [1, 2]. Furthermore, many machine learning-based scene classification techniques were proposed in the detection and classification of acoustic scenes and events (DCASE) challenge 2013 [3, 4, 5].

However, as deep learning techniques have been widely used on various learning tasks, researchers have started to apply them to acoustic scene classification as well [6, 7]. In [8], a DNN-based sound event classification algorithm was performed with several image features.

Deep neural networks (DNNs) are powerful pattern classifier which enables the networks to learn the highly nonlinear relationships between the input features and output targets. Though the DNNs work well in the classification task, they cannot be used to map sequences to sequences because of their structural limitations. To overcome this shortcoming, recurrent neural networks (RNNs) and long short-term memory (LSTM), which is a special type of RNN, have been applied to sequence learning [9].

DNNs can only map from present input vector to output vector, whereas LSTM can map from sequence to output sequence or vector. Therefore, LSTM can learn the temporal information through consecutive input vectors. The authors in [10] and [11] proposed sound event detection techniques based on bi-directional LSTM which yielded higher performance compared to the DNNs. Unlike sound events which occur in a short time frame, acoustic scenes are maintained for relatively longer range. Thus, applying RNNs to the acoustic scene classification will improve the performance.

Other approaches were proposed to use convolutional neural networks (CNNs) with spectrogram image features (SIF) [12]. In [13], the authors addressed the importance of spectro-temporal locality and proposed a CNN-based acoustic event detection algorithm.

In this paper, we propose to combine the LSTM and CNNs in parallel as lower networks in order to exploit sequential correlation and local spectro-temporal information. In the LSTM layers, sequences of Mel-frequency cepstral coefficients (MFCCs) features are utilized as input in order to extract the sequential information. The CNN layers learn the spectro-temporal locality from SIF, and SIF clips are set to have the same length with the timestep of LSTM inputs. The outputs of the two separated layers are combined by the connected layers which are able to learn complementary features of LSTM and CNN. To compare the performance of the proposed method with various neural networks, we conducted a number of experiments on the TUT acoustic scenes 2016 dataset [14]. The results revealed that the combination of LSTM and CNN outperforms the conventional DNN, CNN and LSTM architecture with respect to classification accuracy.

2. LONG SHORT-TERM MEMORY

The key idea of RNN is that the recurrent connections between the hidden layers allow the memory of previous inputs to retain internal state, which can affect the outputs. However, RNN mainly have two issues to solve in the training phase: vanishing gradient and exploding gradient problems [15]. When computing the derivatives of activation function in the back propagation process, long-term components may go exponentially fast to zero. This makes the model hard to learn the correlation between temporally distant inputs. Meanwhile, when the gradient grows exponentially during training, the

exploding gradient problem occurs. In order to solve this problem, the LSTM architecture was proposed [16]. LSTM layers are composed of recurrently connected memory blocks in which one memory cell contains three multiplicative gates. The gates perform continuous analogues of write, read and reset operations which enable the network to utilize the temporal information over a period of time.

3. PARALLEL COMBINATION OF LSTM AND CNN

In this section, we describe our approach to improve the classification accuracy of acoustic scene. The schematic of the proposed neural networks structure can be seen in Figure 1.

3.1. Feature extraction

In the proposed system, different types of neural networks are combined in parallel. Thus, each network accept different form of input feature. The LSTM layers utilize sequence of acoustic feature, but the CNN layers use spectrogram images. As inputs for the CNN layers, the SIF are extracted from the sound spectrogram [8, 12, 17]. Firstly, a spectrogram is generated by short-time Fourier transform. Given audio frame $s(n)$ segmented by length N and Hamming window $w(n)$, the short time spectral column $\mathbf{F}(f,t)$ at time t is computed as,

$$\mathbf{F}(f,t) = \left| \sum_{n=0}^{N-1} s(n)w(n)e^{\frac{-j2\pi nf}{N}} \right| \quad (1)$$

for $f = 0, ..., N/2$. In order to generate a spectrogram image which has K-bin frequency resolution, down sampling is performed by using a window of length $W = N/2K$ as follows:

$$\mathbf{F}_{down}(f,t) = \sum_{i=0}^{W-1} \mathbf{F}(f+i,t)/W, \quad (2)$$

for $f = 0, ..., (K-1)$. Finally, a simple de-noising method is performed by subtracting each minimum frequency bin value in a frame-wise manner as follows:

$$\mathbf{F}_{dn}(f,t) = \mathbf{F}_{down}(f,t) - \min_t\{\mathbf{F}_{down}(f,t)\} \quad (3)$$

for $f = 0, ..., (K-1)$. In the proposed system, the extracted SIF has size of $K \times \tau$, where τ represents the time resolution which is also identical to the timesteps in the LSTM layers.

3.2. LSTM layers

The hidden layers of LSTM have self-recurrent weights. These enable the cell in the memory block to retain previous information. In the proposed system, τ vectors are used for sequential learning. The lower part in Figure 1 depicts how the sequences are trained through the LSTM layers. Previous $\tau - 1$ vectors and one present vector are forwarded to the recurrent layer sequentially. If the MFCC vectors from $x_{t-\tau+1}$ to x_t are used as the present inputs, vectors from $x_{t-\tau+2}$ to x_{t+1} will be used as the next input sequence. The output vector z_t^{LSTM} is extracted from input MFCC sequence x_t^{LSTM} through the LSTM layers, where $x_t^{LSTM} = [x_{t-\tau+1}, ..., x_t]$.

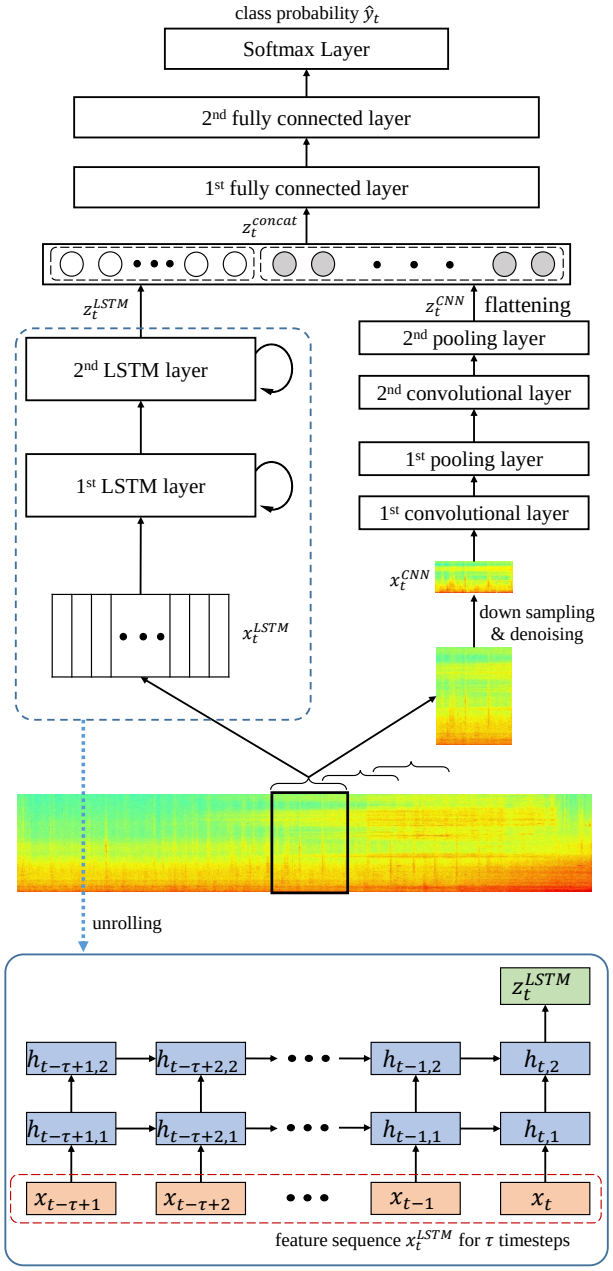


Figure 1: Neural network structure for the proposed technique.

3.3. CNN layers

From Section 3.1, SIF x_t^{CNN}, which is a $F \times \tau$ matrix, are extracted. The convolutional layer performs 2-dimensional convolution between the spectrogram image and the pre-defined linear filters. To enable the network to extract complementary features and learn the characteristics of input SIF, a number of filters with different functions are used. Thus, if we apply K different filters to the spectrogram image, K different filtered images are generated in the convolutional layer. The filtered spectrogram images are forwarded to the pooling layer which conducts down sampling. Especially,

max pooling divides the input image into a set of non-overlapping sub-regions and selects the maximum value. By reducing the spatial size of representation via pooling, the most dominant feature in the sub-region is extracted. The pooling layer operates independently on every filtered image and resizes them spatially. In the last pooling layer, the resized outputs are rearranged in order to fully connect with the upper layer. The flattened output vector z_t^{CNN} is extracted from x_t^{CNN} through the CNN layers

3.4. Connected layer of LSTM and CNN

In [18], long-term recurrent convolution network (LRCN) model was proposed for visual recognition. LRCN is a consecutive structure of CNN and LSTM. LRCN processes the variable-length input with a CNN, whose outputs are fed into LSTM network, which finally predicts the class of the input. In [19], a cascade structure was used for voice search. Compared to the method mentioned above, the proposed network forms a parallel structure in which LSTM and CNN accept different inputs separately. Concatenated vector z_t^{concat} is forwarded to the fully connected layer, where $z_t^{concat} = [z_t^{LSTM}, z_t^{CNN}]$. The connected layers can train the complementary information of LSTM and CNN. These enable the proposed model to learn the sequential information and spectro-temporal information, simultaneously. Finally, the class probability $\hat{y}_t$ is predicted through the softmax layer.

4. EVALUATION

To assess the performance of the proposed method, we conducted a number of experiments on the TUT acoustic scenes 2016 dataset which consists of recordings from various acoustic scenes. The dataset contains 1170 recordings of total 9.75 hours with 15 different classes. Audio signals sampled at 44.1 kHz sampling frequency were divided into 40 ms frames with 50% hop size. Experiments were conducted using 4-fold cross validation. The final results were obtained by averaging over all evaluation folds.

We evaluated the classification accuracy using two measures: frame-based accuracy and segment ($30s$)-based accuracy. Due to the softmax output layer of our networks, probability distributions among the J class labels were obtained individually. Given z_t^{concat}, the predicted class label at t frame was computed by,

$$C_{frame} = \arg\max_j P(\hat{y}_t = j | z_t^{concat}) \quad (4)$$

where j denotes class index. To obtain the class label of the entire audio segment, the likelihood was computed follows as:

$$C_{segment} = \arg\max_j \sum_{t=1}^{T} \log(P(\hat{y}_t = j | z_t^{concat})), \quad (5)$$

where T represents the number of frames in the one audio segment.

4.1. Neural networks setup

All networks in our experiments were trained using mean squared error as the loss function supervised by one-hot encoding class vectors. The randomly ordered mini-batches in each epoch was set to be 256. After a mini-batch was processed, the weights were updated using adadelta [20]. In order to mitigate the over-fitting problem in the training phase, we used the dropout technique which has already proved its regularization capability [21]. The output layer contained 15 softmax nodes identical to the number of scenes.

Table 1: Frame-based classification accuracy (%) on IEEE DCASE 2016 Challenge Task 1 Development Dataset.

Scene	DNN	CNN	LSTM	CNN-LSTM
beach	76.56	65.29	79.86	81.26
bus	44.69	62.61	56.21	60.99
cafe/restaurant	47.79	61.89	57.72	57.12
car	75.49	71.11	85.51	80.57
city center	80.41	79.13	89.26	91.25
forest path	87.24	72.15	91.69	92.22
grocery store	77.19	57.39	83.07	84.71
home	66.28	72.71	52.70	55.39
library	64.07	71.27	69.29	72.55
metro station	85.71	85.76	82.52	82.47
office	83.40	78.93	82.97	89.09
park	38.24	36.11	48.89	43.88
residential area	61.87	51.71	52.54	57.74
train	22.46	38.87	24.42	38.21
tram	73.57	56.82	72.99	76.46
Overall acc	65.66	64.12	68.64	**70.92**

4.1.1. DNN

As a baseline system, we built a DNN which has three hidden layers with 512 hidden units each and used the ReLU activation in the hidden layers. The input features were 60-dimensional MFCC features including both delta and acceleration MFCC coefficients. Input layer was composed of a concatenation of 9 input frames (the current frame and the four previous and four next frames) resulting in 540 input units. To regularize the network, we used dropout with a probability of 40% for all hidden layers.

4.1.2. CNN

The CNN architecture for the baseline system comprised two convolutional layers, two pooling layers and one fully connected layer with softmax layer on the top. The input features were $F \times \tau$ size SIF, where F=40 and τ=40. In the first convolutional layer, the input SIF is convolved with 32 filters of fixed size 5×5. The first pooling layer then reduce the size of filtered SIF. We utilized max-pooling with kernel size 2×2 for all pooling layers. As an activation function, ReLU was applied. The second convolutional layer perform convolution between the output of the pooling layer and 16 filters of fixed size 5×5. After the second pooling is performed, the flattened output is combined with fully connected layer with 512 units. Dropout was only used after the second pooling layer and the fully connected layer with probabilities 30% and 40%, respectively.

4.1.3. LSTM

The network had two hidden layers with 256 LSTM units each and one feed-forward layer with 512 ReLU units. The structure of two LSTM layers is identical to the lower part in Figure 1. The input sequence consisted of 40 frames of 60-dimensional MFCC features. Dropout was applied with a probability of 40% for all layers. The output layer was identical to the mentioned in the previous section.

Table 2: Segment-based (30s) classification accuracy (%) on IEEE DCASE 2016 Challenge Task 1 Development Dataset. Asterisk(*) CNN-LSTM represents the accuracy on Evaluation Dataset.

Scene	Base.	DNN	CNN	LSTM	CNN-LSTM	*CNN-LSTM
beach	69.3	84.62	73.08	88.46	88.46	84.6
bus	79.6	51.28	88.46	67.95	65.38	100
cafe/rest.	83.2	58.97	73.08	67.95	60.26	61.5
car	87.2	78.21	73.08	88.46	89.74	88.5
city center	85.5	92.31	91.03	93.59	97.44	92.3
forest path	81.0	93.59	82.05	98.72	97.44	100
grocery store	65.0	83.33	71.79	85.90	91.03	96.2
home	82.1	80.77	89.74	64.10	70.51	88.5
library	50.4	75.64	83.33	76.92	76.92	46.2
metro station	94.7	94.87	100.0	92.31	94.87	88.5
office	98.6	93.59	96.15	87.18	96.15	100
park	13.9	41.03	43.59	57.69	52.56	96.2
resident. area	77.7	87.18	75.64	73.08	74.36	65.4
train	34.9	25.64	46.15	29.49	43.59	53.8
tram	85.4	88.46	82.05	88.46	88.46	100
correct	-	881	912	905	**926**	-
Overall acc	72.6	75.30	77.95	77.35	**79.15**	**84.1**

4.1.4. Combination of LSTM and CNN

As a proposed system, we built a combined structure of LSTM and CNN in parallel. The network setup and structure of LSTM part and CNN part was identical to the aforementioned networks in Section 4.1.2 and 4.1.3, respectively. To combine and further train the two separated networks, we used fully connected layers. The connected layers were consisted of two hidden layers with 512 ReLU units each.

4.2. Results and discussion

We compared the average accuracies over all scenes for the conventional DNN, CNN, LSTM, and the proposed network. The frame-based classification results are given in Table 1. Table 2 shows the segment-based classification accuracy, where the *correct* represents the number of correctly classified segments among the total 1170 segments. The proposed method achieved higher accuracy than other networks in both frame-based and segment-based classification.

Though the combined neural network achieved higher performance on average, it did not give the best classification results across all scenes. In the *bus* case, CNN outperformed other networks. In the *park* case, LSTM had better result. In the *residential area* case, DNN achieved higher performance. This can be interpreted that the proposed network cannot fully train some acoustic scenes, and these scenes may not contain enough temporal information. Future research will deal with a more robust network architecture to extract distinct features of acoustic scenes.

The proposed method was found to improve classification performance and achieved an average accuracy of 79.15%. The baseline accuracy of audio scene classification task in DCASE 2016 challenge [14], which was based on MFCCs and GMMs, was 72.6%. Our method improved the performance by relative 6.6%. Finally, The accuracy on the evaluation dataset was 84.1%.

5. CONCLUSION

In this paper, in order to enhance the classification accuracy of acoustic scenes, we proposed a novel neural network structure which achieved higher performance compared with the conventional DNN, CNN and LSTM architecture in terms of both frame-based and segment-based accuracy. In the segment-based classification results, the proposed technique obtained improvement of 3.85%, 1.2% and 1.8% in comparison with DNN, CNN and LSTM architecture, respectively. By combining different networks in parallel, the proposed method was able to learn complementary information of LSTM and CNN. Future works will study other neural network architectures in order to extract distinct features of acoustic scenes.

6. ACKNOWLEDGEMENT

This research was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2015R1A2A1A15054343), and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-H8501-16-1016) supervised by the IITP(Institute for Information & communications Technology Promotion).

7. REFERENCES

[1] L. Lu, H.-J. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia systems*, vol. 8, no. 6, pp. 482–492, 2003.

[2] A. Temko and C. Nadeu, "Classification of acoustic events using svm-based clustering schemes," *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006.

[3] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

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

[5] 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, 2015.

[6] Z. Kons and O. Toledo-Ronen, "Audio event classification using deep neural networks," in *INTERSPEECH*, 2013, pp. 1482–1486.

[7] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *European Signal Processing Conference (EUSIPCO)*, 2014, pp. 506–510.

[8] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,*, vol. 23, no. 3, pp. 540–552, 2015.

[9] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 5–13.

[10] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2742–2746.

[11] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6440–6444.

[12] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 559–563.

[13] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, "Exploiting spectro-temporal locality in deep learning based acoustic event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 1–12, 2015.

[14] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *European Signal Processing Conference (EUSIPCO)*, 2016.

[15] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2012.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," *arXiv preprint arXiv:1604.06338*, 2016.

[18] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[20] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

DNN-BASED SOUND EVENT DETECTION WITH EXEMPLAR-BASED APPROACH FOR NOISE REDUCTION

Inkyu Choi, Kisoo Kwon, Soo Hyun Bae and Nam Soo Kim

Seoul National University
Department of Electrical and Computer Engineering and INMC
Gwanak P.O.Box 34, Seoul 151-744, Korea
{ikchoi, kskwon, shbae}@hi.snu.ac.kr, nkim@snu.ac.kr

ABSTRACT

In this paper, we present a sound event detection system based on a deep neural network (DNN). Exemplar-based noise reduction approach is proposed for enhancing mel-band energy feature. Multi-label DNN classifier is trained for polyphonic event detection. The system is evaluated on IEEE DCASE 2016 Challenge Task 2 Datasets. The result on the evaluation set yields up to 0.787 and 0.3660 in terms of F-Score and error rate on segment-based metric, respectively.

Index Terms— Sound event detection, deep neural network, exemplar-based noise reduction

1. INTRODUCTION

Sound event detection (SED) plays an important role in computational auditory scene analysis, with a specific purpose of detecting meaningful sounds, generally referred to sound events. Detecting sound events such as speech, footstep and door slam provides fundamental information for understanding the situation using acoustic signal. Furthermore, SED could be utilized in many applications, including automated surveillance systems, information retrieval, smart home systems and military applications.

Many previous works on SED were based on conventional speech recognition techniques. The most common approach is to use a system based on spectral features such as Mel-Frequency Cepstral Coefficients (MFCCs) and Hidden Markov Models (HMMs) for sound event classification [1, 2]. In recent works, approaches based on Support Vector Machine (SVM) [3, 4, 5] or non-negative matrix factorization (NMF) [6, 7, 8] were also proposed for SED. Most of the previous works were monophonic SED, which focused on detecting a single event at the same time. However, more than two events can happen simultaneously in real environments. In this case, conventional monophonic SED approaches may not be suitable for detecting overlapping events. Polyphonic SED aims to detect multiple sound events in the same time instance of the sound data. A polyphonic AED system that used MFCC for feature and HMMs as classifiers with consecutive passes of the Viterbi algorithm was proposed [9]. In [10], Generalized Hough transform (GHT) voting system has been used to recognize overlapping sound events. In another work, NMF-based approach was used for source separation and then events were detected from each stream [11]. Deep neural networks (DNNs) have shown good performance for polyphonic SED by modeling overlapping sound events in a natural way [12].

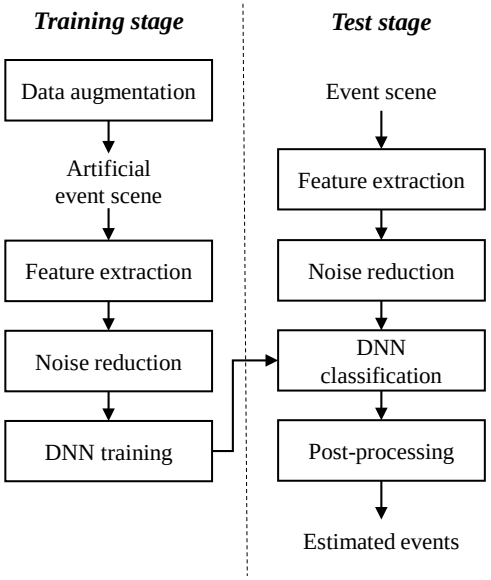


Figure 1: Flowchart of the proposed system

In this paper, we propose a DNN-based SED system. In the proposed system, data augmentation is performed to deal with data sparsity problem in small training dataset and generate polyphonic event examples. Exemplar-based noise reduction algorithm is proposed for feature enhancement. DNN classifier is trained for polyphonic event detection and adaptive thresholding algorithm is applied as a post-processing for robust event detection in noisy condition.

2. THE PROPOSED SED SYSTEM

The proposed system consists of 4 main processing stages. The overall system is illustrated in Fig. 1. First, data augmentation is performed to generate artificial sound event scenes which are used for training the classifier. In the second stage, mel-band energy features are extracted and enhanced by exemplar-based noise reduction. Third, the enhanced feature is fed to a DNN classifier. The features from artificial sound event scene are used for training the DNN classifier. In final stage, the sound events are detected by filtering and thresholding the output of the DNN classifier.

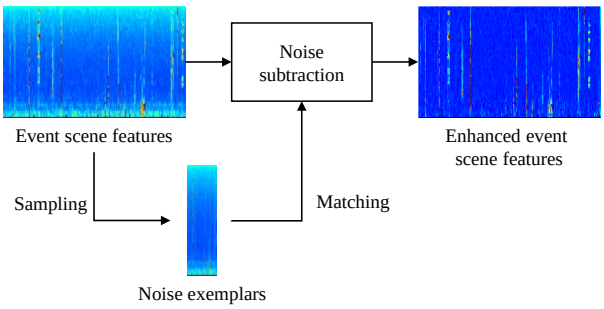


Figure 2: Exemplar-based approach for noise reduction

2.1. Data augmentation

DNNs have shown good performance as classifiers in many applications. When the training data is large, the DNN could learn from the variations presented in the training data under the same labels and make classifications that are robust to intra-class variations. However if the training data from each class is not sufficient to cover its intra-class variations, the DNN classifier trained with the data may have poor generalization ability, leading to low classification performance for test samples. In [13], data augmentation approach was used for training DNNs to deal with the data sparsity problem.

Unlike speech datasets which usually consist of hours of data or more, conventional sound event dataset is not sufficiently long enough to train a robust DNN classifier. Under this condition, data augmentation can help to enhance the performance of the DNN classifier by improving the generalization ability of the neural network. In recent research, data augmentation approach was performed for better performance in polyphonic SED [14]. In this paper, to construct the diverse sound event data from a small dataset, artificial event scenes are generated using data augmentation. In the artificial event scenes, events are overlapped to each other or manipulated by time stretching and power modification for diversity of dataset. These event scenes are corrupted by white, blue and pink noises.

2.2. Exemplar-based approach for noise reduction

In real life recordings, various noises exist and make it difficult to detect sound events correctly. To alleviate the effect of the noises, noise reduction is performed for feature enhancement. Since we assume that the test noise conditions are unknown, model adaptation-based approaches for noise robustness may not be suitable. In order to suppress unseen noises in test conditions, exemplar-based noise reduction approach is proposed. In this approach, noise exemplars are selected from the event scene features, then noise is directly subtracted from the event scene features by using the noise exemplars.

For each event scene, mel-band energy features are extracted and the features that have L1 norm corresponding to the lower 30% are considered to be noise candidates. From the candidates, K noise frames are selected randomly or using K-means algorithm for noise exemplars. For each frame, best matching noise exemplar that minimizes the noise estimation error, defined as in (1), is selected.

$$E_k = ||max(X_t - N_k, 0)||_1 + \alpha \cdot ||max(N_k - X_t, 0)||_1 \ . \quad (1)$$

E_k is the noise estimation error of a noise exemplar N_k and X_t is a feature vector at time index t. Noise estimation error E_k is the summation of under estimation error and over estimation error with ratio α. The selected noise exemplar is subtracted from the frame feature for noise reduction. The proposed noise reduction process is illustrated in Fig. 2.

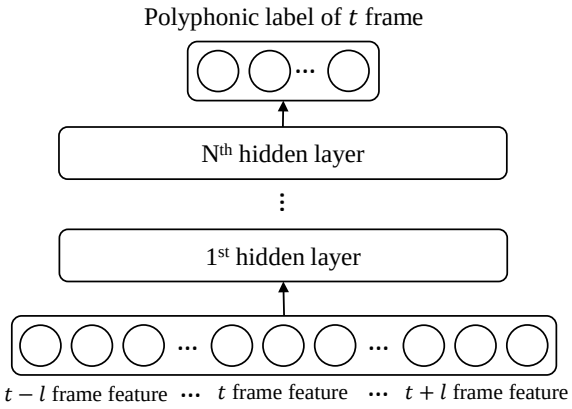


Figure 3: A DNN structure for the proposed SED system

2.3. DNN Classifier

In this paper, we trained a DNN-based classifier for SED. Unlike speech, sound events come from different physical sources so they possess unique characteristics that are distinct from one another. The DNN structure is employed to successfully represent distinct sound events in a single model. The DNN system for SED is illustrated in Fig. 3. The DNN consists of an input layer, a few hidden layers and an output layer which are fully connected to their adjacent layers. As for the input, the mel-band energy features enhanced by the proposed noise reduction approach are used. To consider temporal information, several adjacent frame features are concatenated for a single frame input. The output of the DNN is the estimated labels for input frames. The number of output unit is the same as that of the event classes, and each output unit is matched to each class. When the event exists in input frame, the output unit of the class is set to 1, otherwise it is set to 0. We used rectified linear activation function for hidden layers and sigmoid function units for the output layer.

Artificial event data generated by data augmentation is used for training the DNN classifier. In the fine-tuning stage, backpropagation algorithm with the minimum mean squared error (MMSE) function between the correct label and the estimated label is employed to train to the DNN. A stochastic gradient descent algorithm is performed in mini-batches to improve learning convergence. To deal with overfitting problem, we used the dropout technique which has already proved its regularization capability for training DNN [15].

2.4. Post-processing

The output of the DNN classifier is filtered for robust event detection. An averaging filter may help to remove outliers, but also discourage precise detection in onset or offset period of an event due to non-event periods nearby. For precise onset and offset detection, we used two filters: one of which is a sigmoid function and the other

is the former reflected about the y-axis. The former one is sensitive to the onset and the latter one is sensitive to the offset of an event. To detect both onset and offset of an event correctly, larger values of the output of two filters are taken from both output of the filters.

Generally, static threshold value is used for detection. However, in noisy event scenes, static threshold value can lead to high false detection error rate when the noise has similar characteristic with the events. To consider the noise effect on detection, adaptive threshold value is used as in (2) ,

$$T_i = T_{base} + \beta \cdot S_i \tag{2}$$

where T_i is an adaptive threshold value of class i, T_{base} is a base threshold value, S_i is mean of the DNN output of the class i in the event scene which reflects noise similarity with class i, and β is ratio value for S_i. When noise characteristic is similar to class i, T_i gets higher and reduces false detection error rate of class i.

3. EXPERIMENTAL RESULT

In order to evaluate the performance of the proposed system, we conducted a SED experiment on IEEE DCASE 2016 Challenge Task 2 Train/Development Datasets [16]. The training dataset was composed of mono recordings of isolated acoustic events typically found in an office environment. 11 classes were available: clearthroat, cough, doorslam, drawer, keyboard, keys, knock, laughter, pageturn, phone, speech and each class was represented by 20 recordings in training dataset. The development dataset consisted of 18 two-minute recordings in various noise and event density conditions. Only training dataset and noises sampled from probability density functions were used for learning the system and development dataset was used for evaluation.

Data augmentation was performed for generating the training event scene. Each sound event scene was about two-minute long. All events in the training dataset were normalized to have the same power and 30 of them were randomly selected for one event scene. To diversify the training data, half of the events were manipulated by stretching the time at a $\pm 10\%$ rate and modifying the power in the range of $50\% - 200\%$. One third of the events were overlapped to each other for polyphonic event examples. To consider the effect of noise on events, white Gaussian noise at signal-to-noise ratio (SNR) levels 6 to 18 dB and pink noise and blue noise at SNR level 12 dB were mixed. Total 110 artificial event scenes were generated for training the system.

We used mel-band energy as input features. Instead of original frequency 44.1 kHz, we used the sampling frequency of 30 kHz, spanning 50 bands between 100 Hz and 15 kHz. We used a hamming window with a frame length of 30 ms and a frame shift of 10 ms for frame segmentation. For noise reduction, $K = 100$ noise exemplars are selected and α is set to 0.5. As training data and test data may have power mismatch, the features extracted from each event scene are normalized.

For training the DNN-based classifier, 50-dimensional mel-band energy features were used as input. The input layer for DNN was formed by applying a context window of 11 frames, having 550 visible units for the network. The DNN had 3 hidden layers with 768 hidden rectified linear units with in each layer and the final sigmoid output layer had 11 units, each corresponding to the event classes. The parameters of the network were initialized by random values sampled from zero-mean normal distribution. The fine-tuning of the network was performed using mean squared error as the loss function by error back propagation supervised by the correct label of frames. The mini-batch size for the stochastic gradient descent algorithm was set to be 128. The learning rate was initially set to be 0.015 and exponentially decayed over each epoch with decaying factor 0.99 after fifth iteration. The momentum was set to be 0.7. The training was stopped after 80 epochs. The dropout percentage of 20% was applied for regularization.

Table 1: Average detection results on IEEE DCASE 2016 Challenge Task 2 Development Dataset

Metrics	Segment-based	Event-based
Precision	0.9311	0.7553
Recall	0.9211	0.8367
F-score	0.9261	0.7939
Substitutions	0.0091	0.0152
Deletions	0.0698	0.1481
Insertions	0.0590	0.2559
ER	0.1379	0.4192

Table 2: Average detection results on IEEE DCASE 2016 Challenge Task 2 Evaluation Dataset

Metrics	Segment-based	Event-based
F-score	0.787	0.671
ER	0.3660	0.6178

In post-processing stage, two 21-tap sigmoid shape filters are applied for smoothing output of the DNN. Larger values are taken from both output of the filters and thresholded for event detection. We set T_{base} to 0.6 and α to 0.5 for adaptive thresholding. Same events within 200 ms gap are concatenated and events shorter than 100 ms are removed.

As evaluation measures the F-Score and the error rate (ER) are used on Segment-based level. The F-Score F is the harmonic mean of precision P and recall R. The ER is the total number of insertions I, deletions D and substitutions S relative to the number of reference events N.

$$F = \frac{2P \cdot R}{P + R}, \quad ER = \frac{S + D + I}{N} \tag{3}$$

The results on the development dataset, averaged over the 18 synthetic audio event scenes are shown in Table 1. F-score and ER on segment-based metrics are 0.9261 and 0.1379, respectively. On event-based overall metrics, F-score and ER are 0.7939 and 0.4192, respectively. For DCASE 2016 task 2 challenge evaluation, both training data and development data are used for training DNN classifier. In Table 2, the results on the evaluation dataset are shown. F-score and ER on segment-based metrics are 0.787 and 0.3660, respectively. On event-based overall metrics, F-score and ER are 0.671 and 0.6178, respectively.

4. CONCLUSION

We presented a SED system based on a DNN. We used data augmentation to deal with data sparsity problem and exemplar-based approach for noise reduction. We trained a DNN for classification and filtering and adaptive thresholding are used for detecting events.

The proposed system has shown promising results on IEEE DCASE 2016 Challenge Task 2 Datasets.

5. ACKNOWLEDGMENT

This research was supported in part by the National Research Foundation of Koera (NRF) grant funded by the Korea government (MEST) (NRF-2015R1A2A1A15054343), and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1016) supervised by the IITP(Institute for Information & communications Technology Promotion).

6. REFERENCES

[1] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, "HMM-based acoustic event detection with AdaBoost feature selection," *in Classification of Events, Activities and Relationships Evaluation and Workshop*, 2007.

[2] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," *in Proc. European Signal Processing Conference*, 2010.

[3] A. Temko and C. Nadeu, "Classification of acoustic events using SVM-based clustering schemes," *Pattern Recognition*, vol. 39, pp. 682–694, 2006.

[4] A. Temko and C. Nadeu, "Acoustic event detection in meeting-room environments," *Pattern Recognition Letters*, vol. 30, pp. 1281–1288, 2009.

[5] J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro,, "Non-speech audio event detection," *ICASSP*, 2009.

[6] M. Chin and J. Burred, "Audio event detection based on layered symbolic sequence representations," *ICASSP*, 2012.

[7] J. F. Gemmeke, L. Vuegen, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach for audio event detection," *in Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust. (WASPAA)*, 2013.

[8] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," *ICASSP*, 2015.

[9] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 1–13, 2013.

[10] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[11] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," *ICASSP*, 2013.

[12] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multilabel deep neural networks," *in Int. Joint Conf. on Neural Networks (IJCNN)*, 2015.

[13] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *ICASSP*, 2014.

[14] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," *ICASSP*, 2012.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] IEEE DCASE 2016 Challenge, `http://www.cs.tut.fi/sgn/arg/dcase2016/`, 2016.

EXPERIMENTS ON THE DCASE CHALLENGE 2016: ACOUSTIC SCENE CLASSIFICATION AND SOUND EVENT DETECTION IN REAL LIFE RECORDING

Benjamin Elizalde[1], Anurag Kumar[1], Ankit Shah[2], Rohan Badlani[3], Emmanuel Vincent[4], Bhiksha Raj[1], Ian Lane[1]

[1]Carnegie Mellon University, Pittsburgh, PA, USA, [4]Inria, F-54600 Villers-lès-Nancy, France
[2]NIT Surathkal, India,[3]BITS, Pilani, India
bmartin1,alnu@andrew.cmu.edu, rohan.badlani,ankit.tronix@gmail.com,
emmanuel.vincent@inria.fr, bhiksha@cs.cmu.edu,lane@cmu.edu

ABSTRACT

In this paper we present our work on Task 1 Acoustic Scene Classification and Task 3 Sound Event Detection in Real Life Recordings. Among our experiments we have low-level and high-level features, classifier optimization and other heuristics specific to each task. Our performance for both tasks improved the baseline from DCASE: for Task 1 we achieved an overall accuracy of $\mathbf{78.9}\%$ compared to the baseline of $\mathbf{72.6}\%$ and for Task 3 we achieved a Segment-Based Error Rate of $\mathbf{0.48}$ compared to the baseline of $\mathbf{0.91}$.

Index Terms— audio, scenes, events, features, segmentation, DCASE, bag of audio words, GMMs, sound event detection, acoustic scene classification

1. INTRODUCTION

Audio plays a critical role in understanding the environment around us. This makes audio content analysis research important for tasks related to multimedia [1, 2], and human computer interaction [3, 4] to mention a pair. However, unlike the field of computer vision which has a variety of standard publicly available datasets such as Imagenet, audio event/scene analysis lacks such large dataset. This makes it difficult to compare different approaches and establishing the state of art. The second iteration of DCASE [5], occurring in 2016, offers an opportunity to compare approaches on a standard public dataset. This edition it includes four different tasks: acoustic scene classification, sound event detection– real and synthetic audio, and audio tagging.

The state-of-the-art of the previous DCASE challenge, for both acoustic scenes [6–8] and sound event detection [6, 8, 9], attributed their success mainly to features and audio representations rather than classifiers. Hence, an important aspect in our work is to emphasize on classifier exploration along with features. In this paper we present our work performed on Task 1 and Task 3. We proposed a variety of methods for both tasks and we obtained significant improvement over the baseline methods.

2. TASKS AND DATA

The goal of Task 1, Acoustic Scene Classification, is to classify a test recording into one of predefined classes that characterizes the environment in which it was recorded for example *park*, *home*, *office*. TUT Acoustic Scenes 2016 dataset is used for this task. It consists of recordings from various acoustic scenes. For each recording location, a 3-5 minute long audio recording was captured. The original recordings were then split into 30-second segments for the challenge. There are 15 acoustic scenes for the task.

Task 3, Sound Event Detection in Real Life Recordings, evaluates performance of sound event detection in multi-source conditions similar to our everyday life. There is no control over the number of overlapping sound events at each time, not in the training nor in the audio data. TUT Sound Events 2016 dataset is used for Task 3, which consists of recordings from two acoustic scenes: *Home* and *Residential Area*. There are 18 selected sound event classes, 11 for Home and 7 for Residential Area.

3. TASK 1: ACOUSTIC SCENE CLASSIFICATION

From machine learning perspective, we treated Task 1 as a multi-class classification problem. The first step is to use a suitable method for characterizing acoustic scenes in the audio segments. An effective approach for characterizing audio events is bag-of-audio-words based feature representation [10], which is usually built over low-level features such as MFCCs. Acoustic scenes, however, are more complex mixtures of different audio events and a more robust representation is required. To obtain a more robust representation we use Gaussian Mixture Models (GMMs) for feature representations of audio segments. Broadly, we employed two high-level feature representations to represent audio scenes. On the classification front we used Support Vector Machines (SVMs) as our primary classifier and in combination with other classifiers.

3.1. Feature Representations

Let D-dimensional MFCCs vectors for a recording be represented as $\vec{x}_t$, where $t = 1$ to T, T is the total number of MFCCs vectors for the recording. The major idea behind both high-level feature representation is to capture the distribution of MFCCs vectors of a recording. We will refer to these features as $\vec{\alpha}$ and $\vec{\beta}$ features and the sub-types will be represented using appropriate subscripts and superscripts.

The first step in obtaining high-level fixed dimensional feature representation for audio segments is to train a GMM on MFCC vectors of the training data. Let us represent this GMM by $\mathcal{G} = \{w_k, N(\vec{\mu}_k, \Sigma_k), k = 1\ to\ M\}$, where w_k, $\vec{\mu}_k$ and Σ_k are the mixture weight, mean and covariance parameters of the k^{th} Gaussian in $\mathcal{G}$. We will assume diagonal covariance matrices for all Gaussians and $\vec{\sigma}_k$ will represent the diagonal vector of Σ_k. Given the MFCCs vectors $\vec{x}_t$ of a recording, we computed the probabilistic assignment of $\vec{x}_t$ to the k^{th} Gaussian. These soft assignments are added over all t to obtain the total mass of MFCCs vectors belonging to the k^{th} Gaussian (Eq 1). Normalization by T is used to remove the effect of the duration of recordings.

$$Pr(k|\vec{x}_t) = \frac{w_k N(\vec{x}_t;\vec{\mu}_k,\Sigma_k)}{\sum_{j=1}^{M} w_j N(\vec{x}_t;\vec{\mu}_k,\Sigma_k)}, P(k) = \frac{1}{T}\sum_{i=1}^{T} Pr(k|\vec{x}_t) \quad (1)$$

The soft count histogram features referred to as $\vec{\alpha}$ is, $\vec{\alpha}^M = [P(1), ..P(k)..P(M)]^T$. $\vec{\alpha}^M$ is an M-dimensional feature representation for a given recording. It captures how the MFCC vectors of a recording are distributed across the Guassians in $\mathcal{G}$. $\vec{\alpha}^M$ is normalized to sum to 1 before using it for classifier training.

The next feature ($\vec{\beta}$), also based on the GMM $\mathcal{G}$, tries to capture the actual distribution of the MFCC vectors of a recording. This is done by adapting the parameters of $\mathcal{G}$ to the MFCC vectors of the recording. We employ maximum *a posteriori* (MAP) estimation to for the adaptation [11] [12]. Parameter adaptation for k^{th} Gaussian follows the following steps. First we compute,

$$n_k = \sum_{t=1}^{T} Pr(k|\vec{x}_t), \;\; E_k(\vec{x}) = \frac{1}{n_k}\sum_{t=1}^{T} Pr(k|\vec{x}_t)\vec{x}_t, \;\; E_k(\vec{x}^2) = \frac{1}{n_k}\sum_{t=1}^{T} Pr(k|\vec{x}_t)\vec{x}_t^2 \tag{2}$$

Finally, the updated mean and variances are obtained as

$$\hat{\vec{\mu}}_k = \frac{n_k}{n_k + r}E_k(\vec{x}) + \frac{r}{n_k + r}\vec{\mu}_k \tag{3}$$

$$\hat{\vec{\sigma}}_k = \frac{n_k}{n_k + r}E_k(\vec{x}^2) + \frac{r}{n_k + r}(\vec{\sigma}_k^2 + \vec{\mu}_k^2) - \hat{\vec{\mu}}_k^2 \tag{4}$$

The relevance factor r controls the effect of the original parameters on the new estimates. We obtain 2 different feature representation using the adapted means ($\hat{\vec{\mu}}_k$) and variances ($\hat{\vec{\sigma}}_k$). The first one denoted by $\vec{\beta}^M$ is an $M \times D$ dimensional feature obtained by concatenating the adapted means $\hat{\vec{\mu}}_k$ for all k, that is $\vec{\beta}^M = [\hat{\vec{\mu}}_1^T, ...\hat{\vec{\mu}}_K^T]^T$. In the second $\vec{\beta}$ features adapted $\hat{\vec{\sigma}}_k$ are concatenated along with $\hat{\vec{\mu}}_k$ to obtain a $2 \times M \times D$ dimensional features. This form of $\vec{\beta}$ features are denoted by $\vec{\beta}_\sigma^M$.

3.2. Classification

Once the feature representation for audio segments have been obtained, Task 1 essentially becomes a multi-class classification problem. Our primary classifiers are SVMs where we explore a variety of kernels. For the $\vec{\beta}$ features, we use Linear Kernel (LK) and RBF Kernel (RK). For soft-count histogram $\vec{\alpha}$ features we explore a panoply of kernels. Along with LK and RK we explored the following kernels.

- Exponential χ^2 Distance (ECK): the kernel is computed as $K(\vec{x}, \vec{y}) = \exp^{-\gamma D(\vec{x},\vec{y})}$, where $D(\vec{x}, \vec{y}) = \sum_i (x_i - y_i)^2/(x_i + y_i)$ is χ^2 distance.
- χ^2 Kernel (CK): In this case $K(\vec{x}, \vec{y}) = \sum_i \frac{2x_i y_i}{x_i + y_i}$
- Intersection Kernel (IK): $K(\vec{x}, \vec{y}) = \sum_i \min(x_i, y_i)$
- Exponential Hellinger Distance Kernel (EHK): $K(\vec{x}, \vec{y}) = \exp^{-\gamma D(\vec{x},\vec{y})}$ where $D(\vec{x}, \vec{y}) = \sum_i (\sqrt{x_i} - \sqrt{y_i})^2$
- Hellinger Kernel (HK): $(\vec{x}, \vec{y}) = \sum_i \sqrt{x_i y_i}$

The details of these kernels can be found in [13–15]. For kernels where γ term appears, the optimal value of γ value can be obtained by cross validation over training data. However, setting γ equal to the inverse of average distance $D(\vec{x}, \vec{y})$ between training data points works well in general as well. We use [16] [17] for SVM implementation.

Finally, we have a classifier fusion step where we combined the output of the different classifiers. We combined multiple classifiers by taking prediction vote from each classifier and the final predicted class is the one which gets the maximum vote. We call it the *Fused Classifier* and we observed that the fused classifier can give significant improvement for several acoustic scenes.

Table 1: Task 1 Accuracy for different cases (Single Classifier)

	$\vec{\alpha}^M$							$\vec{\beta}^M$		$\vec{\beta}_\sigma^M$	
M	LK	RK	ECK	CK	IK	EHK	HK	LK	RK	LK	RK
64	62.8	60.6	66.2	66.3	66.0	64.7	65.3	76.8	76.6	75.5	**76.7**
128	63.6	62.3	67.5	67.1	66.4	67.4	66.5	76.5	75.3	77.5	**77.5**
256	63.9	63.9	67.3	67.8	66.5	68.7	67.7	76.5	71.9	**76.6**	75.9
512	65.0	62.9	67.8	67.8	67.1	68.9	69.3	**76.4**	72.2	76.2	75.9

Table 2: Overall Task 1 Accuracy (Fused Classifier)

	Baseline					**Proposed**				
Scene	Fold 1	Fold 2	Fold 3	Fold 4	**Avg.**	Fold 1	Fold 2	Fold 3	Fold 4	**Avg.**
Beach	84.2	66.7	78.9	47.4	69.3	100	71.4	89.5	52.6	78.4
Bus	68.4	65.0	100	85.0	79.6	68.4	50.0	100	95.0	78.4
Cafe/Restaurant	66.7	94.7	71.4	100	83.2	88.9	63.2	76.2	95.0	80.8
Car	70.0	89.5	89.5	100	87.3	80.0	100	100	100	95.0
City Center	83.3	73.7	89.5	95.5	85.5	88.9	84.2	100	95.5	92.1
Forest Path	57.1	100	66.7	100	81.0	81.0	100	100	100	95.2
Grocery Store	52.6	81.0	89.5	36.8	65	89.5	81.0	94.7	84.2	87.3
Home	100	55.6	95.0	77.8	82.1	100	61.1	80.0	44.4	71.4
Library	47.6	38.9	15.0	100	50.4	47.6	33.3	85.0	100	66.5
Metro Station	84.2	94.4	100	100	94.7	94.7	94.4	100	100	97.3
Office	100	100	94.4	100	98.6	78.9	100	72.2	83.3	83.6
Park	10.0	5.6	0	40.0	13.9	65.0	33.3	50.0	30.0	44.6
Residential	78.9	47.6	100	84.2	77.7	84.2	42.9	94.7	57.9	69.9
Train	16.7	31.6	30.4	61.1	34.9	50.0	63.2	34.8	88.9	59.2
Tram	88.9	88.9	63.6	100	85.3	83.3	88.9	63.6	100	84.0
Overall	**67.2**	**68.9**	**72.3**	**81.9**	**72.6**	**80.0**	**71.1**	**82.7**	**81.8**	**78.9**

3.3. Results

Our experimental setup with the folds structure, is same as the one provided by DCASE. We extracted 20 dimensional MFCC features using $30ms$ window and 50% overlap. MFCCs are augmented with their delta and acceleration features. For our final feature representation we experimented with 4 different values of GMM component size $M, 64, 128, 256$ and 512. The relevance factor r for $\vec{\beta}$ is set to 20. Due to space constraints we cannot present fold-and-scene specific results for all cases and hence overall accuracy for all 4 folds is shown. Table 1 shows overall accuracy results for different cases. The accuracy for the MFCC-GMM *baseline* method provided in the challenge is 72.6%.

We can observe from Table 1 that $\vec{\alpha}$ features in general do not perform better than the baseline method for any SVM kernel. However, $\vec{\beta}$ features clearly outperformed baseline method. In the best case, with $M = 128$ and $\vec{\beta}_\sigma^M$ our method outperformed the baseline by an absolute 5%.

Table 2 shows results for the fused classifiers. For the fusion step we did not consider classifiers built over $\vec{\alpha}$ since these classifiers are inferior compared to those using $\vec{\beta}$ features. We can observe that our proposed method beats the baseline method by an absolute $\mathbf{6.3}\%$. Moreover, for scenes such as *Park, Train, Library* where the baseline method gives very poor results, we improved the accuracy by an absolute $\mathbf{16 - 30}\%$. We also obtained superior overall accuracy on all folds which suggests that our proposed method is fairly robust. This is further supported by the fact that on DCASE evaluation set, We achieved an overall accuracy of $\mathbf{85.9}\%$.

4. TASK 3: SOUND EVENT DETECTION IN REAL LIFE RECORDINGS

Detection of sound events in scenes and long recordings have been treated as a multi-class classification problem before in [18–20] where a classifier is trained with the sound segments. For testing, the classifier outputs segment/frame-level predictions for all the classes. In order to follow a similar approach, first we wanted to analyze features' performance for sound events regardless of the scene. This way, we could have an intuition of performance on the harder scenario of Task 3 where not every segment of the scene corresponds to a labeled sound event.

Feature Type	Accuracy%	Classifier
MFCCs	67.7	Logistic Regression
GBFB	52.4	Gradient Boosting
SGBFB	61.5	Gradient Boosting
Scatnet	62.1	Random Forest
Stacked	66.68	Random Forest
Stacked + PCA	66.06	Random Forest

Table 3: Sound-event classification accuracy for different feature types using the 18 sounds and the 75 training - 25 testing ratio. Stacked included normalized MFCCs +SGBFB +Scatnet.

4.1. Features and Classifiers Optimization

For the features we tried the conventional MFCCs with standard parameters such as 12 coefficients plus energy, delta and double delta for a total of 39 dimensions. Moreover, we explored three features addressing the time-frequency acoustic characteristics. The Gabor Filter Bank (GBFB) in [21] have 2D-filters arranged by spectral and temporal modulation frequencies in a filter bank. The Separable Gabor filter bank (SGBFB) features extract spectro-temporal patterns with two separate 1D GBFBs, a spectral one and a temporal one. This approach reduces the complexity of the spectro-temporal feature extraction and further improves robustness as demonstrated in [22]. Both features have the default parameters from the toolbox[1] for a total dimension of 1,020 each. The Scatnet [23] features are generated by a scattering architecture which computes invariants to translations, rotations, scaling and deformations, while keeping enough discriminative information. It can be interpreted as a deep convolution network, where convolutions are performed along spatial, rotation and scaling variables. As opposed to standard convolution networks, the filters are not learned but are scaled and rotated wavelets. The features were extracted with a toolbox[2] using 0.25 second segments. The dimensionality of the three Scatnet components are 2, 84, 435 for a total of 521. Additionally, we included the normalized (mean and variance) Stacked (MFCCs+ SGBFB+ Scatnet) with PCA and also the normalized (mean and variance) Stacked without PCA. For the PCA we used Scikit's [24] and used the full dimensionality of 1,580 as the number of input components and the resultant automatic reduction was 909 dimensions. For all the feature types and for the sake of avoiding the length variability of the temporal dimension, we averaged the vectors across time to end up with one single vector per sound event file.

Then, for the classifiers we considered Tpot [25], built on top of Scikit [24], which is a Python tool that automatically creates and optimizes machine learning pipelines using genetic programming. This toolbox (version 4) considers 12 classifiers such as Decision Tree, Random Forest, Xtreme Gradient Boosting, SVMs, K-Neighbors and Logistic Regression. The main Tpot parameter is "number of generations", which corresponds to the number of iterations carried to tune the classifier, we set it to 15. An example of the best classifier for each feature type can be seen in Table 3. Interestingly, decision tree-based algorithms and logistic regression outperformed others like SVMs.

For our experiments, we extracted the 18 sound events from the two scenes using the annotations, and then we extracted different feature types from these isolated sounds. For each feature type experiment, the sound events' feature files were fed to Tpot in a randomly selected ratio of 75% training and 25% testing, each set with different files. We kept the same partitions across our experiments for consistency. The performance was measured in terms of accuracy and is displayed in Table 3.

[1] http://www.uni-oldenburg.de/mediphysik-akustik/mediphysik/downloads/gabor-filter-bank-features/

[2] http://www.di.ens.fr/data/software/scatnet/

Feature Type	**Accuracy%**	**Classifier**
Home	56.4	Random Forest
Home + G	55.2	Random Forest
Home + G + P	55.7	Random Forest
Residential	53.3	Gradient Boosting
Residential + G	57.8	Decision Tree
Residential + G + P	56.7	Random Forest

Table 4: Sound-event classification accuracy using the DCASE set up with four-folds partitions. The inclusion of the [G]eneric class improved performance for both scenes, whereas the inclusion of the [P]erturbed audio improved only the Home performance.

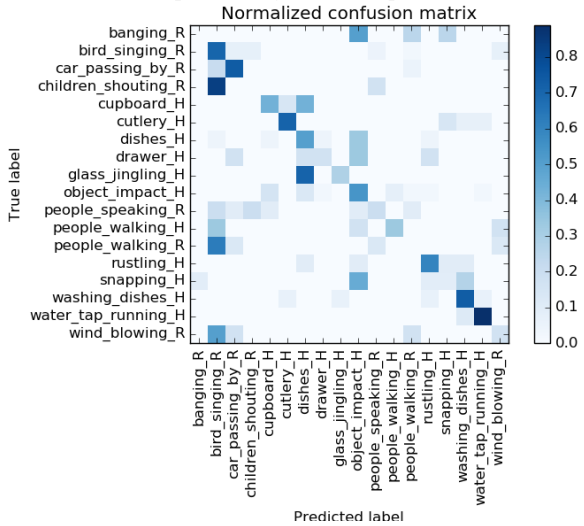


Figure 1: [H]ome and [R]esidential without the generic class. *Object impact* and *bird singing* capture most of ambiguities.

The features with the best performance were MFCCs with 67.7% and thus we keep them for our DCASE evaluation set up. The other features have shown better results than MFCCs on audio classification, but it wasn't the case for this particular dataset. Results for Scatnet was 62.1%, for GBF was 52.4%, and for SGBF was 61.5%. Moreover, the two normalized stacked features performed almost as good as MFCCs with 66.68% for the stacked without PCA and 66.08% for the stacked with PCA. In principle the stacked version contains more information about the acoustics and thus they were expected to perform better. Nevertheless, they didn't outperform MFCCs which is designed for speech and focus on lower frequencies rather than on a wider frequency range. We cannot draw a fundamental conclusion on the performance of these features for sound event classification since the amount of data and classes are determinant.

4.2. Inclusion of Generic Sound Event Class

In the annotated scenes, not every segment of audio corresponds to a labeled sound. Hence, it cannot be assured that any of our sound event classes have to be present on every test segment. To handle out-of-vocabulary segments, we proposed a generic sound event class.

For the first experiment we wanted to analyze the impact of the

generic class together with the 18 sounds in the multi-class classification set up described in Section 4.1 using MFCCs. To create such class, we used the sound events annotations and trimmed out the audio between the labeled segments, which are unlabeled. Then, we randomly selected from both scenes, 60 audio files which is about the average number of sound event samples per class. In order to visualize the performance, we included the normalized confusion matrices (CMs) in Figures 1 and 2. The accuracy performance without the generic class was 67.7% and with the generic class was 60.94%. The performance dropped with the inclusion of the new class, but we can also observe how although the generic class shared the background acoustics with the other sound classes, it didn't significantly ambiguate with them.

The second set of experiments used the DCASE setup of separate scenes and four folds, and utilized the sound events with and without the generic class, but this time the generic class will have files particular to the scene. The results can be seen in Table 4 showing benefit of including the generic class. Moreover, the CMs not included due to space limitations, had cleaner diagonals. The reasons for performance improvements on the DCASE set up are suggested by the utilization of less sound classes, which reduces class ambiguity. The utilization of the generic class built with same-scene files as opposed to a mix of both scenes. As well as the optimization per scene of the classifier using Tpot.

4.3. Generation of Data Through Perturbation

The scarcity of labeled data per event is a common issue as discussed in [26, 27]. Annotations are costly, sounds don't occur with the same frequency and in general it's hard to capture enough variations of the same sound to train robust models. To address this problem, multiple techniques have been explored in the literature such as perturbation of the audio signal as in [28, 29]. The authors presented multiple types of perturbations resulting in improvements of speech separation. For Task 3 we performed time-based perturbation by speeding up and slowing down the sound event samples. We empirically analyzed multiple combinations of speed up-down values for different events. We concluded that speeding up more than 30% the original signal resulted in unintelligible audio and speeding down the signal more than 100% would be unlikely to occur. The range included 13 different speed values and the original version.

The set of experiments used the DCASE setup of separate scenes and four folds, and utilized the time-based perturbed audio. For training, we added to the original files the 13 versions of the perturbed audio files, whereas for testing, the set remained intact. The results can be seen in Table 4, where the performance for *Home* improved, but not for *Residential*. Thus, we decided to use perturbation for the DCASE evaluation.

4.4. Sound Event Detection and Submission Systems

For Task 3, we used the DCASE setup of separate scenes and four folds in a similar setup as the experiments from Table 4. For each scene, we extracted the sound events from the recordings using the annotations from the train set, followed by the extraction of MFCCs features. After, we trained the Tpot optimized multi-class classifier with the event samples. For testing, instead of using sound event files only, we segmented the scene recordings from the test set into one-second consecutive segments. This number was selected due to the metric schema of the DCASE evaluation, which considers one-second segments. After, we extracted audio features from the test segments and evaluate them with the classifier to obtain scores for each trained sound event class. The label corresponding to the highest score was chosen for the segments and then were written down

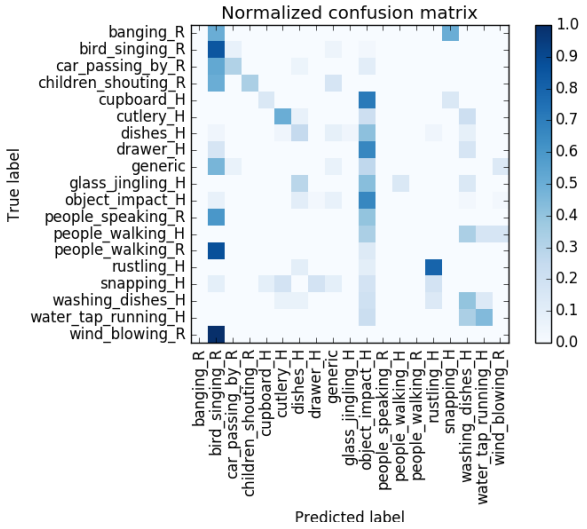


Figure 2: [H]ome and [R]esidential with the generic class. Although this class shared the background acoustics with the 18 sounds, it didn't cause major confusion.

Acoustic Scene	SBER	F-score
Home	0.62	48.8
Home + G	0.70	42.3
Home + G + P	0.72	39.2
Residential	0.34	77.7
Residential + G	0.48	65.2
Residential + G + P	0.56	59.6

Table 5: Our Segment-based Error Rate, using [G]eneric and [P]erturbation, outperformed the baseline.

into the DCASE format output file and fed to the official scoring scripts [3] along with the ground truth to compute performance.

We utilized the pipeline for three experiments, without generic class & without perturbation, with generic class & without perturbation and with generic class & with perturbation. The results using the development-test set is shown in Table 5. Although the inclusion of the generic class and the perturbation did not improve results for SBER, all cases outperformed the baseline method by a significant margin for both *Home* and *Residential* scenes. Our submission consisted on the runs using G and G+P but using the evaluation set. The eval results were SBER of 0.9613 and Fscore of 33.6% given by the G+P version.

5. CONCLUSION

In this paper we showed different approaches for both acoustic scene classification (Task 1) and sound event detection (Task 3) of the 2016 DCASE challenge. On both tasks we were able to obtain significant improvement over the baseline method. For Task 1 we observed that the $\vec{\beta}$ features performed much better than $\vec{\alpha}$ features. Although, linear and RBF kernels with $\vec{\beta}$ features can outperform the baseline by considerable margin on its own, we make note of the fact that a multiple classifier system can give further improvements. For Task 3, we tested different features and classifiers and significantly improved the baseline. Moreover, we explored a way of handling out-of-vocabulary sound segments with the generic class and the inclusion of perturbed audio to add robustness.

[3] http://www.cs.tut.fi/sgn/arg/dcase2016/sound-event-detection-metrics

6. REFERENCES

[1] H. Cheng, J. Liu, S. Ali, O. Javed, Q. Yu, A. Tamrakar, A. Divakaran, H. S. Sawhney, R. Manmatha, J. Allan *et al.*, "Sri-sarnoff aurora system at trecvid 2012: Multimedia event detection and recounting," in *Proceedings of TRECVID*, 2012.

[2] L. Jiang, S.-I. Yu, D. Meng, Y. Yang, T. Mitamura, and A. G. Hauptmann, "Fast and accurate content-based semantic search in 100m internet videos," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015.

[3] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. Mataric, "Where am i? scene recognition for mobile robots using audio features," in *2006 IEEE ICME*. IEEE, 2006, pp. 885–888.

[4] M. Janvier, X. Alameda-Pineda, L. Girin, and R. Horaud, "Sound-Event Recognition with a Companion Humanoid," in *Humanoids 2012 - IEEE International Conference on Humanoid Robotics*. Osaka, Japan: IEEE, 2012, pp. 104–111.

[5] 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, Hungary, 2016.

[6] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat, "Recurrence quantification analysis features for auditory scene classification," *IEEE AASP Challenge:, Tech. Rep*, 2013.

[7] 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.

[8] J. Schröder, N. Moritz, M. R. Schädler, B. Cauchi, K. Adiloglu *et al.*, "On the use of spectro-temporal features for the ieee aasp challenge detection and classification of acoustic scenes and events," in *2013 IEEE WASPAA*. IEEE, 2013.

[9] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste *et al.*, "An exemplar-based nmf approach to audio event detection," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[10] F.-F. Li and P. Perona, "The perceived position of moving objects: Transcranial magnetic stimulation of area MT+ reduces the flash-lag effect," in *IEEE CVPR*, vol. 2, 2005.

[11] J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and audio processing, IEEE Trans. on*, 1994.

[12] F. Bimbot *et al.*, "A tutorial on text-independent speaker verification," *EURASIP journal on applied signal processing*, vol. 2004, pp. 430–451, 2004.

[13] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.

[14] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 480–492, 2012.

[15] P. Li, G. Samorodnitsk, and J. Hopcroft, "Sign cauchy projections and chi-square kernel," in *Advances in Neural Information Processing Systems*, 2013, pp. 2571–2579.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[17] R.-E. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, 2008.

[18] B. Elizalde, M. Ravanelli, K. Ni, D. Borth, and G. Friedland, "Audio-concept features and hidden markov models for multimedia event detection."

[19] B. Elizalde and G. Friedland, "Lost in segmentation: Three approaches for speech/non-speech detection in consumer-produced videos," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.

[20] B. Elizalde, G. Friedland, H. Lei, and A. Divakaran, "There is No Data Like Less Data: Percepts for Video Concept Detection on Consumer-Produced Media," in *ACM International Workshop on Audio and Multimedia Methods for Large-Scale Video Analysis at ACM Multimedia*, 2012.

[21] M. R. Schädler, B. T. Meyer, and B. Kollmeier, "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," *The Journal of the Acoustical Society of America*, pp. 4134–4151, 2012.

[22] M. R. Schädler and B. Kollmeier, "Separable spectro-temporal gabor filter bank features: Reducing the complexity of robust features for automatic speech recognition," *The Journal of the Acoustical Society of America*, 2015.

[23] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in *Proceedings of the IEEE CVPR*, 2013, pp. 1233–1240.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, and others., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] R. S. Olson, R. J. Urbanowicz, P. C. Andrews *et al.*, "Automating biomedical data science through tree-based pipeline optimization," in *Proceedings of the 18th European Conference on the Applications of Evolutionary and Bio-inspired Computation*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2016.

[26] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," in *24th ACM International Conference on Multimedia*. ACM Multimedia, 2016.

[27] ——, "Weakly supervised scalable audio content analysis," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016.

[28] J. Chen, Y. Wang, and D. Wang, "Noise perturbation improves supervised speech separation," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 83–90.

[29] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 309–314.

IMPROVED DICTIONARY SELECTION AND DETECTION SCHEMES IN SPARSE-CNMF-BASED OVERLAPPING ACOUSTIC EVENT DETECTION

Panagiotis Giannoulis [1,3], *Gerasimos Potamianos* [2,3], *Petros Maragos* [1,3], *Athanasios Katsamanis* [1,3]

[1] School of ECE, National Technical University of Athens, 15773 Athens, Greece
[2] Department of ECE, University of Thessaly, 38221 Volos, Greece
[3] Athena Research and Innovation Center, 15125 Maroussi, Greece

paniotis@central.ntua.gr, gpotam@ieee.org, maragos@cs.ntua.gr, nkatsam@cs.ntua.gr

ABSTRACT

In this paper, we investigate sparse convolutive non-negative matrix factorization (sparse-CNMF) for detecting overlapping acoustic events in single-channel audio, within the experimental framework of Task 2 of the DCASE'16 Challenge. In particular, our main focus lies on the efficient creation of the dictionary, as well as the detection scheme associated with the CNMF approach. Specifically, for the dictionary creation stage, we propose a shift-invariant method for its size reduction that outperforms standard CNMF-based dictionary building. Further, for detection, we develop a novel algorithm that combines information from the CNMF activation matrix and atom-based reconstruction residuals, achieving significant improvements over conventional detection based on activations alone. The resulting system, assisted by efficient background noise modeling, outperforms a traditional NMF baseline provided by the Challenge organizers, achieving a 24% relative reduction in the total error rate metric on the Challenge Task 2 test set.

Index Terms— Convolutive Non-Negative Matrix Factorization, Dictionary Building, Overlapping Acoustic Event Detection

1. INTRODUCTION

Acoustic event detection (AED) is a research topic that has attracted significant interest in the literature. Its main goal is the end-pointing and classification of each event present in an audio recording. In its general form, multiple acoustic events may occur simultaneously, making the task extremely challenging. Application areas of AED include, among others, smart home environments, surveillance and security, as well as multimedia database retrieval.

In the case of isolated AED, conventional detection and classification approaches, such as ones based on hidden Markov models (HMMs) in conjunction with traditional audio features (for example MFCCs) achieve satisfactory performance [1]. In the case of overlapping AED however, such methods need to be modified in order to allow multiple event detection. For example, in [2], multiple-path Viterbi decoding is employed to deal with the overlapping scenario. Other works for overlapping AED include multi-label deep neural networks [3], temporally-constrained probabilistic component analysis models [4], generalized Hough-transform based systems [5], and non-negative matrix factorization (NMF) [6].

Among these, NMF-based approaches and their variants have begun to attract interest in the field of both isolated and overlapping AED in recent years. This is due to both their robustness and their natural ability to detect multiple events occurring simultaneously, as long as appropriate non-negative and linear representations of them are available. For example, in [7], a rather small dictionary of events is automatically built using sparse-CNMF, and subsequently the activations produced are used as input for HMM training for each class. Also, in [6], using a large dictionary, NMF activations are directly exploited to perform detection for each event class.

In this paper, overlapping AED is performed on the Task 2 dataset of the DCASE'16 Challenge [8], consisting of single-channel audio that contains eleven office-related events synthetically mixed in various conditions. The detection system proposed is based on the sparse-CNMF framework: Given a dictionary with spectral patches ("atoms") for each class (acoustic event), it determines the activations of each atom over time, thus allowing detection of overlapping events. The main contributions of the work lie in the investigation of methods for efficient dictionary building and in the design of a novel method for the final detection step. In particular, an efficient dictionary selection method based on shift-invariant similarity between atoms is proposed, achieving improved results compared to the standard automatic dictionary building of sparse-CNMF. Also, in the final detection step, a combination of activations with the reconstruction errors for each class is proposed. The approach yields significant improvements over conventional detection employing activations alone, indicating the complementary information contained in the reconstruction errors.

The remainder of the paper is organized as follows: Section 2 overviews the sparse-CNMF framework; Section 3 presents dictionary building for CNMF, including the proposed shift-invariant size reduction approach; Section 4 covers the CNMF detection approaches considered; Section 5 discusses additional system components, such as background noise modeling, feature extraction, and post-processing; Section 6 reviews the experimental framework and reports our results; and, finally, Section 7 concludes the paper.

2. SPARSE-CNMF FOR AED

The application of sparse-CNMF for overlapping AED is based on the idea of linear decomposition of events into spectral patches ("atoms"). Given the linearity of the features employed, mixtures of events will be mainly decomposed into atoms from the mixed classes, therefore indicating their presence. To accomplish this, non-negative features with approximate linearity are required: spectrograms and filterbank energies are typically used for this purpose.

NMF is a linear non-negative approximate factorization of the observed feature matrix. CNMF [9] is its convolutive extension, and

This work has been partially funded by the BabyRobot project, supported by the EU Horizon 2020 Programme under grant 687831.

it is formulated as follows: Given a non-negative data feature matrix $\mathbf{V} \in \Re^{\geq 0, M \times N}$, where M denotes the feature vector size and N the available number of feature vectors, the goal is to approximate $\mathbf{V}$ by matrix $\mathbf{\Lambda}$, derived as a temporal convolutive sum of a "dictionary" and "activations", namely

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_{t=0}^{T-1} \mathbf{W}_t \cdot \overset{t \rightarrow}{\mathbf{H}} \ , \quad (1)$$

where, operator $\overset{t \rightarrow}{\bullet}$ shifts the columns of its matrix argument t places to the right, $\mathbf{W}_t \in \Re^{\geq 0, M \times R}$ denotes the non-negative dictionary matrix at time step t, $\mathbf{H} \in \Re^{\geq 0, R \times N}$ represents the non-negative activation matrix, T is the number of time frames spanned by each dictionary atom, and R stands for the number of atoms in the dictionary. The i-th column of $\mathbf{W}_t$ describes the i-th atom, t time steps after its beginning. The dictionary thus contains R atoms of size $M \times T$ each. Minimization of a suitable error cost function $D(\mathbf{V} || \mathbf{\Lambda})$ results in the iterative estimation of $\mathbf{W}_t$ and $\mathbf{H}$ [9, 10].

For detection, assuming a given dictionary $\mathbf{W}_t$, $t \in [0, T\text{–}1]$, that contains atoms of the various classes of interest, the estimated $\mathbf{H}$ provides the activations of each class through time. Although CNMF produces activation patterns that tend to be sparse, in detection-related tasks sparsity of $\mathbf{H}$ becomes crucial. To achieve it, sparse-CNMF, a variant of CNMF, is often used, minimizing the following objective,

$$G(\mathbf{V} || \mathbf{\Lambda}) = D(\mathbf{V} || \mathbf{\Lambda}) + \lambda \|\mathbf{H}\|_1 \ , \quad (2)$$

with parameter λ controlling the trade-off between sparseness on $\mathbf{H}$ and accurate reconstruction of $\mathbf{V}$ by $\mathbf{\Lambda}$. Depending on the cost function selected (KL-divergence, Euclidean distance), different updating equations result [11, 12].

3. DICTIONARY BUILDING

Dictionary building is a very important step in exemplar-based methods. Representative atoms from each class must be contained in the dictionary matrix, capable of reconstructing unseen data. Using training data consisting of isolated event instances, a sufficient number of atoms is extracted and stored in the dictionary for each class of interest, resulting to matrices

$$\mathbf{W}_t = [\mathbf{W}_t^{(1)}, \ldots, \mathbf{W}_t^{(C)}], \ t \in [0, T\text{–}1] \ , \quad (3)$$

where C is the number of classes. In the case of CNMF-based methods, due to increased computational complexity, we need to create a rather compact dictionary. In the following, we present two alternatives for this task.

3.1. CNMF-based

For each class of interest, the training instances are concatenated to form its data matrix, $\mathbf{V}^{(i)}$. Then, via sparse-CNMF, matrices $\mathbf{W}_t^{(i)}$ and $\mathbf{H}^{(i)}$ are computed (as in [12]), and $\mathbf{W}_t^{(i)} \in \Re^{\geq 0, M \times R_i}$ stored in the dictionary. The duration, T, of each atom and their total number, R_i, are predefined. By extracting the same number of atoms for each class, their total number becomes $R = C \cdot R_i$.

3.2. Shift-invariant dictionary reduction

Here, we propose an alternative way for dictionary creation that selects a group of atoms from the original training data. For each class, first, a large number of atoms is extracted from its data matrix $\mathbf{V}^{(i)}$, using a sliding window of duration T (shifted by one feature frame at a time). Then, only R_i of them are selected by "uniformly sampling" the set of the resulting atoms, as explained next. The process aims at selecting different types of existing atoms based on a similarity measure, appropriate for CNMF. In our case, such similarity should be shift-invariant: i.e., two atoms are considered similar if the Euclidean distance between them, or between their temporally shifted versions, is small.

To achieve atom comparisons in a shift-invariant way, we first rearrange them into vectors of size $M \cdot T$, in a row-wise manner. This way, a time-shift of atoms results to shifts of their corresponding vectors. Then, atom similarity is measured as the Euclidean distance between the magnitudes of the Fourier transforms (DFTs) of the rearranged vectors, based on the well-known shift-invariant property of this transform. The available atoms are thus mapped to their Fourier-magnitude vectors, which are subsequently sorted based on their Euclidean distance from their mean. Finally, R_i atoms are selected by uniformly sampling the resulting sorted list.

The adopted sampling scheme represents a simple approach to desired dictionary size reduction. Alternatively, well-known clustering methods like k-means could also be used for the task.

4. DETECTION APPROACHES

As stated earlier, having created the dictionary matrix $\mathbf{W}_t$, sparse-CNMF accepts as input the data matrix $\mathbf{V}$, and outputs the desired activation matrix $\mathbf{H}$ (following the approach in [11]). The final event detection can occur by exploiting the information in the above matrices. We present two main approaches for accomplishing this.

4.1. Using activations only

Most of NMF-based approaches employ the information in $\mathbf{H}$ directly [6], or indirectly [7]. In our method, activations in $\mathbf{H}$ are directly used for detecting possible events. In particular, for each class, the activations are summed across all their atoms, for each frame, resulting in a new matrix $\mathbf{H}' \in \Re^{\geq 0, C \times N}$, with elements

$$H'(i, n) = \sum_{r \in \{i\}} H(r, n) \ , \quad (4)$$

where i denotes the class ($i = 1, \cdots, C$), $\{i\}$ the set of row indices in $\mathbf{H}$ that correspond to the i-th event atoms, and $n \in \{1, \cdots, N\}$ the time frame. Then, at time n, a class is considered active if $H'(i,n) > \theta_H$, where θ_H is a suitably chosen activation threshold. A post-processing step can also be employed to yield smooth activations. Finally, as activation refers to atoms, $T - 1$ additional frames following the detected activations are considered active.

4.2. Incorporating reconstruction residuals

An alternative method to the above decides for an event activation, not by thresholding the elements of $\mathbf{H}'$, but by measuring KL-divergence between $\mathbf{V}$ and $\mathbf{\Lambda}$, when only the atoms of the event in question and of background noise are used in reconstruction (see Section 5.1 for details on background noise modeling). More specifically, the total reconstruction error of sparse-CNMF over a time-segment, seg, under consideration, is $D(\mathbf{V}_{seg} || \mathbf{\Lambda}_{seg})$, whereas reconstruction error on basis of only the i-th event and noise is $D(\mathbf{V}_{seg} || \mathbf{\Lambda}_{seg}^{(i,bg)})$, where,

$$\mathbf{\Lambda}_{seg}^{(i,bg)} = \sum_{t=0}^{T-1} \mathbf{W}_t^{(i,bg)} \cdot \overset{t \rightarrow}{\mathbf{H}}_{seg}^{(i,bg)} \ , \quad (5)$$

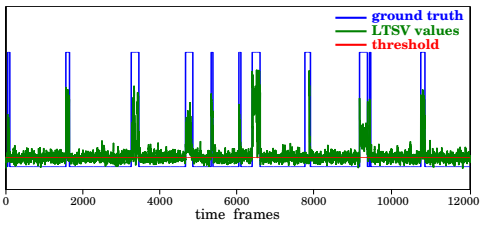


Figure 1: An example of applying the long-term signal variability (LTSV) measure to background noise detection (see Section 5.1). Ground truth peaks correspond to acoustic events.

with $\mathbf{H}_{seg}^{(i,bg)}$ denoting the part of $\mathbf{H}$ that contains only rows corresponding to atoms of the i-th class or background noise and columns that correspond to the time frames of seg. Similarly, in the above, $\mathbf{\Lambda}_{seg}$ and $\mathbf{V}_{seg}$ contain the columns of (1) and of the data matrix, respectively, within the segment under consideration.

We define the "residual ratio" of the i-th event as the ratio between the residual on basis of (5) to the total one, using (1), namely

$$\mathcal{E}(i\,,n) = \frac{D\left(\mathbf{V}_{seg}\,||\,\mathbf{\Lambda}_{seg}^{(i,bg)}\right)}{D\left(\mathbf{V}_{seg}\,||\,\mathbf{\Lambda}_{seg}\right)} \quad , \quad \text{for all} \quad n \in seg \quad . \tag{6}$$

In computing (6), non-overlapping segments of 1 sec. in duration are used. Small residual ratio values for the i-th event in a given segment means that large percentage of the reconstruction in that segment is achieved using only the i-th event (together with background noise). Activations in $\mathbf{H}'$ with large magnitude are also often related with large percentage of reconstruction, but this is not always the case. From the minimization of (2), large magnitude activations may occur for a given event and a given time frame, but with a small corresponding reconstruction contribution.

In our first approach using activations only, the event detection criterion is the activation matrix $\mathbf{H}$ element magnitudes. In the residuals-based approach, instead, the criterion is the accuracy of reconstruction using only atoms and activations of a particular event. In our final system, submitted to the Challenge, we combine both. Thus, the i-th event is considered active at time frame n, if

$$H'(i\,,n) > \theta_H \quad \text{and} \quad \mathcal{E}(i\,,n) < \theta_{\mathcal{E}} \quad . \tag{7}$$

Thresholds θ_H and $\theta_{\mathcal{E}}$ are chosen as explained in Section 5.2.

5. SYSTEM IMPLEMENTATION DETAILS

5.1. Background noise modeling

In addition to modeling the acoustic events by incorporating representative atoms in the dictionary, background noise modeling is necessary for robust AED. With the presence of background noise atoms in the dictionary, false alarm event activations are avoided in areas that events are not present. Also, more reliable reconstruction is possible in active areas, assuming additive noise.

In our approach, and following work in [6], we extract the background noise atoms from the observed data during decoding (on-the-fly). The advantage of this scheme is the adaptation of the background dictionary to slightly different conditions, possibly existing each time. However, instead of assuming background noise present at the beginning and end of the observed data, as in [6], we attempt to extract background atoms from various areas of the signal, by employing the long-term signal variability (LTSV) measure, described in [13]. This measure has been successfully used in voice activity detection, and it is based on the fact that background noise usually exhibits smaller variability through time in its spectrum.

In our system, a frame is considered as noise if its LTSV value is lower than a fixed threshold, θ_L. As before, the shift-invariant dictionary reduction method is applied to areas that noise is detected to help provide background noise atoms. An example of the LTSV based approach is shown in Figure 1, where LTSV values for a Challenge corpus signal are depicted, together with ground-truth locations of acoustic events. As it can be seen, LTSV values and the chosen θ_L ensure that acoustic event time frames are avoided.

5.2. Features, system parameters, and post-processing

We now provide some additional details of our implemented system. Concerning audio feature extraction, we have experimented with various feature sets that satisfy non-negativity and approximate linearity: Mel-filterbank energies, Gammatone-filterbank energies, DFT spectrogram, and the variable Q-Transform (VQT). The first three are computed using 30 msec long frames with a 10 msec shift, whereas VQT is obtained from the baseline system of [8]. Our final submitted system uses 150-dimensional Mel-filterbank energy features ($M = 150$).

Regarding dictionary building, atoms of 200 msec ($T = 17$ frames) in duration are used, and for the CNMF-framework, parameter λ in (2) is set to 0.7. Further, approximately 200 atoms per event class are used ($R_i \approx 200$), with $R \approx 2.4$k total atoms (including background noise modeling).

Concerning the various thresholds employed, θ_H in (7) is computed as a percentage (15%) of the maximum value of matrix $\mathbf{H}'$ elements. Threshold $\theta_{\mathcal{E}}$ in (7) is computed as a percentage (106%) of the minimum of $\mathcal{E}(i\,,n)$ for a given segment. Such values are optimized on available development data (see Section 6.1).

Finally, as a post-processing stage in the detection system, one-dimensional dilation is performed on each row of matrix $\mathbf{H}'$, in order to broaden the intervals of high-peaked activations produced. In the case of the combined method, dilation is performed before the combination with the residuals approach. At the end, $T-1$ frames after each detected activation are also considered as active.

6. EXPERIMENTS

6.1. Database

We perform experiments on the DCASE'16 Challenge database designed for Task 2 – "Sound event detection in synthetic audio" [8]. The corpus contains recordings of eleven office-related acoustic events (see also Figure 2), consisting of three parts: The training set with 20 isolated recordings of each event; a development set with 18 two-minute long recordings of synthetic mixtures of audio events and noise at various SNRs and event overlap conditions ("density" and "polyphony"); and a test set of similar structure to the development set (54 recordings), only used in the Challenge evaluation, with its ground-truth publicly unavailable at the moment.

6.2. Experimental setup

In this paper, we report experiments on both the development and test sets (the latter as only provided by the Challenge organizers). Specifically, for the development set, due to its particularity of containing the same event instances as the training set, we use two different setups, described next.

Table 1: Performance of baseline and proposed systems on 3 sets.

system	setup #1		setup #2		test	
	Fscore	ER	Fscore	ER	Fscore	ER
NMF-baseline	0.42	0.79	0.32	0.87	0.37	0.89
activations-only	0.83	0.30	0.43	0.79	–	–
activations&residuals	0.84	**0.29**	0.55	**0.63**	0.56	**0.68**

Table 2: Performance of different feature sets and dictionary sizes.

features	feat. dim.	dict. size	setup #1		setup #2	
			Fscore	ER	Fscore	ER
VQT	545	200	0.79	0.37	0.29	0.88
Gamma	150	200	0.82	0.33	0.35	0.86
Mel	150	200	0.83	**0.30**	0.43	**0.79**
Mel	150	100	0.81	0.36	0.42	0.85
Mel	100	100	0.83	0.30	0.42	0.82
DFT	545	100	0.78	0.42	0.41	0.83

Table 3: Performance of different dictionary building methods.

dictionary building method	setup #1		setup #2	
	Fscore	ER	Fscore	ER
sparse-CNMF	0.64	0.60	0.29	0.89
shift-invariant reduction	0.83	**0.30**	0.42	**0.82**

- Setup #1: This is identical to the default setup of Task 2. One dictionary is built using all isolated training data, and then AED is performed on all 18 development set recordings.
- Setup #2: Here, to allow testing on unseen event instances, we perform a 18-leave-one-out experiment. In total, 18 dictionaries are built, each tested on a single development set recording, by using each time all available training set instances, except those contained in the particular development set recording.

6.3. Metrics

We report results employing the adopted Challenge metrics [8], namely frame-based Fscore and frame-based total error rate (ER). The latter is defined as $\text{ER} = (I + D + S)/N$, where I denotes acoustic event insertions, D deletions, S substitutions, and N the total number of ground-truth events at a given frame. ER is computed in frames of 1 sec. in length.

6.4. Results

In Table 1, the results using the Challenge-provided NMF baseline, our submitted system, and a variant of it are compared for the different experimental setups considered. Regarding the NMF-baseline, it builds the dictionary using the training data, and extracts 20 atoms per class. Atoms have single-frame duration, and are extracted from the variable-Q transform spectrogram (VQT, 60 bins, 10 msec step). A post-processing stage applies median filtering to the output and allows up to five concurrent events [8].

Both our systems, depicted in Table 1, perform dictionary creation employing the shift-invariant reduction approach, and their details are provided in Section 5.2. It is obvious that both outperform the baseline in all setups. In particular, our submitted system ("activations & residuals") achieves 63.3%, 27.6%, and 23.6% relative reduction in ER over the baseline for setup #1, #2, and the test set, respectively. It seems that the extraction of more atoms per class (almost ten-fold over the baseline), combined with the incorporation of temporal structure under the CNMF-framework, lead to major improvements.

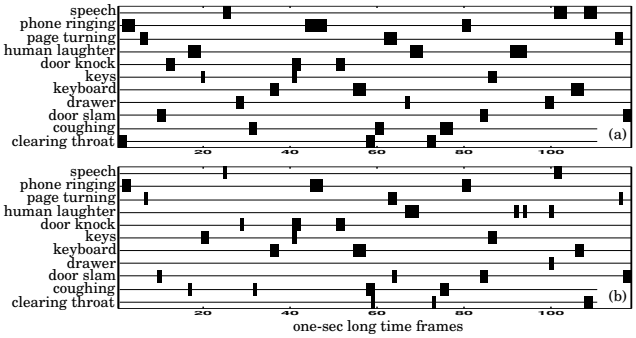


Figure 2: AED on the "dev_1_ebr_6_nec_3_poly_0.wav" Challenge recording: (a) ground-truth; (b) output of our submitted system. Acoustic event labels are also shown.

Comparing our two detection approaches, we can observe that the system using the combination of activations and reconstruction residuals (submitted to the Challenge) achieves a 20% ER relative reduction in setup #2, compared to the system using activations only. This highlights the complementarity of the two methods. The improvement is mainly due to the elimination of false activations, exhibiting large peaks in $\mathbf{H}'$ but also having a large residual ratio.

In Table 2, we show experimentation regarding different audio feature sets, together with variations in their dimensionality and dictionary size (number of atoms per class is depicted). We can observe that Mel-filterbank energies achieve the best performance among the different sets considered. It thus seems that they are more appropriate for the set of acoustic events considered in the Challenge. Also from the Mel feature results (150-dimensional), we can observe that increasing dictionary size leads to slight improvements.

A comparison of the different dictionary building methods is shown in Table 3, using the same detection system in both cases (a 100-dimensional Mel-filterbank, activations-only system, with 100 atoms per class). Clearly, the shift-invariant dictionary size reduction approach outperforms conventional CNMF-based dictionary building. This provides evidence that accurate representation of event atoms (instead of approximate) is beneficial to detection, as long as we have a way to select appropriate atoms.

Finally, in Figure 2, the output of our system is shown against ground-truth for a particular audio recording of the development set.

7. CONCLUSIONS

We presented a sparse-CNMF based system for overlapping audio event detection, employing an efficient dictionary building method and a novel detection approach. Attention was also given to background noise modeling and on experimentation with different possible feature sets for the CNMF framework. Results obtained on Task 2 of the DCASE'16 Challenge were promising, significantly outperforming the NMF-baseline provided.

In future work, better ways to combine activation-based and residual-based approaches will be investigated. Also the performance of our system will be tested in more datasets relevant to overlapping acoustic event detection.

8. REFERENCES

[1] P. Giannoulis, G. Potamianos, A. Katsamanis, and P. Maragos, "Multi-microphone fusion for detection of speech and acoustic events in smart spaces," in *Proc. 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 2375–2379.

[2] A. Diment, T. Heittola, and T. Virtanen, "Sound event detection for office live and office synthetic AASP challenge," *Proc. IEEE AASP Challenge on Detection Classif. Acoust. Scenes Events (WASPAA)*, 2013.

[3] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. International Joint Conference on Neural networks (IJCNN)*, 2015, pp. 1–7.

[4] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6450–6454.

[5] J. Dennis, H. Tran, and E. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised Hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[6] J. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach to audio event detection," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[7] C. Cotton and D. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011, pp. 69–72.

[8] Detection and Classification of Acoustic Scenes and Events 2016. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/

[9] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *International Conference on Independent Component Analysis and Signal Separation*, 2004, pp. 494–499.

[10] W. Wang, A. Cichocki, and J. Chambers, "A multiplicative algorithm for convolutive non-negative matrix factorization based on squared Euclidean distance," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2858–2864, 2009.

[11] P. O'Grady and B. Pearlmutter, "Convolutive non-negative matrix factorisation with a sparseness constraint," in *Proc. 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 2006, pp. 427–432.

[12] W. Wang, "Convolutive non-negative sparse coding," in *Proc. IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 3681–3684.

[13] P. Ghosh, A. Tsiartas, and S. Narayanan, "Robust voice activity detection using long-term signal variability," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 600–613, 2011.

SYNTHETIC SOUND EVENT DETECTION BASED ON MFCC

J.M. Gutiérrez-Arriola, R. Fraile, A. Camacho, T. Durand, J.L. Jarrín, S.R. Mendoza

Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación
Universidad Politécnica de Madrid

ABSTRACT

This paper presents a sound event detection system based on mel-frequency cepstral coefficients and a non-parametric classifier. System performance is tested using the training and development datasets corresponding to the second task of the DCASE 2016 challenge. Results indicate that the most relevant spectral information for event detection is below 8000 Hz and that the general shape of the spectral envelope is much more relevant than its fine details.

Index Terms— Sound event detection, spectral envelope, cepstral analysis

1. INTRODUCTION

Automatic sound event detection is a rather recent research issue and any advance related to it may impact a variety of application fields [1]. Probably, the most intuitive approach to sound description for event detection consists in parameterising its spectrum. Specifically, mel-frequency cepstral coefficients (MFCC) provide a low-dimensional procedure for coding the shape of the spectral envelope that has been successfully applied to speech processing tasks such as speaker verification [2] or laryngeal pathology detection [3]. In fact, this type of coefficients has also been applied to sound event detection [1, 4, 5]. Yet, it is known that sound perception not only works in spectral domain, but also in temporal domain [6]. Such temporal dimension may be included in sound event detection by different means such as calculating MFCC derivatives, training hidden Markov models for classification, or both [1, 4].

When it comes to detecting several sound events happening simultaneously, proposed approaches include decomposition of sound spectra in several components prior to classification [7], adding complexity to the classification stage to allow for multiple event detection [1], or combinations of both [8].

In our view, *a priori* decomposition of sound spectra in several components is problematic, since the addition of two signals in temporal domain does not necessarily result in the addition of their power spectra. For this reason, we approach the problem by directly coding the spectrum of the recorded signal using MFCC. The temporal dimension of the event detection problem is acknowledged by calculating the first derivatives of MFCCs and by splitting the sound signal into frames before processing. In this work, we concentrate on the design of the datasets and the signal analysis; consequently no assumption is made regarding the distribution of the calculated signal parameters. For this reason, a non-parametric classifier is chosen.

This work has been partially financed by the Spanish Government, through project grant number TEC2012-38630-C04-01.

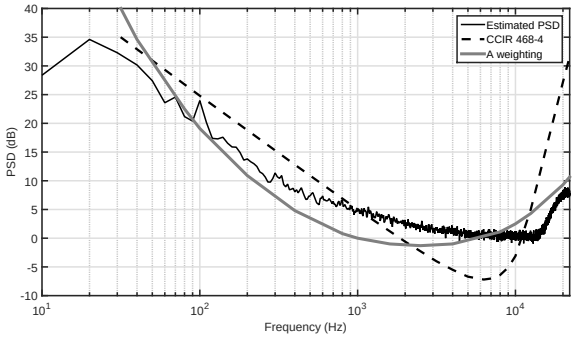


Figure 1: Power spectral density (PSD) of synthetic noise, estimated from a 6.5 second-length fragment using the Welch method [9]. For reference purposes, 'A Weighting' and 'CCIRR 468-4' curves [10] have also been plotted.

2. MATERIALS

Audio recordings were provided by IRCCYN, École Centrale de Nantes. They correspond to 11 sound event types (see Tab.1) recorded in a quiet environment, using a condenser microphone (AT8035, manufactured by Audio-Technica) connected to a portable recorder (H4n, manufactured by Zoom). Audio signals were sampled at 44.1 kHz and recorded with a single microphone (monophonic recordings). The microphone pass band ranges from 40 to 20,000 Hz.

20 events from each type were recorded, hence resulting 220 recordings each one containing a single sound event. For validation purposes, an additional dataset was built using the previous 220 recordings as a basis. This consists of 18 recordings with 2 minute durations. These were obtained by combining some of the single-event recordings into a single file and adding noise recorded in an independent session. Overlapping between events was allowed in 50% of the resulting files. Noise was approximately grey (Fig.1) and several levels of event-to-background ratio (EBR) were allowed: -6, 0 and 6 dB.

3. SIGNAL ANALYSIS

3.1. Inspection of sound spectra

Fig.2 depicts the estimated spectra, averaged for each type of event. While some types have distinct spectral envelope shapes, such as key drops or phone ringing, there are others for which the spectral envelopes are similar. This is especially the case of cough, throat clearing, laughter and speech, since all these sounds are produced as outputs of the same acoustic filter: the human vocal tract. Such fact

Type#	Type name	Event
1	Clearthroat	Throat clearing
2	Cough	Cough
3	Doorslam	Door slam
4	Drawer	Drawer sliding
5	Keyboard	Typewritting
6	Keys	Keys dropping on a desk
7	Knock	Knocking on a door
8	Laughter	Laughter
9	Pageturn	Paper page turning
10	Phone	Phone ringing
11	Speech	French speech
12	Back	Background noise

Table 1: Event types. Recordings corresponding to the 12^{th} type (*back*) were obtained by cutting out event-free segments from the validation dataset.

suggests that parameterisation schemes based only on estimating the average spectral envelope are likely to have poor performances.

From another point of view, all spectra exhibit a decay at frequencies above 13 kHz. However, the power spectral density of background noise (*back* type in Fig.2) grows from 13 to 22 kHz, as also shown in Fig.1. As a consequence, the EBR above 13 kHz is a decreasing function of frequency.

3.2. Parameter computation

Considering aforementioned characteristics of the target sound event spectra, we propose a parameterisation scheme based on the calculation of mel-frequency cepstral coefficients (MFCCs) and their derivatives. The proposed signal processing scheme comprises the next stages:

1. *Windowing*: Each digital audio signal is first normalised to yield a unit power discrete-time signal $x\,[n]$, composed by N samples ($n = 0 \dots N-1$). This signal is segmented in speech frames of length equal to L samples through multiplication by a framing window $w\,[n]$:

$$x_p\,[n] = x\,[n + p\,(L - l_0)] \cdot w\,[n] \tag{1}$$

where l_0 is the number of overlapping samples between consecutive frames and p is the frame index.

2. *Fourier transform*: From each speech frame, the short-term Discrete Fourier Transform (stDFT) is computed as:

$$X_p\,(k) = \sum_{n=0}^{L-1} x_p\,[n] \cdot e^{-j\frac{2\pi nk}{N_{\text{DFT}}}} \tag{2}$$

where N_{DFT} is the number of points of the stDFT, $N_{\text{DFT}} \geq L$ and $k = 0 \dots N_{\text{DFT}} - 1$.

The absolute frequency value that corresponds to each stDFT coefficient is:

$$f_k = \begin{cases} f_{\text{s}} \cdot \frac{k}{N_{\text{DFT}}} & \text{if } k \leq \frac{N_{\text{DFT}}}{2} \\ f_{\text{s}} \cdot \frac{k - N_{\text{DFT}}}{N_{\text{DFT}}} & \text{if } k > \frac{N_{\text{DFT}}}{2} \end{cases} \tag{3}$$

being f_{s} the sampling frequency.

3. *Mel distortion*: After the computation of the stDFT, the next step is frequency distortion in spectral domain. This is made according to [11, chap. 2]:

$$f_k^{\text{mel}} = \text{sgn}\,[f_k] \cdot 2595 \cdot \log_{10}\left(1 + \frac{|f_k|}{700}\right) \tag{4}$$

4. *Mel spectrum smoothing*: This is done by integrating the energy present in the spectrum of the processed speech frame along a set of pre-defined mel-frequency bands. These are M equal-width bands linearly distributed between $f_{\text{MIN}}^{\text{mel}}$ and $f_{\text{MAX}}^{\text{mel}}$ with 50% overlap between consecutive bands. Each one is characterised by its centre mel frequency and its width. The i^{th} centre frequency is

$$f_{\text{c},i}^{\text{mel}} = f_{\text{MIN}}^{\text{mel}} + \left(f_{MAX}^{\text{mel}} - f_{\text{MIN}}^{\text{mel}}\right) \cdot \frac{i}{M+1} \tag{5}$$

where $i = 1 \dots M$. Thus, each band covers the range $I_i^{\text{mel}} = \left[f_{\text{c},i-1}^{\text{mel}}, f_{\text{c},i+1}^{\text{mel}}\right]$, yielding bandwidth

$$\Delta f^{\text{mel}} = 2 \cdot \frac{f_{MAX}^{\text{mel}} - f_{\text{MIN}}^{\text{mel}}}{M+1} \tag{6}$$

Integration along bands is commonly done using triangular windows [12, chap. 6]. Thus, the result for each band is:

$$\tilde{X}_p\,(i) = \frac{1}{A_i} \cdot \sum_{f_k^{\text{mel}} \in I_i^{\text{mel}}} \left| \frac{f_k^{\text{mel}} - f_{\text{c},i-1}^{\text{mel}}}{\frac{\Delta f^{\text{mel}}}{2}} - 1 \right| |X_p\,(k)| \tag{7}$$

where the normalising term A_i ensures that for each band the mean energy is computed without any bias:

$$A_i = \sum_{f_k^{\text{mel}} \in I_i^{\text{mel}}} \left| \frac{f_k^{\text{mel}} - f_{\text{c},i-1}^{\text{mel}}}{\frac{\Delta f^{\text{mel}}}{2}} - 1 \right| \tag{8}$$

5. *Transformation into cepstral domain*: The last step in MFCC computation is transformation of the afore-mentioned smoothed mel spectrum into cepstral domain. Such transformation can be realised by calculating the inverse DFT of the logarithm of the power spectrum [13]. Given that the speech signal is real-valued, it may be assumed that its spectrum is symmetric. Furthermore, if $\tilde{X}_p\,(0)$ is defined to be equal to 1, which simply means adding a constant value to the signal in temporal domain, then the power cepstrum of the mel-wrapped and spectrally smoothed signal can be written as:

$$\begin{aligned} \mathcal{X}_p\,[q] &= \tfrac{1}{2M+1} \textstyle\sum_{i=-M}^{M} \log\left(\tilde{X}_p\,(i)\right) e^{j\frac{2\pi i}{2M+1}q} \\ &= \tfrac{1}{M+\frac{1}{2}} \textstyle\sum_{i=1}^{M} \log\left(\tilde{X}_p\,(i)\right) \cos\left(\frac{\pi i q}{M+\frac{1}{2}}\right) \end{aligned} \tag{9}$$

The coefficients $\mathcal{X}_p\,[q]$ are called MFCC and they may be computed using an expression that resembles the discrete cosine transform (DCT) of the logarithm of the smoothed mel-wrapped spectrum of the speech frame $x_p\,[n]$. In fact, the original MFCC formulation [14] directly uses the second form of the DCT (DCT-2) [15, chap. 8]. Herein, (9) is preferred because it has a simpler relation to the DFT.

6. *Derivation*: Derivation of MFCC to obtain ΔMFCC is performed using a eighth-order discrete differentiating filter:

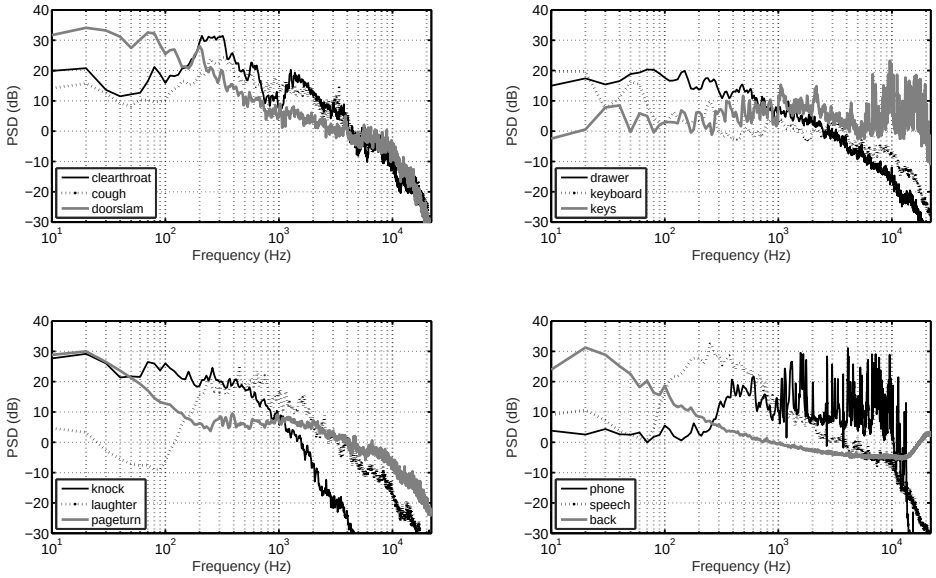


Figure 2: Average power spectral density for each type of event. Spectra have been averaged for all 20 recordings belonging to each type. Estimation has been carried out using the Welch method [9].

$$\begin{aligned}\Delta\mathcal{X}_p[q] &= \frac{1}{4}\mathcal{X}_{p-4}[q] - \frac{1}{3}\mathcal{X}_{p-3}[q] + \frac{1}{2}\mathcal{X}_{p-2}[q] \\ &- \mathcal{X}_{p-1}[q] + \mathcal{X}_{p+1}[q] - \frac{1}{2}\mathcal{X}_{p+2}[q] \\ &+ \frac{1}{3}\mathcal{X}_{p-3}[q] - \frac{1}{4}\mathcal{X}_{p+4}[q] \end{aligned} \quad (10)$$

4. CLASSIFICATION

The feature vectors describing sound frames that result from the previous signal analysis scheme have probability distributions with shapes that significantly differ between distinct sound events. For instance, the distributions for the *speech* and *keys* classes illustrated in Fig.3 present different shapes. From another point of view, it is known that for classification problems, the choice of classifier is much less relevant than the availability of as many data as possible [16]. For these reasons, a non-parametric discriminant approach based on the k-nearest-neighbours (kNN) rule [17] was selected.

5. POST-PROCESSING

Let $N_t(p)$ be the number of neighbours belonging to event type t assigned to the p^{th} sound frame by the kNN rule, therefore:

$$\sum_{t=1}^{12} N_t(p) = k \quad (11)$$

Then, a straightforward application of this classification rule would lead to assigning event type $\mathcal{T}(p)$ to the p^{th} sound frame such that:

$$\mathcal{T}(p) = \arg\max_t N_t(p) \quad (12)$$

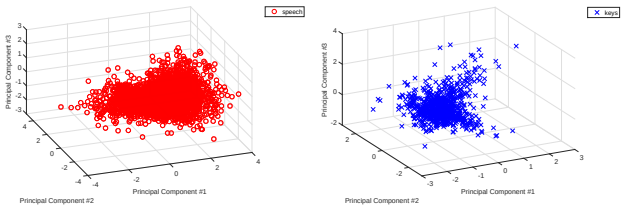


Figure 3: Distribution of frames belonging to *speech* (left) and *keys* (right) classes in the feature space defined by the three first principal components of the feature vectors including 15 MFCC + 15 ΔMFCC parameters.

However, the following procedure was used in order to smooth the effect of outlier frames:

1. Low-pass filtering of the number of neighbours by computing the local average using a sliding Hamming window w_{h}:

$$\tilde{N}_t(p) = \frac{\sum_{\Delta p=-P_1}^{P_1} N_t(p+\Delta p)\cdot w_{\text{h}}[\Delta p]}{\sum_{\Delta p=-P_1}^{P_1} w_{\text{h}}[\Delta p]} \quad (13)$$

2. Discarding events for which the filtered number of neighbours is below a certain threshold:

$$\hat{N}_t(p) = \begin{cases} \tilde{N}_t(p) & \text{if } \tilde{N}_t(p) \geq N_{\text{thres}} \\ 0 & \text{otherwise} \end{cases}, \; t = 1\ldots 11 \quad (14)$$

3. Assigning an event type to each frame, in case the smoothed number of neighbours corresponding to some class is above the threshold; otherwise, the frame is considered to belong to the *back* class:

Param.	Value	Explanation
L	1324	30 ms frames with $f_s = 44.1$ kHz
l_0	331	25% overlap between adjacent frames
$\omega[n]$		Hamming window
N_{DFT}	1324	Same as frame length
$f_{\mathrm{MIN}}^{\mathrm{mel}}$	62.63 mel	Corresponding to $f = 40$ Hz
$f_{\mathrm{MAX}}^{\mathrm{mel}}$	3582 mel	Corresponding to $f = 13$ kHz
M	40	
k	25	
P_1	5	200 ms filter length
N_{thres}	6.75	
P_2	1	Corresponding to 25 ms
$\Delta\tau_{\mathrm{MIN}}$	2 s	
T_{MIN}	300 ms	

Table 2: Parameter values for the reference system.

$$\mathcal{T}(p) = \begin{cases} \arg\max_t \hat{N}_t(p) & \text{if } \max_t \hat{N}_t(p) > 0 \\ 12 & \text{otherwise} \end{cases} \quad t = 1 \ldots 12 \tag{15}$$

4. Discarding events that are not detected in a minimum number of consecutive frames:

$$\tilde{\mathcal{T}}(p) = \begin{cases} \mathcal{T}(p) & \text{if } \sum_{\Delta p=-P_2}^{P_2} \mathcal{T}(p+\Delta p) = 2P_2 + 1 \\ 12 & \text{otherwise} \end{cases} \tag{16}$$

After classification of every sound frame, decision on the on/off times of sound events is made based on the next rules:

A. An event is considered to be formed by a set of consecutive frames corresponding to the same value of $\tilde{\mathcal{T}}(p)$. In such a case, the event type is defined by $\tilde{\mathcal{T}}(p)$ and its starting and ending times are defined by the central time instants of the first and last frames of the set, respectively.

B. Two events of the same type are merged into a single one if the time difference between the starting time of the second one and the ending time of the first one is less than a certain threshold $\Delta\tau_{\mathrm{MIN}}$. The resulting event duration is from the starting time of the first original event to the ending time of the second one.

C. A minimum event duration T_{MIN} is defined. If the duration of a given event is shorter, then its starting point is advanced and its ending point delayed so that its duration equals T_{MIN}.

6. EXPERIMENTS & RESULTS

The previously described system, with the parameter values summarised in Tab.2, was used as a reference and applied to the detection of sound events in the additional dataset described in section 2. Results for 20 MFCC + 20 ΔMFCC are summarised on the left column of Tab.3.

System performance can be significantly improved by building a training dataset with features as similar as possible to those of the validation dataset. In this case, if noise sequences extracted from the additional dataset are added to the 220 training recordings with the same levels of SNR as in the validation dataset, namely -6, 0 and 6 dB, and the resulting 660 sound signals are used as the new training dataset then the system performance can be significantly improved, as shown in the middle column of Tab.3.

Results in the right column of Tab.3 indicate that the most relevant information is concentrated below 8000 Hz ($f_{\mathrm{MAX}}^{\mathrm{mel}} = 2840$) and that it can be described using only 15 MFCCs plus their derivatives without any big loss of performance. This being a simpler configuration, a more robust performance is to be expected.

Last, it should be noted that the post-processing rule B implicitly allows event overlapping. In fact, detection performance for the recordings in the validation set with overlapped events ($F = 65.84\%$, $ER = 0.5030$ for the segment-based evaluation; $F = 59.18\%$, $ER = 0.6869$ for the event-based evaluation) is similar to the overall performance (Tab.3).

	Perform. Measure	Refer. System	Training with noise	8000 Hz 15 MFCC
Segment based	F	9.61%	70.06%	67.65%
	ER	0.9569	0.4706	0.4973
Event based	F	6.07 %	62.99%	60.22 %
	ER	1.059	0.6616	0.6902

Table 3: Event detection results in terms of F-score (F) and Error Rate (ER).

	Average	Clearthroat	Cough	Knock
F	34.2%	54.0%	25.0%	42.4%
ER	2.2537	0.7510	0.9053	1.8566
	Doorslam	**Drawer**	**Keyboard**	**Keys**
F	4.5%	33.1%	67.5%	12.5%
ER	3.0628	0.9033	0.5426	1.1156
	Laughter	**Pageturn**	**Phone**	**Speech**
F	47.6%	5.6%	71.3%	12.8%
ER	0.7647	0.9744	0.4793	13.4350

Table 4: Class average evaluation results (segment-based).

7. CONCLUSIONS

The reported results in sound event detection, obtained using a system based on MFCC parameters and a non-parametric classifier lead to two main conclusions. In the first place, system performance is critically affected by a proper selection of the sound recordings used for training the system. In this particular case, using recordings with noise levels similar to those in the testing set has allowed a significant improvement in performance. Secondly, the key spectral information for sound event detection seems to be concentrated below 8000 Hz. Additionally, the fact that 15 MFCCs provide almost the same performance as 20 MFCCs reveals that the essential information is in the overall shape of the spectral envelope and not in its fine details, be them either narrow peaks or narrow valleys.

APPENDIX: EVALUATION RESULTS

Performance of the proposed system (15 MFCCs; 40-8000 Hz) for the DCASE 2016 evaluation dataset is reported in [18]. The overall indicators for the segment-based evaluation were $ER = 2.0870$ and $F = 25.0\%$; for the event-based evaluations, they were $ER = 1.3064$ and $F = 25.7\%$. Per-class results are summarised in Tab.4.

8. REFERENCES

[1] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP J. Audio Speech Music Process.*, vol. 2013, no. 1, pp. 1–13, 2013.

[2] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 4, pp. 1–22, 2004.

[3] R. Fraile, N. Sáenz-Lechon, J. I. Godino-Llorente, V. Osma-Ruiz, and C. Fredouille, "Automatic detection of laryngeal pathologies in records of sustained vowels by means of mel-frequency cepstral coefficient parameters and differentiation of patients by sex," *Folia Phoniatr. Logop.*, vol. 61, no. 3, pp. 146–152, 2009.

[4] A. Diment, T. Heittola, and T. Virtanen, "Sound event detection for office live and office synthetic AASP challenge," in *Proc. IEEE AASP Challenge Detection Classif. Acoust. Scenes Events (WASPAA)*, 2013.

[5] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Internat. Joint Conf. Neural Networks (IJCNN)*, 2015, pp. 1–7.

[6] P. Gómez-Vilda, J. M. Ferrández-Vicente, V. Rodellar-Biarge, A. Álvarez-Marquina, L. M. Mazaira-Fernández, R. Martínez-Olalla, and C. Muñoz-Mulas, "Detection of speech dynamics by neuromorphic units," in *Internat. Work-Conf. Interplay between Natural and Artificial Comput.* Springer, 2009, pp. 67–78.

[7] O. Dikmen and A. Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *IEEE Workshop Appl. Signal Process. Audio Acoust.* IEEE, 2013, pp. 1–4.

[8] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Workshop on Machine Listening in Multisource Environm.*, 2011, pp. 36–40.

[9] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles Algorithms and Applications*. Macmillan Publishing Company, 1988.

[10] P. Skirrow, "Audio measurements and test equipment," in *Audio Engineers Reference Book*, M. Talbot-Smith, Ed. Focal Press, Oxford, 1999, ch. 3.6.

[11] X. D. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice-Hall, 2001.

[12] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. Macmillan Publishing Company, 1993.

[13] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," *Proc. IEEE*, vol. 65, no. 10, pp. 1428–1443, 1977.

[14] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 4, pp. 357–366, 1980.

[15] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice-Hall, 1989, vol. 2.

[16] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proc. 39^{th} Annual Meeting Assoc. Computational Linguistics*, 2001, pp. 26–33.

[17] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press - Elsevier, 2003.

[18] "Sound event detection in synthetic audio. Task results," Tampere University of Technology," DCASE, 2016. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2016/task-results-sound-event-detection-in-synthetic-audio[Visited: 04/08/2016]

BIDIRECTIONAL LSTM-HMM HYBRID SYSTEM FOR POLYPHONIC SOUND EVENT DETECTION

Tomoki Hayashi[1], *Shinji Watanabe*[2], *Tomoki Toda*[1], *Takaaki Hori*[2], *Jonathan Le Roux*[2], *Kazuya Takeda*[1]

[1]Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8603, Japan
[2]Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA

hayashi.tomoki@g.sp.m.is.nagoya-u.ac.jp,
{takeda,tomoki}@is.nagoya-u.ac.jp, {watanabe,thori,leroux}@merl.com

ABSTRACT

In this study, we propose a new method of polyphonic sound event detection based on a Bidirectional Long Short-Term Memory Hidden Markov Model hybrid system (BLSTM-HMM). We extend the hybrid model of neural network and HMM, which achieved state-of-the-art performance in the field of speech recognition, to the multi-label classification problem. This extension provides an explicit duration model for output labels, unlike the straightforward application of BLSTM-RNN. We compare the performance of our proposed method to conventional methods such as non-negative matrix factorization (NMF) and standard BLSTM-RNN, using the DCASE2016 task 2 dataset. Our proposed method outperformed conventional approaches in both monophonic and polyphonic tasks, and finally achieved an average F1 score of 67.1 % (error rate of 64.5 %) on the event-based evaluation, and an average F1-score of 76.0 % (error rate of 50.0 %) on the segment-based evaluation.

Index Terms— Polyphonic Sound Event Detection, Bidirectional Long Short-Term Memory, Hidden Markov Model, multi-label classification

1. INTRODUCTION

Sounds include important information for various applications such as life-log, environmental context understanding, and monitoring system. To realize these applications, It is necessary to extract internal information automatically from not only speech and music, which have been studied for long time, but also other various types of sounds.

Recently, studies related to sound event detection (SED) attracted much interest to aim for understanding various sounds. The objective of SED systems is to identify the beginning and end of sound events and to identify and label these sounds. SED is divided into two scenarios, monophonic and polyphonic. Monophonic sound event detection is under the restricted condition that the number of simultaneous active events is only one. On the other hand, in polyphonic sound event detection, the number of simultaneous active events is unknown. We can say that polyphonic SED is a more realistic task than monophonic SED because in real situations, it is more likely that several sound events may happen simultaneously, or multiple sound events are overlapped.

The most typical approach to SED is to use a Hidden Markov Model (HMM), where the emission probability distribution is represented by Gaussian Mixture Models (GMM-HMM), with Mel Frequency Cepstral Coefficients (MFCCs) as features [1, 2]. Another approach is to utilize Non-negative Matrix Factorization (NMF) [3, 4, 5]. In the NMF approaches, a dictionary of basis vectors is learned by decomposing the spectrum of each single sound event into the product of a basis matrix and an activation matrix, then combining the basis matrices. The activation matrix at test time is estimated using the basis vector dictionary. More recently, methods based on neural networks have achieved good performance for sound event classification and detection using acoustic signals [7, 8, 9, 10, 11, 12]. In the first two of these studies [7, 8], the network was trained to be able to deal with a multi-label classification problem for polyphonic sound event detection. Although these networks provide good performance, they do not have an explicit duration model for the output label sequence, and the actual output needs to be smoothed with careful thresholding to achieve the best performance.

In this paper, we propose a new polyphonic sound event detection method based on a hybrid system of bidirectional long short-term memory recurrent neural network and HMM (BLSTM-HMM). The proposed hybrid system is inspired by the BLSTM-HMM hybrid system used in speech recognition [13, 14, 15, 16], where the output duration is controlled by an HMM on top of a BLSTM network. We extend the hybrid system to polyphonic SED, and more generally to the multi-label classification problem. Our approach allows the smoothing of the frame-wise outputs without post-processing and does not require thresholding.

The rest of this paper is organized as follows: Section 2 presents various types of recurrent neural networks and the concept of long short term memory. Section 3 describes our proposed method in detail. Section 4 describes the design of our experiment and evaluates the performance of the proposed method and conventional methods. Finally, we conclude this paper and discuss future work in Section 5.

2. RECURRENT NEURAL NETWORKS

2.1. Recurrent Neural Network

A Recurrent Neural Network (RNN) is a layered neural network which has a feedback structure. The structure of a simple RNN is shown in Fig. 1. In comparison to feed-forward layered neural networks, RNNs can propagate prior time information forward to the current time, enabling them to understand context information in a sequence of feature vectors. In other words, the hidden layer of an RNN serves as a memory function.

An RNN can be described mathematically as follows. Let us denote a sequence of feature vectors as $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$. An RNN with a hidden layer output vector $\mathbf{h}_t$ and output layer one $\mathbf{y}_t$ are

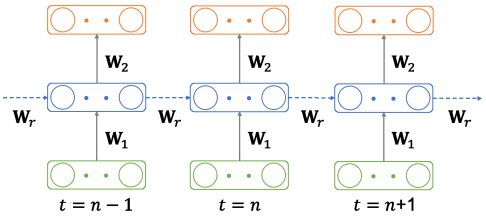


Figure 1: Recurrent Neural Network

calculated as follows:

$$\mathbf{h}_t = f(\mathbf{W}_1\mathbf{x}_t + \mathbf{W}_r\mathbf{h}_{t-1} + \mathbf{b}_1), \quad (1)$$

$$\mathbf{y}_t = g(\mathbf{W}_2\mathbf{h}_t + \mathbf{b}_2), \quad (2)$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ represent the input weight matrix and bias vector of the i-th layer, respectively, $\mathbf{W}_r$ represents a recurrent weight matrix, and f and g represent activation functions of the hidden layer and output layer, respectively.

2.2. Bidirectional Recurrent Neural Network

A Bidirectional Recurrent Neural Network (BRNN) [13, 17] is a layered neural network which not only has feedback from the previous time period, but also from the following time period. The structure of a BRNN is shown in Fig. 2. The hidden layer which connects to the following time period is called the *forward layer*, while the layer which connects to the previous time period is called the *backward layer*. Compared with conventional RNNs, BRNNs can propagate information not only from the past but also from the future, and therefore have the ability to understand and exploit the full context in an input sequence.

2.3. Long Short-Term Memory RNNs

One major problem with RNNs is that they cannot learn context information over long stretches of time because of the so-called *vanishing gradient* problem [19]. One effective solution to this problem is to use Long Short-Term Memory (LSTM) architectures [20, 21]. LSTM architectures prevent vanishing gradient issues and allow the memorization of long term context information. As illustrated in Fig. 3, LSTM layers are characterized by a *memory cell* $\mathbf{s}_t$, and three gates: 1) an *input gate* $\mathbf{g}_t^I$, 2) a *forget gate* $\mathbf{g}_t^F$, and 3) an *output gate* $\mathbf{g}_t^O$. Each gate $\mathbf{g}^*$ has a value between 0 and 1. The value 0 means that the gate is closed, while the value 1 means that the gate is open. In an LSTM layer, the hidden layer output $\mathbf{h}_t$ in

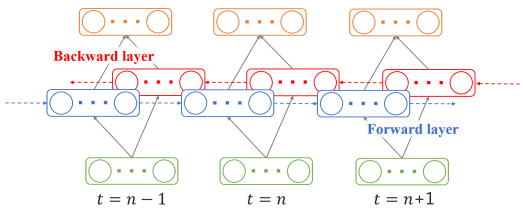


Figure 2: Bidirectional Recurrent Neural Network

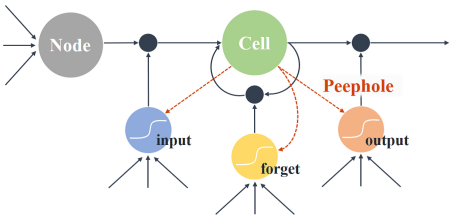


Figure 3: Long Short-Term Memory

Eq. 1 is replaced by the following equations:

$$\mathbf{g}_t^I = \sigma(\mathbf{W}^I\mathbf{x}_t + \mathbf{W}_r^I\mathbf{h}_{t-1} + \mathbf{s}_{t-1}), \quad (3)$$

$$\mathbf{g}_t^F = \sigma(\mathbf{W}^F\mathbf{x}_t + \mathbf{W}_r^F\mathbf{h}_{t-1} + \mathbf{s}_{t-1}), \quad (4)$$

$$\mathbf{s}_t = \mathbf{g}_t^I \odot f(\mathbf{W}_1\mathbf{x}_t + \mathbf{W}_r\mathbf{h}_{t-1} + \mathbf{b}_1) + \mathbf{g}_t^F \odot \mathbf{s}_{t-1}, \quad (5)$$

$$\mathbf{g}_t^O = \sigma(\mathbf{W}^O\mathbf{x}_t + \mathbf{W}_r^O\mathbf{h}_{t-1} + \mathbf{s}_{t-1}), \quad (6)$$

$$\mathbf{h}_t = \mathbf{g}_t^O \odot \tanh(\mathbf{s}_t), \quad (7)$$

where $\mathbf{W}$ and $\mathbf{W}_r$ denote input weight matrices and recurrent weight matrices, respectively, subscripts I, F, and O represent the input, forget, and output gates, respectively, $\odot$ represents point-wise multiplication, and σ represents a logistic sigmoid function.

2.4. Projection Layer

Use of a projection layer is a technique which reduces the computational complexity of deep recurrent network structures, which allows the creation of very deep LSTM networks [14, 15]. The architecture of an LSTM-RNN with a projection layer (LSTMP-RNN) is shown in Fig. 4. The projection layer, which is a linear transformation layer, is inserted after an LSTM layer, and the projection layer outputs feedback to the LSTM layer. With the insertion of a projection layer, the hidden layer output $\mathbf{h}_{t-1}$ in Eqs. 3-6 is replaced with $\mathbf{p}_{t-1}$ and the following equation is added:

$$\mathbf{p}_t = \mathbf{W}_I\mathbf{h}_t, \quad (8)$$

where $\mathbf{W}_l$ represents a projection weight matrix, and $\mathbf{p}_t$ represents a projection layer output.

3. PROPOSED METHOD

3.1. Data generation

There are only 20 clean samples per sound event in the DCASE2016 task 2 training dataset. Since this is not enough data to train a deeply structured recurrent neural network, we synthetically generated our own training data from the provided data. The training data generation procedure is as follows: 1) generate a silence signal of a

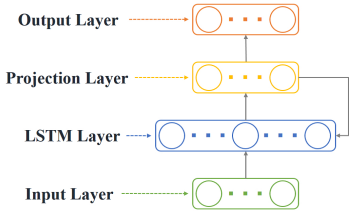


Figure 4: Long Short-Term Memory Recurrent Neural Network with Projection Layer

predetermined length, 2) randomly select a sound event from the training dataset, 3) add the selected sound event to the generated silence signal at a predetermined location, 4) repeat Steps 2 and 3 a predetermined number of time, 5) add a background noise signal extracted from the development set at a predetermined signal to noise ratio (SNR).

In this data generation operation, there are four hyper-parameters; signal length, number of events in a signal, number of overlaps, and SNR between sound events and background noise. We set signal length to 4 seconds, number of events to a value from 3 to 5, number of overlaps to a value from 1 to 5, and SNR to a value from -9 dB to 9 dB. We then generated 100,000 training samples of 4 seconds length, hence, about 111 hours of training data.

3.2. Feature extraction

First, we modified the amplitude of the input sound signals to adjust for the differences in recording conditions by normalizing the signals using the maximum amplitude of the input sound signals. Second, the input signal was divided into 25 ms windows with a 40 % overlap, and we calculated a log filterbank feature for each window in 100 Mel bands (more bands than usual since high frequency components are more important than low frequency ones for SED). Finally, we conducted cepstral mean normalization (CMN) for each piece of training data. Feature vectors were calculated using HTK [22].

3.3. Model

We extended the hybrid HMM/neural network model in order to handle a multi-label classification problem. To do this, we built a three state left-to-right HMM with a non-active state for each sound event. The structure of our HMM is shown in Fig. 5, where $n = 0$, $n = 5$ and $n = 4$ represent the *initial state*, *final state*, and *non-active state*, respectively. Notice that the non-active state represents not only the case where there is no active event, but also the case where other events are active. Therefore, the non-active state of each sound event HMM has a different meaning from the silence. In this study, we fix all transition probabilities to a constant value of 0.5.

Using Bayes' theorem, HMM state emission probability $P(\mathbf{x}_t|s_{c,t} = n)$ can be approximated as follows

$$P(\mathbf{x}_t|s_{c,t} = n) = \frac{P(s_{c,t} = n|\mathbf{x}_t)P(\mathbf{x}_t)}{P(s_{c,t} = n)} \simeq \frac{P(s_{c,t} = n|\mathbf{x}_t)}{P(s_{c,t} = n)} \quad (9)$$

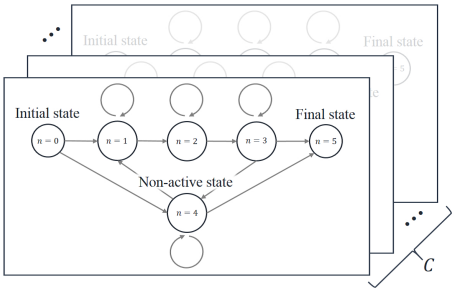


Figure 5: Hidden Markov Model of each sound event

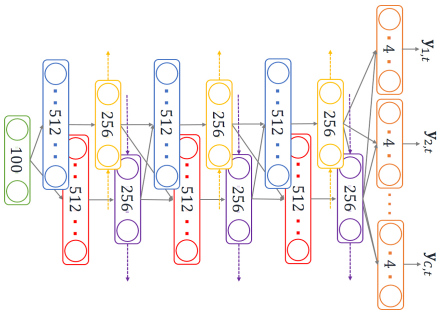


Figure 6: Proposed model structure

where $c \in \{1, 2, \ldots C\}$ represents the index of sound events, and $n \in \{1, 2, \ldots, N\}$ represents the index of HMM states, hence, $P(s_{c,t} = n|\mathbf{x}_t)$ satisfies the sum-to-one condition of $\sum_n P(s_{c,t} = n|\mathbf{x}_t) = 1$. In the BLSTM-HMM Hybrid model, HMM state posterior $P(s_{c,t} = n|\mathbf{x}_t)$ is calculated using a BLSTM-RNN. The structure of the network is shown in Fig. 6. This network has three hidden layers which consist of an LSTM layer, a projection layer, and the number of output layer nodes is $C \times N$. All values of the posterior $P(s_{c,t}|\mathbf{x}_t)$ have the sum-to-one condition for each sound event c at frame t, it is obtained by the following softmax operations

$$P(s_{c,t} = n|\mathbf{x}_t) = \frac{\exp(a_{c,n,t})}{\sum_{n'=1}^{N} \exp(a_{c,n',t})}, \quad (10)$$

where a represents the activation of output layer node. The network was optimized using back-propagation through time (BPTT) with Stochastic Gradient Descent (SGD) and dropout under the cross-entropy for *multi-class multi-label* objective function

$$E(\mathbf{\Theta}) = \sum_{c=1}^{C} \sum_{n=1}^{N} \sum_{t=1}^{T} y_{c,n,t} \ln(P(s_{c,t} = n|\mathbf{x}_t)), \quad (11)$$

where $\mathbf{\Theta}$ represents the set of network parameters, and $y_{c,n,t}$ is the HMM state label obtained from the maximum likelihood path at frame t. (Note that this is not the same as the multi-class objective function in conventional DNN-HMM.) HMM state prior $P(s_{c,t})$ is calculated by counting the number of occurrence of each HMM state. However, in this study, because our synthetic training data does not represent the actual sound event occurrences, the prior obtained from occurrences of HMM states has to be made less sensitive. Therefore, we smoothed $P(s_{c,t})$ as follows

$$\hat{P}(s_{c,t}) = P(s_{c,t})^{\alpha}, \quad (12)$$

where α is a smoothing coefficient. In this study, we set $\alpha = 0.01$. Finally, we calculated the HMM state emission probability using Eq. 9 and obtained the maximum likelihood path using the Viterbi algorithm.

4. EXPERIMENTS

4.1. Experimental condition

We evaluated our proposed method by using the DCASE2016 task 2 dataset [18, 6]. In this study, we randomly selected 5 samples per event from training data, and generated 18 samples which have 120 sec length just like DCASE2016 task 2 development set using selected samples. These generated samples are used as development set for open condition evaluation, and remaining 15 samples

Table 1: Experimental conditions

Sampling rate	44,100 Hz
Frame size	25 ms
Shift size	10 ms
Learning rate	0.0005
Initial scale	0.001
Gradient clipping norm	5
Batch size	64
Time steps	400
Epoch	20

per class are used for training. Evaluation is conducted by using two metrics: *event-based* evaluation, and *segment-based* evaluation, where an F1-score (F1) and an error rate (ER) are utilized as evaluation criteria (see [24] for more details).

We built our proposed model using following procedure: 1) divide an active event into three segments with equal intervals in order to assign left-to-right HMM state labels, 2) train the BLSTM-RNN using these HMM state labels as supervised data, 3) calculate the maximum likelihood path with the Viterbi algorithm using RNN output posterior, 4) train the BLSTM-RNN by using the obtained maximum likelihood path as supervised data, 5) repeat step 3 and step 4. In this study, when calculating the maximum likelihood path, we fixed the alignment of non-active states, i.e., we just aligned event active HMM states. When training networks, we checked the error for test data every epoch, and if the error became bigger than in previous epoch, we restored the parameters of previous epoch and re-train the network with a halved learning rate. All networks were trained using the open source toolkit TensorFlow [23] with a single GPU (Nvidia Titan X) . Details of the experimental conditions are shown in Table 1.

4.2. Comparison with conventional methods

To confirm the performance of our proposed method, we compared it with the following four methods: 1) NMF (DCASE2016 task2 baseline), 2) BLSTM-RNN, 3) BLSTM-RNN disregarding a few missing frames, 4) BLSTM-RNN with median filter smoothing. NMF is trained using remaining 15 samples per class by DCASE2016 task2 baseline script. In this study, **we do not change any settings** except for the number of training samples. BLSTM-RNN has the same network structure as BLSTM-HMM with the exception that the number of output layer nodes which have a sigmoid function as an activation function corresponds to the number of sound events C. Each node conducts a binary classification, hence, each output node y_c is between 0 and 1. We set the threshold as 0.5, i.e., $y_c > 0.5$ represents sound event c being active, and $y_c \leq 0.5$ non-active. For post-processing, we applied two methods, median filtering, and disregarding a few missing frames. In this time, we set the degree of median filtering to 9, and the number of disregarded frames to 10.

Experimental results are shown in Table 2. Note that the results of test set are provided by DCASE2016 organizers [18]. From the results, we can see that the methods based on BLSTM are significantly better than NMF-based method in polyphonic sound event detection. As regards post-processing, in study [8], the authors reported that they did not require post-processing since RNN outputs have already been smoothed. However, we confirmed that post-processing is still effective, especially for event-based evaluation. In addition, although RNN outputs are smoother than the outputs of neural networks without a recurrent structure, there is still room for improvement by smoothing RNN outputs. Our proposed method achieved the best performance for development set on event-based evaluation, which supports this assertion.

Table 2: Experimental results

	Event-based (dev / test)		Segment-based (dev / test)	
	F1 [%]	ER [%]	F1 [%]	ER [%]
NMF (Baseline)	14.6 / 24.2	665.4 / 168.5	35.9 / 37.0	183.4 / 89.3
BLSTM	66.5 / -	85.3 / -	87.0 / -	25.9 / -
BLSTM (w/ disregard)	75.9 / -	52.5 / -	87.0 / -	25.9 / -
BLSTM (w/ median)	75.8 / **68.2**	53.2 / **60.0**	**87.7** / **78.1**	**24.2** / **40.8**
BLSTM-HMM	**76.6** / 67.1	**51.1** / 64.4	87.2 / 76.0	25.9 / 50.0

Table 3: Effect of background noise

	Event-based		Segment-based	
EBR [dB]	F1 [%]	ER [%]	F1 [%]	ER [%]
-6	73.7	58.0	86.0	28.0
0	76.7	51.3	87.4	25.9
6	79.6	44.1	88.1	23.9

4.3. Analysis

In this section, we focus on the factors which influenced the performance of our proposed method using development set. The first factor is SNR between background noise and events. The performance of our proposed method for development set of each SNR condition is shown in Table 3. From these results, there are clear differences in performance between the different SNR conditions. This is because the loud background noise caused more insertion errors, especially small loudness events such as *doorslam* and *pageturn*.

The second factor is the difference in performance between the monophonic and polyphonic tasks. The performance of our proposed method on each type of tasks is shown in Table 4. In general, polyphonic task is more difficult than monophonic task. However, we observed a strange behavior with better scores in the polyphonic task than in the monophonic task, while the opposite is normally to be expected. We will investigate the reason as a future work.

5. CONCLUSION

We proposed a new method of polyphonic sound event detection based on a Bidirectional Long Short-Term Memory Hidden Markov Model hybrid system (BLSTM-HMM), and applied it to the DCACE2016 challenge task 2. We compared our proposed method to baseline non-negative matrix factorization (NMF) and standard BLSTM-RNN methods. Our proposed method outperformed them in both monophonic and polyphonic tasks, and finally achieved an average F1-score of 67.1% (error rate of 64.5%) on the event-based evaluation, and an average F1-score 76.0% (error rate of 50.0%) on the segment-based evaluation.

In future work, we will investigate the reason for the counter-intuitive results in the difference between monophonic and polyphonic task, the use of sequence discriminative training for BLSTM-HMM, and we will apply our proposed method to a real-recording dataset.

Table 4: Difference in the performance between monophonic and polyphonic task

	Event-based		Segment-based	
	F1 [%]	ER [%]	F1 [%]	ER [%]
Monophonic	76.2	54.0	84.3	32.4
Polyphonic	76.9	49.6	88.7	22.5

6. REFERENCES

[1] J. Schrder, B. Cauchi, M. R. Schdler,N. Moritz, K. Adiloglu, J. Anemller, and S. Goetze, "Acoustic event detection using signal enhancement and spectro-temporal feature extraction," in *Proc. WASPAA*, 2013.

[2] T. Heittola, A. Mesaros, A. Eronen, and T. Virtunen, "Context-dependent Sound Event Detection," *EURASIP Journal on Audio, Speech, and Music Processing*, Vol. 2013, No.1, 2013, pp. 1–13.

[3] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Workshop on machine listening in Multisource Environments*, 2011, pp. 36–40.

[4] S. Innami, and H. Kasai, "NMF-based environmental sound source separation using time-variant gain features," *Computers & Mathematics with Applications*, Vol. 64, No. 5, 2012, pp.1333–1342.

[5] A. Dessein, A. Cont, and G. Lemaitre, "Real-time detection of overlapping sound events with non-negative matrix factorization," *Springer Matrix Information Geometry*, 2013, pp. 341–371.

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

[7] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. IEEE IJCNN*, 2015, pp. 1–7.

[8] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *Proc. IEEE ICASSP*, 2016, pp. 6440–6444.

[9] T. Hayashi, M. Nishida, N. Kitaoka, and K. Takeda, "Daily activity recognition based on DNN using environmental sound and acceleration signals, " in *Proc. EUSIPCO*, 2015, pp. 2306–2310.

[10] M. Espi, M. Fujimoto, K. Kinoshita, T. Nakatani, "Exploiting spectro-temporal locality in deep learning based acoustic event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, Vol.1, No.1, 2015.

[11] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks," in *IEEE ICASSP*, 2016, pp. 2742–2746.

[12] F. Eyben, S. Bck, B. Schuller, and A. Graves, "Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks," in *ISMIR*, 2010, pp. 589–594.

[13] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE ICASSP*, 2013 , pp. 6645–6649.

[14] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, " *ArXiv e-prints arXiv:1402.1128*, 2014.

[15] H. Sak *et. al.,* "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. IEEE INTERSPEECH*, 2014.

[16] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey, "Integration of speech enhancement and recognition using long Short-term memory recurrent neural network," in *Proc. IEEE INTERSPEECH*, 2015, pp. 3274–3278.

[17] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, 1997, pp. 2673–2681.

[18] `http://www.cs.tut.fi/sgn/arg/dcase2016/`.

[19] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, Vol. 5, No. 2, 1994, pp.157–166.

[20] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural Computation*, No. 9, Vol. 8 ,1997, pp. 1735–1780.

[21] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Artificial Neural Networks*, Vol. 12, No. 10, 1999, pp. 2451–2471.

[22] `http://htk.eng.cam.ac.uk`

[23] `https://www.tensorflow.org`

[24] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection", *Applied Sciences*, Vol. 6, No. 6, 2016, 162.

ESTIMATING TRAFFIC NOISE LEVELS USING ACOUSTIC MONITORING: A PRELIMINARY STUDY

Jean-Rémy Gloaguen, Arnaud Can

Ifsttar - LAE
Route de Bouaye - CS4
44344, Bouguenais, FR
jean-remy.gloaguen@ifsttar.fr

Mathieu Lagrange, Jean-François Petiot

IRCCyN, UMR CNRS 6597
École Centrale de Nantes
1 rue de la Noe
44321, Nantes, FR

ABSTRACT

In this paper, Non-negative Matrix Factorization is applied for isolating the contribution of road traffic from acoustic measurements in urban sound mixtures. This method is tested on simulated scenes to enable a better control of the presence of different sound sources. The presented first results show the potential of the method.

Index Terms— Non-negative Matrix Factorization, road traffic noise mapping, urban measurements

1. INTRODUCTION

Noise in cities is one of the main sources of annoyance essentially caused by road, air an rail traffic. To know better the noise spatial distribution, the number of people impacted and to preserve quiet areas, the European Directive 2002/49/EC [1] requires that cities over 250 000 inhabitants produce noise maps for road, air and rail traffic. Road traffic noise maps are produced based on a census of the traffic volumes and mean speeds along the main roads which allows estimating their acoustic emission. Assuming knowledge of the city topography, the acoustic propagation within the streets is then calculated. In addition, noise observatories are being deployed in some agglomerations. They aim to facilitate both the mandatory five year update of maps and the validation of the simulated noise maps. Combining classical noise maps with measures would also be a promising approach to go towards more accurate noise maps [2] [3].

However, to achieve those important goals, we have to isolate the road traffic contribution from measurements of the sound mixture that contain many other sources. Indeed, urban sound environments are composed of a large variety of sounds as traffic noise, horn, bird whistles, foot steps, construction sound noise, voices ... Each has its own spectral properties and temporal structure and may overlap with the other sound sources. Without distinction between these, the traffic noise level estimation is calculated with some sources which do not belong to a traffic car class and is then overestimated. In this study car horn and braking noise are not considered as a traffic car noise as they are not taken into account in traffic noise map.

Different techniques exist and were shown relevant for recognition or detection in urban environment [4] [5] but they do not take into account the overlap between the sources. Methods for source separation, such as Computational Auditory Scene Analysis [6] or Independent Component Analysis [7] are efficient but are, to the best of our understanding, not suitable for urban applications. Indeed, the first one has been primarily developed to simulate the human auditory system whereas the second one requires as many sensors as sound sources, which is unrealistic in a urban context.

Non-negative Matrix Factorization (NMF) [8] has the advantage to deal with the overlap between the sound sources. It has been used for many applications in audio domain such as polyphonic music transcription [9] or for source separation of musical content [10]. Thus the NMF seems to be a suitable method for the isolation of the contribution of road traffic from measurements. We propose to apply an NMF scheme on a corpus of urban sound mixtures to validate its ability to estimate the noise level of road traffic. The specificity of urban sound environments, and the fact that the method has, to the best of our knowledge, never been used in this setting, stands as a challenge and requires specific adaptations.

In this paper, we present the implementation of our experimental plan and some first results. Section 2 exposes the structure of the proposed system based on the NMF framework. Then the experimental protocol is presented in Section 3 and preliminary results are discussed in section 4.

2. PROPOSED APPROACH

The aim of the system is to estimate the level of some predefined sources in the mixture coming from measurements of the urban scene. As can be seen in Figure 1, the signal is

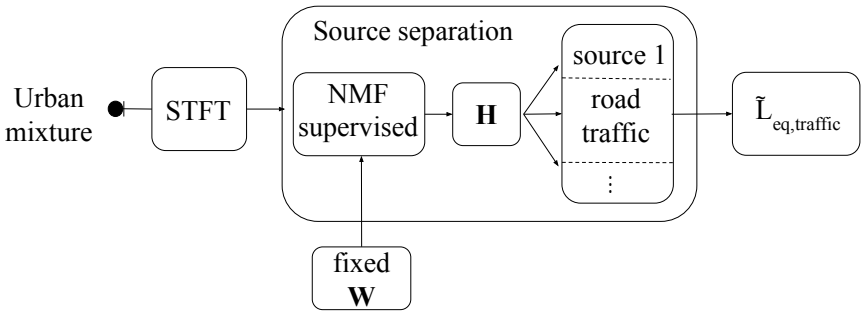


Figure 1: Block diagram of the proposed method

first mapped to a time-frequency plane using the Short Time Fourier Transform. Using the NMF framework, the contribution of the road traffic is isolated and its level, $\tilde{L}_{eq,tr}$, is estimated.

2.1. Non-Negative Matrix Factorization

Non-negative Matrix Factorization is a dimension-reduction technique expressed by

$$\mathbf{V} \approx \tilde{\mathbf{V}} = \mathbf{WH} \tag{1}$$

where $\mathbf{V}_{F\times N}$, is the power spectrogram of an audio, $\tilde{\mathbf{V}}$ is the approximate power spectrogram determined by the NMF, $\mathbf{W}_{F\times K}$, is the basis matrix (called dictionary), in our case, representing a set of sound spectra usually found in urban areas. $\mathbf{H}_{K\times N}$ is the feature matrix standing for the temporal variation of each spectrum. All these elements are constrained to be positive leading to additive combinations only. The approximation (1) is determined by a minimization problem

$$\min_{\mathbf{W},\mathbf{H}\geq 0} D(\mathbf{V}||\mathbf{WH}). \tag{2}$$

$D(\mathbf{V}||\mathbf{WH})$ is called cost function, a dissimilarity measure usually belonging to the β-divergence for the NMF. 3 popular expressions are compared in this study namely the Euclidean distance ($\beta = 2$),

$$D_{EUC}(\mathbf{V}||\mathbf{WH}) = ||\mathbf{V} - \mathbf{WH}||, \tag{3}$$

the Kullback-Leibler divergence, ($\beta = 1$),

$$D_{KL}(V||WH) = \mathbf{V}\log\frac{\mathbf{V}}{\mathbf{WH}} - \mathbf{V} + \mathbf{WH}, \tag{4}$$

and the Itakura-Sato divergence, ($\beta = 0$),

$$D_{IS}(V||WH) = \frac{\mathbf{V}}{\mathbf{WH}} - \log\frac{\mathbf{V}}{\mathbf{WH}} - 1. \tag{5}$$

Note that decimal β values between 0 and 2 will be investigated in a further study. Here, the supervised NMF is considered where $\mathbf{W}$ is fixed and only $\mathbf{H}$ is updated iteratively. The choosen algorithm is the maximisation-minimisation algorithm proposed by Févotte and Idier [11].

$$\mathbf{H}^{k+1} \longleftarrow \mathbf{H}^k.\left(\frac{\mathbf{W}^T\left[(\mathbf{WH}^k)^{\beta-2}.\mathbf{V}\right]}{\mathbf{W}^T\left[\mathbf{WH}^k\right]^{\beta-1}}\right)^{\gamma(\beta)} \tag{6}$$

where $\gamma(\beta) = 1$ for $\beta \in [1\ 2]$ and $\gamma(\beta) = \frac{1}{2}$ for $\beta = 0$.

2.2. Method

Our approach consists in considering an audio signal recorded in an urban context, sampled at $44,1$ kHz and expressed in the time-frequency plan using a Short Time Fourier Transform. The size of the Hanning window is 5000 points with an overlap of 50 % and $NFFT = 4096$ points. The temporal resolution chosen is, for the moment, very low ($\Delta t \approx 0,05$ s).

The supervised NMF is then performed with the spectrogram $\mathbf{V}$ in the input, a fixed dictionary $\mathbf{W}$ and $\tilde{\mathbf{V}}$ in the output. Currently, $\mathbf{H}$ is updated for a number of iterations fixed at 100. When the iteration is over, it is possible to estimate the level of the elements of interest. In the case of road traffic, $\tilde{\mathbf{V}}_{tr} = [\mathbf{WH}]_{tr}$ which allows to calculate the sound pressure level $\tilde{L}_p$ for each temporal frame

$$\tilde{L}_{p,tr,n} = 20\log\frac{\sum_f \tilde{\mathbf{v}}_{\mathbf{n},\mathbf{tr}}}{p_0} \tag{7}$$

with $\tilde{\mathbf{v}}_{\mathbf{n},\mathbf{tr}}$, the n-th temporal frame of the matrix $\tilde{\mathbf{V}}_{tr}$ and $p_0 = 2\times 10^{-5}$ Pa, the reference sound pressure. The equivalent traffic sound level estimated, $\tilde{L}_{eq,tr}$, is then determined by

$$\tilde{L}_{eq,tr} = \frac{1}{T} \sum_{n} 10 \log \left(10^{\tilde{L}_{p,tr,n}/10} \right) \quad (8)$$

where T is the duration of $\mathbf{V}$.

3. EXPERIMENT

To evaluate the ability of the NMF framework to estimate the road traffic level, we consider simulated sounds mixtures where the actual level of contribution of the traffic is known. This solution ensures controlling the road traffic level, $L_{eq,tr}$, relatively to the other sources in comparison to real recordings where it would not be correctly determined. Furthermore, working on simulated sound mixtures will create a controlled framework where the time of presence of each source is exactly known. Thus allows the production of specific sound environments (animated streets, parks ...).

The mixtures are simulated with *simScene* software developed by Mathias Rossignol and Gregoire Lafay [12][1] which synthesizes sound mixtures from a sound database of isoled sound events. This tool can control multiple parameters as the event/background ratio, the sample duration, the time between samples ... Each of these parameters is coupled with a standard deviation to bring some variability between the scenes produced. In the output, an audio file of each sound class is created that allows us to compute the specific contribution of each class present in the scene. The sound database we use is composed of sound samples provided with the software and completed by others sounds found online[2]. The scenes are built with the first half of the database, the second half being considered as the dictionary $\mathbf{W}$. For tests of feasibility, the first constructed scenes are simple but more realistic scenes fully consistent will be soon produced.
For this preliminary study, 20 scenes are created with a duration of 15 s. Each one is composed of 3 classes of sounds that can typically be heard in urban areas: *car*, *bird* and *car horn* and a noise background (voice hubbub). Currently, our dataset for creating these scenes is composed of 30 audio samples for the *car* class and 3 samples for *bird* and *horn* classes. The dictionary W is then composed of the same number of samples but extracted form the second half of our database. The aim of this preliminary study is to see the influence of some parameters of the NMF (such as the divergence calculation or the number of iteration) on the quality of the traffic noise levels estimation. The NMF is performed on each scene i and $L^i_{eq,tr}$ is compared with the computational level $\tilde{L^i}_{eq,tr}$ to evaluate the performance of the method by computing the error,

$$RSME = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (L^i_{eq,tr} - \tilde{L}^i_{eq,tr})^2} \quad (9)$$

where N, the number of scenes created.

4. RESULTS

Figure 2 presents the spectrograms obtained by *simScene* (on left) and by the NMF (on right) for one scene with the Euclidean distance (3) after 100 updates of $\mathbf{H}$. We can observe the bird on the frequency range $[3000 - 6000]$ Hz, the horn is characterized by its harmonic content whereas the car is mainly composed of low frequencies with a slower temporal evolution. From each sound mixture, comparison between $L_{p,n}$ and $\tilde{L}_{p,n}$ for Euclidean distance (EUC), Kullback-Leibler (K-L) and Itakura-Sato (I-S) divergences can be made (Figure 3 for the scene presented in Figure 2).

For this scene, in the time interval $[1.5 - 4.5]\, s$, there is no traffic, the actual sound level is then zero. But we can see in Figure 3 that the class *car* contributes to describe the noise background level. This result is the consequence of the minimization problem (2) where this sound class is activated to reduce the cost function even though there is no traffic. Nevertheless, the noise background is low enough in comparison with the other class sounds to not distort the estimations.

Let us now consider the error RMSE with respect to the number of iterations of the NMF computed on the N scenes for the three β-divergences on Figure 4. The error between $L_{eq,tr}$ and the equivalent sound pressure of the global mixture, L_{eq} (global error), is added. This corresponds to the error that would be made if no source separation was done and all the sound sources were taken into account without distinction.

Even if the global error is low ($\approx 2dB$), the use of the NMF to compute the traffic noise level produces a better estimation than taking the sound mixture with all the sound source. The Kullback-Leibler divergence produced the most interesting results with the lowest and the most stable RMSE. Surprisingly, the Itakura-divergence, despite its scale invariant property [11], has an error similar to the Euclidean distance. This result may change in the future with more complex and more realistic scenes.

5. CONCLUSION

In this article, we proposed to use the supervised NMF framework to estimate the road traffic noise levels based on acoustic measurements achieved in an urban context. In

[1] Open-source project available at: `https://bitbucket.org/mlagrange/simscene`

[2] `www.freesound.org`

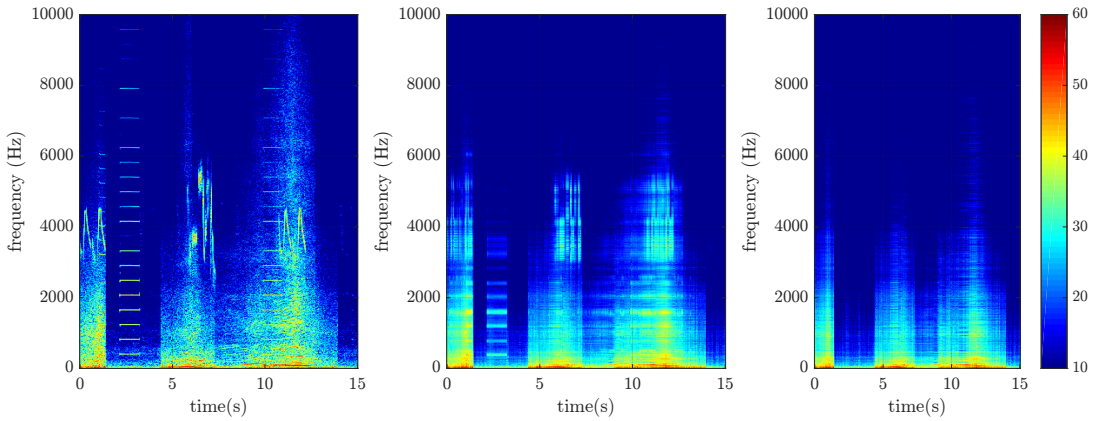


Figure 2: Spectrograms of a sound mixture composed with 3 sound classes (*car*, *horn*, *bird*). On the left, the initial audio spectrogram given by *simScene*, in the middle, the estimation $\tilde{\mathbf{V}}$ given by the NMF, on the right, the traffic car noise estimated $\tilde{\mathbf{V}}_{\mathbf{tr}}$ after the source separation.

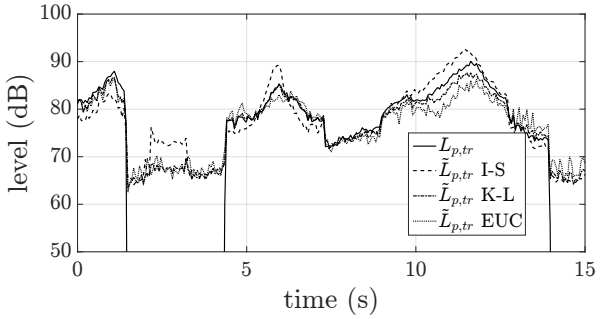


Figure 3: Evolution, according to time, of the actual sound pressure level, $L_{p,tr}$, and the estimated levels .

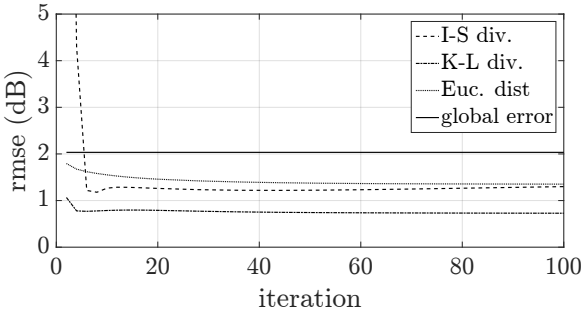


Figure 4: RMSE evolution

our opinion, such approach would find many applications in the environmental acoustics field such other than improving noise maps with acoustic measurements, for example acoustic biodiversity monitoring.

This method is tested on sound mixtures simulated using the *simScene* software which allows us to get the exact traffic contribution separately from the other sounds. The method is tested by comparing the equivalent sound level between the traffic element of *simScene* and the estimation given by the NMF for three cost functions. The first results show that this method gives a better estimation of the sound level than if the source separation is not done, thus demonstrating its interest. Both the road traffic time of presence and amplitude are accurately estimated, advocating for the use of the NMF for isolating the road traffic contribution. The Kullback-Leibler divergence results in the lowest errors and will therefore receive specific attention for future work.

Further investigations with more realistic and complex scenes are now required to confirm the behavior of the Kullabck-divergence. Then, some refinements of the NMF including acoustics considerations should improve the goodness of the road traffic noise levels estimation. For instance, the addition of some temporal constraints with a smoothness constraint within the NMF framework such as [13] [14] [15] to better model the temporal evolution of the traffic elements is currently investigated.

6. REFERENCES

[1] "Directive 2002/49/EC relating to the assessment and management of environmental noise." [Online]. Available: http://ec.europa.eu/environment/noise/directive_en.htm

[2] A. Can, L. Dekoninck, and D. Botteldooren, "Measurement network for urban noise assessment: Comparison of mobile measurements and spatial interpolation ap-

proaches," *Applied Acoustics*, vol. 83, pp. 32–39, Sept. 2014.

[3] W. Wei, T. Van Renterghem, B. De Coensel, and D. Botteldooren, "Dynamic noise mapping: A map-based interpolation between noise measurements with high temporal resolution," *Applied Acoustics*, vol. 101, pp. 127–140, Jan. 2016.

[4] 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, no. 2, pp. 881–891, 2007.

[5] B. Defreville, F. Pachet, C. Rosin, and P. Roy, "Automatic Recognition of Urban Sound Sources." Audio Engineering Society, 2006.

[6] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech & Language*, vol. 8, no. 4, pp. 297–336, Oct. 1994.

[7] P. Comon, "Higher Order Statistics Independent component analysis, A new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[8] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[9] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," pp. 177–180, Oct. 2003.

[10] M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *Proc. EUSIPCO2005.*, 2005.

[11] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[12] M. Rossignol, G. Lafay, M. Lagrange, and N. Misdariis, "SimScene: a web-based acoustic scenes simulator," in *1st Web Audio Conference (WAC)*, 2015.

[13] C. Févotte, "Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 International Conference on IEEE*. IEEE, 2011, pp. 1980–1983.

[14] S. Essid and C. Févotte, "Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 415–425, 2013.

[15] T. Virtanen, "Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

ACOUSTIC EVENT DETECTION METHOD USING SEMI-SUPERVISED NON-NEGATIVE MATRIX FACTORIZATION WITH A MIXTURE OF LOCAL DICTIONARIES

Tatsuya Komatsu, Takahiro Toizumi, Reishi Kondo, and Yuzo Senda

Data Science Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki 211-8666, Japan

ABSTRACT

This paper proposes an acoustic event detection (AED) method using semi-supervised non-negative matrix factorization (NMF) with a mixture of local dictionaries (MLD). The proposed method based on semi-supervised NMF newly introduces a noise dictionary and a noise activation matrix both dedicated to unknown acoustic atoms which are not included in the MLD. Because unknown acoustic atoms are better modeled by the new noise dictionary learned upon classification and the new activation matrix, the proposed method provides a higher classification performance for event classes modeled by the MLD when a signal to be classified is contaminated by unknown acoustic atoms. Evaluation results using DCASE2016 task 2 Dataset show that F-measure by the proposed method with semi-supervised NMF is improved by as much as 11.1% compared to that by the conventional method with supervised NMF.

Index Terms— Acoustic event detection, Non-negative matrix factorization, Semi-supervised NMF, Mixture of local dictionaries

1. INTRODUCTION

To identify a physical event or a sound source by which an observed acoustic signal has been produced, acoustic event detection (AED) is studied in various research fields such as smart home systems [1, 2], environmental and ecological surveillance [3, 4], and audio and video indexing [5, 6, 7]. Particularly, to make cities safer, AED as part of a monitoring system is expected to find hazardous sounds related to crimes, accidents, and incidents in public spaces [8, 9]. Environmental sound coexisting with a target acoustic signal causes wrong feature extraction and results in failure of detection. AED methods based on non-negative matrix factorization (NMF) have been proposed as promissing solutions [10, 11, 12, 13]. For AED, NMF models an acoustic event as a combination of acoustic atoms which constitutes spectra of acoustic events. NMF-based methods learn a dictionary of acoustic atoms by decomposing training signals into their spectral bases. A signal to be classified is decomposed into bases of the dictionary and the corresponding activation matrix by supervised NMF. The extracted activation matrix represents a mixture ratio of acoustic atoms in the signal and is used as a feature vector.

One of the most important points for an NMF-based AED method is how to learn a dictionary of acoustic atoms. Gemmeke et al. [14] made a dictionary by concatenating event specific basis matrices which were extracted by performing NMF on each acoustic event individually. However, when different acoustic events share the same acoustic atoms, the dictionary becomes redundant. This redundancy prevents proper extraction of an activation matrix. Komatsu et al. [15] used a mixture of local dictionaries (MLD) [16] constituting sub-groups of bases which directly models acoustic atoms. The MLD is learned with constrained NMF using a prior knowledge of acoustic atoms, which is obtained from clustered spectra of training signals. Modeling acoustic atoms directly by sub-groups of basis, the MLD has less redundancy and performs more accurate feature extraction. However, the conventional method performs supervised NMF [17, 18] using their fixed dictionaries upon classification. When a signal to be classified has unknown spectra (e.g. environmental sound) which are not included in training signals, the unknown spectra are expressed by acoustic atoms in the training signals. The extracted activation matrix is contaminated by unknown spectra and leads to failure of detection.

This paper proposes an AED method using semi-supervised NMF with the MLD. The proposed method based on semi-supervised NMF newly introduces a noise dictionary and a noise activation matrix both dedicated to unknown acoustic atoms which are not included in training data. Because unknown acoustic atoms are better modeled by the new noise dictionary learned upon classification and the new activation matrix, the proposed method provides a higher classification capability for event classes modeled by MLD when a signal to be classified are contaminated by unknown acoustic atoms.

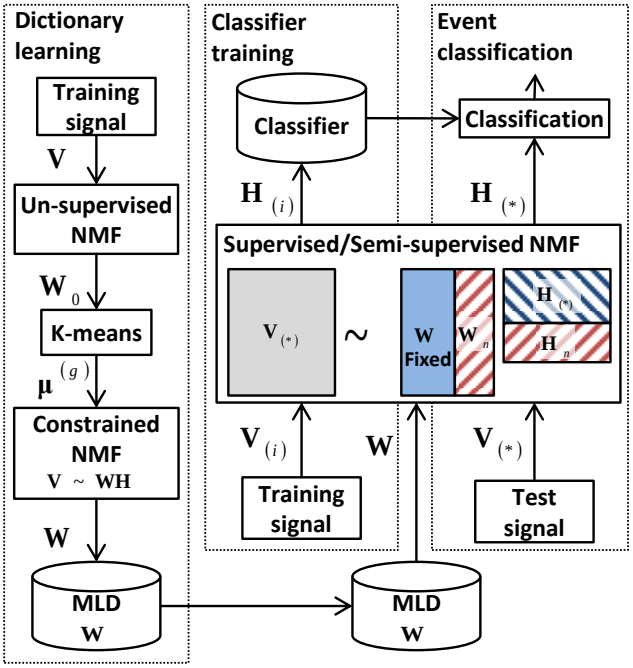


Figure 1: Block diagram of the proposed acoustic event detection.The supervised NMF is a special case of semi supervised NMF without a noise dictionary $\mathbf{W}_n$ and a noise activation matrix $\mathbf{H}_n$.

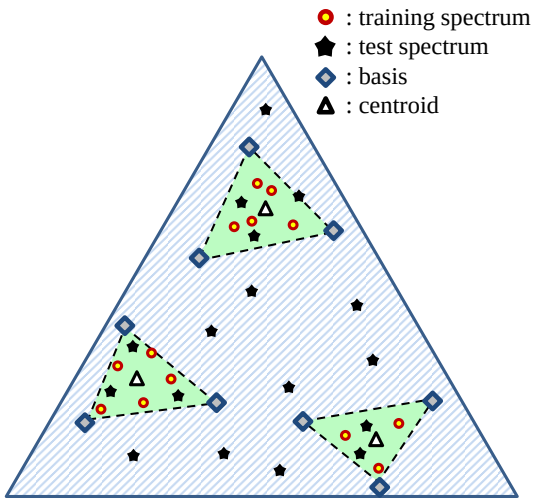


Figure 2: Relationship among training/test spectrum, the MLD, and the noise dictionary

2. PROPOSED METHOD

Figure 1 shows a block diagram of the proposed method. It consists of three parts, dictionary learning, classifier training, and event classification. Acoustic signals are used after being transformed to spectrograms.

In dictionary learning, the training spectrogram $\mathbf{V}$ is decomposed into an initial basis matrix $\mathbf{W}_0$ by basic un-supervised NMF [19]. Next, K-means clustering is applied to $\mathbf{W}_0$, and G centroids $\boldsymbol{\mu}^{(g)}$ are obtained where $g \in \{1, ..., G\}$ denotes an index of centroid. A MLD $\mathbf{W}$ is learned by constrained NMF using $\boldsymbol{\mu}^{(g)}$ as prior knowledge.

In classifier training, an event-specific activation matrix $\mathbf{H}_{(i)}$ is extracted from the corresponding spectrogram $\mathbf{V}_{(i)}$ with supervised NMF using the MLD $\mathbf{W}$ where i denotes an index of each acoustic event class. Column vectors of $\mathbf{H}_{(i)}$ at each time frame are used as feature vectors for training the classifier.

In event classification, unlike classifier training, semi-supervised NMF is applied to a test spectrogram $\mathbf{V}_{(*)}$ with the MLD $\mathbf{W}$ and a noise dictionary $\mathbf{W}_n$ which is learned from $\mathbf{V}_{(*)}$. $\mathbf{W}_n$ and $[\mathbf{H}_{(*)}, \mathbf{H}_n]$ which are activation matrices of the MLD and the noise dictionary are alternately updated. Unknown spectra included in $\mathbf{V}_{(*)}$ are expressed by $\mathbf{W}_n$ and $\mathbf{H}_n$, so that $\mathbf{H}_{(*)}$ is extracted properly. The classifier uses only $\mathbf{H}_{(*)}$ as a feature vector for classification of acoustic event classes.

2.1. Dictionary learning

MLD consists of G sub-groups of bases which model acoustic atoms $\mathbf{W} = [\mathbf{W}^{(1)}, ..., \mathbf{W}^{(G)}]$. A basis matrix $\mathbf{W}^{(g)} \in \mathcal{R}_+^{F \times K_g}$ consists of K_g basis vectors where $\mathcal{R}_+^{F \times K_g}$, F, and g denote a set of non-negative $F \times K_g$ matrices, the number of frequency bins, and an index of each acoustic atom.

To determine acoustic atoms, an initial basis matrix $\mathbf{W}_0$ is first extracted from the entire training data spectrogram $\mathbf{V} \in \mathcal{R}_+^{F \times T}$ with the basic un-supervised NMF where T denotes its number of time frames. K-means clustering is then applied to bases in $\mathbf{W}_0$ to select G centroids $\boldsymbol{\mu}^{(g)}$ which represent centroids of acoustic atoms.

NMF is again applied to $\mathbf{V}$ with the centroids $\boldsymbol{\mu}^{(g)}$ and the following cost function $\mathcal{D}(\mathbf{V}|\boldsymbol{\Lambda})$:

$$\mathcal{D}(\mathbf{V}|\boldsymbol{\Lambda}) = \mathcal{D}_{\mathcal{KL}}(\mathbf{V}|\boldsymbol{\Lambda}) + \eta \sum_g \mathcal{D}_{\mathcal{KL}}(\boldsymbol{\mu}^{(g)}|\mathbf{W}^{(g)}) + \lambda \sum_t \Omega(\mathbf{h}_t), \quad (1)$$

where $\boldsymbol{\Lambda} = \mathbf{WH}$ is approximation of $\mathbf{V}$ and $\mathbf{H}$ is an activation matrix of MLD $\mathbf{W}$. A column vector $\mathbf{h}_t$ of $\mathbf{H}$ at time frame t consists of activations $\mathbf{h}_t^{(g)}$ for $\mathbf{W}^{(g)}$ (g = 1,...,G),

$$\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_t, ..., \mathbf{h}_T], \quad (2)$$

$$\mathbf{h}_t^\top = [\mathbf{h}_t^{(1)\top}, ..., \mathbf{h}_t^{(g)\top}, ..., \mathbf{h}_t^{(G)\top}], \quad (3)$$

where $[\cdot]^\top$ denotes a matrix transpose.

Cost function in (2) consists of three terms; a generalized Kullback-Leibler(KL) divergence $\mathcal{D}_{\mathcal{KL}}(\mathbf{V}|\boldsymbol{\Lambda})$ between $\mathbf{V}$ and $\boldsymbol{\Lambda}$, a constraint $\sum_g \mathcal{D}_{\mathcal{KL}}(\boldsymbol{\mu}^{(g)}|\mathbf{W}^{(g)})$, and a group sparsity constraint $\sum_t \Omega(\mathbf{h}_t)$. The first term is a generalized KL divergence used by the basic un-supervised NMF algorithm. The second term is a constraint which allocates sub-groups of bases $\mathbf{W}^{(g)}$ to g th acoustic atoms characterized by the centroid $\boldsymbol{\mu}^{(g)}$. The strength of constraint is controlled by η. The third term represents group sparsity constraint at time t controlled by λ, where

$$\Omega(\mathbf{h}_t) = \sum_g \log(\epsilon + \|\mathbf{h}_t^{(g)}\|_1) \quad (4)$$

is used in prior arts [16, 20] to turn off activation of the irrelevant acoustic atoms.

To minimize the cost function in (2), the following update rules are iteratively applied:

$$\mathbf{W}^{(g)} \leftarrow \mathbf{W}^{(g)} \odot \left\{ \left(\frac{\mathbf{V}}{\boldsymbol{\Lambda}} \right) \mathbf{H}^\top + \eta \frac{\boldsymbol{\mu}^{(g)}}{\mathbf{W}^{(g)}} \right\} \Big/ \left\{ \mathbf{1} \left(\mathbf{H}^\top + \eta \right) \right\}, \quad (5)$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left\{ \mathbf{W}^\top \left(\frac{\mathbf{V}}{\boldsymbol{\Lambda}} \right) \right\} \Big/ \left\{ \mathbf{W}^\top \mathbf{1} \right\}, \quad (6)$$

$$\mathbf{h}_t^{(g)} \leftarrow \mathbf{h}_t^{(g)} \frac{1}{1 + \lambda \Big/ \left\{ \epsilon + ||\mathbf{h}_t^{(g)}||_1 \right\}} \quad (7)$$

where $\mathbf{1}$ is a matrix with all elements equal to 1 and with a dimension of $\mathbf{V}$. $\mathbf{A} \odot \mathbf{B}$ represents element wise multiplication, $\mathbf{A}/\mathbf{B}$ and $\frac{\mathbf{A}}{\mathbf{B}}$ represent element wise division. The procedure of dictionary learning is shown in Algorithm 1.

Algorithm 1 Dictionary learning for MLD

1: INPUT: $\mathbf{V}$
2: Obtain $\mathbf{W}_0$ by a basic NMF
3: Obtain $\boldsymbol{\mu}^{(g)}$ by using K-means to $\mathbf{W}_0$
4: Initialize $\mathbf{W}$ and $\mathbf{H}$ with random values.
5: **repeat**
6: Update $\mathbf{W}$ using (5).
7: Update $\mathbf{H}$ using (6) and (7).
8: **until** Convergence
9: OUTPUT: $\mathbf{W}$

2.2. Classifier training

In classifier training, an activation matrix $\mathbf{H}_{(i)}$ is extracted from the corresponding training spectrogram $\mathbf{V}_{(i)}$ by supervised NMF with MLD $\mathbf{W}$ and a classifier is trained using the activation matrices where $i \in \{1, ..., I\}$ represents an event-class index. In supervised NMF, $\mathbf{V}_{(i)}$ is approximated by a product of $\mathbf{W}$ and $\mathbf{H}_{(i)}$,

$$\mathbf{V}_{(i)} \sim \mathbf{W}\mathbf{H}_{(i)}. \tag{8}$$

For a given $\mathbf{W}$ by dictionary learning, $\mathbf{H}_{(i)}$ is updated using (6) and the group sparsity constraint in (7). The procedure is shown in Algorithm 2.

Once $\mathbf{H}_{(i)}$ has been obtained, column vectors $\mathbf{h}_{t(i)}$ of $\mathbf{H}_{(i)}$ at each time frame t are used as feature vectors to train the classifier. Simple linear support vector machine(SVM) [21] is used for classifier. Multi-class SVM is trained based on the one-against-all approach.

Algorithm 2 Feature extraction with supervised NMF

1: INPUT: $\mathbf{V}_{(i)}$ and $\mathbf{W}$
2: Initialize $\mathbf{H}_{(i)}$ with random values.
3: **repeat**
4: Update $\mathbf{H}_{(i)}$ using (6) and (7). with fixed $\mathbf{W}$
5: **until** Convergence
6: OUTPUT: $\mathbf{H}_{(i)}$

2.3. Event classification

In event classification, the proposed method extracts an activation matrix from a test spectrogram using semi-supervised NMF with MLD. A noise dictionary is learned concurrently with extracing the activation matrix. Unknown spectra included in a test spectrogram is expressed by the noise dictionary and the corresponding activation matrix, so that an activation matrix of acoustic atoms are extracted properly.

Let $\mathbf{V}_{(*)}$ and $\mathbf{W}_n \in \mathcal{R}_+^{F \times K_n}$ denote the test spectrogram and the noise dictionary, respectively, where K_n is the number of bases in the noise dictionary. $\mathbf{H}_{(*)}$ and $\mathbf{H}_n$ denote activation matrices of MLD and $\mathbf{W}_n$, respectively. The relationship among these matrices is described as in the following approximation:

$$\mathbf{V}_{(*)} \sim \mathbf{\Lambda}_{(*)} = [\mathbf{W}, \mathbf{W}_n] \begin{bmatrix} \mathbf{H}_{(*)} \\ \mathbf{H}_n \end{bmatrix}. \tag{9}$$

In semi-supervised NMF, $\mathbf{H}_{(*)}$, $\mathbf{H}_n$ and $\mathbf{W}_n$ are updated to minimize a generalized KL divergence $\mathcal{D}_{\mathcal{KL}}\left(\mathbf{V}_{(*)}|\mathbf{\Lambda}_{(*)}\right)$, applying an update rule for the activation matrix in (6), a group sparsity constraint in (7) and the following update rule for $\mathbf{W}_n$:

$$\mathbf{W}_n \leftarrow \mathbf{W}_n \odot \left\{ \left(\frac{\mathbf{V}_{(*)}}{\mathbf{\Lambda}_{(*)}} \right) \mathbf{H}_n^\top \right\} \Big/ \left\{ \mathbf{1}\mathbf{H}_n^\top \right\}. \tag{10}$$

The procedure is shown in Algorithm 3.

Figure 2 is a simple illustration of the relationship among training/test spectrum, MLD, and the noise dictionary. The relationship is explained as data points on the 3-dimentional simplex [22, 23]. ◯ and ★ represent training and test spectrum, respectively, ◊ and △ represent bases and centroids of MLD, respectively. In dictionary learning, sub-groups of bases ◊ in MLD are learned to span convex hulls enclosing training spectra ◯. In event classification, the noise dictionary is learned from unknown test spectra ★ lying outside the convex hulls which is indicated with the shaded area. Therefore unknown spectra included in the test spectrogram are expressed by the noise dictionary and MLD can extract a proper activation matrix of acoustic atoms.

After extracting $\mathbf{H}_{(*)}$, the classifier receives $\mathbf{H}_{(*)}$ as a feature and outputs a $T \times I$ binary classification-result matrix $\mathbf{R}$, where I represents the number of event classes for classification. A binary column vector of $\mathbf{R}$ per frame corresponds to the presence of each event class. When a column of $\mathbf{R}$ contains two non-zero elements for example, there are two detected events in that frame. A non-zero and a zero column vector stand for event-detected and event-undetected status, respectively.

Algorithm 3 Feature extraction with semi-supervised NMF

1: INPUT: $\mathbf{V}_{(*)}$ and $\mathbf{W}$
2: Initialize $\mathbf{W}_n$, $\mathbf{H}_{(*)}$ and $\mathbf{H}_n$ with random values.
3: **repeat**
4: Update $\mathbf{W}_n$ using (10)
5: Update $\mathbf{H}_{(*)}$ and $\mathbf{H}_n$ using (6) and (7).
6: **until** Convergence
7: OUTPUT: $\mathbf{H}_{(*)}$

3. ADDITIONAL PROCESSING SPECIFIC TO EVALUATION

DCASE 2016 task 2 Dataset is used for evaluating the proposed method. The Dataset includes 11 sound classes, which are typically found in the office and shown on the left side of Figure 4. The task 2 has two types of dataset; Training Dataset used for generating MLD and training SVM classifiers, and Development Dataset used for classification.

Training Dataset consists of 20 noise-free files for each sound class totaling 220 files. Development Dataset includes 18 files to cover 6 event occurrence patterns and three SNRs, namely, -6, 0, and 6 dB, each of which contains all 11 sound classes. Development Dataset also has an annotation file for each sound data file to evaluate classification result.

Because DCASE 2016 task 2 Dataset is used for evaluation with the annotation file and classification results, each classification result needs to be expressed in the format of the annotation file, which is defined as columns of sound class name, onset time, and offset time. The classification-result matrix $\mathbf{R}$ is applied a median filter in a row-wise manner. Columns of $\mathbf{R}$ are further replaced with zero column vectors when the corresponding frame is determined as silent by an integrated spectral intensity (ISI) or as a gap shorter than 0.1 second (10 frames). Values of F-measure are calculated by sed_eval tools [24] for evaluation on segment based metrics over 1 second for each SNR and each event.

4. EVALUATION AND DISCUSSION

Table 1 shows a parameter setting used in the evaluation. For generating spectrograms from sound files, a variable q transform (VQT) [25] was used. VQT spectrograms were extracted for all files of DCASE task 2 Dataset. MLD was generated from the obtained VQT spectrograms. The number of bases in the noise dictionary for semi-supervised NMF was set to the one with the best performance for each event class.

Figure 3 compares F-measure values calculated by the conventional AED with supervised NMF[15] and the proposed AED with semi-supervised NMF for different SNRs. The F-measures by the proposed method are 4.7%, 7.7%, and 11.1% higher than those by the conventional method at SNRs of 6, 0, and -6 dB, respectively. The degradation of F-measure from 6 to 0 dB is 2.0% and that from

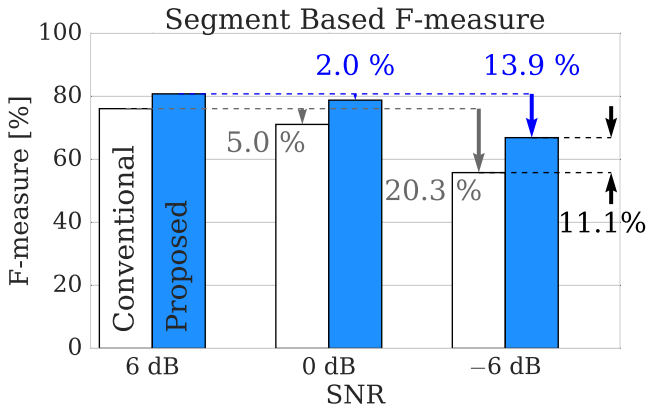


Figure 3: Evaluation results for all event classes at three different SNRs using DCASE2016 task 2 Dataset.

Table 1: Parameter setting for the evaluation.

Parameter	value	
Sampling rate	44.1	kHz
F_{min} for VQT	27.5	Hz
Number of bins per octave for VQT	60	
γ for VQT	30.0	
Number of basis for MLD	46	
Number of group basis for MLD	4	

6 to -6 dB is 13.9% for the proposed method. These values are smaller than those for the conventional method.

Conventionally, the input spectrogram including the noise is modeled by fixed MLD, which is learned without noise, and the activation matrix of MLD, so that the activation matrix of MLD includes errors. The proposed method dedicates both a noise dictionary and its activation matrix to the noise. Because noise spectra are better modeled by the noise dictionary learned upon classification and its activation matrix, the proposed method provides a higher F-measure values than conventional method at each SNR, when known acoustic atoms in the learning data are contaminated by noise in event classification. Therefore, the proposed method is robust to the noise.

Results for each event class are compared in Figure 4. It shows big improvement for cough and page turn. Especially, the F-measure of page turn is improved by 24.4%. The F-measure for clear throat, keyboard, keys, laughter, phone, and speech show small improvement. Door slam, drawer, and knock did not improve at all. The proposed method generally provides better results than the conventional method for each event class, because the conventional method is a special case of the proposed method with no noise dictionary and no noise activation matrix. The effect of the proposed method changes according to similarities between an event-class spectrum and an unknown noise spectrum. It seems that an event class with big improvement by the proposed method has MLD that can be easily activated by the noise spectrum. The proposed method reduces such erroneous activation with a help of the noise dictionary. In contrast, when the spectrum of an event class are clearly different from the noise spectrum, the proposed method is not as effective as for the similar spectrum case. Further investigation is left for future study.

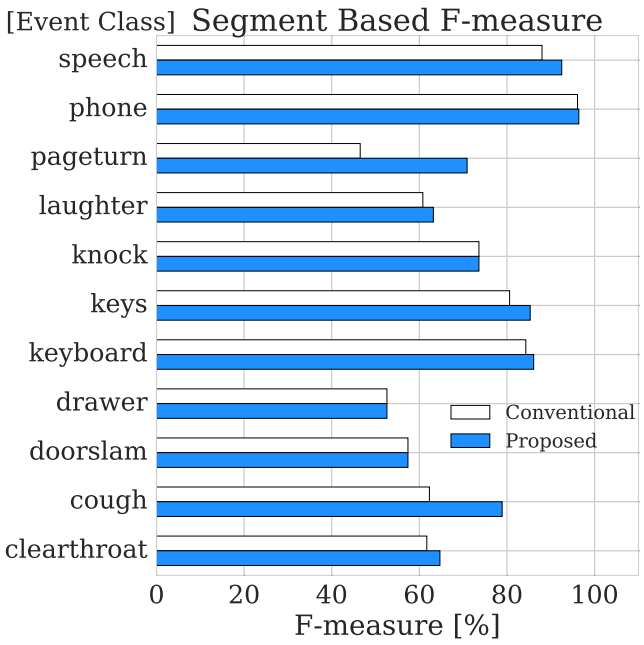


Figure 4: Evaluation results for each acoustic event of DCASE2016 task 2 Dataset.

Table 2: DCASE 2016 Challenge results.

	Error rate	F-measure
Development dataset	0.27	86.7 %
Evaluation dataset	0.33	80.2 %

5. DCASE 2016 CHALLENGE

Table 2 shows segment-based overall results of our sysmtem. To obtain better performance for DCASE 2016 Challenge, we additionally incorporate noise suppression [26] for test spectrogram $\mathbf{V}_{(*)}$ as preprocessing to suppress stationary noise in $\mathbf{V}_{(*)}$. Parameters of the proposed method are optimized for the development dataset. In particular, event classes have their respective optimal number of bases in the noise dictionary for semi-supervised NMF. Therefore, for detection of each event class, the corresponding optimal numbers of bases are used ; 1 for cough, doorslam, pageturn, and phone, 2 for clearthroat, 3 for drawer and keyboard, 4 for keys, and 10 for knock, laughter, and speech.

6. CONCLUSIONS

An acoustic event detection (AED) method using semi-supervised non-negative matrix factorization (NMF) with mixture of local dictionaries (MLD) has been proposed. The proposed method has newly introduced a noise dictionary and a noise activation matrix both dedicated to unknown acoustic atoms which are not included in the learning data. Because unknown acoustic atoms are better modeled by the new noise dictionary learned upon classification and the new activation matrix, the proposed method provides a higher classification capability for event classes modeled by MLD when a signal to be classified is contaminated by unknown acoustic atoms. Evaluation results using DCASE2016 task 2 Dataset have shown that F-measure by the proposed method with semi-supervised NMF has been improved by as much as 11.1% compared to that by the conventional method with supervised NMF.

7. REFERENCES

[1] D. Hollosi, J. Schröder, S. Goetze, and J. -E. Appell, "Voice activity detection driven acoustic event classification for monitoring in smart homes," in *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*. IEEE, 2010, pp. 1–5.

[2] J. Schröeder, S. Wabnik, P. W. Van Hengel, and S. Goetze, "Detection and classification of acoustic events for in-home care," in *Ambient Assisted Living*. Springer, 2011, pp. 181–195.

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

[4] G. Roma, J. Janer, S. Kersten, M. Schirosa, P. Herrera, and X. Serra, "Ecological acoustics perspective for content-based retrieval of environmental sounds," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, no. 1, p. 1, 2010.

[5] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, "Audio keywords generation for sports video analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 4, no. 2, p. 11, 2008.

[6] Y. Ohishi, D. Mochihashi, T. Matsui, M. Nakano, H. Kameoka, T. Izumitani, and K. Kashino, "Bayesian semi-supervised audio event transcription based on markov indian buffet process," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3163–3167.

[7] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Sparse representation based on a bag of spectral exemplars for acoustic event detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6255–6259.

[8] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "On acoustic surveillance of hazardous situations," in *Acoustics, Speech and Signal Processing (ICASSP), 2009 IEEE International Conference on*. IEEE, 2009, pp. 165–168.

[9] P. Laffitte, D. Sodoyer, C. Tatkeu, and L. Girin, "Deep neural networks for automatic detection of screams and shouted speech in subway trains," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6460–6464.

[10] C. V. Cotton and D. P. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE, 2011, pp. 69–72.

[11] A. Dessein, A. Cont, and G. Lemaitre, "Real-time detection of overlapping sound events with non-negative matrix factorization," in *Matrix Information Geometry*. Springer, 2013, pp. 341–371.

[12] O. Dikmen and A. Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[13] E. Benetos, G. Lafay, M. Lagrange, and M. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6450–6454.

[14] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, *et al.*, "An exemplar-based nmf approach to audio event detection," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[15] T. Komatsu, Y. Senda, and R. Kondo, "Acoustic event detection based on non-negative matrix factorization with mixtures of local dictionaries and activation aggregation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2259–2263.

[16] M. Kim and P. Smaragdis, "Mixtures of local dictionaries for unsupervised speech enhancement," *Signal Processing Letters, IEEE*, vol. 22, no. 3, pp. 293–297, 2015.

[17] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, 2013.

[18] P. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 414–421.

[19] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, 2001, pp. 556–562.

[20] A. Lefevre, F. Bach, and C. Févotte, "Itakura-saito nonnegative matrix factorization with group sparsity," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 21–24.

[21] R. -E. Fan, K. -W. Chang, C. -J. Hsieh, X. -R. Wang, and C. -J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[22] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Advances in Neural Information Processing Systems 16*, 2004, pp. 1141–1148.

[23] C. Bauckhage, "A purely geometric approach to non-negative matrix factorization," in *16th LWA Workshops: KDML, IR and FGWM*, 2014.

[24] DCASE 2016, "Detection and classification of acoustic scenes and events 2016," http://www.cs.tut.fi/sgn/arg/dcase2016/.

[25] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference, 53rd International Conference on*, 2014.

[26] M. Kato, A. Sugiyama and M. Serizawa, "Noise suppression with high speech quality based on weighted noise estimation and MMSE STSA," Proc. IWAENC2001, pp. 183-186, Sep. 2001.

DEEP NEURAL NETWORK BASELINE FOR DCASE CHALLENGE 2016

Qiuqiang Kong, Iwona Sobieraj, Wenwu Wang, Mark D. Plumbley

Centre for Vision, Speech and Signal Processing, University of Surrey, UK
{q.kong, iwona.sobieraj, w.wang, m.plumbley}@surrey.ac.uk

ABSTRACT

The DCASE Challenge 2016 contains tasks for Acoustic Scene Classification (ASC), Acoustic Event Detection (AED), and audio tagging. Since 2006, Deep Neural Networks (DNNs) have been widely applied to computer visions, speech recognition and natural language processing tasks. In this paper, we provide DNN baselines for the DCASE Challenge 2016. In Task 1 we obtained accuracy of 81.0% using Mel + DNN against 77.2% by using Mel Frequency Cepstral Coefficients (MFCCs) + Gaussian Mixture Model (GMM). In Task 2 we obtained F value of 12.6% using Mel + DNN against 37.0% by using Constant Q Transform (CQT) + Nonnegative Matrix Factorization (NMF). In Task 3 we obtained F value of 36.3% using Mel + DNN against 23.7% by using MFCCs + GMM. In Task 4 we obtained Equal Error Rate (ERR) of 18.9% using Mel + DNN against 20.9% by using MFCCs + GMM. Therefore the DNN improves the baseline in Task 1, 3, and 4, although it is worse than the baseline in Task 2. This indicates that DNNs can be successful in many of these tasks, but may not always perform bette than the baselines.

Index Terms— Deep Neural Network (DNN), Acoustic Scene Classification (ASC), Acoustic Event Detection (AED), Audio Tagging

1. INTRODUCTION

Sounds carry a large amount of information about our everyday environment. Humans can perceive the sound scene around them (busy street and office, etc.), and recognize individual sound events (car passing by and footsteps). Although image classification and detection have been popular in recent years, audio classification and detection have not attracted a similar level of attention. In the past years, CLEAR 2007 was a challenge on detecting events and activities [1]. The DCASE Challenge 2013 [2] contained challenge for scene classification and synthetic acoustic classification . The DCASE Challenge 2016[1] has four tasks in acoustic related problems. Task 1 is Acoustic Scene Classification (ASC), the goal of which is to classify a test recording into one of the predefined classes that characterize the environment in which it was recorded, for example "park", "home", "office". Task 2 is Acoustic Event Detection (AED) in Synthetic audio, which aims at detecting sound events in synthetic mixture (e.g. "door slam", "human speaking") that are present within an audio. Task 3 is Sound Event Detection in Real Life Audio. In contrast to Task 2, it aims to detect acoustic events in real life, such as "bird singing", "car passing by". Task 4 is Domestic Audio Tagging, the goal of which is to perform multi-label classification on short recordings collected in a domestic environment.

ASC and AED are intimately related to industry applications. They have applications in audio indexing [3], audio classification [4], audio tagging [5], audio segmentation [6], surveillance, military and public abnormal event detection [7], etc. In previous work, Mel Frequency Ceptral Coefficients (MFCCs) and Gaussian Mixture Model (GMM) were used for ASC [8]. McLoughlin *et al*. improved on this result by using auditory features and Deep Neural Network (DNN) classifier [9]. Unsupervised learning proposed by Lee *et al*. [4] uses convolutional deep belief networks to learn audio features. In AED, the Constant Q Transform (CQT) and Nonnegative Matrix Factorization (NMF) are widely used to detect sound events in a recording [10]. Hidden Markov Models (HMM) with Viterbi decoding have been proposed in [7], where a universal background model (UBM) is used to model background sound. In [11], a Bidirectional Long Short Term Memory (BLSTM) is proposed, which yields better result than the HMM. In audio tagging, MFCCs + GMM is a standard method to detect whether or not tags occur in the audio [12]. Recently Convolution Neural Networks (CNNs) have been used for audio tagging in [13].

This work aims at providing DNN baseline for all four tasks of the DCASE Challenge 2016. The remainder of the paper is organized as follows. Section 2 discusses related works. Section 3 describes the deep DNN structure. Section 4 presents experimental results we obtained on Task 1 - 4 of DCASE Challenge 2016. Section 5 draws conclusion of our work and future research.

2. DEEP NEURAL NETWORKS

DNNs have been widely used in Computer Vision (CV), Natural Language Processing (NLP), *etc*. since 2006. Their vari-

[1] http://www.cs.tut.fi/sgn/arg/dcase2016/

ants include CNNs and Recurrent Neural Networks (RNNs). In this paper, we propose to use the same features and the same structures of DNN for all of the four tasks in the DCASE Challenge 2016. This is aimed at evaluating how DNN performs in these tasks compared with original baseline methods, as well as providing a baseline for other researchers to compare.

2.1. Features

In audio processing, MFCCs are widely used in speech recognition. They are developed with the assumption that sounds are produced by glottal pulse passing through vocal tract filter. However, with MFCCs some useful information about the sound may be lost, which restricts its ability for recognition and classification. In recent years, Mel Filter Bank Features have been widely used in speaker recognition [14]. Other features such as Constant Q Transform (CQT) [15] are used in music related tasks, which has good resolution in low frequency. In this paper, we apply Mel-filter bank features with 40 channels to all of the four tasks. Features extraction code is based on *librosa*[2].

2.2. DNN structure

The DNN we used in our experiment is a fully connected neural network with 3 hidden layers. As the bag of frames feature cannot capture time dependency, the input to the DNN is taken as a concatenation of 10 frames mel-bank features so there are 400 input nodes (10 frames * 40 Mel-filter banks). We use 500 hidden units per layer. ReLU [16] activation function is used. For Task 1, softmax output and categorical cross-entropy loss function are used. For Tasks 2, 3, and 4, binary output and binary cross-entropy function are used. Dropout [17] with value of 0.1 is used to avoid overfitting. RMSProp [18] optimizer is used since it is generally faster than Stochastic Gradient Descend (SGD). The DNN structure is shown in Figure 1.

3. EXPERIMENTS

In this section we evaluate the performance of Mel-filter bank features plus DNN on DCASE Challenge 2016 Tasks 1 - 4 on ASC, AED and audio tagging. We use 40 Mel-filter bank features. Then we apply the DNN shown in Figure 1 to all of the four tasks. These systems are implemented in python. The source code can be found in Task 1[3], Task 2[4], Task 3[5], Task 4[6]. Our DNN implementation is based on *Hat*[7], which

[2]https://github.com/librosa
[3]https://github.com/qiuqiangkong/DCASE2016_Task1
[4]https://github.com/qiuqiangkong/DCASE2016_Task2
[5]https://github.com/qiuqiangkong/DCASE2016_Task3
[6]https://github.com/qiuqiangkong/DCASE2016_Task4
[7]https://github.com/qiuqiangkong/Hat

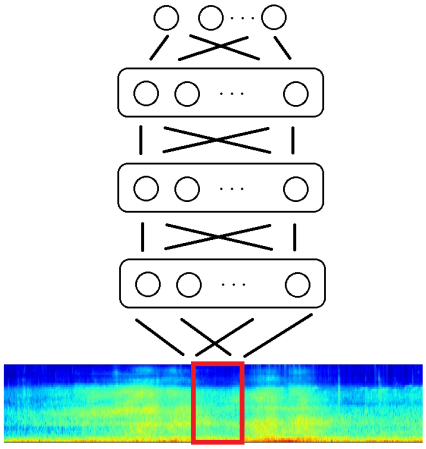


Figure 1: DNN used for Task 1 - 4

is an open source deep learning framework built on top of *Theano*[8].

3.1. Task 1: Acoustic Scene Classification

The TUT Acoustic scenes 2016 datasets [19] is used in this task. The dataset consists of recordings from various acoustic scenes, all having distinct recording locations. Each recording contains 30-second segments. There are altogether 15 classes with 4 fold cross validation. For training DNN, the batch size is set to 100. RMSProp (Section 3.2) learning rate is set to 10^{-3} at beginning then is tuned to 10^{-4} after 30 epochs. The maximum number of epochs is set to 100. Time consumption is 3 s/epoch on Tesla 2090. The results are shown in Table 1. NG is the abbreviation of Not Given. Dev., Test means development dataset and private dataset, respectively.

Table 1: Accuracy of Task 1

	Chunk based acc. (Dev.)	Segment based acc. (Dev.)	Segment based acc. (Test)
MFCCs + GMM (Baseline)	NG	72.5%	77.2%
Mel + DNN	**63.3%**	**76.4%**	**81.0%**

From this table, it can be observed that using the Mel + DNN obtains an accuracy of 81.0%, outperforms MFCCs + GMM baseline (77.2%) in test dataset. Detailed results of development set on each fold are shown in Table 2.

[8]http://deeplearning.net/software/theano/

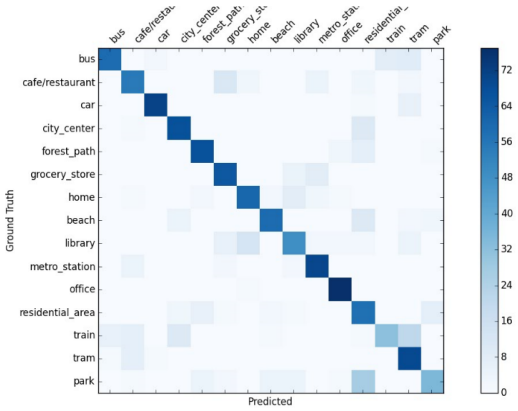


Figure 2: Confusion matrix of segment based accuracy in Task 1.

Table 2: Fold wise accuracy of Task 1 using Mel + DNN

	Frame based acc.	Segment based acc.
fold 1	65.2%	80.0%
fold 2	61.5%	70.7%
fold 3	62.0%	74.8%
fold 4	64.6%	80.1%
average	**63.3%**	**76.4%**

Table 2 shows that the accuracy of different folds (development dataset) are different, with frame based accuracy ranging from 61.5% to 65.2% and segment based accuracy ranging from 70.7% to 80.1%. This indicates the dataset is not homogeneous. The overall confusion matrix is shown in Figure 2. We can see that "park" is easily mis-recognized as "residential area". This may result from the fact these scenes share similar features, which are difficult to classify using the bag of words model.

3.2. Task 2: AED in Synthetic Audio

A dataset provided by IRCCYN Ecole Centrale de Nantes is used in Task 2 [19]. The training set includes 11 classes of sound events. There are 20 samples provided for each sound event class in the training set, plus a development set consisting of 18 minutes of synthetic mixture material in 2 minute length audio files. The event-to-background ratio (EBR)[9] is set to -6, 0, +6 dB. In this task, we set the RMSProp learning rate to 10^{-3}, the batch size to 20, the number of epochs to 20, respectively. Binary output and sigmoid cost function are used. Time consumption in Tesla 2090 GPU is 0.1 s/epoch (one processor). Results are shown in Table 3.

[9]http://www.cs.tut.fi/sgn/arg/dcase2016/task-sound-event-detection-in-synthetic-audio

Table 3: F value of Task 2

EBR	F value (Dev.)	F value (Test)
CQT + NMF (Baseline)	**41.6%**	**37.0%**
Mel + DNN	17.4%	12.6%

Table 3 shows that using Mel + DNN yields an F value of 12.6% which is worse than CQT + NMF baseline (37.0%). One possible explanation for this underperformance is that without data augmentation, DNN is not good at classifying the samples with additive noise, while the NMF based technique has better ability in modeling sounds with additive noise. Detailed results on development dataset on different EBR levels of -6, 0, +6 dB are shown in Table 4.

Table 4: Fold wise F value of Task 2 using Mel + DNN

	F value
-6 dB	16.0%
0 dB	17.6%
+6 dB	18.8%
Average	**17.4%**

3.3. Task 3: AED in Real Life Audio

The TUT Sound events 2016 dataset [19] is used in this task. Audio in the dataset is a subset of TUT Acoustic scenes 2016 dataset (used for task 1). The TUT Sound events 2016 dataset consists of recordings from two acoustic scenes: Home (indoor) and Residential area (outdoor). In this task, we set the RMSProp learning rate to 10^{-3}, the batch size to 20, the number of epochs to 50. Results are shown in Table 5.

Table 5: F value of Task 3

	Home (Dev.)	Residential area (Dev.)	Average (Dev.)	Average (Test)
MFCCs + GMM (baseline)	15.9%	31.5%	23.7%	34.3%
Mel + DNN	**29.2%**	**47.0%**	**38.1%**	**36.3%**

Table 5 shows that for real life event detection using Mel + DNN yields an F value of 36.3%, which outperforms MFCCs + GMM baseline (23.7%). Detailed results on development dataset on each fold are shown in Table 6.

Table 6: Fold wise F value of Task 3 using Mel + DNN

	Home	Residential area
fold 1	28.0%	62.4%
fold 2	28.8%	34.5%
fold 3	22.3%	43.7%
fold 4	37.5%	47.5%
average	**29.2%**	**47.0%**

3.4. Task 4: Domestic audio tagging

The CHiMe-Home dataset is used in Task 4 . The objective of this task is to perform multi-label classification on 4-second audio chunks. There are 7 labels occurring in audio segments including child speech and adult male, *etc*. Binary output and binary cross-entropy loss function are used because the labels can occur simultaneously. We set the RMSProp learning rate to 10^{-3}, the batch size to 500, the number of epoch to 100. Cross validation with 4 folds is used. Results are shown in Table 7.

Table 7: F value of Task 4

	EER (Dev.)	(Test)
MFCCs + GMM (baseline)	**21.0%**	20.9%
Mel + DNN	20.9%	**18.9%**

Table 7 shows that we obtain Equal Error Rate (ERR) of 18.9% using Mel + DNN, which is similar to MFCCs + GMM baseline (20.9%). Detailed results on development dataset on four folds are shown in Table 8.

Table 8: Fold wise EER of Task 4 using Mel + DNN

	EER
fold 1	19.3%
fold 2	15.6%
fold 3	26.3%
fold 4	22.4%
average	**20.9%**

4. CONCLUSION

In this paper, we have applied the same DNN structure to Task 1 - 4 in the DCASE Challenge 2016 as a DNN baseline for future research. In summary, in Task 1, Mel + DNN is better than MFCCs + GMM (accuracy of 81.0% against 77.2%). In task 2, Mel + DNN is worse than the CQT + NMF baseline (F value 12.6% against 37.0%). In task 3, Mel + DNN is better than the MFCCs + GMM baseline (F value 36.3% against 23.7%). In task 4, Mel + DNN is better than MFCCs + DNN baseline (18.9% against 20.9%). We publish our codes of Task 1 - 4 and hope this will attract interests from other institutions to do further research.

5. ACKNOWLEDGMENT

This work is sponsored by "Making Sense of Sounds", CVSSP, University of Surrey and China Scholarship Council (CSC), European Union's H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement n 642685 MacSeNet. MP is partly supported by EPSRC grante EP/N014111/1.

6. REFERENCES

[1] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, R Travis Rose, Martial Michel, and John Garofolo. The CLEAR 2007 Evaluation, Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007, Baltimore, MD, USA, May 8-11, 2007, Revised Selected Papers, 2008.

[2] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, 2015.

[3] Rui Cai, Lie Lu, Alan Hanjalic, Hong-Jiang Zhang, and Lian-Hong Cai. A flexible framework for key audio effects detection and auditory context inference. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):1026–1039, 2006.

[4] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1096–1104, 2009.

[5] Mohit Shah, Brian Mears, Chaitali Chakrabarti, and Andreas Spanias. Lifelogging: Archival and retrieval of continuously recorded audio using wearable devices. In *IEEE International Conference on Emerging Signal Processing Applications (ESPA), 2012*, pages 99–102. IEEE, 2012.

[6] Gordon Wichern, Jiachen Xue, Harvey Thornburg, Brandon Mechtley, and Andreas Spanias. Segmentation, indexing, and retrieval for environmental and natural sounds. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):688–707, 2010.

[7] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. Context-dependent sound event detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013(1):1–13, 2013.

[8] Burak Uzkent, Buket D Barkana, and Hakan Cevikalp. Non-speech environmental sound classification using svms with a new set of features. *International Journal of Innovative Computing, Information and Control*, 8(5):3511–3524, 2012.

[9] Ian McLoughlin, Haomin Zhang, Zhipeng Xie, Yan Song, and Wei Xiao. Robust sound event classification using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):540–552, 2015.

[10] Arnaud Dessein, Arshia Cont, and Guillaume Lemaitre. Real-time detection of overlapping sound events with non-negative matrix factorization. In *Matrix Information Geometry*, pages 341–371. Springer, 2013.

[11] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Recurrent neural networks for polyphonic sound event detection in real life recordings. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6440–6444. IEEE, 2016.

[12] Peter Foster, Siddharth Sigtia, Sacha Krstulovic, Jon Barker, and Mark D Plumbley. CHiME-home: A dataset for sound source recognition in a domestic environment. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5. IEEE, 2015.

[13] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

[14] Li Deng, Ossama Abdel-Hamid, and Dong Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6669–6673. IEEE, 2013.

[15] Judith C Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant Q transform. *The Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

[16] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[17] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[18] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[19] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Tut database for acoustic scene classification and sound event detection. In *24th European Signal Processing Conference*, 2016.

BAG-OF-FEATURES ACOUSTIC EVENT DETECTION FOR SENSOR NETWORKS

Julian Kürby, Rene Grzeszick, Axel Plinge, and Gernot A. Fink

TU Dortmund University, Dortmund, Germany

ABSTRACT

In this paper a novel approach for acoustic event detection in sensor networks is presented. Improved and more robust recognition is achieved by making use of the signals from multiple sensors. To this end, various known fusion strategies are evaluated along with a novel method using classifier stacking. A comparative evaluation of these fusion strategies is performed on two different datasets: the ITC-Irst database, and a set of smart room recordings. In both datasets, 32 distributed microphones were used for recording. Furthermore, the effect of previously observed as well as unobserved locations is investigated. The proposed stacking yields a notable improvement. The performance of recognizing events at previously unobserved locations can be improved by sorting the channels according to their posterior probabilities.

Index Terms— Bag-of-Features, Acoustic Event Detection, Sensor Arrays, Robustness, Acoustic Sensor Networks

1. INTRODUCTION

The detection and classification of acoustic events is important for many practical applications in various environments: The recognition of such events can be used for meeting and online lecture analysis and annotation [1]. Surveillance in cluttered scenes can be improved by an acoustic analysis in order to detect unexpected scenarios that are not easily visually recognizable (e. g. screams or glass breaking) [2]. In a slightly different field of research outdoor applications are addressed. These include mobile robots for security [3], urban planning [4], and the analysis of possible noise complaints [5]. It can also be used to improve the robustness of different real world applications, such as speech enhancement, speaker tracking, or the calibration of microphone arrays [6–8]. What makes this problem difficult is the vast diversity of the acoustic events.

Methods for online analysis of acoustic events are typically applied over short time windows and combined with a sliding window approach. One common approach stems from speaker identification [9]. A Gaussian mixture model (GMM) is trained for each class. The estimates of all GMMs are summed up over all frames and the class with the highest likelihood is chosen. These methods are sometimes termed 'Bag-of-Frames' [10,11]. Over the last years, methods that build on the Bag-of-Features (BoF) principle have emerged in the field of acoustic event detection [12,13]. There, features are clustered in order to obtain a histogram representation which is then classified. The BoF principle has been proven to generalize well with respect to the diversity of the acoustic events.

Many methods in acoustic event detection focus on a single signal. However, in many scenarios a sensor network with multiple microphones is available (cf. [8,14]). In [15] multiple channels are used to extract features describing spatial information. These features work well for classifying scenes where the sound sources occur at distinct locations. In [16] a multi-channel approach that uses Regression Forests is proposed. The confidence scores of different channels are accumulated and then the presence is predicted using a pre-defined threshold. In [17] different combination strategies including accumulation of log probabilities, the maximum rule, and majority voting are evaluated. Both works show that the combination of information obtained from multiple channels improves the robustness of the system and the detection results.

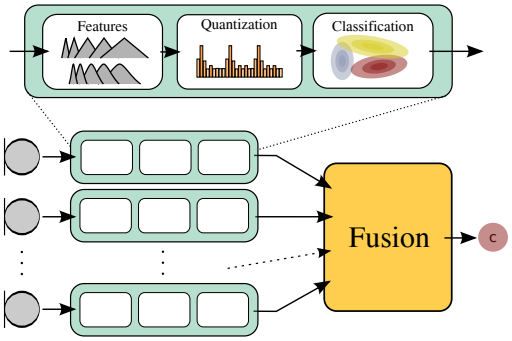


Figure 1: Overview of the proposed method. A three-step BoF approach for acoustic event detection is applied to every source in a sensor network comprised of many microphones, before the results are combined by a fourth fusion step.

This paper extends the BoF approach discussed in [13,18] and provides a thorough evaluation of different multi-channel fusion strategies in the context of acoustic event detection. The heuristic combination strategies presented in [17] are compared with a novel method based on classifier stacking. A comparative evaluation on two different datasets is given. Furthermore, different training and test setups are evaluated having a closer look at the prerequisites necessary for successfully exploiting information from multiple sources.

2. METHOD

For the acoustic event detection in sensor networks, a single channel Bag-of-Features (BoF) approach is extended to multiple channels by adding an additional fusion step that combines the information from different microphones. A sliding window approach is used for detection. For each window, four basic processing steps are applied, as shown in Fig. 1:

1. Given an input signal and a short time window, a set of feature vectors is calculated for all frames in this window.
2. The feature vectors of all frames in the training set are clustered in a supervised manner using a GMM for each class. The features within one window are assigned to the clusters using soft assignment. These are accumulated in a histogram, the BoF representation.

3. These representations are then used for classification, applying maximum likelihood classification.
4. The results from multiple channels are fused in order to get a more robust classification. A novel fusion strategy based on classifier stacking is proposed.

2.1. Single-channel BoF acoustic event classification

For the single-channel BoF based acoustic event classification, a single microphone or beamformed signal is processed in short time windows. The processing steps are explained in more detail in the following.

Features Given an input signal and a time window n of w milliseconds, a set of feature vectors $Y_n = (y_1 \dots y_K)$ is calculated. For sound and especially speech processing, the mel frequency cepstral coefficients (MFCCs) are one of the most widely used features. The input signal is filtered by a mel frequency filter bank, from the logarithm of its magnitude the discrete cosine transform (DCT) is computed and its second to 13^{th} coefficient is used. From that the gammatone frequency cepstral coefficients (GFCCs) were derived in [19]. Here, the filterbank of the MFCCs is replaced by linear phase gammatone filters. As for the MFCCs, the second to 13^{th} GFCC coefficients are used. In addition, a single loudness filter is evaluated. In total the feature vector has a dimensionality of 27. A whitening transformation is computed on the training data which is applied to all feature vectors.

Feature Representation A BoF approach is used for building a codebook of *acoustic words* from the training set. While the classical BoF uses hard quantization via the k-Means algorithm, soft quantization by GMMs has been shown to improve the performance [13,20]. The basic principle also employs a globally estimated codebook which can lead to mitigation of significant differences. A remedy for this effect is to build codebooks of size I for all C classes Ω_c separately and then concatenating them into a large super-codebook [13]. Here, the expectation maximization (EM) algorithm is applied to all feature vectors $\boldsymbol{y}_k$ for each class Ω_c in order to estimate I means and standard deviations $\mu_{i,c}, \sigma_{i,c}$ for all C classes. All means and deviations are concatenated into a super-codebook $\boldsymbol{v}$ with $V = I \cdot C$ elements

$$v_{j=(I \cdot c+i)} = (\mu_{i,c}, \sigma_{i,c}) \tag{1}$$

where the index j is computed from the class index c and the Gaussian index i as $j = I \cdot c + i$. Using this codebook, a soft quantization of a feature vector $\boldsymbol{y}_k$ can be computed as

$$q(\boldsymbol{y}_k, v_j) = \mathcal{N}(\boldsymbol{y}_k | \mu_j, \sigma_j) \; / \; \sum_{j'} \mathcal{N}(\boldsymbol{y}_k | \mu_{j'}, \sigma_{j'}) \; . \tag{2}$$

Then, a histogram $\boldsymbol{b}$ can be computed over all K frames of the input window by

$$b(Y_n, v_j) = \frac{1}{K} \sum_k q(\boldsymbol{y}_k, v_j) \; . \tag{3}$$

Classification The probability $P(v_j|\Omega_c)$ of an acoustic word v_j given class Ω_c is estimated using a set of training samples $Y_n \in \Omega_c$ for each class c by Lidstone smoothing:

$$P(v_j|\Omega_c) = \frac{\alpha + \sum_{Y_n \in \Omega_c} b(Y_n, v_j)}{\alpha V + \sum_{m=1}^{V} \sum_{Y_n \in \Omega_c} b(Y_n, v_m)} \tag{4}$$

A typical choice for the smoothing factor α is in the range of $[0, 1]$. Here, α is set to 0.5. Since all classes are assumed to be equally likely and have the same prior, maximum likelihood classification is used. The posterior is estimated using the relative frequency of all acoustic words

$$P(Y_n|\Omega_c) = \prod_{v_j \in \boldsymbol{v}} P(v_j|\Omega_c)^{b(Y_n, v_j)} \; . \tag{5}$$

2.2. Multi-channel fusion

In a sensor network containing M microphones the approach can be evaluated for each microphone m individually. It is assumed, that all microphones are synchronized at least at a frame level. The results can then be combined in order to obtain a more robust classification. In the following three traditional heuristic fusion strategies (cf. [17]) will be reviewed and a novel approach based on classifier stacking will be introduced.

Majority voting A straightforward fusion approach is evaluating each channel separately so that a set of class labels

$$\hat{c}_{(m)} = \underset{c}{\operatorname{argmax}} \; P_m(Y_n|\Omega_c) \tag{6}$$

is estimated. Then, a majority voting over all decisions $\hat{c}_{(m)}$ is performed. This assumes that most microphones are able to detect the correct event. However, it discards the posterior probabilities which might carry important information about the confidence of the single channels.

Maximum rule The maximum rule is a fusion strategy that considers the posterior probabilities of each channel instead of the labels. It chooses the class with the overall highest posterior probability. For each class the maximum over all channels is computed and then the class with the highest probability in the complete sensor network is chosen:

$$\hat{c} = \underset{c}{\operatorname{argmax}} \; \max_m \; P_m(Y_n|\Omega_c) \; . \tag{7}$$

This approach can be highly influenced by positive outliers. It is assumed that at least one microphone is positioned well with respect to the acoustic event.

Product rule Alternatively the product of the posterior probabilities is used. For each of the classes the product of the posteriors of all channels is computed. Then, the class with the highest probability product in the complete sensor network is chosen:

$$\hat{c} = \underset{c}{\operatorname{argmax}} \; \prod_m \; P_m(Y_n|\Omega_c) \; . \tag{8}$$

In contrast to taking the highest probability this strategy is strongly influenced by negative outliers.

Classifier stacking While the previous approaches are mere heuristic approaches that decide on a fusion strategy, it is also possible to learn a combination strategy from the training data. A second classifier is trained that uses the posterior probabilities from all microphones in the sensor network as input features. The learned classification function $\mathcal{F}$ is then used for predicting the class:

$$\hat{c} = \mathcal{F}\left((P_m(Y_n|\Omega_c))_{(c,m)} \right) \; . \tag{9}$$

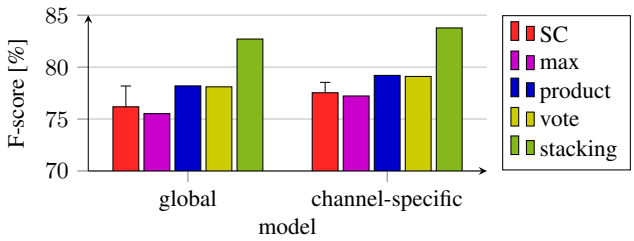


Figure 2: Frame-wise F-score [%] comparing fusion strategies with the single-channel (SC) baseline on the ITC-Irst dataset. For the single-channel results the mean and standard deviation are plotted.

Two thirds of the training data are used for training the single-channel BoF models and the single-channel classifiers and the last third is used in order to train a Random Forest classifier on the posterior probabilities of the single-channel evaluations.

Note that the classifier learns the probabilities based on their ordering. Therefore, it implicitly learns the position of the microphones and also the locations at which the different acoustic events occur. This can be an advantage for events or especially noise sources with a fixed location (e. g. doors or windows). However, it can be a limitation for events that can occur at arbitrary locations such as speech. A remedy for this effect is ordering the M channels according to the highest posterior probability. Sorting the channels descending by probability provides a new ordering:

$$\mathcal{M} = \left[\underset{m}{\operatorname{argsort}} \max_{c} \; P_m(Y_n|\Omega_c) \right] \tag{10}$$

After re-ordering of the channels, the posterior probabilities are again used as input for the classifier $\mathcal{F}$. Thus, the indices (c, m) in eq. 9 are replaced by $(c, \mathcal{M}_m)$.

2.3. Detection

Due to its simplicity and rapid computation, the BoF approach can easily be adapted to event detection, where a sequence of acoustic events is given. It currently runs in approx. 20% real time on a single core i7 cpu. The classification window is moved forward in a sliding window approach by one frame k at a time. The recognition result is used for the frame that is centered in the window so that context information is available for a short time before and after the frame. As the window has a length of w milliseconds, there is a processing delay of only $w/2$ milliseconds.

3. EVALUATION

The experiments are conducted on two different datasets for acoustic event detection, the ITC-Irst dataset [14] as well as a set of recordings conducted in a smart conference room at TU Dortmund University. On these datasets the detection performance of the presented multi-channel approaches are evaluated and compared to a single-channel baseline. All channels were synchronized with a global clock in both datasets.

3.1. ITC-Irst Dataset

The ITC-Irst dataset is comprised of 16 different acoustic events, including *door knock, door slam, steps, chair moving, spoon (cup jingle), paper wrapping, key jingle, keyboard typing, phone ring, applause, cough, laugh, door open, phone vibration, mimo pen buzz, falling object,* and *unknown/background*. The recording room was equipped with 32 microphones, 28 of which were located in seven T-shaped arrays on the walls and four were table microphones. The experiments consist of twelve recording session on three different days. The first three sessions of each day are considered as training and the fourth session is used for testing.

The first experiments were conducted using all sounds except silence and unknown as classes of interest. Then, in order to allow for comparability with existing experiments [14,16], only the first twelve classes were considered as foreground and the remaining ones as background.

evaluation	method	channels	error	F-score
event-based	RF [16]	mean (4)	15.4%	91.8%
	RF [16]	fusion (4)	**13.0%**	**93.3%**
	HMM2 [14]	SC (1)	23.6%	-
	HMM1 [14]	SC (1)	45.2%	-
	SVM [14]	SC (1)	64.4%	-
frame-based	RF [16]	fusion (4)	30.7%	82.8%
	proposed	mean (32)	39.0%	77.4%
	proposed	stacking (32)	**25.6%**	**84.2%**

Table 1: Results on the ITC-Irst dataset using the CLEAR evaluation protocol with the first 12 classes as foreground in comparison to literature results. The methods use either a single channel or different fusion approaches (number of channels in parentheses).

Baseline For the evaluation, two different setups were considered. First, the BoF model is trained on the events of all microphones yielding a global model. Second, a separate model is trained for each microphone in the sensor network. In both cases, the BoF model is computed using a codebook size of $I = 30$ centroids for each class and a window size of $w = 600$ ms based on the results in [18]. For the baseline each channel is evaluated separately and the average over all microphones is reported (single channel is denoted as SC).

Fusion experiments In the following the multi-channel fusion strategies are compared with each other and to the baseline of single-channel results. The first two sessions of each day are used for training the base classifier, the third session for training the stacking classifier. Since the positions of the acoustic events were changed for each of the three recording days, the stacking classifier is able to learn different acoustic locations. The fourth session is used for testing so that it contains the different locations from all three days. Note that the single-channel and heuristic approaches are trained on the complete training set. The frame-wise F-scores are shown in Fig. 2. The models that are trained for every channel separately perform much better than a single global model. Furthermore, it can be seen that the classifier stacking that learns a fusion strategy from the training data outperforms the heuristic approaches.

Literature comparison For comparison with the literature, only the first twelve classes are used as foreground (cf. [14,16]). Regression Forest (RF) were evaluated in combination with a multi-channel fusion approach using this setup [16]. Note that only four channels were used for evaluating the RF while the proposed approach is able to incorporate all 32 microphones. In contrast to [16], a frame-based evaluation protocol is used in this paper. The

set	model	SC	max	prod.	vote	stacking	sorted (32)	sorted (5)
separate	global	$10.7 \pm 4.2\%$	$8.5 \pm 3.5\%$	$9.3 \pm 3.3\%$	$9.5 \pm 3.2\%$	$7.6 \pm 3.0\%$	$7.2 \pm 2.3\%$	$\mathbf{7.1 \pm 2.4\%}$
	channel-wise	$12.2 \pm 4.0\%$	$9.2 \pm 3.8\%$	$9.4 \pm 3.3\%$	$9.8 \pm 3.1\%$	$10.0 \pm 2.9\%$	$9.5 \pm 2.7\%$	$\mathbf{8.4 \pm 2.7\%}$
mixed	global	$7.6 \pm 3.2\%$	$5.1 \pm 1.3\%$	$6.2 \pm 1.7\%$	$6.5 \pm 1.7\%$	$\mathbf{3.2 \pm 0.9\%}$	$3.4 \pm 1.0\%$	$3.8 \pm 1.1\%$
	channel-wise	$6.3 \pm 2.6\%$	$4.6 \pm 1.6\%$	$5.3 \pm 1.8\%$	$5.5 \pm 1.7\%$	$\mathbf{2.7 \pm 1.0\%}$	$2.7 \pm 0.9\%$	$2.8 \pm 0.8\%$

Table 2: Mean frame-wise classification error and its standard deviation over five splits of the position experiments. In "separate", the events of the training and test set occur on different sides of the smart room, in "mixed" on both.

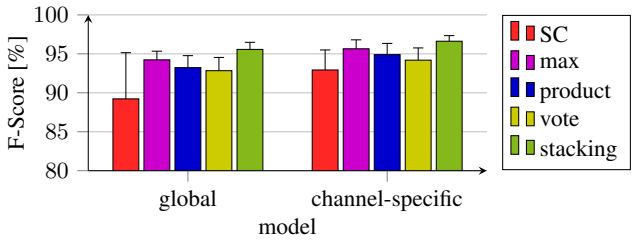


Figure 3: Mean F-score [%] and the standard deviation over the five splits of smart room recordings using different fusion strategies on the dataset.

Acoustic Frame Error Rate (AFER) is calculated analogously to the Acoustic Event Error Rate (AEER) [14], but with respect to the frames[1]. The frame-wise results of [16] are calculated on the resulting sequences, that were kindly provided by the authors. The results are shown in Tab. 1, reporting F-Scores, AEER (event-based error) and AFER (frame-based error). Additionally the event-based results of the RF and the CLEAR evaluation [14] are shown in Tab. 1. The CLEAR evaluation compared two different HMM and an SVM approach for acoustic event detection on this setup using a single microphone. The event-based results show that RFs achieve state of the art results.

On a frame level the proposed approach yields similar performance to the RF. The proposed classifier stacking shows the best results with 25.6% AFER and 84.2% F-score. The performance which is obtained using the stacking improves the results by a margin compared to the single-channel performance.

3.2. Smart room recordings

An additional set of acoustic events has been recorded in a smart room at TU Dortmund University.[2] The room is equipped with 32 microphones of which 16 are located at the table and the remaining 16 are mounted at the ceiling. The acoustic events were located at multiple positions in the room without any overlap. There is a lot of structure-borne noise changing the characteristics of sounds based on the microphones location. 19 sound categories have been recorded: *applause, chairs, cups, door, doorbell, doorknock, keyboard, knock, music, paper, phonering, phonevibration, pouring, screen, speech, steps, streetnoise, touching, ventilator,* and *silence*. Following the approach proposed in [5], acoustic events that are longer than five seconds were split into blocks of up to four seconds.

[1] A similar evaluation has been proposed for the DCASE2016 challenge referred to as a segment based metric.

[2] The dataset is publicly available as *Multi-channel acoustic event dataset* at http://patrec.cs.tu-dortmund.de/cms/en/home/Resources/

General experiments For generating different training and test sets five random splits were performed. Each split randomly selects two thirds of the data from each class for training and the remaining third for testing. For the stacking experiments the training data is randomly divided in two thirds for training the single-channel models and classifiers and the other third for training the stacking classifier. The sliding window is evaluated within the annotated four second blocks. All classes are considered as foreground events, using *silence* as background. The results are shown in Fig. 3. As for the ITC-Irst experiments, the stacking approach outperforms the heuristic fusion strategies. The best results are obtained using channel-specific models and classifier stacking which yields a frame-wise F-Score of $96.6 \pm 0.7\%$. This is an improvement of 3.7% compared to the mean results of the single-channel evaluation.

Position experiments The dataset contains a set of nine classes occurring on multiple positions (*applause, door, doorknock, keyboard, music, phonering, phonevibration, speech,* and *ventilator*). Here, the data has been recorded on different sides of the room (left & right respectively). Again five splits have been computed. Each split randomly selects the data of an acoustic event from the left or right side for training and the other side for testing, and vice versa. Hence, the robustness of the stacking classifier toward location changes can be investigated. The classification results are reported as the frame-wise classification error in Tab. 2. The classification error is used, because in this experiment all classes are considered as foreground. For comparison, five mixed sets using data from both sides for testing and training are also shown. As expected, the proposed stacking approach works well if a diverse set of training samples is provided. However, there is a drop in the performance when the locations in the test differ from the training set. This limitation can be overcome by sorting the input for the stacking classifier. In Tab. 2 the results for all 32 and the first 5 microphones of the sorted set $\mathcal{M}$ (denoted as sorted (32) and sorted (5) respectively) are shown (see eq. 10). Interestingly, the global model seems more robust toward reducing the information covered by the training set. This is probably due to the fact that multiple event locations and all microphones at different postions are used for training the model.

4. CONCLUSION

In this paper a multi-channel approach for acoustic event detection in sensor networks that builds on the Bag-of-Features principle has been presented. It was shown that combining the information from different channels allows for improving the performance of the recognition system. A novel fusion strategy that uses classifier stacking has been introduced which yields state-of-the-art results. Sorting the ordering of the microphones according to the posterior probability can overcome the requirement of having all locations in the training set.

5. REFERENCES

[1] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang, "Automatic analysis of multimodal group actions in meetings." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 305–17, Mar. 2005.

[2] V. Carletti, P. Foggia, G. Percannella, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance using a bag of aural words classifier," in *IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, Aug. 2013, pp. 81–86.

[3] S. H. Young and M. V. Scanlon, "Robotic vehicle uses acoustic array for detection and localization in urban environments," *SPIE Proc. Mobile Robot Perception*, vol. 4364, pp. 264–273, Sept. 2001.

[4] D. Steele, J. D. Krijnders, and C. Guastavino, "The sensor city initiative: Cognitive sensors for soundscape transformations," GIS Ostrava, 2013.

[5] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *ACM Int. Conf. on Multimedia*, 2014.

[6] A. Plinge and S. Gannot, "Multi-microphone speech enhancement informed by auditory scene analysis," in *Sensor Array and Multichannel Signal Process. Workshop*, Rio de Janeiro, Brazil, 2016.

[7] A. Plinge and G. A. Fink, "Multi-Speaker tracking using multiple distributed microphone arrays," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proces.*, May 2014.

[8] A. Plinge and G. A. Fink, "Geometry calibration of multiple microphone arrays in highly reverberant environments," in *Int. Workshop on Acoustic Signal Enhancement*, Sept. 2014.

[9] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23–61, 2011.

[10] 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, no. 2, pp. 881–891, 2007.

[11] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *IEEE Workshop on Applications of Signal Process. to Audio and Acoustics*, 2013.

[12] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Interspeech*, 2012.

[13] A. Plinge, R. Grzeszick, and G. Fink, "A Bag-of-Features approach to acoustic event detection," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proces.*, May 2014.

[14] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," in *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science, R. Stiefelhagen and J. Garofolo, Eds. Springer Berlin Heidelberg, 2007, vol. 4122, pp. 311–322.

[15] K. Imoto and N. Ono, "Spatial-feature-based acoustic scene analysis using distributed microphone array," in *European Signal Process. Conf.* IEEE, 2015, pp. 734–738.

[16] H. Phan, M. Maass, L. Hertel, R. Mazur, and A. Mertins, "A multi-channel fusion framework for audio event detection," in *IEEE Workshop on Applications of Signal Process. to Audio and Acoustics*, 2015.

[17] P. Giannoulis, G. Potamianos, A. Katsamanis, and P. Maragos, "Multi-microphone fusion for detection of speech and acoustic events in smart spaces," in *European Signal Process. Conf.* IEEE, 2014, pp. 2375–2379.

[18] R. Grzeszick, A. Plinge, and G. A. Fink, "Temporal acoustic words for online acoustic event detection," in *German Conf. on Pattern Recognition*, 2015.

[19] Y. Shao, S. Srinivasan, and D. Wang, "Incorporating auditory feature uncertainties in robust speaker identification," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proces.*, 2007, pp. 277–280.

[20] S. Pancoast and M. Akbacak, "Softening Quatization in Bag-of-Audio-Words," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Proces.*, 2014, pp. 1384–1388.

CQT-BASED CONVOLUTIONAL NEURAL NETWORKS FOR AUDIO SCENE CLASSIFICATION

Thomas Lidy

Vienna University of Technology
Institute of Software Technology
Vienna, Austria
lidy@ifs.tuwien.ac.at

Alexander Schindler

Austrian Institute of Technology
Digital Safety and Security
Vienna, Austria
alexander.schindler@ait.ac.at

ABSTRACT

In this paper, we propose a parallel Convolutional Neural Network architecture for the task of classifying acoustic scenes and urban sound scapes. A popular choice for input to a Convolutional Neural Network in audio classification problems are Mel-transformed spectrograms. We, however, show in this paper that a Constant-Q-transformed input improves results. Furthermore, we evaluated critical parameters such as the number of necessary bands and filter sizes in a Convolutional Neural Network. These are non-trivial in audio tasks due to the different semantics of the two axes of the input data: time vs. frequency. Finally, we propose a parallel (graph-based) neural network architecture which captures relevant audio characteristics both in time and in frequency. Our approach shows a 10.7 % relative improvement of the baseline system of the DCASE 2016 Acoustic Scenes Classification task [1].

Index Terms— Deep Learning, Constant-Q-Transform, Convolutional Neural Networks, Audio Event Classification

1. INTRODUCTION

Recent advances with Deep Learning approaches in image retrieval have fueled the interest as well in audio-based tasks such as speech recognition and music information retrieval. A particular sub-task in the audio domain is the detection and classification of acoustic sound events and scenes, such as the recognition of urban city sounds, vehicles, or life forms, such as birds.[1] The IEEE AASP Challenge DCASE 2016 is a benchmarking challenge for the "Detection and Classification of Acoustic Scenes and Events". It comprises four tasks, which include acoustic scene classification in urban environments (task 1), sound event detection in synthetic and real audio (tasks 2 and 3) and audio tagging of human activity in a domestic environment (task 4). In this paper we focus particularly on acoustic scene classification in urban environments (task 1). The goal of this task is to classify test recordings into one of predefined classes that characterizes the environment in which it was recorded, for example "metro station", "beach", "bus", etc. [1].

A popular choice for applying Deep Learning to audio is the use of Convolutional Neural Networks (CNN). The apparent method is to use an audio spectrogram (derived from the Fast Fourier Transform and/or other transformations) as an input to a CNN and to apply convolving filter kernels that extract patterns in 2D, similar as being done for image analysis and object recognition. Yet, audio has a fundamental difference to images: The two axes in a spectrogram do not represent a spatial coherence of visual data, but exhibit two completely different semantics: time and frequency. Approaches have been reported applying convolutions directly on the wave form (i.e. time domain) data, however with not fully satisfying success so far [2]. Therefore, typically audio is transformed into the time-frequency domain, with some (optional) further processing steps, such as the Mel transform and/or a Log transform.

In an earlier publication related to our participation in the MIREX benchmarking contest ("Music Information Retrieval Evaluation eXchange") [3] we have shown the successful application of Mel-spectrogram based Convolutional Neural Networks on music/speech classification (discrimination) [4]. Our approach won the MIREX 2015 music/speech classification task with 99.73 % accuracy.[2] As our background is the recognition of semantic high-level concepts in music (e.g. genre, or mood, c.f. [5, 6]), and Mel Frequency Cepstral Coefficients (MFCCs) are used in both music and speech recognition, the use of the Mel scale was an evident choice.

However, we realized in the course of developing a solution for the task of classifying acoustic scenes from urban sounds that an adaptation was necessary to cover activity in very low and very high frequencies that may or may not be rhythmical. Our research and experimentation led us to applying the Constant-Q-Transform (CQT), which captures low and mid-to-low frequencies better than the Mel scale. We also did a number of alterations in the architecture of the Convolutional Neural Network. Earlier research [7] showed that a combination of a CNN that captures temporal information and another one that captures timbral relations in the frequency domain is a promising approach for music genre recognition, in which typically both tempo and timbre (e.g. particular instruments) play an important role. Again, this had to be adapted for the task of audio scene classification.

In Section 2 we will give a brief overview of related work. Section 3 describes the data set and the task's challenge. Section 4 describes our method in detail, while Section 5 presents preliminary results. Finally, Section 6 summarizes the paper and provides conclusions.

2. RELATED WORK

A variety of publications study the modeling of audio signals in time and/or frequency domain for the purpose of acoustic scene recognition and event detection. Mel-frequency Cepstral Coefficients (MFCCs) typically model the frequency relations very well and are

[1] http://www.imageclef.org/lifeclef/2016/bird

[2] http://www.music-ir.org/mirex/wiki/2015:Music/Speech_Classification_and_Detection_Results

used frequently as part of an audio event detection system. However, MFCCs without any derivatives are not performing very well. Only by including derivatives of first and second order the temporal context is included to a certain extent [8]. The authors of [9] propose to use the Matching Pursuit (MP) algorithm to supplement MFCC features to yield higher accuracy. They demonstrate the effectiveness of joint MP + MFCC features for unstructured environmental sound classification, e.g. chirpings of insects and sounds of rain, which are investigated for their temporal domain signatures. Cotton and Ellis [10] propose an approach modeling acoustic events directly describing temporal context. They use convolutive non-negative matrix factorization (NMF) to discover spectro-temporal patch bases, which correspond to event-like structures. Features are derived from the activations of these patch bases. Mesaros et al present a combination of MFCC features with a Hidden Markov Models (HMM) based audio event detection system [11]. They test it on a diverse set of 61 classes of isolated events (54 % accuracy) as well as real life recordings (23.8 % avg. accuracy).

In 2013, the IEEE AASP Challenge "Detection and Classification of Acoustic Scenes and Events" (DCASE) was organized for the first time to help move forward the research in this domain [12, 13]. The dataset used in the acoustic scene classification task comprised 10 classes of indoor and outdoor urban and office sounds, similar to the current one. The authors of [12] also provide an overview of previous approaches in literature: The two main methodologies are 1) the bag-of-frames approach using a set of low-level features (e.g. MFCCs), modeling long-term statistical distribution of the local features and 2) the use of an intermediate representation that models the scene using higher level features that are usually captured by a vocabulary of "acoustic atoms", which represent audio events that are learned in an unsupervised manner from the data (e.g. by NMF). The authors also present a NMF-based system for the event detection task, in which the constant-Q transform (CQT) is used for the time-frequency representation, with a log-frequency resolution of 60 bins per octave. The best performing system in the DCASE 2013 office live event detection task [8] laid the focus on spectro-temporal features and used a two-layer HMM. The authors compare amplitude modulation spectrograms, Gabor filterbank features and MFCCs and employ various noise reduction / signal enhancement strategies. The use of Gabor filters is motivated by their similarity to spectro-temporal patterns of neurons in the auditory cortex of mammals. Their proposed spectro-temporal features achieve a better recognition accuracy than MFCCs. Another work that explores the temporal dynamics in the audio and tackles the sensitivity of MFCCs to background noise is found in [14]. The authors present a work on unsupervised feature learning for urban sound classification, employing the spherical k-means algorithm for feature learning. It is shown that classification accuracy can be significantly improved by feature learning if the domain specificities are taken into account – in this case capturing the temporal dynamics of urban sound sources.

The authors of [15] use a framework of spectrogram image-based SIF features and human auditory system modeling SAI features (stabilized auditory image) in various configurations together with Support Vector Machines (SVM) and Deep Neural Networks (DNN). The DNN uses 5 to 6 fully connected layers with 100 to 300 hidden units each and is shown to perform better than the SVM. A comparison is done to a range of other systems on various noise levels. One finding is that the SIF features used in conjunction with DNN incorporate additional temporal context being advantageous for classification in noisy environments.

3. DATA SET

For both the development and the evaluation of the system we describe in Section 4 we used the TUT Acoustic scenes 2016 dataset provided by the DCASE 2016 organizers for task 1 on Acoustic scene classification [1]. The goal of this task is to classify a recording into one of 15 different classes that represent urban and some non-urban environments. The 15 classes are: beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train and tram. In this task 1, individual train and test files are exclusively labeled with one class.

The sounds were recorded from different locations (mostly in Finland) and use 44.1 kHz sampling rate and a 24 bit resolution. For each location, a 3-5 minute long audio recording was captured. The original recordings were then split into 30-second segments for the challenge. This imposes the need for particular attention when doing train/test set splits or cross-validation: one needs to make sure that recordings from the same location are not to be found in different sets, as it introduces a beneficial bias. Thus, the task organizers made sure that all segments from the same original recording are included in a single subset – either development dataset or evaluation dataset. They also provide a 4-fold cross-validation setup for the development set which ensures this correct splitting.

For each acoustic scene, 78 segments (39 minutes of audio) were included in the development dataset and 26 segments (13 minutes of audio) were kept for evaluation. The development set contains 1170 30-sec segments (in total 9h 45mins of audio), and the evaluation set 390 30-sec segments (3h 15mins).

Full annotations for the *development set* were available, but no annotations for the *evaluation set*, as the task was still open at the time of this writing. We therefore exclusively used the *development set* of this data set to create, improve and evaluate our methodology for acoustic scene classification described in the next section, using the 4-fold cross-validation splits provided by the organizers.

4. METHOD

For the task of acoustic scene classification we use Convolutional Neural Networks, which we trained on CQT-transformed audio input. We describe these two parts in more detail.

4.1. Audio Preprocessing: CQT

Before being input to the neural network, a few preprocessing steps are carried out on the original audio which are depicted in Figure 1. First of all, a stereo audio signal is transformed to mono by averaging the two channels. Then, we apply the Constant-Q-Transform. The Constant-Q-Transform (CQT) is a time-frequency representation where the frequency bins are geometrically spaced and the so called Q-factors (ratios of the center frequencies to bandwidths) of all bins are equal [16]. The CQT is essentially a wavelet transform, which means that the frequency resolution is better for low frequencies and the time resolution is better for high frequencies. The CQT is motivated from both musical and perceptual viewpoints: The human auditory system is approximately "constant Q" in most of the

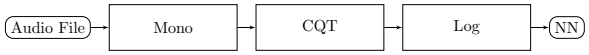


Figure 1: Preprocessing of audio before input to CNN

audible frequency range, and also the fundamental frequencies of the tones in Western music are geometrically spaced along the standard 12-tone scale [16]. Thus, the CQT typically captures 84 bands covering 7 octaves of 12 semi-tones each, however, it allows to set a different number of bands and also a higher number of bands per octave. In our approach, we use a total number of 80 bands, with the standard setting of 12 bands per octave, meaning that the 4 highest bands will be cut off. We use a hop length of 512 samples (similar as it is typically used when a fast Fourier transform is applied on 1024 samples long windows to calculate a spectrogram), i.e. a CQT is computed every 512 samples (11.6 milliseconds). Following the CQT, we perform a Log_{10} transform of all values derived from the CQT. This process is performed on chunks, or segments, of 41472 samples length (0.94 seconds), resulting in 82 CQT frames (analogously to FFT frames). The idea is to process a multitude of short-term segments from an audio example to be learned by the neural network. In this case, a 30 second input file results in 31 CQT excerpts of shape 80 bands $\times$ 82 frames.

4.2. Convolutional Neural Network

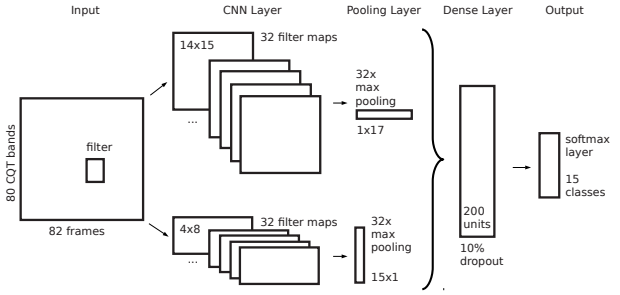


Figure 2: CNN architecture

Following [7] we created a parallel CNN architecture, which comprises a CNN Layer which is optimized for processing and recognizing relations in frequency domain, and a parallel one which is aimed at capturing temporal relations (c.f. Figure 2). Both parts of the CNN architecture use the same input, i.e. the 80 bands $\times$ 82 frames CQT matrix as output of step 1 described in Subsection 4.1. In each epoch of the training, multiple training examples, sampled from the segment-wise CQT extraction of all files in the training set, are presented to both pipelines of the neural network. Both CNN layers are followed by a Max Pooling layer, which performs a sub-sampling of the matrices that are output after applying the CNN's filter kernels. We describe this in more detail: In a Convolutional Neural Network, weights are essentially learned in a filter kernel of a particular shape. Multiple of such filter kernels – in our approach 32 in each pipeline – are applied to the input data, by convolving over the input image. Convolution means multiplication of the filter kernel with an equal sized portion of the input image. This filter kernel window is then moved sequentially over the input data (typically from left to right, top to bottom), producing an output of either equal size (when padding is used at the borders), or reduced by filter-length - 1 on each axis (when no padding is used, and the filter kernel is kept inside the borders of the input).

The particularity of this process is that the weights that are stored in each filter kernel are shared among the "input units" regardless of their input location. The filter weights are updated after each training epoch using back-propagation. Thus, by convolving over the input data, the filter kernels learn characteristic structures of the input data. The subsequent Max Pooling step serves as a data aggregation and reduction step. The pooling length in each direction determines how many "pixels" are aggregated together in the output. Max pooling thereby preserves only the maximum value from the input within its pooling window. Note that Max Pooling is applied to all 32 filter outputs (even though not visible in Figure 2).

In our CNN architecture, depicted in Figure 2, we use two pipelines of CNN Layer with 32 filter kernels each, following by a Max Pooling on all of these filter kernels. The upper pipeline is aimed at capturing frequency relations. Its filter kernel sizes are set to 14$\times$15 and the Max Pooling size to 1$\times$17. This means that the output of the filtering step is 32 matrices of shape 67$\times$68, which are then "pooled" to 32 matrices of shape 67$\times$4, preserving more information on the frequency axis than in time. On the contrary, the lower pipeline uses filter sizes of 4$\times$8 and pooling of 15$\times$1, aggregating on the frequency axis and therefore retaining more information on the time axis: Its output shape is 5$\times$75 (32 times).

In the next step, the parallel architecture is merged into a single pipeline, by flattening all the matrices from both previous pipelines, concatenating them and feeding them into a dense (fully connected) layer with 200 units. Note that the input to this layer is 20,576 weights (the flattened output of the two previous pipelines) and with 200 fully connected units (and one bias) this layer has 4,115,400 parameters. The complete network has 4,126,223 parameters, which hints at the power of Convolutional Layers: to drastically reduce the weights that are needed to make the network learn, through the spatial weight sharing principle of the filter convolution approach. Note, however, that setting the filter and pooling parameters is less straight-forward than in image retrieval where typically quadratic shapes are used for both the filter and the pooling shapes, due to the different semantics of the two axes while in images the axes have the same semantics. The parameters we described were found after a larger set of experiments (not described in this paper).

Recently, a number of techniques have been presented that make Deep Neural Networks generalize faster and better. One such technique is *Dropout*: it can be applied to any layer and reduces overfitting by dropping a percentage of random units at each weight update [17, 18]. Dropping means that it disregards these units in both input and output, so that they do not contribute to activation, nor to any weight updates. In terms of activation of a unit's output, the traditional Sigmoid function has been widely replaced by the ReLU: The *Rectified Linear Unit* simplifies and speeds up the learning process by using the activation function $f(x) = max(0, x)$ [19]. Due to its sparse activation (in a randomly initialized network) only about 50% of hidden units are activated, which makes the network generalize much faster [20]. The *Leaky ReLU* [21] is an extension to the ReLU that does not completely cut off activation for negative values, but allows for negative values close to zero to pass through. It is defined by adding a coefficient α in $f(x) = \alpha x$, for $x < 0$, while keeping $f(x) = x$, for $x \geq 0$ as for the ReLU.

In our architecture, we apply Leaky ReLU activation with $\alpha = 0.3$ in both Convolutional layers, and Sigmoid activation in the dense layer. We apply a Dropout value of 0.1 to the fully connected layer. The last layer is a so-called Softmax layer: It connects the 200 units of the preceding layer with as many units as the number of output classes (15), and applies the Softmax function to guarantee that the output activations to always sum up to 1 [20]. The output from the Softmax layer can be thought of as a probability distribution and is typically used for single-label classification problems. All layers are initialized with the Glorot uniform initialization [22].

For the experiments presented in Section 5 this CNN architec-

ture was trained over 100 epochs with a constant learning rate of 0.02. The model is adapted in each epoch using Stochastic Gradient Descent (SGD) and a mini-batch-size of 40 instances.

The system is implemented in Python and using *librosa* for the CQT-transform and *Theano*-based library *Keras* for Deep Learning.

5. RESULTS

5.1. Data set and Baseline

For our experimental results, we used exclusively the *development* dataset that was provided by the DCASE 2016 Acoustic Scene Classification task organizers, which was described in Section 4.The task organizers also provide a cross-validation setup for this development dataset which consists of 4 folds distributing the 78 available segments based on location, to ensure that all files recorded in same location are placed on the same side of the evaluation, in order to prevent bias from recognizing the recording location.We used the provided fold splits in order to make results comparable to other work, including the baseline system that was also provided by the task organizers. The baseline system is a GMM classifier using MFCC audio features calculated using frames of 40 ms with a Hamming window and 50 % overlap. 40 Mel bands are extracted but only the first 20 coefficients are kept, plus delta and acceleration coefficients (60 values in total). The system learns one acoustic model per acoustic scene class (GMM with 32 components) and performs the classification using a maximum likelihood classification scheme (expectation maximization) [1]. The reported average classification accuracy over 4 folds is 72.5 %.

5.2. Evaluation

As our system analyzes and predicts multiple audio segments per input audio file, there are several ways to perform the final prediction of an input instance:

Maximum Probability: The output probabilities of the Softmax layer for the 15 classes are summed up for all segments belonging to the same input file. The predicted class is determined by the maximum probability among the classes from the summed probabilities.

Majority Vote: Here, the predictions are made for each segment processed from the audio file as input instance to the network. The class of an audio segment is determined by the maximum probability as output by the Softmax layer for this segment instance. Then, a majority vote is taken on all predicted classes from all segments of the same input file. Majority vote determines the class that occurs most often.

In both cases, the resulting accuracy is determined by comparing the file-based predictions to the groundtruth provided by the task organizers. We present the result achieved by the system described in Section 4 and compare the impact of different audio transformations as an input step to the CNN. Table 1 shows the results. The Mel frequency transforms have been computing using a Fast Fourier Transform (FFT) with a Hanning window of 1024 samples and 50 % overlap. The segment size of the audio chunks has been chosen to be equal to the one used for CQT, which results in 80 frames. The FFT spectrogram frequency bands are transformed to Mel scale by applying 40 or 80 Mel filters. Subsequently a Log_{10} transform is applied. From the results table we see that the approach with 80 Mel filters performed only slightly better. Yet, we also see that using the CQT instead of the Mel-transform has a beneficial impact. The best result is achieved with 80 CQT bands. It is 80.25 % accuracy with the Maximum Probability strategy and 80.07 % with Majority Vote. Applying the full standard CQT of 7 octaves with 12 semi-tones each performed worse. Extending the 12 semi-tones to 18 bands per octave, with 126 CQT bands in total (covering the same 7 octaves) did also not improve the results.

Transform	Bands / Frames	$A_{maxprob}$	$A_{majvote}$
Mel	40×80	76.23%	75.62%
Mel	80×80	76.55%	76.38%
CQT	80×82	**80.25%**	**80.07**%
CQT	84×82	78.11%	77.59%
CQT	126×82	79.39%	79.14%

Table 1: Different input transformations to the parallel CNN

	li	ci	tr	pa	fo	gr	re	ca	tr	of	be	me	bu	ho	ca
library	61	0	0	1	0	3	0	0	3	0	0	10	0	0	0
city_center	0	76	0	0	0	0	1	0	0	0	0	1	0	0	0
tram	0	0	70	1	0	1	0	3	2	0	0	0	1	0	0
park	5	0	0	35	0	0	26	0	0	0	7	1	2	2	0
forest_path	0	0	0	0	75	0	2	0	0	0	0	0	0	0	1
grocery_store	0	3	0	0	0	73	0	0	0	0	0	2	0	0	0
residential_area	0	3	0	20	7	0	48	0	0	0	0	0	0	0	0
car	0	0	3	0	0	0	0	75	0	0	0	0	0	0	0
train	0	0	11	3	0	0	2	0	51	0	0	1	9	0	1
office	0	0	0	0	0	0	0	0	0	68	0	0	0	10	0
beach	0	1	0	5	0	0	6	2	0	0	64	0	0	0	0
metro_station	6	0	0	5	0	0	0	0	0	0	0	66	0	1	0
bus	0	0	2	2	0	0	1	1	11	0	0	0	61	0	0
home	0	0	3	0	1	0	2	0	0	1	0	0	0	71	0
cafe/restaurant	0	0	0	0	0	20	0	0	1	0	0	3	3	6	45

Figure 3: Confusion Matrix of the best model of Table 1.

Next, we investigate the per-class accuracies by having a look at the confusion matrix in Figure 3. It can be observed that the best configuration of the proposed system excels for the classes *city center*, *forest path*, *grocery store*, *car* and *home* with accuracies ranging from 91% to 97.4%. The largest confusions are between the classes *residential area* and *park*, *cafe/restaurant* and *grocery store* as well as *tram*, *train* and *bus*.

6. SUMMARY

We have shown how we adapted a musically inspired Convolutional Neural Network approach to recognize acoustic scenes from recordings of urban environments. The crucial adaptations were the utilization of the Constant-Q-Transform to capture essential audio information from both low and high frequencies in sufficient resolution and the creation of a parallel CNN architecture which is capable of capturing both relations in time and frequency. The presented Deep Neural Network architecture has shown a 10.7 % relative improvement over the baseline system provided by the DCASE 2016 Acoustic Scene Classification task organizers, achieving 80.25 % on the same 4-fold cross-validation setup as provided. With a few additional optimizations, described in our submission abstract [23], our approach achieved 83.3 % accuracy on the evaluation set in the challenge (rank 14 of 35). On the domestic audio tagging task, our improved approach is the winning algorithm (rank 1 of 9) with 16.6 % equal error rate. We conclude that this system is capable of detecting urban acoustic settings, yet there is ample room for improving the system further.

7. ACKNOWLEDGMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

8. REFERENCES

[1] T. H. Annamaria Mesaros and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[2] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 6964–6968, 2014.

[3] J. Downie, K. West, A. Ehmann, and E. Vincent, "The 2005 music information retrieval evaluation exchange (MIREX 2005): preliminary overview," in *6th Int. Conf. on Music Information Retrieval (ISMIR)*, 2005, pp. 320–323.

[4] T. Lidy, "Spectral convolutional neural network for music classification," in *Music Information Retrieval Evaluation eXchange (MIREX)*, Malaga, Spain, October 2015.

[5] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, September 11-15 2005, pp. 34–41.

[6] T. Lidy, C. N. S. Jr., O. Cornelis, F. Gouyon, A. Rauber, C. A. A. Kaestner, and A. L. Koerich, "On the suitability of state-of-the-art music information retrieval methods for analyzing, categorizing, structuring and accessing non-western and ethnic music collections," *Signal Processing*, vol. 90, no. 4, pp. 1032 – 1048, April 2010, Special section: ethnic music audio documents: from the peservation to the fruition.

[7] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *Proceedings of the 14th International Workshop on Content-based Multimedia Indexing (CBMI 2016)*, Bucharest, Romania, June 2016.

[8] J. Schröder, N. Moritz, M. R. Schadler, B. Cauchi, K. Adiloglu, J. Anemuller, S. Doclo, B. Kollmeier, and S. Goetze, "On the use of spectro-temporal features for the IEEE AASP challenge 'detection and classification of acoustic scenes and events'," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA.

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

[10] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2011, pp. 69–72.

[11] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *European Signal Processing Conference*, 2010, pp. 1267–1271.

[12] D. Giannoulis, D. Stowell, E. Benetos, M. Rossignol, M. Lagrange, and M. D. Plumbley, "A database and challenge for acoustic scene classification and event detection," in *European Signal Processing Conference*, 2013.

[13] 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, 2015.

[14] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. April, South Brisbane, Australia, 2015, pp. 171–175.

[15] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust Sound Event Classification Using Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 540–552, Mar 2015.

[16] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," *7th Sound and Music Computing Conference*, pp. 3–64, Jan 2010.

[17] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv: 1207.0580*, pp. 1–18, 2012.

[18] N. Srivastava, G. E. 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.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th International Conference on Machine Learning*, no. 3, pp. 807–814, 2010.

[20] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com

[21] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *ICML 2013*, vol. 28, 2013.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, 2010, pp. 249–256.

[23] 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 (DCASE 2016), Budapest, Hungary, Tech. Rep., September 3 2016.

PAIRWISE DECOMPOSITION WITH DEEP NEURAL NETWORKS AND MULTISCALE KERNEL SUBSPACE LEARNING FOR ACOUSTIC SCENE CLASSIFICATION

Erik Marchi[1,3]*, Dario Tonelli*[2]*, Xinzhou Xu*[1]*, Fabien Ringeval*[1,3]*, Jun Deng*[1]*, Stefano Squartini*[2]*, Björn Schuller*[1,3,4]

[1] University of Passau, Chair of Complex and Intelligent Systems, Germany
[2]A3LAB, Department of Information Engineering, Universitá Politecnica delle Marche, Italy
[3]audEERING GmbH, Gilching, Germany
[4]Imperial College London, Department of Computing, London, United Kingdom
erik.marchi@tum.de

ABSTRACT

We propose a system for acoustic scene classification using pairwise decomposition with deep neural networks and dimensionality reduction by multiscale kernel subspace learning. It is our contribution to the Acoustic Scene Classification task of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE2016). The system classifies 15 different acoustic scenes. First, auditory spectral features are extracted and fed into 15 binary deep multilayer perceptron neural networks (MLP). MLP are trained with the 'one-against-all' paradigm to perform a pairwise decomposition. In a second stage, a large number of spectral, cepstral, energy and voicing-related audio features are extracted. Multiscale Gaussian kernels are then used in constructing optimal linear combination of Gram matrices for multiple kernel subspace learning. The reduced feature set is fed into a nearest-neighbour classifier. Predictions from the two systems are then combined by a threshold-based decision function. On the official development set of the challenge, an accuracy of 81.4% is achieved.

Index Terms— Computational Acoustic Scene Analysis, Acoustic Scene Classification, Multilayer Perceptron, Deep Neural Networks, Multiscale Kernel Analysis

1. INTRODUCTION

Acoustic scene classisfication aims at recognising the acoustic background and goes under the field of Computational Auditory Scene Analysis (CASA) [1]. Acoustic scene analysis is a challenging task since a plethora of different overlapping sound sources are composing the acoustic mark of a certain scene, making it a complex combination of various acoustic events.

In the past years, we observed an increasing interest on intelligent audio-based systems able to recognise an environment around a device [2]. This has stimulated the research community to find more robust and efficient methods ranging from unsupervised approaches such as acoustic novelty detection [3, 4] to supervised approaches such as acoustic scene classification and sound even detection [5].

Several works on acoustic scene classification applied different spectral, energy and voicing-related features, in conjunction with neural networks [6]. A system for acoustic scene recognition is described and evaluated in [7]. That system uses several audio features and a nearest neighbour (NN) classifier. In [8], a system for acoustic scene classification is described. The approach relies on Support Vector Machines (SVM), embedded in a hierarchical or parallel framework. In [9], the detection and classification of acoustic events is evaluated by providing a testbed. In [10], it was shown how, in the case of small amounts of training data, new acoustic events can be *learned* by a system. In [11], large-scale acoustic features are used in combination with SVM for the task of acoustic scene analysis.

Acoustic scene classification is applicable in several fields such as intelligent user interfaces [12], serious games [13], automotive [14], and street routing [15], where the context can be recognised using acoustic scene classification techniques.

In the scene classification task of the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE2016), systems for acoustic scene recognition are compared. The provided corpus is divided into a development set and a non-public evaluation set. The dataset is categorised into 15 different classes of acoustic scenes.

This contribution describes our investigated method for acoustic scene classification. From the recordings, auditory spectral features and a large number of spectral, cepstral, energy and voicing-related audio features are extracted. Fifteen binary deep MLP neural networks are trained in a 'one-against-all' fashion to perform a pairwise decomposition instead of simply training a multi-class neural network. In a second stage, multiscale Gaussian kernels are used for multiple kernel suspace learning in order to decrease the dimensionality of the feature space. The reduced feature set is fed into a nearest-neighbour classifier. Finally, the predictions from the two system are then combined with a threshold-based decision function. To our best knowledge, little research focuses on multikernel subspace learning for CASA. Thus, we aim at filling this white spot in the litterature in order to verify if this method can significantly improve the generalisation abilities of a system for acoustic scene classification.

On the official development set of the challenge, an accuracy of 81.4 % is achieved. The employed database, audio features and classification methods are described in Section 2. Experimental results are presented in Section 3, and conclusions are given in Section 4.

2. METHODOLOGY

2.1. Database

For evaluation of our system, we employ the official dataset of the IEEE AASP Challenge on Detection and Classification of Acoustic

Scenes and Events [5]. Thereby, we use only the data of the scene classification task. This dataset contains 30 s recordings of various acoustic scenes, categorised into fifteen different classes. For each of the fifteen classes, the database contains 39 minutes of recordings in the development set, summing up to 9 hours and 45 minutes total duration of the development set. In addition, for the challenge, the systems were evaluated with a non-public test set containing similar data. Sounds were recorded with a high-quality binaural recording system, whereby the portability and subtlety of the system allowed to obtain unobstructed everyday recordings with relative ease. Since the recordings were performed with binaural microphones on the ears of a person, the head-related transfer function (HRTF) of that person is intrinsically incorporated.

2.2. Acoustic features

Auditory Spectral Features (ASF) [16, 17] are computed by applying the Short Time Fourier Transformation (STFT) using a frame size of 40 ms and a frame step of 20 ms. Each STFT yields the power spectrogram which is converted to the Mel-Frequency scale using a filter-bank with 26 triangular filters obtaining the Mel spectrograms $M_{40}(n,m)$. Finally, to match the human perception of loudness, a logarithmic representation is chosen:

$$M_{log}^{40}(n,m) = log(M_{40}(n,m) + 1.0). \quad (1)$$

In addition, the positive first order differences $D_{40}(n,m)$ are calculated from each Mel spectrogram as follows:

$$D_{40}(n,m) = M_{log}^{40}(n,m) - M_{log}^{40}(n-10,m), \quad (2)$$

with n being the frame index, and k the frequency bin index. Furthermore, the frame energy and the log frame energy are also included as a feature leading to a total number of 56 features. The features are extracted with our open-source audio feature extractor openSMILE [18].

We separately consider the feature sets of the 'emobase' configuration [19]. These features are obtained by extracting the following Low-Level Descriptors (LLDs): intensity, loudness, 12 MFCC, fundamental frequency (F0), probability of voicing, F0 envelope, 8 line spectral frequencies, zero-crossing rate. Statistical functionals are then applied to the LLDs and their first order differences. The following functionals have been used: max./min. value and respective relative position within input, range, arithmetic mean, two linear regression coefficients, and linear and quadratic error, standard deviation, skewness, kurtosis, quartile 1–3, and 3 inter-quartile ranges resulting in a total of 988 features.

2.3. Pairwise Decomposition

Multi-class neural learning [20] can be implemented via several paradigms. One of those is the so called 'one-against-all' paradigm. It consists in decomposing an N-class pattern recognition problem into a system of $L > 1$ neural networks. The L neural networks are trained using a given data set with the assumption of using different class lables. A decision function is usually applied to fuse the results of L neural networks and provide the final system prediction. The 'one-against-all' modelling paradigm employs an ensamble of $L = N$ binary neural networks, $ANN_i, i = 1, ..., N$, each with one unit output layer Y_i with output function f_i (usually sigmoid function) that provides $f_i(\bar{x}) = 1$ or 0 whether the input vector $\bar{x}$ belongs to class i or does not belong to class i. In order to train the i-th neural network ANN_i, the training set S_{tr} is relabelled in two sets, $S_{tr} = {S_{tr}}^i \cup \bar{S}_{tr}^i$, where ${S_{tr}}^i$ consists of all the reference patterns belonging to class i (labelled as '1'), and $\bar{S}_{tr}^i$ consists of all the reference patterns belonging to remaining other classes (labelled as '0'). The decision module in this paradigm should be designed to face the following three output scenarios: The first scenario consists in obtaining $f_i = 1$, and $f_j = 0$ for all j given $i \neq j$. The decision function D can be easily implemented as $D(\bar{x}, f_1, f_2, ..., f_L) = argmax_{i=1,...,L}(f_i)$. The second scenario consists in obtaining all $f_i = 0$ for $i = 1, ..., L$. The third one consists in having more than one neural networks output '1'. In both last scenarios the system is uncertain and the decision function outputs the class label that corresponds to the neural network that shows the largest output value by the activation function at the output layer unit:

$$D(\bar{x}, y_1, y_2, ..., y_L) = argmax_{i=1,...,L}(y_i), \quad (3)$$

where y_i is the output of the activation function used in the output layer of the i-th neural network.

Since we used fully connected MLP feed-forward neural networks, we will refer to this approach as pairwise decomposition with MLP (PDMLP).

In our final system we applied an enhanced decision function that relies on an auxiliary system when the PDMLP is uncertain. The adopted decision function is described in Section 2.5.

2.4. Auxiliary Systems

A system of N binary neural networks trained with 'one-against-all' is indeed a more flexible system and allows for a better discrimination of one class. However, it has one major drawback, the system decision borders generated by the N binary neural networks are suffering from overlapping or uncovering regions in a feature space. In order to mitigate this drawback we introduced some auxiliary systems when the N binary neural networks are providing uncertain outputs. We applied three different auxiliary system: selected 'one-against-one' (OAO) classification, SVM, and Multiple kernel leaning.

2.4.1. Selected 'one-against-one'

A first auxiliary system decomposes an N-class pattern classification problem into $N(N-1)/2$ two-class classification problems using the OAO paradigm. Let's define the $N(N-1)/2$ two-class neural networks as $ANN_k(i,j)$, with $1 \leq k \leq L = N(N-1)/2$. An $ANN_k(i,j)$ represents a neural network trained to discriminate class i from class j , for $1 \leq i < j \leq N$. An $ANN_k(i,j)$ is trained with reference patterns of class i and j, and its output, $f_k(i,j)$, is indicating whether the input pattern $\bar{x}$ is either class i or j. In our case, we refer to selected OAO (sOAO) since we just use as an auxiliary system the binary classifier $\bar{ANN}_k(i,j)$ where i and j are the two more likely classes resulting from the output of the PDMLP in Section 2.3.

2.4.2. Support Vector Machines

As a second auxiliary system we applied the traditional SVM approach trained on the high dimensional feature set 'emobase'.

SVMs have shown to achieve good performances for the task of acoustic scene analysis [10, 11], and are used in this contribution as a comparison to a state-of-the-art method.

2.4.3. Multiple Kernel Learning

The third auxiliary system is based on MultiScale-Kernel Fisher Discriminant Analysis (MSKFDA). This methods was recently

proven to be effective in solving emotion recognition in speech [21]. To our best knowledge, little research focuses on multiscale representation in CASA and we believe that this method for subspace learning may significantly improve the generalisation of the system by learning more robust features. This method benefits from alternatively optimising two variables, namely the kernelised mapping directions and a nonnegative linear combination for kernels with different scaling parameters.

The research of MSKFDA provides the possibility of solving multiscale analysis of acoustic scene factors. For Gaussian kernels, it is easy to draw the multiscale case by regulating scaling parameters. The kernel transforming between samples x_i and x is shown in Eq. (4), with the parameters $\sigma_m > 0$, $m = 1, 2, \ldots, M$ and $i = 1, 2, \ldots, N$:

$$(\Omega_{xi})_m = \phi_m^T(x_i)\phi_m(x) = e^{-\frac{(x_i - x)^2}{\sigma_m^2}}, \quad (4)$$

where Ω_{xi} is the multiple kernel coordinate matrix, and $\phi_m(x)$ is the high dimensional form of x. Kernel methods are originally represented as high-dimensional space by adopting inner product forms in RKHS. However, it can be also assumed that kernel methods bring a dimension-limited feature transformation in graph embedding. This transformation constructs a new feature space for each sample by kernel functions and training samples. Thus, the relationship between a given sample and each training sample leads to the new features. Then, the scales of kernels are mainly determined by the parameters of respective kernels.

As is shown in Figure 1, for sample x, the original features x are transformed into new features $\Omega_x\beta$ by linearly combining multiscale kernels, where $\beta \in \Re^{M\times 1}$ is the column vector with corresponding elements $\beta_m \geq 0$ for kernel m. Then, for the new features of x, the dimensionality-reduced sample can be achieved by using $A^T\Omega_x\beta$ in bilateral ways, where A contains the kernel mappings.

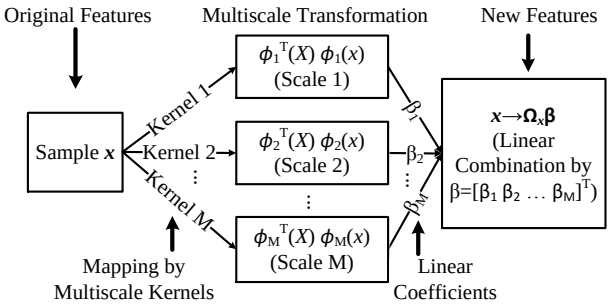


Figure 1: Schematic diagram of learning multiscale kernels. The original features x are transformed into new features $\Omega_x\beta$ by linearly combining multiscale kernels.

High-dimensional acoustic features inevitably include much interference resulting from the factors of background environment sound, speakers, etc., in spite of state-of-the-art feature acquisition ways. Therefore, CASA systems would benefit from the suggested novel feature reduction method in combination with the embedding graphs of FDA and multiple kernel learning. In addition, few parameters need to be regulated in FDA. For these reasons, we utilise MSKFDA as an auxiliary tool in order to obtain better performance as second stage of our algorithm.

2.5. Decision Functions

In our final system, the decision function is obtained by first applying a threshold to the output activations of a PDMLP. If the number of outputs above the threshold is 1, then the predicted class is the one corresponding to the neural network that generated that output. Otherwise, if more outputs are above the threshold or none of the outputs are above the threshold we only rely on the auxiliary system predicted class. The value of threshold is set to 0.3 and was optimised on the development set.

3. EXPERIMENTS

This section contains the experimental setup and the evaluation of different approaches on the development set of DCASE.

3.1. Setup

In the fifteen binary classifiers composing the PDMLP system, MLP were trained on 100 parallel sequences per batch, using Stochastic Gradient Descent with Adam by applying a fixed learning rate of 0.001 and a binary cross-entropy objective function. We used rectified linear units as activation function. Weights were initialized with Gaussian normal distribution (σ = 0.1, μ = 0). For better generalization, the networks were trained using early stopping on the corresponding test set per each fold. Furthermore, the early stopping criterion was applied considering the sum of all validation errors at each epoch of each network. In this way, we first reduced the training time by a factor of 3 and we also avoided potential overfitting. The training procedure stopped after a maximum number of 1000 epochs. The 'selected OAO' auxiliary networks were trained in the same fashion. In order to compare the proposed approaches with state-of-the-art methods, we also evaluated traditional multi-class MLP with exactly the same training algorithm and parameters, except that we used a multi-class cross entropy error as objective function. All networks were trained using Theano [22] and Lasagne[1]. We also evaluated SVM with a linear kernel and complexity value $C = 0.001, 0.01, 0.1, 1.0, 10.0$. SVM are trained with the sequential minimal optimisation (SMO) algorithm using the training data. The parameters in the MSKFDA are set as follows. The number of scales is set as $M = 21$, with the Gaussian scaling parameters σ_m ($m = 1, 2, \ldots, M$) selected as $0.0001n$, $0.0003n$, $0.0005n$, $0.0007n$, $0.001n$, $0.003n$, $0.005n$, $0.007n$, $0.01n$, $0.03n$, $0.05n$, $0.07n$, $0.1n$, $0.3n$, $0.5n$, $0.7n$, n, $3n$, $5n$, $7n$, and $10n$, respectively, where n is the number of original features. We employ openSMILE's '*emobase*' feature set which results in $n = 988$ here. The dimensions d of the dimensionality-reduced feature space are selected no larger than 21. The number of iterations is set as 7. A Nearest-Neighbour classifier is selected as the final decision maker.

3.2. Results

We first tested traditional multi-class approaches by using SVM and MLP in order to compare the performance of PDMLP with state-of-the-art methods. Table 1 reports performances on the development set using the 4-fold cross validation as specified in the challenge baseline. By applying PDMLP, we can observe an absolute improvement of 7% accuracy over the baseline of the DCASE challenge [5]. SVMs perform slightly better than the baseline with up to 74% accuracy (best performance obtained with $C = 0.1$).

[1] https://lasagne.readthedocs.io

Table 1: Comparison of performances between traditional multi-class systems and the proposed method with 15 binary classifiers (PDMLP). Multi-class classifiers: Gaussian Mixture Models (Baseline), Support Vector Machines (SVM), and Multi Layer Perceptron (MLP). For neural networks, the layout is indicated in parenthesis (*number of units* × *number of layers*). Results are given in terms of accuracy [%].

Method	**Fold1**	**Fold2**	**Fold3**	**Fold4**	**Mean**
Baseline [5]	67.2	68.9	72.2	81.9	72.5
SVM	72.2	73.9	77.1	72.4	74.0
MSKFDA	76.8	73.7	79.5	79.7	77.5
MLP (*54x3*)	78.5	70.8	77.7	75.9	75.9
MLP (*256x3*)	78.6	77.7	76.2	77.1	77.4
PDMLP (*54x3*)	82.6	76.9	77.5	77.0	78.5
PDMLP (*256x3*)	81.4	78.2	77.5	80.8	**79.5**

Table 2: Combination of the PDMLP system with different auxiliary systems. Best accuracy (%) obtained on the development set with 4-fold cross validation. Auxiliary systems: Support Vector Machines (SVM), selected 'one-against-one' classifier (sOAO), and MultiScale-Kernel Fisher Discriminant Analysis (MSKFDA) with nearest-neighbours. Results are given in terms of accuracy [%].

	Folds				
Method	1	2	3	4	**Mean**
PDMLP	81.4	78.2	77.5	80.8	79.5
PDMLP-SVM	79.4	75.8	78.6	80.0	78.4
PDMLP-sOAO	81.9	80.4	75.8	81.2	80.8
PDMLP-MSKFDA	81.5	79.8	81.5	82.9	**81.4**

MSKFDA performs significantly better than SVM with an auccuracy of 77.5%, corroborating our assumption that multikernel subspace learning is effective for acoustic scene classification. Multi-class MLP are evaluated using different layouts. For simplicity, we only report the best results obtained with three hidden layers composed by 54 units and 256 units. We can observe that, increasing the dimension of the hidden layer to 256 units brings better performances up to 77.4% accuracy, however, no more improvement was observed by further augmenting the dimensionality of the hidden layer. The same layout (256-256-256) also brought about an increase in performance in the PDMLP method of up to 79.5% accuracy. We kept this layout for the PDMLP as final first stage system and applied the auxiliary systems by adopting the enhanced decision function described in Section 2.5.

Table 2 shows the results obtained from the fusion with SVM, sOAO and MSKFDA. We observe that the combination with SVM is not fruitful and led to a decrease in performance down to 78.4% accuracy. However, combining PDMLP and sOAO seems to increase performances up to 80.8% with an absolute improvement of 1.3% accuracy. Further improvement is observed with the combination of PDMLP and MSKFDA up to 81.4% with an absolute improvement of 8.9% accuracy over the challenge baseline.

Table 3 shows the confusion matrix for the best-performing system, using PDMLP and MSKFDA. Some classes (*office*, *car*) are recognised with high accuracy, while for others (*park, restaurant, train*), low scores are obtained. Most confusions are made between the classes *park* and *residential area* or *city* and *train*. The recordings of the classes *park* and *residential area* are partly very similar.

Summing up, we believe that such a system is more robust to variation since it relies on two generalised system. In fact, PDMLP can be considered already a very flexible system given that it was tailored to discriminate one class against the rest. Additionally, we carefully trained the 15 MLP considering the overall validation error, avoiding individual training and subsequent overfitting. Furthermore, by selecting MSKFDA as auxiliary system we relied on another well-generalised model trained on a reduced and rubust feature set obtained via multi kernel subspace learning.

Table 3: Confusion Matrix of the development data for the proposed system, achieving an accuracy of 81.4 %.

	beach	*bus*	*cafe*	*car*	*city*	*forest*	*grocery*	*home*	*library*	*metro*	*office*	*park*	*resid.*	*train*	*tram*
beach	**62**	0	0	0	4	0	0	0	0	0	0	7	4	0	1
bus	0	**68**	0	6	0	0	0	0	0	0	0	0	0	2	2
cafe/rest.	0	0	**59**	0	0	0	6	2	0	7	0	1	0	0	3
car	0	4	0	**71**	0	0	0	0	0	0	0	0	0	3	0
city	0	0	0	0	**74**	0	1	0	0	0	0	0	3	0	0
forest	1	0	0	0	0	**73**	0	1	0	0	0	1	2	0	0
grocery	2	0	9	0	0	0	**56**	0	0	11	0	0	0	0	0
home	5	0	1	0	0	1	0	**67**	2	0	1	1	0	0	0
library	0	0	3	0	0	0	2	0	**71**	0	0	0	0	2	0
metro st.	0	0	0	0	0	0	3	5	0	**70**	0	0	0	0	0
office	0	0	0	0	0	0	0	0	5	0	**73**	0	0	0	0
park	4	0	0	0	2	1	0	0	4	0	0	**47**	20	0	0
res. area	2	0	0	0	1	4	0	0	1	0	1	15	**54**	0	0
train	0	9	6	0	15	0	1	0	1	0	0	0	0	**39**	7
tram	0	1	0	0	2	0	5	0	0	0	0	0	0	1	**69**

4. CONCLUSIONS

We presented and evaluated a system for acoustic scene classification. Combining pairwise decomposition with deep neural networks and dimensionality reduction by multiscale kernels, an accuracy of 81.4 % is obtained on the development set of the D-CASE challenge. A comparison with stat-of-the-art approaches showed that the pairwise decomposition can alone bring a significant (one-tailed z-test [23], $p<0.001$) improvement. Furthermore, we found that a dimensionality reduction via multiple kernel learning is also effective and outperforms the baseline significantly. The two methods seems to be complementary and thus – if combined – they provide a more robust system. Some acoustic scenes (*park, restaurant, train, city*) are difficult to recognise due to the high variability in the class and the similarity between the different classes. In future works, we will focus on new acoustic features and enhanced decision functions for the late fusion stage.

5. ACKNOWLEDGMENT

The research leading to these results has received funding from the EU's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 338164 (ERC Starting Grant iHEARu), and the EU's Horizon 2020 Programme agreements No. 645378 (RIA ARIA VALUSPA), and No. 688835 (RIA DE-ENIGMA).

6. REFERENCES

[1] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley interscience, 2006.

[2] 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. on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, Jan 2006.

[3] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A Novel Approach for Automatic Acoustic Novelty Detection Using a Denoising Autoencoder with Bidirectional LSTM Neural Networks," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. Brisbane, Australia: IEEE, Apr 2015, p. 5.

[4] E. Marchi, F. Vesperini, F. Weninger, F. Eyben, S. Squartini, and B. Schuller, "Non-Linear Prediction with LSTM Recurrent Neural Networks for Acoustic Novelty Detection," in *Proc. 2015 Int. Joint Conference on Neural Networks, IJCNN*, IEEE. Killarney, Ireland: IEEE, Jul 2015, p. 5.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *Proc. 24th European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[6] Z. Liu, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene segmentation and classification," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 20, no. 1-2, pp. 61–79, 1998.

[7] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Orlando, FL, USA, 2002.

[8] H. Jiang, J. Bai, S. Zhang, and B. Xu, "Svm-based audio scene classification," in *Proc. Natural Language Processing and Knowledge Engineering (NLP-KE)*. IEEE, 2005, pp. 131–136.

[9] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo, "Clear evaluation of acoustic event detection and classification systems," in *Multimodal Technologies for Perception of Humans*. Springer, 2007, pp. 311–322.

[10] J. T. Geiger, M. A. Lakhal, B. Schuller, and G. Rigoll, "Learning new acoustic events in an hmm-based system using map adaptation," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 293–296.

[11] J. T. Geiger, B. Schuller, and G. Rigoll, "Large-Scale Audio Feature Extraction and SVM for Acoustic Scene Classification," in *Proc. of the 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2013*, IEEE. New Paltz, NY: IEEE, October 2013, pp. 1–4.

[12] N. Sabouret, L. Paletta, B. Schuller, E. Marchi, H. Jones, and A. B. Youssef, "Intelligent User Interfaces in Digital Games for Empowerment and Inclusion," in *Proc. of the 12th Int. Conference on Advancement in Computer Entertainment Technology, ACE 2015*, ACM. Iskandar, Malaysia: ACM, November 2015, p. 8.

[13] B. Schuller, E. Marchi, S. Baron-Cohen, A. Lassalle, H. O'Reilly, D. Pigat, P. Robinson, I. Davies, T. Baltrusaitis, M. Mahmoud, O. Golan, S. Friedenson, S. Tal, S. Newman, N. Meir, R. Shillo, A. Camurri, S. Piana, A. Staglianò, S. Bölte, D. Lundqvist, S. Berggren, A. Baranger, N. Sullings, M. Sezgin, N. Alyuz, A. Rynkiewicz, K. Ptaszek, and K. Ligmann, "Recent developments and results of ASC-Inclusion: An Integrated Internet-Based Environment for Social Inclusion of Children with Autism Spectrum Conditions," in *Proc. 3rd Int. Workshop on Intelligent Digital Games for Empowerment and Inclusion (IDGEI 2015) as part of the 20th ACM IUI 2015*, ACM. Atlanta, GA: ACM, March 2015.

[14] I. Abdić, L. Fridman, E. Marchi, D. E. Brown, W. Angell, B. Reimer, and B. Schuller, "Detecting Road Surface Wetness from Audio: A Deep Learning Approach," *arxiv.org*, no. 1511.07035, p. 5, December 2015.

[15] B. Schuller, F. Pokorny, S. Ladsttter, M. Fellner, F. Graf, and L. Paletta, "Acoustic geo-sensing: Recognising cyclists' route, route direction, and route progress from cell-phone audio," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013.

[16] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller, "Multi-resolution Linear Prediction Based Features for Audio Onset Detection with Bidirectional LSTM Neural Networks," in *Proc. Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. Florence, Italy: IEEE, 2014, pp. 2183–2187.

[17] E. Marchi, G. Ferroni, F. Eyben, S. Squartini, and B. Schuller, "Audio Onset Detection: A Wavelet Packet Based Approach with Recurrent Neural Networks," in *Proc. 2014 Int. Joint Conference on Neural Networks (IJCNN) as part of the IEEE World Congress on Computational Intelligence (IEEE WCCI)*, IEEE. Beijing, China: IEEE, Jul 2014, pp. 3585–3591.

[18] F. Eyben, F. Weninger, F. Groß, and B. Schuller, "Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor," in *Proc. of the 21st ACM Int. Conference on Multimedia, MM 2013*, ACM. Barcelona, Spain: ACM, Oct 2013, pp. 835–838.

[19] F. Eyben, M. Wöllmer, and B. Schuller, "OpenEAR–introducing the Munich open-source emotion and affect recognition toolkit," in *Affective Computing and Intelligent Interaction and Workshops*. Amsterdam, The Netherlands: IEEE, 2009, pp. 576–581.

[20] G. Ou and Y. L. Murphey, "Multi-class pattern classification using neural networks," *Pattern Recognition*, vol. 40, no. 1, pp. 4 – 18, 2007.

[21] X. Xu, J. Deng, W. Zheng, L. Zhao, and B. Schuller, "Dimensionality reduction for speech emotion features by multiscale kernels," in *Proc. Annual Conference of the Int. Speech Communication Association (INTERSPEECH)*. Dresden, Germany: ISCA, 2015, pp. 1532–1536.

[22] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[23] M. D. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *Proc. of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, ser. CIKM '07. New York, NY, USA: ACM, 2007, pp. 623–632.

Acoustic Scene Classification using Time-Delay Neural Networks and Amplitude Modulation Filter Bank Features

Niko Moritz, Jens Schröder, Stefan Goetze [1]

Fraunhofer IDMT, Project Group for Hearing, Speech, and Audio Technology
Marie-Curie-Str. 2, 26121 Oldenburg, Germany
niko.moritz@idmt.fraunhofer.de

Jörn Anemüller, Birger Kollmeier

University of Oldenburg
Medizinische Physik & Hearing4all
Carl-von-Ossietzky-Str. 9-11
26129 Oldenburg, Germany

ABSTRACT

This paper presents a system for acoustic scene classification (ASC) that is applied to data of the ASC task of the DCASE'16 challenge (Task 1). The proposed method is based on extracting acoustic features that employ a relatively long temporal context, i.e., amplitude modulation filer bank (AMFB) features, prior to detection of acoustic scenes using a neural network (NN) based classification approach. Recurrent neural networks (RNN) are well suited to model long-term acoustic dependencies that are known to encode important information for ASC tasks. However, RNNs require a relatively large amount of training data in comparison to feed-forward deep neural networks (DNNs). Hence, the time-delay neural network (TDNN) approach is used in the present work that enables analysis of long contextual information similar to RNNs but with training efforts comparable to conventional DNNs. The proposed ASC system attains a recognition accuracy of 76.5 % on the development set, which is 4.0 % higher compared to the DCASE'16 baseline system.

Index Terms— Time-delay neural networks, acoustic scene classification, DCASE, amplitude modulation filter bank features.

1. INTRODUCTION

Machine listening for automatic scene classification (ASC) becomes increasingly popular, e.g., as reflected by a past ASC challenge that compared research results of many international research teams [1]. Devices like hearing-aids, smart-phones, and robotic platforms are equipped with microphones and applications analyzing the acoustical environment, e.g., to allow for switching parameters of signal processing schemes [2,3]. Hence, in many situations it is of interest to know the environment in which an electronic device is used, e.g., to distinguish acoustic conditions of a conference room, cafeteria or subway. ASC algorithms aim at classifying the surrounding environment automatically by identifying acoustic events and sound characteristics that are specific for the environment. In contrast to acoustic event detection (AED) [4,5,6], individual events are of minor interest and since acoustic scenes do not change rapidly, constraints on temporal resolution for ASC are more relaxed than for AED and often comprise lengths of 30 seconds [1,7,8] up to 3 minutes [9].

Different approaches have been proposed for the purpose of automatic ASC such as the use of a bag-of frames approach [9], for which a Gaussian mixture model (GMM) in combination with Mel-frequency cepstral coefficients (MFCCs) are adopted. This approach has established itself in the field of scene classification and till today is still accepted as a reasonable baseline system for the DCASE challenges 2013 [1] and 2016 [7], though most of the systems in the DCASE'13 challenge could outperform the baseline results. Proposed features within DCASE'13 ranged from standard features such as MFCCs [10,11] and low-level features like energy, spectral flux etc. [12,13] over cochleograms [14] to histogram of gradients (HOG) features [8] and Gabor filter bank (GFB) features [15] that both have been derived from computer vision. Most back-end classifiers used for the DCASE'13 challenge were based on support vector machines (SVM) [16,12,14,8,11].

In a recent publication [17], the idea of using HOG features was revisited and improved by using them in conjunction with the subband power distribution (SPD). Other common approaches for ASC apply non-negative matrix factorization (NMF) to spectrograms to decompose features before classification [18,19].

In this contribution, we propose the use of amplitude modulation filter bank (AMFB) features [20] in combination with a neural network (NN) based classifier for the task of ASC. AMFB features analyze temporal amplitude fluctuations of static MFCCs within modulation frequency subbands. In combination with GMM and deep neural network (DNN) based systems, AMFB features have demonstrated to outperform numerous other common feature extraction methods in automatic speech recognition (ASR) [21,20,22]. In addition to AMFB features, spectral flux, spectral centroid, and spectral entropy features are calculated and appended.

DNNs are well established in, e.g., ASR [23,24] and have recently received increased attention also in the field of AED [25,26]. In ASR and AED, DNNs have proven to outperform conventional GMM-HMM approaches [27,25] and NMF-based features [26] under the constrained of availability of sufficient training data. Hence, DNNs may also be well suited for acoustic ASC, since ASC corpora mostly comprise several hours of data, e.g., the LITIS Rouen dataset [8] that comprises 25 hours of urban sound scenes, which is necessary to train a reasonable NN-based system.

Here, we report on our work on the DCASE'16 challenge and results are shown using a time-delay neural network (TDNN) architecture [28] that relies on AMFB features as an input for the

[1] This work was funded in parts by the European Commission (project EcoShopping, project no.609180) and the Federal Ministry for Education and Research (BMBF), project ACME 4.0, FKZ 16ES0469).

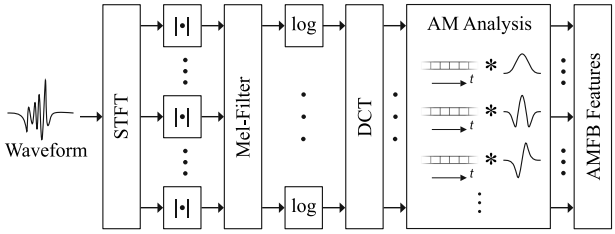


Fig. 1. Signal processing scheme to extract amplitude modulation filter bank features.

Task 1 of the DCASE'16 challenge, which comprises less than 10 hours of recordings [7]. Results are compared to the DCASE'16 baseline system that applies GMM acoustic models in combination with MFCC features.

2. METHODS

2.1. Extraction of Amplitude Modulation Filter Bank Features

The acoustic feature extraction scheme employs the amplitude modulation filter bank (AMFB) to decompose short-term spectral features into AM frequency components [20]. Signal processing steps are depicted in Fig. 1. The short-term spectral representation $Y_k(l)$ for block l is calculated by applying a discrete Fourier transform (DFT) on audio segments of 25 ms length with a hop size of 10 ms. Segments are windowed by the Hann function $w_b(n)$ to minimize the spectral leakage effect.

$$Y_k(l) = \sum_{n=-\infty}^{\infty} y(n) \cdot w_b(n-l) \cdot e^{-\frac{j2\pi kn}{N}} \quad ,0 \le k \le N-1 \tag{1}$$

$$w_b(n) = \begin{cases} 0.5 - 0.5 \cdot \cos\left(\frac{2\pi n}{b}\right) & ,0 \le n \le b \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In (1) and (2), n, k, b, and N represent the discrete time and frequency indices, the analysis window length, and the DFT length, respectively.

The magnitude of the complex valued spectrum $Y_k(l)$ is passed to the triangular-shaped Mel filters $F_{k,m}$ that integrate DFT bins into $M = 40$ critical spectral bands. Mel-spectral energies are compressed using a logarithmic function, whereby the log-Mel-spectrogram $\hat{Y}_m(l)$ is derived for each Mel band m.

$$\hat{Y}_m(l) = \log\left(\sum_{k=0}^{N-1} \left|Y_k(l)\right| \cdot F_{k,m}\right) \quad ,0 \le m \le M-1 \tag{3}$$

Log-Mel-spectral energies are analyzed by a discrete cosine transform (DCT), which leads to the cepstrogram $\tilde{Y}_c(l)$ with C being the DCT length.

$$\tilde{Y}_c(l) = \sum_{m=0}^{M-1} \hat{Y}_m(l) \cdot \cos\left(\frac{\pi}{M}\left(m+\frac{1}{2}\right)c\right) \quad ,0 \le c \le C-1 \tag{4}$$

Temporal dynamics of the cepstrogram are analyzed using the AMFB. The AMFB consists of I complex exponential functions $q_i(l_0)$, that are windowed by the zero-phase Hann envelope $W_i(l_0)$.

Table 1. Center frequency (CF) and bandwidth (BW) parameters of the amplitude modulation filter bank.

i	0	1	2	3	4
CF [Hz]	0	5.5	10.15	15.91	27.03
BW [Hz]	8.25	5.5	6.13	8.27	19.52

$$q_i(l_0) = e^{-j\Omega_i l_0 \cdot T} \cdot W_i(l_0) \quad ,0 \le i \le I-1 \tag{5}$$

$$W_i(l_0) = \begin{cases} 0.5 + 0.5\cos\left(\frac{2\pi l_0}{B_i}\right) & ,-\left\lceil\frac{B_i-1}{2}\right\rceil < l_0 < \left\lceil\frac{B_i-1}{2}\right\rceil \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$B_i = \frac{9.06}{2\pi \cdot \beta_i \cdot T} \tag{7}$$

B_i determines the AM filter length with the sampling period T. Ω_i and β_i are the angular AM frequency and the -3 dB AM filter bandwidth, respectively. Convolution of $q_i(l_0)$ and $\tilde{Y}_c(l_0)$ yields the AM frequency decomposition of the cepstrum.

$$Q_{c,i}(l) = \left(\tilde{Y}_c * q_i\right)(l) \tag{8}$$

Center frequency (CF) and bandwidth (BW) settings of the employed AM filters are presented in Table 1, which are derived by an ASR study on finding optimal AMFB parameters using different ASR corpora [22]. The last step of AMFB feature extraction is the concatenation of real and imaginary AM filter outputs to form a feature vector. Note that the imaginary part of the DC filter is zero, and thus is not taken into account.

2.2. Other Features

Spectral *flux*, spectral *centroid*, and spectral *entropy* features are derived according to Eq. 9-11 and appended to AMFB features.

$$Centroid(l) = \frac{\sum_{k=0}^{N-1}(k+1) \cdot \left|Y_k(l)\right|}{\sum_{k=0}^{N-1}\left|Y_k(l)\right|} \tag{9}$$

$$Flux(l) = \sum_{k=0}^{N-1}\left(\left|Y_k(l)\right| - \left|Y_k(l-1)\right|\right)^2 \tag{10}$$

$$Entropy(l) = -\sum_{k=0}^{N-1}\left|Y_k(l)\right|^2 \cdot \log_2\left(\left|Y_k(l)\right|^2\right) \tag{11}$$

These three feature types are used to measure the spectral "center of mass", the spectral "rate of change", and the spectral "complexity" [12,13].

2.3. Classification

Extracted features are fed into a time-delay neural network (TDNN) to extract further acoustic cues and to perform the classification task. The TDNN differs from a conventional DNN by the multi-splicing concept that enables an efficient way of modelling a large temporal context [28,29]. Multi-splicing denotes a method by which feature frames and intermediate DNN-layer

Table 2. Multi-splicing configuration of the TDNN system. Numbers in brackets indicate frame indices that are spliced together at each neural net layer.

NN-Layer	Input Context [Frames]
1	[-6 ,0 ,4]
2	[-12 ,0 ,12]
3	[-24 ,0 ,24]
4	[-50 ,0 ,50]
5	[0]

outputs are time-delayed and stacked to form the input to an upstream neural network (NN) layer. Splicing configurations per NN-layer are presented in Table 2. For example, the splicing notation [-6, 0, 4] in the first NN-layer denotes that the current frame minus six, the current frame itself, and the current frame plus 4 are spliced together by stacking input feature frames. We do not splice consecutive frames in the first layer, since AMFB features are used as input that already capture a temporal context of +/- 13 time frames and, thus, consecutive AMFB feature frames have highly overlapping filter functions and a high redundancy, respectively. The same principle applies to outputs of deeper NN-layers that capture an increasing temporal context due to the previous splicing stages. In total the TDNN captures feature frames ranging from -92 to +90, which corresponds with the feature frame rate of 100 Hz to a total temporal context of approx. 1.8 seconds.

The TDNN training is based on the greedy layer-wise supervised training [30] and the layer-wise backpropagation algorithm [27], respectively. As nonlinear activation units we are using the *p*-norm function that effect a dimension reduction of NN-layer outputs that each consist of 576 neurons in our setup. For example, for a group of G neurons x_i the p-norm output y is being computed by Eq. 12 with $p = 2$ and G = 6.

$$y = \|x\|_p = \left(\sum_{i=1}^{G} |x_i|^p \right)^{1/p} \tag{12}$$

Thus, the output of each NN-layer is reduced from 576 to 96. The final TDNN output layer has 15 neurons representing the 15 acoustic scenes that need to be discriminated.

3. EXPERIEMNTAL SETUP

For evaluating the algorithms, the database provided within the DCASE'16 challenge is used [7]. It consists of 15 scene classes: *lakeside beach*, *bus*, *cafe/restaurant*, *car*, *city center*, *forest path*, *grocery store*, *home*, *library*, *metro station*, *office*, *urban park*, *residential area*, *train*, and *tram*. Each scene is composed of 39 minutes of stereo recordings at 44.1 kHz sampling frequency that are trimmed to 30 second files. The data is divided into four disjoint sets to conduct a four-fold cross-validation, where all files belonging to one specific time/location are part of one set.

Evaluation is conducted file-wise applying the accuracy measure, i.e., the number of correctly classified files in ratio to the total number of files.

Table 3. Acoustic scene classification results of the DCASE'16 baseline system and the proposed TDNN-based system.

	Hit Rates [%]			
	Development (Cross-Validation)		Evaluation	
Environment	Baseline	Proposed Method	Baseline	Proposed Method
Beach	69.3	79.5	84.6	88.5
Bus	79.6	56.4	88.5	100.0
Café/Restaurant	83.2	44.9	69.2	19.2
Car	87.2	96.2	96.2	100.0
City Center	85.5	88.5	80.8	92.3
Forest Path	81.0	98.7	65.4	100.0
Grocery Store	65.0	87.2	88.5	88.5
Home	82.1	76.9	92.3	92.3
Library	50.4	69.2	26.9	38.5
Metro Station	94.7	79.5	100.0	80.8
Office	98.6	76.9	96.2	100.0
Park	13.9	56.4	53.8	61.5
Residential Area	77.7	88.5	88.5	76.9
Train	33.6	64.1	30.8	46.2
Tram	85.4	84.6	96.2	100.0
Average	72.5	76.5	77.2	79.0

4. RESULTS

In order to artificially augment the number of training frames the left and right channel of the stereo audio data is used in addition to the mean of both channels. In the testing phase the TDNN output for each of these three audio tracks is computed and the detected acoustic scene within an audio test file is based on a majority vote across frames and audio tracks. Note that prior to feature extraction we resampled data of the DCASE'16 challenge to 16 kHz.

Results of the proposed method and the DCASE'16 baseline system are presented in Table 3. On the cross-validation development set, the average improvement of the TDNN system amounts 4 % compared to the baseline. Particular strength can be noted for the environments *beach, car*, *forest path*, *grocery store*, *library*, *park*, *residential area*, and *train*. A decreased performance is found for the environments *bus*, *café/restaurant*, *home*, *metro station*, and *office*. Fig. 2 depicts the confusion matrix of the proposed classification system. It shows that some environments with relatively low recognition rates, i.e., *café/restaurant, bus*, *library*, *park*, and *train*, are mostly confused with similar or related environments such as *café*/*restaurant* > *grocery store*, *bus* > *tram/train*, *library* > *home*, *park* > *residential area*, and *train* > *tram/bus*.

Scene classification results of the evaluation test data are shown in Table 3. The average recognition score of the proposed TDNN system constitutes 79.0 %, which is 1.8 % higher compared to the baseline results. Whereas in most acoustic scenes the TDNN system scored significantly better or with comparable

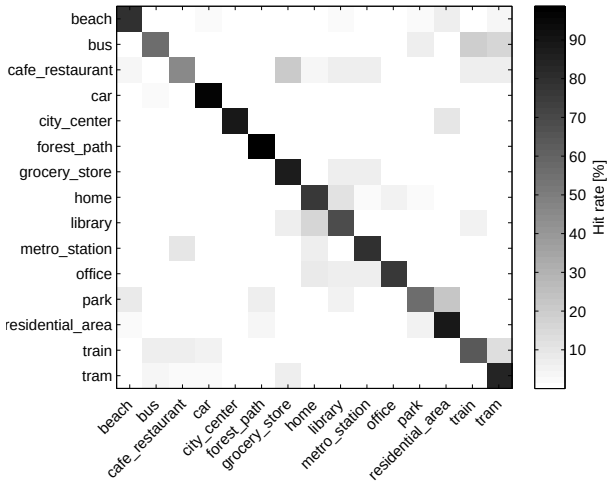


Fig. 2. Aggregate confusion matrix of the four-fold cross-validation results. Rows are ground truths and columns recognized scenes.

accuracy as the baseline system, classification results of the *café/restaurant* environment are clearly deteriorated. A closer investigation of why this acoustic scene has not been detected well enough is still pending. Possibly it has been confused with the *grocery store* (cf. Fig. 2), which exhibits similar acoustic conditions and events.

5. DISCUSSION AND CONCLUSIONS

A time-delay neural network (TDNN) based acoustic scene classification approach is proposed that employs the amplitude modulation filter bank (AMFB) as well as spectral flux, centroid, and entropy features. The system aims at analyzing a relatively long temporal context to identify the acoustic environments. It is shown that the AMFB-TDNN system improves over a MFCC-GMM baseline system by approximately 4.0 % and 1.8 % on the development and evaluation test data, respectively. Further improvements may be attained by additionally utilizing binaural cues of the stereo DCASE'16 data that is recorded using a manikin head with in-ear microphones and by emphasizing other features such as iVectors, for example.

6. REFERENCES

[1] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and D. Plumbley, "Detection and classification of audio scenes and events," *IEEE Transaction on Multimedia*, vol. 17, no. 10, pp. 1733-1746, 2015.

[2] J. Rennies, S. Goetze, and J. .-E. Appell, "Personalized Acoustic Interfaces for Human-Computer Interaction," in *Human-Centered Design of E-Health Technologies: Concepts, Methods and Applications*. IGI Global, 2011, ch. 8, pp. 180-207.

[3] B. Cauchi, S. Goetze, and S. Doclo, "Reduction of non-stationary noise for a robotic living assistant using sparse non-negative matrix factorization," in *Speech and Multimodal Interaction in Assistive Environments*, Jeju Island, 2012.

[4] D. Giannoulis, et al., "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, 2013.

[5] J. Schröder, et al., "On the use of spectro-temporal features for the IEEE AASP challenge 'detection and classification of acoustic scenes and events'," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, 2013.

[6] R. Stiefelhagen, et al., "The clear 2006 evaluation," in *Multimodal technologies for perception of humans*. Springer Berlin Heidelberg, 2007, pp. 1-44.

[7] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24th European Signal Processing Conference*, Budapest, 2016.

[8] A. Rakotomamonjy and G. Gasso, "Historgram of gradients of time-frequency representations for audio scene classification," *IEEE/ACM Transaction on Audio, Speech, and Signal Processing*, vol. 23, no. 1, pp. 142-153, 2015.

[9] J. .-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-grames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *Journal of the Acoustical Society of America*, vol. 122, no. 2, pp. 881-891, 2007.

[10] W. Nogueira, G. Roma, and P. Herrera, "Sound scene identification based on MFCC, binaural features and a support vector machine classifier," technical report, 2013.

[11] G. Roma, W. Nogueira, and P. Herrera, "Recurrence quantification analysis features for audiotory scene classification," technical report, 2013.

[12] J. T. Geiger, B. Schuller, and G. Rigoll, "Recognizing acoustic scenes with large-scale audio feature extraction and SVM," TUM, technical report, 2013.

[13] D. Li, J. Tam, and D. Toub, "Auditory scene classification using machine learning techniques," technical report, 2013.

[14] J. D. Krijnders and G. A. ten Holt, "A tone-fit feature representation for scene classification," technical report, 2013.

[15] J. Schröder, S. Goetze, and J. Anemüller, "Spectro-temporal Gabor filterbank features for acoustic event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 2198-2208, 2016.

[16] M. Chum, A. Habshush, A. Rahman, and C. Sang, "IEEE AASP scene classification challenge using hidden Markov models and frame based classification," technical report, 2013.

[17] V. Bisot, S. Essid, and G. Richard, "HOG and subband power distribution image features for acoustic scene classificantion," in *23rd European Signal Processing Conference*, Nice, 2015, pp. 2551-2555.

[18] B. Cauchi, "Non-negative matrix factorisation applied to auditory scenes classfication," M.S. thesis, ATIAM ParisTech, Paris, 2011.

[19] V. Bisot, R. Sterizal, S. Essid, and G. Richard, "Acoustic

scene classification with matrix factorization for unsupervised feature learning," in *International Conference on Acoustics, Speech, and Signal Processing*, Shanghai, 2016, pp. 6445-6449.

[20] N. Moritz, J. Anemüller, and B. Kollmeier, "An auditory inspired amplitude modulation filter bank for robust feature extraction in automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1926-1937, 2015.

[21] N. Moritz, et al., "A CHiME-3 challenge system: Long-term acoustic features for noise robust automatic speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Phoenix, 2015.

[22] N. Moritz, B. Kollmeier, and J. Anemüller, "Integration of optimized modulation filter sets into deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[23] G. Hinton, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.

[24] M. Seltzer, Y. Dong, and Y. Wang, "An investiagtion of deep neural networks for noise robust speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, 2013, pp. 7398-7402.

[25] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *22nd European Signal Processing Conference*, Lisbon, 2014, pp. 506-510.

[26] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Multi-label vs. combined single-label sound event detection with deep neural networks," in *23rd European Signal Processing Conference*, Nice, 2015, pp. 2551-2555.

[27] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Interspeech*, Florence, 2011, pp. 437-440.

[28] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transaction on Acoustics, Speech, and Language Processing*, vol. 37, no. 3, pp. 328-339, 1989.

[29] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of a long temporal contexts," in *Interspeech*, Dresden, 2015, pp. 2440-2444.

[30] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, vol. 19, Vancouver, 2007, pp. 153-160.

A REAL-TIME ENVIRONMENTAL SOUND RECOGNITION SYSTEM FOR THE ANDROID OS

Angelos Pillos †, *Khalid Alghamidi* †, *Noura Alzamel* †, *Veselin Pavlov* †, *Swetha Machanavajhala* ‡

† Computer Science Department, University College London, UK
angelos.pillos.15, khalid.alghamidi.15, noura.alzamel.15, veselin.pavlov.15@ucl.ac.uk
‡ Microsoft, Redmond, USA, swmachan@microsoft.com

ABSTRACT

Sounds around us convey the context of daily life activities. There are 360 million individuals [1] worldwide who experience some form of deafness. For them, missing these contexts such as fire alarm can not only be inconvenient but also life threatening. In this paper, we explore a combination of different audio feature extraction algorithms that would aid in increasing the accuracy of identifying environmental sounds and also reduce power consumption. We also design a simple approach that alleviates some of the privacy concerns, and evaluate the implemented real-time environmental sound recognition system on Android mobile devices. Our solution works in embedded mode where sound processing and recognition are performed directly on a mobile device in a way that conserves battery power. Sound signals were detected using standard deviation of normalized power sequences. Multiple feature extraction techniques like zero crossing rate, Mel-frequency cepstral coefficient (MFCC), spectral flatness, and spectral centroid were applied on the raw sound signal. Multi-layer perceptron classifier was used to identify the sound. Experimental results show improvement over state-of-the-art.

Index Terms— Environmental sound recognition, signal processing, machine learning, Android OS

1. INTRODUCTION

Understanding or recognizing context of sounds in environmental surroundings is very important in terms of making the next move based on a sound that occurred. Examples include evacuating a building on hearing a fire alarm or attending to a baby on hearing the cries. Such activities depend on human hearing which intelligently filters out sounds and quickly recognizes it thereby signaling the brain to take the next step. According to the World Health Organization [1], over 360 million individuals worldwide suffer from some form of disabling hearing loss. Deaf individuals can neither hear sounds nor distinguish between many sounds. Such situations could be life threatening at times and also inconvenient.

Providing access to sound in some other form of communication are in demand for its practical applications towards assisting deaf individuals in their daily activities. Capturing, detecting, identifying sounds and alerting users in the form of visual notifications, and vibrations is a concept known as Environmental sound recognition. Although auditory recognition is an active research area where extensive efforts have been made over fifty years, the focus has largely been on recognizing human speech or music, and much less efforts have been directed towards acoustic event recognition [2, 3]. Over the past decade, environmental sound recognition has become popular leading to a considerable amount of research.

Technology has progressed to a stage where a powerful computational device can be easily carried in your pocket. Therefore, one of the emerging trends is the increased demand for sound recognition applications to be available on these portable devices. Despite the existence of non-portable applications that are capable of acoustic environmental recognition, it decreases the practical application of being able to move anywhere thereby hindering the freedom of deaf individuals. Additionally, reproduction of non-portable systems is considered either not suitable or leads to low performance on mobile devices [4, 5].

In this paper, we carry out investigations, experiments and propose a simple but effective approach to recognizing everyday environmental sounds using mobile devices, in particular on the Android platform. Our goal is to notify the user about an acoustic event that just occurred in the users surroundings.

Our main contribution is to deliver a real-time sound recognition system on Android devices following several principles. First, the fundamental nature of our system is a sound processing system and its key performance metric is the relative accuracy of correctly recognized sounds. With systems which provide only conceptually similar features [6], we managed to get similar and even a higher level of accuracy. Second, we designed a battery friendly approach and as a result of various tests, our application managed to be approximately two times more battery efficient than other similar applications. Third, it was proved that mobile devices were hardware efficient by optimizing the computational resources. Fourth, our sound recognition system is network independent indicating that it will be always available. Fifth, the machine learning algorithm that we used for the application has the capability to learn new types of sounds and update the model on Android platform. Last, we provide a simple design that alleviates privacy concerns over audio.

The rest of this paper is organized as follows. Section 2 motivates the problem of classifying environmental sounds. Section 3 provides a system overview. Section 4 discusses the sound datasets used in experiments. Section 5 describes our approach in designing the sound recognition system and explains in detail about the system functionality such as sound detection, feature extraction and classification techniques. Section 6 provides an evaluation of approaches used in related work and our approach. Section 7 concludes the research and proposes future work.

2. ACOUSTIC EVENT RECOGNITION OVERVIEW

Natural sounds (dog barking, rain, rooster, sea waves) and artificial sounds (helicopter, siren etc.,) that might be heard in a given environment [2] are often referred to as acoustic events. Unlike music signals, the characteristics of environmental sounds do not

exhibit meaningful stationary patterns such as melody and rhythm [7]. Moreover, acoustic events differ from human speech because there is no existing sub-word dictionary for sounds in the same way that is possible to decompose words into their constituent phonemes [2]. Usually, environmental sound recognition systems have three main components: acoustic events detection as they occur, processing of these events in real-time by extracting useful information to create an acoustic feature for classification and then the determination of the most suitable category for that event, based on training carried out with similar samples [2].

3. SYSTEM OVERVIEW

Different mobile recognition system design choices were explored in prior related works [4, 8]. For example, feature extraction and sound recognition could be done directly on the phone and this mode is called embedded sound recognition. Another architecture is where both the feature extraction and recognition are done in cloud which is synoymous with a back-end server. [8]. In addition, we can have a combination of both approaches, distributed sound recognition, where the recognition is split across the mobile device and the server.

In our case, due to the high requirement of system availability we have decided to use the embedded mobile sound recognition architecture where both feature extraction and recognition are implemented on the mobile device alone. The main advantage of this mode is that the application will work in conditions where there is no Wi-Fi connection [4]. Mobile internet will not be sufficient since we are using 44100 Hz and 16 bits per sample. Mathematically, it is equivalent to 705600 bits per second which averages to 5 MB per minute and that requires a lot of internet bandwidth.

Our system consists of two main deliverables: A desktop application that considers a set of environmental sounds [10] as training data and extracts features, then trains the model to classify sounds based on the features and an Android application component that loads the machine learning model generated by the desktop application. The goal is to provide a single application that identifies multiple sounds such that the user can use the application like a baby monitor or a fire alarm alerter. To achieve this, the android application first detects the sound activity in the environment and captures the sound using a microphone, then extracts the features from the captured sound. Finally, it classifies the sound into its correspondent class using Multi-Layer Perceptron classifier.

4. SOUND DATASET

The dataset used in this paper is the ESC-10 [10] which represents three general groups of sounds. It includes transient/percussive sounds, sometimes with very meaningful temporal patterns (sneezing, dog barking, clock ticking) and sound events with strong harmonic content (crying baby, crowing rooster). Additionally, it also contains more or less structured noise/soundscapes (rain, sea waves, fire crackling, helicopter, chainsaw). This dataset was available under a Creative Commons non-commercial license through the Harvard Dataverse project [9]. Furthermore, the recordings are available in a unified format (5- second-long clips, sampled at 44.1kHz, 16 bits resolution, single channel) [10].

5. SYSTEM APPROACH

Most sound recognition systems follow either the Simple Classifier approach, Gaussian Mixture Model approach or HMM Model approach [11]. In our system we followed the approach with the lowest energy consumption and relatively high accuracy where the system detection component will continuously record the environmental sounds in chunks of one second each. Then, each captured sound signal will be processed by splitting it into frames (windows) of 1024 samples (around 25 millisecond (ms) considering the 44.1kHz sampling rate). In the case of training, splitting is done with an overlap of 50%, whereas in the testing there is no overlap in the splitting process. Each frame is smoothed with a Hanning window which is used with gated continuous signals and long transients to give them a slow onset and cutoff in order to reduce the generation of side lobes in their frequency spectrum.

Next, each frame is passed to the MFCC, ZCR, Spectral Centroid and Spectral Flatness feature extractors. Then, statistical values such as mean and standard deviation of the extracted features are calculated. The feature extraction stage plays an important part for the recognition process (explained in Subsection 5.2). At the end of this stage, there is a constant size feature vector for the whole one second sound signal. By analyzing each frame individually, the spectrogram type B solution technique is applied for selecting the features set [11]. We have considered this solution since from an implementation point of view, this technique does not need much memory [11]. Finally, the feature vector will be passed to the classification model to get the best match. The described recognition approach is shown in Fig. 1.

5.1. Sound Detection

The detection technique implemented in the system is based on the standard deviation of normalized power sequences. This method immediately signals a possible impulsive sound [11]. The signal is based on the normalization of successive windowed power sequences where the detection flag is signaled when the power exceeds the threshold value, which in our case is 0.015 [11] This technique aims to be simple yet an efficient detection scheme with a low computational load since it intends to be running all the time. Moreover, the performance turns out to be relatively the same as other more energy consuming advanced methods as described in [11].

5.2. Feature Extraction

As most recent research propose, feature analysis is considered the most crucial and important part in building a robust and effective recognition system [11, 12]. The aim of feature extraction is to convert the sound signal into a sequence of feature vectors in order to produce a set of characteristic features that describe the sound signal [13, 14].

From a physical point of view, signals can be represented in different domains: time or frequency domain. Therefore, acoustic features can be grouped into two groups, temporal and spectral features that can be extracted using time and frequency domain respectively [12]. Due to the nature of the environmental sound, which is considered as unstructured data and no assumptions can be made about detectable repetitions or harmonic structure in the signal, many features are needed to describe the audio signals [12].

In this work, several analysis methods were considered, inspired from the speech recognition community. As mentioned in

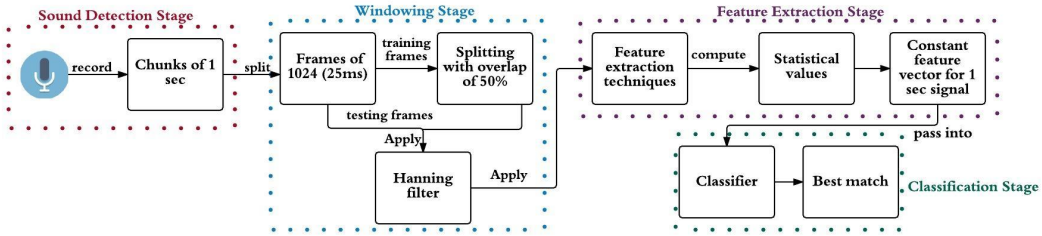


Figure 1: System Architecture.

Section 5, the feature extraction is applied on frame level. Time domain features such as Zero Crossing Rate, Spectral Centroid and Flatness were considered. Zero crossing rate (ZCR) is a measure of the number of times the signal value crosses the zero axe [14]. It is considered a very simple, yet useful feature. ZCR is defined formally as illustrated in (1)

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1}\prod S_t S_{t-1} < 0 \quad (1)$$

Where s is a signal of length T and the indicator function PA is 1 if its argument A is true and 0 otherwise. After the ZCR is applied for each frame of the 1 second signal, the mean and standard deviation were computed for all of the ZCR coefficients.

In order to obtain frequency domain features, we first applied the Fast Fourier Transform (FFT) algorithm to convert the sound signal from its original time domain into magnitude spectrum. Three types of features were extracted from each frame: Mel-frequency cepstral coefficient (MFCC), spectral centroid, and spectral flatness.

MFCC is the most common feature used in audio classification [15, 10, 11, 8]. The idea of the MFCC technique is to distribute the cepstral coefficients according to the critical bands, instead of the traditional linear distribution. Usually, it is mentioned as calculation of 13 Mel-frequency cepstral coefficients and discarding 0th order coefficient for each of the 25 ms frames. After that, the mean and standard deviation of each frame element are computed which results in 12 values representing means and 12 values representing standard deviation.

Another widely used feature is spectral centroid, which measures the brightness of a sound. Brightness is especially relevant for continuous instrumental sounds and is calculated as the amplitude-weighted average of all partials for the whole duration of the event [16]. In spectral centroid, the higher the centroid, the brighter the sound is [12]. It is calculated as the weighted mean of the frequencies present in the signal, determined using a Fourier transform, with their magnitudes as weights [14].

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \quad (2)$$

where $x(n)$ represents the weighted frequency value, or magnitude, of bin number n, and $f(n)$ represents the center frequency of that bin. Similarly, spectral flatness has been useful in audio signal processing which quantifies the tonal quality; namely, how much tone-like the sound is as opposed to being noise-like [12]. It is calculated by dividing the geometric mean of the power spectrum by the arithmetic mean of the power spectrum [14]:

$$Flatness = \frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\frac{\sum_{n=0}^{N-1} x(n)}{N}} = \frac{\exp(\frac{1}{N}\sum_{n=0}^{N-1}\ln x(n))}{\frac{1}{N}\sum_{n=0}^{N-1} x(n)} \quad (3)$$

where x(n) represents the magnitude of bin number n. Note that a single (or more) empty bin yields a flatness of 0, so this measure is most useful when bins are generally not empty.

As in MFCC, mean and standard deviation were calculated for spectral centroid. Moreover, min and max were calculated for spectral flatness. In total we obtained a vector of features with 30 elements which describe the signal to help the classifier provide the best match.

5.3. Classification

For the classification stage, we worked with the Weka Classification Library [17]. It is a collection of machine learning algorithms for data mining tasks which provides a number of classification models. We used an unofficial stripped down version of the library [18] which was compatible on the Android platform. The author of the project claims that this is the same Weka project with the GUI components removed so that it can work on Android.

As mentioned previously, a desktop application has been developed to extract the features from the datasets mentioned in Section 4 and then saves the training and test data into separate ARFF files (Attribute-Relation File Format) for training the model and then evaluating it. Another option is to save the extracted features from both datasets and then save them into one file for cross validation.

The extracted features were input to five classifiers: Multi-layer Perceptron, SVM (SMO in Weka), RandomForest, BayesNet and NaiveBayes. Weka uses Sequential Minimal Optimization (SMO) to solve the SVM training problem by using heuristics to partition the training problem into smaller problems that can be solved analytically. Training and testing were performed on ESC-10 dataset, using 2, 3, 4 and 5 fold cross validation. The results for the 10 class datasets are shown in figure 2.

5.4. Design to Alleviate Privacy Concerns

Gray [19] gives a nice explanation on privacy implications of having always-on microphone enabled devices. We utilized some of the design approaches in our application to alleviate privacy concerns. For

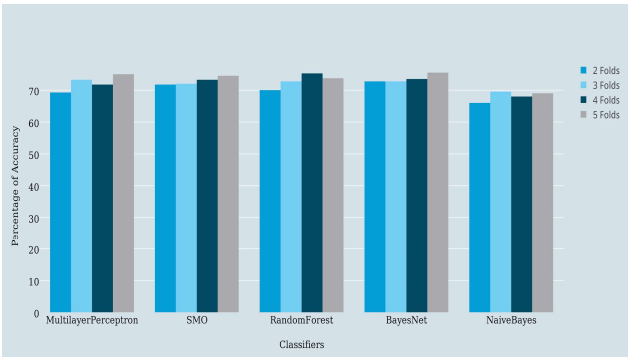


Figure 2: Cross Validation of 10 class dataset.

Class-Fold	10 Classes
2 folds	69.25%
3 folds	73.25%
4 folds	71.75%
5 folds	74.5%

Table 1: Multi-layer Perceptron Cross Validation Results

	ESC-10	
	Random forest	Multi-Layer Perceptron
Proposed system	73.75%	74.50%
ESC paper results [10]	73.70%	62.50%

Table 2: Comparison with previous research studies

example, the user would know when the application is listening to a sound, when it has detected a sound and finally when it's classifying the sound; all of which are provided by flashing lights. Additionally, we also give the user an option to turn the sound recognition on or off. This makes it clear that user has given consent for the application to be always-on listening.

6. EVALUATION

We evaluate our proposed system's accuracy by comparing the results obtained using a combination of feature extraction algorithms and Multi-Layer perceptron classifier with the results obtained from previous studies [10]. The same dataset [10] was used in the evaluation.

Additionally, one of the main factors of a successful application is the battery consumption. We evaluate the battery efficiency of our proposed application with another similar application.

6.1. Multi-layer Perceptron Accuracy

Fig. 2 lists the accuracy of each classifier on 10 classes dataset. As a result of training the models, Multi-layer Perceptron was chosen as it performed with the best overall accuracy with 74.5% for the 10 class dataset and also due to the ability to derive meaningful patterns from unstructured data.

We perform multiple versions of cross validation to ensure comparability with ESC paper results [10]. Though a large number of folds is more expensive, it does however give a less biased estimate of the model performance. Additionally, the variance of the resulting estimate for different samples or partitions of data to form training and test sets is reduced as the number of folds are increased. As a result, cross validating the classifiers with more folds provide more accurate results. Table (1) shows an example of this.

6.2. Accuracy Comparision

The results of the proposed system were compared to results in the research paper considering environmental sound recognition [9] using Random Forest and Multi-layer perceptron classifiers. The results were acquired by using 5-fold cross validation. For the purpose of objective results the same dataset was used: ESC-10 dataset [10]. From Table (2), we can see that our system performs with a better accuracy using Multi-Layer Perceptron classifier.

6.3. Battery Consumption

We have compared battery consumption of our system with Otosense [6] which is also an environmental sound recognition system. Battery consumption was tested by measuring the mobile device runtime for 5 repetitions. The tests were held by using two smartphones: LG Nexus 5 and LG Nexus 4. During all of the tests the mobile devices were initially fully charged, the connection to internet was switched off and no background processes were stopped. The runtime was measured until the mobile devices switched off. Table (3) shows the average runtime for the both systems using the mobile devices stated above. Both of the mobile devices perform almost twice better when running our proposed approach.

7. CONCLUSION

In this paper, a viable real-time environmental sound recognition system for Android mobile devices was developed. The system uses a sound detection algorithm which helps reduce power consumption. In addition, a combination of several feature extracting algorithms were applied to accurately identify sounds. Finally, several classifiers were compared with described features and Multi-Layer Perceptron classifer performed best with 74.5% accuracy on 10-class dataset. The recognition accuracy of the proposed system exceeds the results of previous efforts using the same dataset [10].

The current dataset was chosen specifically for benchmarking purposes with prior papers. Looking into the future we propose to filter out sounds, just as fire crackling, clock-tickling, which are not really useful for people who are deaf or hard of hearing.

	LG Nexus 5	LG Nexus 4
Proposed System	5 hours 27 minutes	7 hours 15 minutes
Otosense [6]	3 hours 4 minutes	3 hours 50 minutes

Table 3: Average runtime duration

8. REFERENCES

[1] "World Health Organization", http://www.who.int/en/ [accessed on: 30 July 2016].

[2] J. Dennis, "Sound Event Recognition And Classification In Unstructured Environments", PhD thesis, Nanyang Technological University, 2011.

[3] R. Goldhor, "Recognition of environmental sounds", in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference*, vol. 1, pp. 149–152, 1993.

[4] A. Kumar, A. Tewari, S. Horrigan, M. Kam, F. Metze and J. Canny, "Rethinking speech recognition on mobile devices", in *Proceedings of 2nd International Workshop on Intelligent User Interfaces for Developing Regions*, Palo Alto, CA, pp. 10-15, February 2011.

[5] A. Schmitt, D. Zaykovskiy and W. Minker, "Speech recognition for mobile devices", in *International Journal of Speech Technology 11*, no. 2, pp. 6372, 2008.

[6] "Otosense", http://www.otosense.com/ [accessed on: 30 July 2016].

[7] N. Scaringella, G. Zoia and D. Mlynek, "Automatic genre classification of music content: a survey", in *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 133141, March 2006. doi: 10.1109/MSP.2006.1598089.

[8] R. Mattia, S. Feese, O. Amft, N. Braune, S. Martis and G. Troster, "AmbientSense: A realtime ambient sound recognition system for smartphones", in *Pervasive Computing and Communications Workshops (PERCOM Workshops)* 2013 IEEE International Conference on. IEEE, 2013.

[9] "The Dataverse project", http://dataverse.org/ [accessed on: 30 July 2016].

[10] K. Piczak,"ESC: Dataset for Environmental Sound Classification", in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 10151018, ACM, 2015.

[11] A. Dufaux, "Detection And Recognition Of Impulsive Sounds Signals", PhD thesis, University of Neuchtel, 2001.

[12] S. Chu, S. Narayanan and C. Kuo, "Environmental Sound Recognition With TimeFrequency Audio Features", in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 11421158, Aug. 2009. doi: 10.1109/TASL.2009.2017438.

[13] M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition", in *Pattern recognition letters*, vol. 24, no.15, pp. 2895-2907, 2003.

[14] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project", Technical report, IRCAM, 2004.

[15] X. Zhang and Y. Li, "Environmental Sound Recognition Using DoubleLevel Energy Detection", in *Journal of Signal and Information Processing*, Vol. 4 No. 3B, pp. 1924, 2013. doi: 10.4236/jsip.2013.43B004.

[16] D. Keller and J. Berger, "Everyday sounds: synthesis parameters and perceptual correlates", in *Proceedings of the VIII Brazilian Symposium of Computer Music*, 2001.

[17] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java", http://www.cs.waikato.ac.nz/ml/weka/ [accessed on: 30 July 2016].

[18] "Weka for Android", https://github.com/rjmarsan/WekaforAndroid [accessed on: 30 July 2016].

[19] S. Gray, "Always On: Privacy Implications of Microphone-Enabled Devices", in *Future of privacy forum*, April 2016.

PERFORMANCE COMPARISON OF GMM, HMM AND DNN BASED APPROACHES FOR ACOUSTIC EVENT DETECTION WITHIN TASK 3 OF THE DCASE 2016 CHALLENGE

Jens Schröder[1,3]*, *Jörn Anemüller*[2,3], *Stefan Goetze*[1,3]

[1] Fraunhofer Institute for Digital Media Technology IDMT, Oldenburg, Germany
[2] University of Oldenburg, Department of Medical Physics and Acoustics, Oldenburg, Germany
[3] Cluster of Excellence, Hearing4all, Germany
jens.schroeder@idmt.fraunhofer.de

ABSTRACT

This contribution reports on the performance of systems for polyphonic acoustic event detection (AED) compared within the framework of the "detection and classification of acoustic scenes and events 2016" (DCASE'16) challenge. State-of-the-art Gaussian mixture model (GMM) and GMM-hidden Markov model (HMM) approaches are applied using Mel-frequency cepstral coefficients (MFCCs) and Gabor filterbank (GFB) features and a non-negative matrix factorization (NMF) based system. Furthermore, tandem and hybrid deep neural network (DNN)-HMM systems are adopted. All HMM systems that usually are of single label type, i.e., systems that only output one label per time segment from a set of possible classes, are extended to multi label classification systems that are a compound of single binary classifiers classifying between target and non-target classes and, thus, are capable of multi labeling. These systems are evaluated for the data of residential areas of Task 3 from the DCASE'16 challenge. It is shown that the DNN based system performs worse than the traditional systems for this task. Best results are achieved using GFB features in combination with a single label GMM-HMM approach.

***Index Terms*—** acoustic event detection, DCASE'16, Gabor filterbank, deep neural network

1. INTRODUCTION

Acoustic event detection (AED) denotes the automatic identification of sound events in audio signals. Commonly, the acoustic event's category as well as its time of occurrence are to be recognized. Application fields for AED are e.g., surveillance of public spaces for security issues [1–3], monitoring of health states e.g. in care systems [4–6] or condition monitoring of technical systems [7, 8].

AED in monophonic environments, i.e., for settings in which only single, isolated acoustic sources are active for a given time interval, has been the main focus of research in the past, with prominent comparisons of competitive systems in, e.g., the "classification of events, activities and relationships" (CLEAR'07) and "detection and classification of acoustic scenes and events 2013" (DCASE'13) challenges. Established methods for detecting acoustic events in monophonic environments are often based on Mel-frequency cepstral coefficient (MFCC) features and hidden Markov models (HMMs) using Gaussian mixture models (GMMs) as observation probability functions (GMM-HMM) [9–11]. These systems are denoted as single label classification systems since for a certain time segment they select one and only one label from a set of pre-trained classes based on maximum likelihood criteria or comparable scores. However, in many realistic environments rarely only a single source is active per time instance. Instead, usually multiple sources emit sound waves simultaneously leading to a mixed sound signal at a receiver. This case of multiple and overlapping sound signals is commonly referred to as polyphony. For acoustic event detection systems this case is by far more challenging than the monophonic case, not only because of the pure signal mixture of an unknown number of acoustic events present in the signal but also because training and test data can be considerably different due to the vast number of possibilities of event mixtures. Recently, polyphonic acoustic event detection has gained considerable attention, e.g., by being addressed in the DCASE'16 challenge. Some approaches for polyphonic event detection are based on MFCC and GMM-HMM classifiers. Using these back-ends, either binary classification between target events and universal background model is performed [12] or classification on multiple streams separated by non-negative matrix factorization (NMF) is conducted [11]. Further approaches apply NMF as part of feature extraction by thresholding the activations of the source code book [13, 14]. In recent publications, deep neural networks (DNNs) are used [3, 15, 16]. The output of the DNNs replaces the NMF-features, while the classification continues to rely on thresholded feature values. In the field of automatic speech recognition (ASR), DNNs are well-established and constitute the state-of-the-art baseline. Incorporation into recognition systems is based on two paradigms, tandem and hybrid approaches [17]. For the tandem approach, DNN features replace the MFCC features while the back-end is a conventional GMM-HMM classifier. Commonly, bottleneck features are used, for which one layer of the DNN acts as "bottleneck" with only a small number of neurons compared to the preceding and subsequent layers [17]. The hybrid approach uses DNNs as observation functions replacing the GMMs leading to DNN-HMM back-ends. They can be used with any kind of features. A common observation with DNNs is that they need more training data than for example GMM-HMM systems with MFCCs features.

This paper describes the authors contribution to the DCASE'16 challenge. It focuses on the subtask of Task 3 containing acoustic data recorded in residential environments (cf. Section 2). We investigate the performance of GMM-HMM systems using MFCCs and Gabor filterbank (GFB) features, the best scoring system of the DCASE'13 challenge, as well as NMF. Furthermore, we examine

*This work was funded in parts by the European Commission (project EcoShopping, (no. 609180) and the Federal Ministry for Education and Research (BMBF), project ACME 4.0, FKZ 16ES0469)

Table 1: Event statistics of Task 3 for the residential area. Given are the number of events ('num. ev.'), the average duration ('av. dur.') and the total duration ('tot. dur.') of each class individually and overall as mean and standard deviation.

	num. ev.	av. dur. [s]	tot. dur. [s]
bird singing	130	7.55 ± 25.19	981.21
car passing by	57	9.16 ± 4.81	521.94
children shouting	23	2.00 ± 1.68	46.16
object banging	15	0.76 ± 0.70	11.33
people speaking	40	8.08 ± 24.42	323.08
people walking	32	5.50 ± 5.94	176.11
wind blowing	22	6.09 ± 5.98	133.96
overall	319	6.88 ± 18.59	2193.79

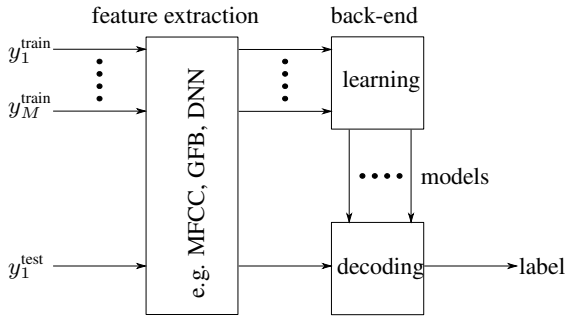


Figure 1: General schematic of the applied classification systems.

the performance of DNN tandem and hybrid approaches. Single label and multi label classification systems are used.

The remaining of this paper is structured as follows. The experimental setup including the dataset 'residential area' of Task 3 from the DCASE'16 challenge is outlined in Section 2. The concept of single label and multi label systems is explained in Section 3. The individual classification systems are detailed in Section 4. The results for these systems are shown in Section 5. Conclusions are drawn in Section 6.

2. EXPERIMENTAL SETUP

The following experiments are based on the setup and data of Task 3 of the DCASE'16 challenge [18]. Task 3, called 'Sound event detection in real life audio', consists of stereo data recorded at 44.1 kHz and in a home environment and in a residential area. Only the first channel is used in our contribution. The dataset of the home environment comprises eleven classes of a total duration of 36 min whilst the dataset of the residential area is a compound of seven classes and a total duration of 42 min. Since these are relatively few data especially for training of DNNs, we will just show results for the larger subset 'residential area'. Details of this subset are given in Table 1. The proposed four cross-validation sets from the challenge are used as well as the evaluation measures F-Score and the acoustic event error rate (AEER) [18]. The F-Score F represents the relation between the precision P and the recall R, i.e.,

$$P = \frac{N_{\text{corr}}}{N_{\text{est}}}; \quad R = \frac{N_{\text{corr}}}{N_{\text{ref}}}; \quad F = \frac{2 \cdot P \cdot R}{P + R}, \tag{1}$$

where N_{corr} denotes the number of correct hits, N_{est} the number of estimated events and N_{ref} the number of reference events. The AEER is the sum of insertions I, deletions D and substitutions S relative to the number of reference events N_{ref}, i.e.

$$\text{AEER} = \frac{I + D + S}{N_{\text{ref}}}. \tag{2}$$

Both measures are applied on 1 sec segments and averaged over all crossvalidation folds.

3. SINGLE AND MULTI LABEL SYSTEMS

For detecting events, two main classification systems will be tested: Single label classification and multi label classification systems. A single label classification system consists of multiple models for different classes. The model yielding highest probability for a time segment is selected as label. Thus, such approaches are not capable of detecting simultaneous or overlapping events. Commonly, HMM systems are single label classification systems. To overcome this disadvantage and get multiple labels per time segment, the single label systems can be extended to multi label classification systems. A single binary classifier consists of a target class model and a garbage or background model that covers all non-target classes. Hence, a compound of such binary classifiers in a classification system is able to label each time segment with multiple labels.

4. CLASSIFICATION SYSTEMS

The commonly applied classification systems consist of a feature extraction step and a back-end (cf. Figure 1). In the training phase, the extracted features, e.g. MFCCs, GFB features, DNN features etc., are used to create class models for the back-end that can be, e.g., HMMs. In the testing phase, these models are applied to the extracted features of the test data to decode it and output labels for time segments. The adopted systems of this contribution will be detailed in the following.

4.1. Baseline System

As baseline we use the provided baseline system from Task 3 of the DCASE'16 challenge [18]. It is composed of a GMM model using MFCCs. The MFCC features use 40 ms windows with 50% shift. The first 19 coefficients and the 0th energy coefficient plus derivations of first (Δ) and second order ($\Delta\Delta$) are used, that are computed over 9 time frames. The GMM is based on 16 Gaussian mixtures per class model and is applied on sliding windows of 1 second. The baseline is just applied in a binary classification system.

4.2. NMF System

The NMF system is based on the baseline system of Task 2 of the DCASE'16 challenge. It uses variable Q-transform (VQT) spectrograms of 60 bins per octave and a step size of 10 ms. The NMF codebook consists of 20 spectral templates per class that are learned during a training phase. For the original baseline, the 20 spectral templates were generated by averaging the delivered 20 event files

Table 2: Results of Task 3 for the residential area. In each row, the performance of the respective system is given in terms of AEER and F-Score. Both measures are divided into the total average and the class-wise average. A check mark in column 'multi label' indicates that the system is capable of making multi label outputs, e.g. the binary systems, otherwise systems produce single label output. For DNN features, the underlying features are given in brackets. Note: The baseline system uses other parameters for MFCCs than the other MFCC based systems depicted in rows 3, 4, 7, 8, 10 and 11. Best scores are highlighted by bold numbers.

no.	multi-label	back-end	feature	AEER		F-Score [%]	
				total	class	total	class
1	√	baseline	MFCC	0.86	1.16	34.6	19.9
2	√	NMF	VQT	1.35	2.11	14.8	8.7
3		GMM-HMM	MFCC	0.77	1.02	41.2	15.8
4	√	GMM-HMM	MFCC	0.81	1.08	41.0	16.6
5		DNN-HMM	log-Mel	1.02	3.47	17.2	8.5
6	√	DNN-HMM	log-Mel	1.78	5.87	16.0	13.3
7		DNN-HMM	MFCC	1.04	3.75	10.7	7.9
8	√	DNN-HMM	MFCC	1.22	2.86	22.6	14.7
9	√	GMM-HMM	DNN(log-Mel)	2.17	6.23	28.6	19.6
10	√	GMM-HMM	DNN(MFCC)	1.87	6.08	30.8	18.8
11	√	GMM-HMM	MFCC+DNN(log-Mel)	2.37	5.89	32.4	**24.1**
12		GMM-HMM	GFB	**0.74**	**1.01**	**48.5**	19.2
13	√	GMM-HMM	GFB	0.93	1.44	44.2	17.6

per class. Hence, the codebook size depended on the amount of files. To avoid the dependency on the dataset size, we modified the training phase by applying a GMM with 20 mixture components to the complete spectrogram data of each class to create the desired number of spectral templates. Based on these templates, data is decoded by a NMF. The NMF output is postprocessed using a threshold (1.0), a minimum event length of 60 ms and a maximum number of concurrent events (5).

4.3. DNN-HMM Hybrid System

For the DNN-HMM hybrid system, the commonly applied GMM observation function of an HMM is replaced by a DNN. The HMM for each class is modeled by one transition state, i.e., it is actually a GMM. Viterbi-decoding is applied with multiple, unlimited number of repetitions of events per file to get time segment labels. The input layer consists of the current time frame plus 4 frames before and after, thus, extending the feature dimensionality by a factor of 9. Several different combinations of number of layers (2,3,4), number of neurons per layer (20, 32, 39, 64, 128, 256) and characteristics like a bottleneck have been tested. Here, only the results of the DNN yielding best performance using three hidden layers with 128, 20, and 39 neurons will be shown. The hidden layers use the rectified linear unit (ReLU) as activation function, whilst the output function applies the softmax function.

Two types of features are investigated. One feature type is based on static MFCCs, i.e., a window length of 25 ms and 10 ms shift is used to compute the twelve first coefficients as well as the 0th. The other feature type is a logarithmic Mel (log-Mel)-spectrogam with 40 frequency bins (window length of 25 ms and shift of 10 ms).

4.4. GMM-HMM Sytem

The GMM-HMM systems use GMMs as observation functions for HMMs. The HMM of each class is modeled by one transition state. The best number of mixtures is evaluated on the validation fold, i.e., the performance of the mixture yielding the best total performance will be shown. In contrast to the baseline, the decoding is done using Viterbi-decoding with multiple repetitions of events per file.

Several different features are used for this system. Basic MFCCs as for the DNN-HMM hybrid system (cf. Section 4.3) with additional Δ and $\Delta\Delta$ features. Another feature type is based on the GFB. The GMM-HMM(GFB) system [19] achieved highest performance on the previous DCASE'13 challenge [20]. Here we use the GFB optimized for AED that has been shown to improve the results for the acdcase2013 challenge [21].

Furthermore, features are derived from DNNs, thus building a tandem system. Therefore, the DNNs of the hybrid systems are applied, and, hence, are either based on MFCCs or on the log-Mel-spectrogram. The hybrid DNNs are modified by deleting the output layer that represents the class probabilities, and replacing it by the second last layer containing 39 neurons. Furthermore, the activation function ReLU is replaced by a linear activation function to produce features with better discriminative abilities [22]. We used HTK [22] to adapt HMMs and DNNs.

5. RESULTS

The results of the tested systems are given in Table 2. The AEER and the F-Score are shown. They are divided into a total average over all frames and into a class-wise average, i.e., the score for each class is computed and the average of these numbers are depicted. Hence, effects on scores resulting from different amount of data per class are avoided. Each row describes a system. A check mark (√) in column 'multi label' indicates that a system has multi label output, which are the baseline system, the NMF system and the multi label versions of the HMM approaches. No check mark indicates that a system has single label output, which are the standard HMM versions.

It can be seen that the NMF based system (cf. Row 2), which is the baseline system of Task 2 of the DCASE'16 challenge, performs relatively poorly compared to the GMM(MFCC) baseline (Row 1). This might result from the polyphonic training data. Commonly, the training data for NMF approaches consist of isolated events.

However, the data for Task 3 was polyphonic. Thus, a proper codebook is unlikely to be generated leading to much confusion between classes. Another reason for the inaccuracy of the approach might be that beeing the baseline system of Task 2, the used parameters for the classifier may not be optimal for Task 3.

The GMM-HMM-systems using MFCCs (cf. Rows 3 and 4) achieve better results for the AEER and for the total F-Score. For the class-wise F-Score, they are slightly worse. This is a result of the unequal amount of data per class. Both GMM-HMM-systems are particularly good in recognition of the classes with most data 'bird singing' and 'car passing by'. For class 'bird singing', the single label GMM-HMM-systems (cf. Row 3) yields a class-wise F-Score of 56.3% whereas the baseline yields F-Score of 35.5%. This imbalance leads to a better total F-Score for the GMM-HMM-systems but a worse class-wise F-Score than for the baseline.

Against expectation, the single label approach of the GMM-HMM-system yields better performance than the multi label approach, though the single label approach is not capable of detecting multiple overlapping events and, thus, in contrast to the binary approach, by its nature can never yield 100% accuracy. For the applied dataset, it seems to be beneficial to just output one label with maximum likelihood than to try to detect multiple concurrent events.

However, for the DNN-HMM hybrid systems (cf. Rows 5 to 8), the binary versions yield better F-Scores than the single label systems. In comparison to the baseline, they perform worse for all shown measures. The tandem systems (cf. Rows 9 to 11) yield worse AEER scores. However, the F-Scores are relatively high. For the system with concatenated MFCC and DNN features (cf. Row 11), even the highest class-wise F-Score of all examined systems is achieved. The reason for the low AEER but high F-Score lies in the high number of label outputs that are generated by the tandem system. It causes many errors but also a high recall R forcing a relative high F-Score.

The best scores except for the class-wise F-Score are achieved by the single label GMM-HMM with GFB features (cf. Row 12). Especially the total F-Score is much higher than for all other tested systems. Similar to the GMM-HMM-systems using MFCCs (cf. Row 4), the multi label version of the GMM-HMM system adopting GFB features (cf. Row 13) performs less well than the single label version.

6. CONCLUSIONS

This study reports system performances for different acoustic event detection strategies applied to Task 3 ('residential area' data) of the DCASE'16 challenge. We compared commonly used GMM systems to tandem and hybrid DNN systems. Single and multi label systems were applied. We showed that for this task, DNNs are less accurate than GMM-HMM systems. Probably, the amount of data available for Task 3 is too low to train DNNs properly. The GMM-HMM system in combination with GFB features which was developed for the DCASE'13 challenge [19] for isolated events and that was meanwhile improved as described in [21], yields best performance of all tested systems. Furthermore, a single label system that is only able to output one label per time segment seems not to be inferior to a multi label classification system for polyphonic data.

A drawback for all classification systems is the little amount of available data within this challenge. As it could be observed, the two classes with most data, which are 'bird singing' and 'car passing by', achieved best recognition results for nearly all classification systems. Testing the approaches on larger corpora is thus subject of future work.

7. REFERENCES

[1] C. Clavel, T. Ehrette, and G. Richard, "Events detection for an audio-based surveillance system," in *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME)*, Amsterdam, The Netherlands, Jul. 2005, pp. 1306–1309.

[2] J. Schröder, S. Goetze, V. Grützmacher, and J. Anemüller, "Automatic acoustic siren detection in traffic noise by part-based models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 493–497.

[3] P. Laffitte, D. Sodoyer, C. Tatkeu, and L. Girin, "Deep neural networks for automatic detection of screams and shouted speech in subway trains," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6460–6464.

[4] S. Päßler and W. Fischer, "Food intake monitoring: Automated chew event detection in chewing sounds," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 278–289, 2014.

[5] J. Schröder, J. Anemüller, and S. Goetze, "Classification of human cough signals using spectro-temporal Gabor filterbank features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6455–6459.

[6] S. Matos, S. S. Birring, I. D. Pavord, and D. H. Evans, "Detection of cough signals in continuous audio recordings using hidden Markov models," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1078–1083, 2006.

[7] J. Schröder, M. Brandes, D. Hollosi, J. Wellmann, M. Wittorf, O. Jung, V. Grützmacher, and S. Goetze, "Foreign object detection in tires by acoustic event detection," in *DAGA 2015*, Nuremberg, Germany, Mar. 2015, pp. 1266–1269.

[8] N. K. Verma, R. K. Sevakula, S. Dixit, and A. Salour, "Intelligent condition based monitoring using acoustic signals for air compressors," *IEEE Trans. Reliability*, vol. 65, no. 1, pp. 291–309, 2016.

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

[10] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real-life recordings," in *18th European Signal Processing Conference (EUSIPCO 2010)*, Aalborg, Denmark, Aug. 2010, pp. 1267–1271.

[11] T. Heittola, A. Mesaros, A. J. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal of Audio, Speech and Music Processing*, vol. 2013, p. 1, 2013.

[12] J. Schröder, F. X. Nsabimana, J. Rennies, D. Hollosi, and S. Goetze, "Automatic detection of relevant acoustic events in kindergarten noisy environments," in *DAGA 2015*, Nuremberg, Germany, Mar. 2015, pp. 1525–1528.

[13] O. Dikmen and A. Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2013.

[14] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, Apr. 2015, pp. 151–155.

[15] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Multi-label vs. combined single-label sound event detection with deep neural networks," in *23rd European Signal Processing Conference, EUSIPCO 2015*, Nice, France, Aug. 2015, pp. 2551–2555.

[16] A. Diment, E. Cakir, T. Heittola, and T. Virtanen, "Automatic recognition of environmental sound events using all-pole group delay features," in *23rd European Signal Processing Conference, EUSIPCO*, Nice, France, Aug. 2015, pp. 729–733.

[17] Z. Tüske, R. Schlüter, H. Ney, and M. Sundermeyer, "Context-dependent MLPs for LVCSR: TANDEM, hybrid or both?" in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association*, Portland, Oregon, USA, Sep. 2012, pp. 18–21.

[18] A. Mesaros, T. Heittola, , and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24rd European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, Sep. 2016, p. ?, *accepted*.

[19] J. Schröder, N. Moritz, M. R. Schädler, B. Cauchi, K. Adiloglu, J. Anemüller, S. Doclo, B. Kollmeier, and S. Goetze, "On the use of spectro-temporal features for the IEEE AASP challenge 'detection and classification of acoustic scenes and events'," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2013.

[20] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2013.

[21] J. Schröder, S. Goetze, and J. Anemüller, "Spectro-temporal Gabor filterbank features for acoustic event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 12, pp. 2198–2208, Dec. 2015.

[22] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, A. R. V. Valtchev, P. Woodland, and C. Zhang, *The HTK Book (for HTK Version 3.5alpha)*, 2015.

ACOUSTIC SCENE CLASSIFICATION: AN EVALUATION OF AN EXTREMELY COMPACT FEATURE REPRESENTATION

*Gustavo Sena Mafra**

Universidade Federal de Santa Catarina
Florianópolis, Santa Catarina, Brazil
gsenamafra@gmail.com

Ngoc Q. K. Duong†, Alexey Ozerov, Patrick Pérez

Technicolor
975 Avenue des Champs Blancs CS 17616
35576 Cesson-Sévigné, France
firstname.lastname@technicolor.com

ABSTRACT

This paper investigates several approaches to address the acoustic scene classification (ASC) task. We start from low-level feature representation for segmented audio frames and investigate different time granularity for feature aggregation. We study the use of support vector machine (SVM), as a well-known classifier, together with two popular neural network (NN) architectures, namely multilayer perceptron (MLP) and convolutional neural network (CNN). We evaluate the performance of these approaches on benchmark datasets provided from the 2013 and 2016 Detection and Classification of Acoustic Scenes and Events (DCASE) challenges. We observe that a simple approach exploiting averaged Mel-log-spectrograms and SVM can obtain even better results than NN-based approaches and comparable performance with the best systems in the DCASE 2013 challenge.

Index Terms— Acoustic scene classification, Audio features, Multilayer Perceptron, Convolutional Neural Network, Support Vector Machine.

1. INTRODUCTION

Acoustic scene classification (ASC), a particular form of audio classification, consists in using acoustic information (audio signals) to infer the context of the recorded environment [1]. Examples of such environments are bus, office, street, etc... It offers a wide range of applications in connected home, *e.g.* expensive video cameras can be replaced by cheap microphones for monitoring daily activity, and for smartphones, *e.g.* they could automatically switch to silence mode during a meeting or automatically increase the sound volume in a noisy environment. However, real-life ASC is not a trivial task as recognising a greater variety of sounds in both indoor and outdoor environments would require a new set of strategies and adjustments of existing machine learning techniques to make the most out of the available data.

While speaker identification [2], speech recognition [3], and some audio classification tasks in music information retrieval such as music genre recognition [4, 5] or music instrument recognition [5] have been studied for a long time, the real-life ASC task has become active quite recently in the research community. While the classification task itself has been studied since at least 2002 [6], it was only recently that efforts were made to provide a benchmark for the task, with the new initiative of the DCASE challenges in 2013 and 2016. Various techniques have been proposed to tackle the problem with the use of different acoustic features (*e.g.* cochleogram representation, wavelets, auditory-motivated representation, features learned by neural networks) and different classifiers (*e.g.* Support Vector Machine (SVM), Gaussian Mixture Model (GMM), Hidden Markov Model (HMM)) [7]. One of the most popular approaches, known as bag-of-frames (BOF) approach [8, 9] is used as a baseline in the DCASE challenge, and exploits the long-term statistical distribution (by GMM) of the short-term MFCCs.

Besides the DCASE challenge, nonnegative matrix factorization (NMF) was recently exploited for sound event detection in real life recordings [10]; recurrent neural networks (RNN) were investigated for polyphonic sound event detection in real life recordings [11]; and deep neural networks (DNN) have been developed for sensing acoustic environment [12]. It would be interesting to note that while DNNs [13, 14] were recently applied with great success to many different audio, visual and multimedia tasks, it was less investigated within the DCASE 2013 challenge and one of the reasons would be the lack of a substantial amount of labeled data for training.

This paper aims to study the use of well-established low-level acoustic feature representations and different machine learning techniques, including DNN-based methods and SVM, for the ASC task. While most existing approaches extract an acoustic feature vector for each short-term audio frame, then perform a frame-based classification based either on BOF over GMMs [8, 9] or simple majority voting [15, 16, 7], we investigate the use of an another feature representation, *i.e.* a single vector for a whole audio scene, aiming an extremely compact representation that greatly reduces the computational cost for the whole ASC system, since the number of examples to be used to train the classifier is drastically reduced. We evaluate the use of this compact feature with SVM and MLP on both DCASE 2013 and DCASE 2016 datasets and the performance are more or less equivalent to a frame-based approach with majority voting strategy. Furthermore, it results in classification accuracy comparable to the best systems participating in the DCASE 2013 challenge.

The rest of the paper is organized as follows. In Section 2 we present the general framework which involves different approaches for feature extraction and classification. Experiment results on DCASE dataset obtained by our approaches and some state-of-the-art methods are discussed in Section 3. We finally conclude in Section 4.

*Part of this work has been done while the first author was with Technicolor.

†Email:quang-khanh-ngoc.duong@technicolor.com

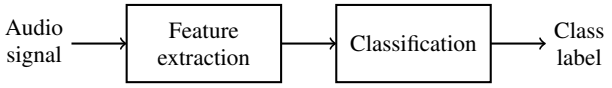


Figure 1: General workflow of the acoustic scene classification framework.

2. ACOUSTIC SCENE CLASSIFICATION FRAMEWORK

The general workflow of an ASC system is usually divided into two major steps as shown in Fig. 1. In the first, the feature extraction step, various types of hand-crafted representations have been considered in the literature such as chroma, pitch, spectrograms, zero-crossing rate, and linear predictive coding coefficients [1]. Among them, features based on Mel-frequency Cepstrum Coefficients (MFCCs) computed for each short-time frame are arguably the most common one, as it can be seen by the DCASE 2013 Challenge, where out of 12 systems submitted, at least 7 involved MFCCs [1]. More recent DNN-based approaches usually attempt to learn higher level features from these low-level signal representations [12, 17]. In the classification step, popular classifiers include SVM and GMM [7]. In the following, we will first describe the standard Mel-log-spectrogram, as the low-level feature used in this work, and the proposed compact representation from it in Section 2.1. We then briefly present some exploited classification approaches in Section 2.2. The choice of hyperparameters for both feature extraction and classifiers is discussed in Section 2.3.

2.1. Feature extraction

The time domain audio signal $x(n)$ is first transformed into the frequency domain by means of the short-term Fourier transform (STFT) as

$$\mathrm{STFT}\{x\}(m,\omega) = \sum_{n=-\infty}^{+\infty} x(n)w(n-mL)e^{-j\omega n} \quad (1)$$

where $w(n)$ is a window function (which is Hanning window in our implementation), m denotes frame index and L the frame shift. The spectrogram is then defined as

$$\mathbf{S}(m,\omega) = |\,\mathrm{STFT}\{x\}(m,\omega)|^2 \quad (2)$$

In our DNN-based system, we use spectrogram with a logarithmic amplitude scale (named log-spectrogram) as the frame input feature which is computed as

$$\mathbf{F}_{\text{Log-spec}}(m,\omega) = \log(\mathbf{S}(m,\omega)). \quad (3)$$

In our other systems, we first map the spectrogram $\mathbf{S}(m,\omega)$ into the auditory-motivated Mel frequency scale - denoted by $\mathbf{MS}(m,\omega)$, then transform it into logarithmic scale as

$$\mathbf{F}_{\text{Mel-log-spec}}(m,\omega) = \log(\mathbf{MS}(m,\omega)). \quad (4)$$

Note that with the CNN-based system, we use the raw log-spectrogram as the input feature in order to give flexibility for the CNN to learn a higher level feature representation optimized for the ASC task. For SVM-based systems we have tested four different features: spectrogram, log-spectrogram, Mel-log-spectrogram, and MFCC, and found that the two last ones result in a very similar ASC performance that outperforms the two first ones. As the Mel-log-spectrogram is simpler to compute than the MFCC, we use it as the main acoustic feature in this paper. Finally, we propose to average the feature vectors for all frames so as to present a whole audio example by an extremely compact feature vector whose entries are computed as

$$\mathbf{f}_{Avg-mel-log-spec}(\omega) = \frac{1}{M}\sum_{m=1}^{M} \mathbf{F}_{\text{Mel-log-spec}}(m,\omega). \quad (5)$$

This type of averaging of features is very straight-forward and has already been used in past works [18][19] in ASC. However, no submitted systems in the DCASE 2013 Challenge made use of it, misleadingly pointing to a lack of efficiency of this method.

2.2. Classification approaches

2.2.1. Support vector machine

SVM has been known as one of the most popular classifiers for many different tasks. It was also widely used in the DCASE 2013 challenge [7]. In our work, we used SVM as a benchmark classifier to evaluate the effectiveness of different features, as mentioned in Section 2.1, as well as to obtain the optimal choice of hyperparameters (*e.g.* the window size and the number of Mel-frequency coefficients) for the considered task.

In our implementation, we train SVMs using a coordinate descent algorithm [20] and following a one-vs-the-rest scheme to perform classification of multiple classes [21]. We have tested SVM with linear kernel and Gaussian radial basis function (RBF) kernel and found that the linear kernel works slightly better than RBF kernel for the DCASE 2013 dataset.

2.2.2. Multilayer Perceptron

Multilayer Perceptron (MLP) is a fully connected feedforward artificial neural network architecture that maps sets of input data onto a set of appropriate outputs. It can be seen as a logistic regression classifier where the input is first transformed using a non-linear transformation [22, 23]. A typical set of equations for an MLP is the following. Layer k computes an output vector $\mathbf{h}^k$ using the output $\mathbf{h}^{k-1}$ of the previous layer, starting with the input $\mathbf{x} = \mathbf{h}^0$,

$$\mathbf{h}^k = f(\mathbf{W}^k\mathbf{h}^{k-1} + \mathbf{b}^k) \quad (6)$$

where $\mathbf{b}^k$ denotes a vector of offsets (or biases) and $\mathbf{W}^k$ a matrix of weights. The function f is called the activation function and it is applied element-wise. Common options for it are sigmoid function, hyperbolic tangent, and rectified linear unit (ReLU). The latter, *i.e.* $f(x) = \max(0, x)$, was used to obtain the results reported in this document.

The top layer output is used for making a prediction and is combined with the groundtruth label into a loss function. We use softmax as the classification layer and the log-likelihood loss function regularized with ℓ_1 and ℓ_2 penalties. This cost function is then optimized using mini-batch stochastic gradient descent (SGD) with an adaptive learning rate [24] and dropout is performed between the hidden layers [25].

2.2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of neural network designed to exploit the redundancy and correlation between neighbour units. It has gained great success in different fields such as image and video recognition, natural language processing, speech recognition, etc., [14]. This motivates us to investigate the use of CNN for the ASC task in this work.

We trained CNNs over the log-spectrogram of the signals with a structure of vertical filters, *i.e.* the frequency bins can also be interpreted as a CNN channel (as in RGB channels for images) instead of a dimension and the convolution is ran over the time axis. This type of structure was proposed for music recommendation in Spotify[1] and is justifiable by the fact that an audio "pattern" detected in a high-frequency region is usually different from that same pattern in a low-frequency region. Thus it is desirable to model the vertical filters to extract more meaningful information from the spectral representation. More details about the implemented CNN architecture can be found in Section 3.1.

2.3. Hyperparameter optimization

The choice of hyperparameters in each step of the ASC system or in any machine learning task can significantly affect the final classification result. Such hyperparameters are *e.g.* the window length and hop length in the STFT computation for feature extraction, the regularization parameter for SVM, the number of hidden units in an MLP, and the step size for the SGD algorithm in DNN based methods. The conventional strategy of tuning these parameters manually would not be feasible as it requires a great number of trials so that all parameters can be optimized together. Thus, in this work we incorporate a Bayesian optimization [27] method to find these parameters altogether. The algorithm models the generalization accuracy of a classifier as a function of the corresponding parameters, and finds the optimal parameters that maximize the expected accuracy given the observed dataset.

In our implementation, we use Hyperopt [28], a Python library for optimizing hyperparameters in machine learning algorithms, with the Tree of Parzen Estimators (TPE) [29], an algorithm that falls into the class of sequential model-based optimization (SMBO) [30] algorithms. The TPE algorithm performs cross-validation with the development datasets of DCASE 2013 and DCASE 2016 and finds an optimal set of hyperparameter values. It is interesting to note that the optimal window size for STFT computation found by the TPE algorithm is quite long, *i.e.* about half of a second. This can be explained by the fact that the acoustic events are more spread in time compared to *e.g.* speech which is very localized so as the window length used for STFT is usually much smaller.

3. EXPERIMENTS

We evaluate the ASC performance of our four implementing systems with the benchmark DCASE 2013 dataset, which allows to compare with the state-of-the-art approaches participating in the challenge, in Section 3.1. We then present the result with DCASE 2016 dataset in Section 3.2. Our first system (named *Proposed SVM-A*) uses an extremely compact feature as the Mel-log-spectrogram coefficients averaged for all frames, and SVM with a linear kernel as classifier. The second system (named *Proposed SVM-V*) performs frame classification by SVM with a linear kernel, then majority voting in the end. The third system (named *Proposed MLP*) takes the compact averaged Mel-log-spectrogram as input, learns an intermediate feature representation by MLP, then classifies by softmax as the last layer of the MLP. The fourth system (named *Proposed CNN*) takes log-spectrogram as low-level input feature, learns higher feature representations by CNN layers, then classifies by softmax.

[1] http://benanne.github.io/2014/08/05/spotify-cnns.html

3.1. Results with the DCASE 2013 dataset

The DCASE 2013 dataset consists of 30-second audio segments belonging to 10 classes. Each class has 10 segments in the development set and 10 other examples in the test set [31].

The ASC performance was evaluated in terms of the classification accuracy, averaged over all classes, and shown in Table 2. Note that as we did not have access to the groundtruth labels of the test set at the time of these experiments, we evaluated our systems averaging with the standard 5-fold cross-validation on the development set only, while results for most other approaches in the table are obtained with the test set [7]. Some hyperparameters for each systems were found by the Bayesian optimization method presented in Section 2.3. More detailed settings for each system are as follows. The window length for the STFT was set by 0.57 seconds and 0.41 seconds for the SVM-A and SVM-V system, respectively, the number of Mel-frequency coefficients is about 1900, the regularization parameter C in SVM for SVM-A and SVM-V were 0.98 and 0.62, respectively. MLP had one hidden layer with 677 units, dropout rate and learning rate for parameter training was set by 0.08 and 0.011, respectively. CNN had 3 convolutional layers, the number of filters for each layer are 50, 29, and 19, respectively, and the max-pooling ratios between layers are 3, 4, and 3.

As it can be seen, the two systems based on SVMs outperform the ones based on NNs. This can be explained by the fact that the dataset may be not large enough for training DNNs directly. Three of our proposed systems (SVM-A, SVM-V, and MLP) achieve comparable performance with some of the best performing approaches in the DCASE 2013 Challenge - as we suppose that there is not much difference between development set and the test set. Moreover, we achieve higher accuracy than Li *et al.* [16] in the same development set. Finally, we note that the proposed feature, which is extremely compact so as to represent a whole 30-second audio segment by just a single vector, can be sufficient for the classification as the SVM-A and MLP obtained 75% and 72% accuracy, respectively.

3.2. Results with the DCASE 2016 dataset

The DCASE 2016 dataset is structured in a similar way as the DCASE 2013 dataset. However the number of acoustic classes is extended to 15, and the number of examples for each class is significantly enlarged to 78 for the development set and 26 for the test set.

The results for development set obtained by our four systems are shown in Table 2, where the best performance of 80% is achieved by the SVM-A system with a window length of 0.42 seconds and a hop size of 0.14 seconds for the STFT computation. This result confirms again the benefit of using the proposed compact feature representation and the use of a long window for the spectral transformation. The MLP, which obtains similar performance as the baseline, had two hidden layers with 66 and 199 units, respectively, SGD was used for parameter training with learning rate of

Method	Acoustic feature	Classifier	Accuracy
Baseline	MFCC	”bag-of-frames” GMM	55%
Geiger *et al.* [15]	Diverse features	SVM + majority voting	69%
Roma *et al.* [26]	MFCC with Recurrence Quantification Analysis	SVM	76%
Proposed SVM-A	**Averaged Mel-log-spectrogram**	**Linear SVM**	**75%**
Proposed SVM-V	**Frame Mel-log-spectrogram**	**Linear SVM + majority voting**	**78%**
Proposed MLP	Averaged Mel-log-spectrogram	MLP with softmax as classification layer	72%
Proposed CNN	Log-spectrogram	CNN with softmax as classification layer	62%

Table 1: Acoustic scene classification results with DCASE 2013 test dataset (for state-of-the-art approaches) and development dataset (for our proposed approaches). Note that other submitting systems resulting in less classification accuracy are not mentioned in the table.

Method	Acoustic feature	Classifier	Accuracy
Baseline	MFCC	”bag-of-frames” GMM	75%
Proposed SVM-A	**Averaged Mel-log-spectrogram**	**Linear SVM**	**80%**
Proposed SVM-V	**Frame Mel-log-spectrogram**	**Linear SVM + majority voting**	**78%**
Proposed MLP	Averaged Mel-log-spectrogram	MLP with softmax as classification layer	75%
Proposed CNN	Log-spectrogram	CNN with softmax as classification layer	59%

Table 2: Acoustic scene classification results with DCASE 2016 development dataset.

	Beach	Bus	Cafe/restaurant	Car	City center	Forest path	Grocery store	Home	Library	Metro station	Office	Park	Residential area	Train	Tram
Beach	47	0	3	4	5	4	1	0	1	0	3	4	2	0	2
Bus	2	61	0	4	1	0	0	0	0	2	0	2	2	3	1
Cafe/restaurant	4	0	46	0	0	2	19	4	0	0	0	2	0	0	1
Car	0	2	0	71	0	0	0	0	0	0	0	0	1	0	4
City center	0	0	0	0	75	0	0	0	0	2	0	0	1	0	0
Forest path	1	0	0	0	0	75	0	0	0	0	0	0	2	0	0
Grocery store	0	0	0	0	2	0	76	0	0	0	0	0	0	0	0
Home	6	0	1	0	0	13	1	30	7	0	3	6	9	1	1
Library	0	1	0	0	0	0	0	0	72	1	0	0	0	4	0
Metro station	0	0	0	0	0	0	0	0	0	76	0	2	0	0	0
Office	0	0	0	0	0	0	0	8	1	0	69	0	0	0	0
Park	9	0	5	0	2	1	0	1	0	0	0	49	11	0	0
Residential area	5	0	0	0	9	7	0	0	2	0	0	19	35	0	1
Train	17	3	0	0	2	0	0	0	1	0	0	1	0	49	5
Tram	2	0	0	0	0	0	0	0	0	2	0	0	0	0	74

Figure 2: Confusion matrix of the SVM-A method in the development set of the DCASE 2016 database after a 4-fold cross-validation over 78 samples of each class.

0.003 and batch size of 100, weights for ℓ_1 and ℓ_2 penalties were 10^{-5} and 10^{-4}, respectively. The CNN, with the same configuration used for the DCASE 2013 dataset, still resulted in the lowest performance. These four systems will also be tested with the test set for participating in the DCASE 2016 challenge.

The confusion matrix for SVM-A is shown in Fig. 2, where rows are groundtruth, columns are the inferred class label, and values are number of the classified acoustic scene. As it can be seen, some environments containing a specific type of noise (such as car, metro station, forest path) are quite easy to recognize, while some others (such as home, residential area, park) are quite confusing.

4. CONCLUSION

In this article we present several approaches for the ASC task, targeting on fast systems working with very compact feature representations so that ASC can be implemented *e.g.* in smartphones. We investigate the use of Bayesian optimization for hyperparameter optimization and find its benefit in *e.g.* choosing the optimal window length for STFT or setting DNN parameters. By evaluating on benchmark DCASE datasets, we find that (1) a long window size for spectral transformation is more relevant for the environmental acoustic scenes, (2) a very compact feature representation by long-term temporal averaging of Mel-log-spectrogam coefficients would be sufficient for the task compared to more complicated approaches, and (3) similar accuracies are found for the two datasets, possibly owing to the similarities between these datasets or to the robustness of the proposed systems. Finally, it is worth noting that DNN approaches have not reached the same performance of the more classical SVM based systems so far. Thus future work would be devoted to investigate transfer learning strategies for DNN based systems where part of the DNN can be initially learned by a large amount of external audio data.

5. REFERENCES

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

[2] D. Reynolds, R. C. Rose, *et al.*, “Robust text-independent speaker identification using gaussian mixture speaker models,” *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[3] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Prentice Hall, 1993.

[4] T. Li, M. Ogihara, and Q. Li, “A comparative study on content-based music genre classification,” in *Proceedings of the 26th*

annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003, pp. 282–289.

[5] B. Kostek, A. Kupryjanow, P. Zwan, W. Jiang, Z. W. Raś, M. Wojnarski, and J. Swietlicka, "Report of the ISMIR 2011 contest: music information retrieval," in *Foundations of Intelligent Systems*. Springer Berlin Heidelberg, 2011, pp. 715–724.

[6] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, "Computational auditory scene recognition," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 2. IEEE, 2002, pp. II–1941.

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

[8] 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, no. 2, pp. 881–891, 2007.

[9] M. Lagrange, G. Lafay, B. Defreville, and J.-J. Aucouturier, "The bag-of-frames approach: a not so sufficient model for urban soundscapes," *The Journal of the Acoustical Society of America*, vol. 138, no. 5, pp. EL487–EL492, 2015.

[10] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 151–155.

[11] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6440–6444.

[12] N. D. Lane, P. Georgiev, and L. Qendro, "Deepear: Robust smartphone audio sensing in unconstrained acoustic environments using deep learning," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '15. ACM, 2015, pp. 283–294.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[16] 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.

[17] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1798–1828, 2013.

[18] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1015–1018.

[19] 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, 2014, pp. 1041–1044.

[20] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 408–415.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, p. 3, 1988.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[26] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat, "Recurrence quantification analysis features for auditory scene classification," *IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events, Tech. Rep*, 2013.

[27] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[28] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*. Citeseer, 2013, pp. 13–20.

[29] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 2546–2554.

[30] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *International Conference on Learning and Intelligent Optimization*. Springer, 2011, pp. 507–523.

[31] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

COUPLED SPARSE NMF VS. RANDOM FOREST CLASSIFICATION FOR REAL LIFE ACOUSTIC EVENT DETECTION

Iwona Sobieraj *Mark D. Plumbley*

i.sobieraj@surrey.ac.uk m.plumbley@surrey.ac.uk

University of Surrey
Centre for Vision Speech and Signal Processing
Guildford, Surrey GU2 7XH, United Kingdom

ABSTRACT

In this paper, we propose two methods for polyphonic Acoustic Event Detection (AED) in real life environments. The first method is based on Coupled Sparse Non-negative Matrix Factorization (CSNMF) of spectral representations and their corresponding class activity annotations. The second method is based on Multi-class Random Forest (MRF) classification of time-frequency patches. We compare the performance of the two methods on a recently published dataset TUT Sound Events 2016 containing data from home and residential area environments. Both methods show comparable performance to the baseline system proposed for DCASE 2016 Challenge on the development dataset with MRF outperforming the baseline on the evaluation dataset.

***Index Terms*—** Acoustic event detection, random forest classifier, non-negative matrix factorization, sparse representation

1. INTRODUCTION

Acoustic Event Detection (AED) is an important task in machine listening. It aims to automatically recognise, label, and estimate the position in time in a continuous audio signal of meaningful sounds, referred to as acoustic events. There exists a number of real-world applications for AED such as home-care [1], surveillance [2], multimedia retrieval [3], urban traffic control [4], to name just a few. The task of AED can be broadly classified into *monophonic* and *polyphonic* detection. Monophonic detection, which has been the major area of research in this field, aims to recognize only one prominent event at a time [5, 6, 7]. However, in real-life environments multiple events occur at the same time making the task challenging. Polyphonic detection aims to identify these several overlapping events at the same time.

Several solutions have been proposed for polyphonic AED. Some approaches were strongly inspired by speech recognition systems, using mel frequency cepstral coefficients (MFCCs) with Gaussian Mixture Models (GMMs) combined with Hidden Markov Models (HMM) [8, 9]. Another popular technique for AED is matrix factorization of time-frequency spectra, especially Non-negative Matrix Factorization (NMF) [10]. NMF has been used to extract dictionaries for each acoustic event class in a supervised manner using isolated sounds [11, 12]. This approach served as a baseline for the Event Detection - Office Synthetic subtask of DCASE2013 Challenge [13]. In the same challenge, the best performance on polyphonic AED was achieved by examplar-based NMF decomposition followed by HMM postprocessing [14]. Another system used NMF to separate sound into different tracks and then detected events in each track separately assuming prior knowledge of the number of overlapping sources [15]. Probabilistic Latent Component Analysis (PLCA), the probabilistic counterpart of NMF, was also used for polyphonic AED using isolated sounds as training data [16].

Recently, several approaches for polyphonic AED that learn models directly from the mixture of sounds have been proposed. NMF applied to annotated overlapping events was used directly on the mixture of sounds, without the need to learn from isolated samples [17, 18]. Feed-forward deep neural networks (FNNs) trained on mixtures of sounds for multi-label AED achieved better performance than NMF [19]. Recurrent neural networks (RNNs) using bidirectional long short-term memory (BLSTM), which can directly model the sequential information of audio, were reported to outperform FNNs on the same dataset [20]. Despite their successes DNNs have several drawbacks. They are computationally complex and rely on huge amounts of data, hence data augmentation is often necessary to achieve better results [20]. Moreover, it is often difficult to interpret, what features does a DNN learn. On the contrary, methods such as NMF or multi-class classification are less computationally complex, and, in the case of NMF, may offer interpretable dictionaries.

In this paper, we propose two methods for polyphonic AED. The first method is inspired by promising results of coupled matrix factorization in [18]. We explore this idea by modifying the learning algorithm to explicitly sparse NMF. The second method is based on a multi-class random forest classification [21]. The random forest classifier has proved efficient for environmental sound classification and for monophonic AED [22, 6]. Therefore, we explore its application to polyphonic AED. As feature input for both methods we chose 2D time-frequency patches, which have proven effective for standard NMF [7]. We investigate the influence of the size of the patch. We compare our results on the TUT Sound Events 2016 introduced for the DCASE2016 Challenge [23].

This paper is organised as follows: Section 2 presents the details of training and testing procedures of the two methods. The experimental setup and the results of the evaluation are shown in Section 3. Section 4 contains the discussion of the results and conclusions. Conclusions and future work are presented in Section 5.

The research leading to these results has received funding from the European Union's H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement no 642685 MacSeNet. MDP is also partly supported by EPSRC grant EP/NO14111/1

2. METHOD

We propose two methods for polyphonic AED and compare them on the development set of Task 3 of the DCASE2016 Challenge. The first method is based on sparse dictionary learning using non-negative matrix factorization (NMF) on 2D spectral patches coupled with class annotations. The second is based on a simple multi class random forest classification of 2D spectral patches. Both methods classify directly the mixture of events instead of building separate models for each sound event. The output of each method per frame is directly a set of multiple labels class activity. In order to make a fair comparison between the methods, we use the same preprocessing and post processing for both approaches. Both methods are implemented using *python*. For audio processing we used *librosa* [24] and for machine learning *scikit-learn* [25] libraries.

2.1. Preprocessing

We pre-process the data to reduce the dimensionality but at the same time remain meaningful representation appropriate for environmental sounds. Therefore, as a feature representation we choose to use the perceptually motivated mel scale [26]. We extract mel-spectrograms with 40 components, using a window size of 23 ms, hop size of the same duration and sampling frequency of 44.1 kHz. In order to model temporal dynamics of environmental sounds we choose a spectro-temporal representation of the data, which is achieved by grouping several consecutive frames into 2D spectral patches, also known as *shingling*, which has proven to be a discriminative feature for environmental audio classification [22]. We investigate the size of patches as it may differ depending on the characteristics of the sounds that we are trying to detect. Finally, the patches are normalised to account for intensity level difference among different occurrences of the events.

2.2. Coupled Sparse Non-negative Matrix Factorization

Coupled Sparse Non-negative Matrix Factorization (CSNMF) is inspired by the approach by Dikmen et al. [17]. The system presented by the authors uses coupled matrix factorization to learn dictionaries based on spectral representation of signals and the corresponding labels. In our method, we use sparse NMF to learn the coupled dictionaries in an analogical way.

The aim of a standard NMF is to find a low-rank representation of a matrix $\mathbf{V}$ by approximating it as a product of a non-negative dictionary $\mathbf{W}$ and its non-negative activation matrix $\mathbf{H}$, so that:

$$\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH} \tag{1}$$

where $\mathbf{V} \in R_+^{F \times N}$, $\mathbf{W} \in R_+^{F \times K}$ and $\mathbf{H} \in R_+^{K \times N}$.

In a coupled matrix factorization problem we are given two different matrices $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$, which we want to decompose into non-negative dictionaries $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ which share a common activation matrix $\mathbf{H}$. For polyphonic AED, $\mathbf{V}^{(1)}$ is a spectral representation of the signal of size $F \times N$, $\mathbf{V}^{(2)}$ is a binary matrix of class activation of size $E \times N$. F is the number of frequency bins, N number of frames and E number of classes of events. The dictionaries $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are found by minimizing the following objective function:

$$\eta_1 D^{(1)}(\mathbf{V}^{(1)}||\mathbf{W}^{(1)}\mathbf{H}) + \eta_2 D^{(2)}(\mathbf{V}^{(2)}||\mathbf{W}^{(2)}\mathbf{H}) + \lambda||\mathbf{H}||_1 \tag{2}$$

where $D(\mathbf{V}||\hat{\mathbf{V}})$ is chosen to be Kullbeck-Leibler (KL) divergence between the data V and the approximation $\hat{\mathbf{V}}$ and λ is a regularization parameter that penalizes over the l_1-norm, which induces sparsity on activation matrix $\mathbf{H}$. To facilitate computation the weighting parameters are chosen to be equal, i.e $\eta_1 = \eta_2 = 1$. We choose 60 bases for NMF, the number selected empirically.

The authors of [17] used an estimator based on maximum marginal likelihood (MMLE), which was shown to return sparse solutions. In order to investigate the influence of sparsity on the performance of the algorithm, we introduce the sparse regularisation explicitly and minimise the objective function in (2) using the multiplicative update rule for NMF with a sparsity constraint λ on the activation matrix [10, 27]:

$$\begin{aligned} \mathbf{W} &\leftarrow \mathbf{W} \odot \frac{\mathbf{V}\mathbf{H}^T}{1 \cdot \mathbf{H}^T} \\ \mathbf{H} &\leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T \cdot 1 + \lambda} \end{aligned} \tag{3}$$

where $\mathbf{V}$ is a concatenation of $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$, $\mathbf{W}$ is a concatenation of $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$, $\mathbf{H}$ the common activation matrix, λ the sparsity regularizer and 1 is a matrix of ones of the size of V. $A \odot B$ denotes a Hadamard product of two matrices, A/B - Hadamard division and other multiplications are matrix multiplications.

Having learnt the dictionaries $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$, in the testing phase we obtain an activation matrix $\mathbf{H}_{test}$ based on spectral representation of the test data, $\mathbf{V}^{(1)}$, and its dictionary $\mathbf{W}^{(1)}$. Next, we obtain the class activity matrix, $\mathbf{V}^{(2)}$, by multiplying the activation matrix $\mathbf{H}_{test}$ with the label dictionary $\mathbf{W}^{(2)}$. More details can be found in [17]. The estimated class activity matrix needs to be binarised using an arbitrary threshold to show the presence or absence of the sound.

2.3. Multi-class random forest classification

The multi-class random forest classification (MRF) method is based on classification of spectro-temporal patches using random forest classifier of 500 trees. This combination of features and classifier has proved to perform well on environmental sound classification task [22]. As the authors of [22] classify single events only, we modify the method to perform polyphonic AED. For E events in the dataset, we model all their possible combinations, i.e. we construct $M = 2^E$ classes. That means, that the set M of possible classes is a Cartesian product of $\{0, 1\}^E$. We classify each patch as belonging to one of the M product classes and concatenate all the estimates to form a class activation matrix. We can only model combinations of sounds already seen in the training set. Hence, any new combination of audio events will not be recognised correctly.

2.4. Postprocessing

We post-process the annotation matrices obtained by both methods using the baseline approach for DCASE2016, i.e. discarding events shorter than 100 ms and removing gaps shorter than 100 ms between the events.

3. EXPERIMENTS

3.1. Experimental setup and metrics

The two methods, CNMF and MRF, are tested on the TUT Sound Events 2016 dataset provided as a development set for Task 3 of

DCASE2016 Challenge [23]. The dataset consists of two everyday environments: one outdoor environment which is residential area with 7 classes, and one indoor environment, home with 11 classes. Table 1 shows the list of classes and the number of instances for both acoustic scenes. We can clearly see that the dataset is unbalanced, especially the residential area acoustic scene. Two classes, namely "bird singing" and "cars passing by", account for 74% of all the class instances. The home acoustic scene dataset is more balanced, but the two most appearing classes, i.e. "dishes", and "object impact", account for 42% of all instances.

Residential area		**Home**	
Event class	instances	Event class	instances
object (banging)	23	(object) rustling	60
bird singing	271	(object) snapping	57
car passing by	108	cupboard	40
children shouting	31	cutlery	76
people speaking	52	dishes	151
people walking	44	drawer	51
wind blowing	30	glass jingling	36
		object impact	250
		people walking	54
		washing dishes	84
		water tap running	47

Table 1: TUT Sound Events 2016: event classes and number of instances

As a metric for evaluating the methods we chose the official measure proposed for DCASE2016 Challenge, that is segment-based F-score given by the formula:

$$F = \frac{2P \cdot R}{P + R}, P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \quad (4)$$

where TP is the number of true positives, FP - false positives, and FN - false negatives.

The second metric that we report is Error Rate (ER), proposed for DCASE2016, which measures the amount of errors in terms of a number of insertions $I(k)$, deletions $D(k)$ and substitutions $S(k)$ in a segment k [23]. The Error Rate is then calculated by integrating segment-wise counts over the total number of segments K, with $N(k)$ being the number of active ground truth events in segment k:

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)} \quad (5)$$

Both F-score and Error Rate are calculated in 1 second segments.

3.2. Results

3.2.1. Binarization of activity matrix

The activation matrix that we get from CSNMF method needs to be binarized with a certain threshold in order to determine the presence of events. To determine the threshold value, we search among a number of threshold values to find the best balance between the F-score and Error Rate. For residential area environment the threshold is chosen to be of 30% of the maximum activation value, whereas for home environment - 20% of the maximum activation value.

3.2.2. Temporal context and sparsity

Table 2 and 3 show the influence of the sparsity regularizer and size of the 2D time-frequency patches on the performance of the proposed CSNMF method in residential area and home environment respectively. For the residential area we can see that adding some temporal context improves the F-score. The best result is achieved for 4 concatenated frames. Longer context is not beneficial for the method. Similarly, enforcing sparsity improves the results with the highest performance for $\lambda = 0.3$. However, for home environment the best performance is achieved with high sparsity, $\lambda = 0.5$, but using a single time frame as an input.

λ	0	0.1	0.2	0.3	0.5	1
1 frame	22.8%	30.5%	31.9%	19.3%	25.4%	23.5%
	2.45	1.37	1.13	1.03	1.11	0.95
4 frames	32.7%	33.9%	32.9%	**35.8%**	29.8%	11.7%
	1.46	1.09	0.95	**0.86**	0.95	1.03
6 frames	36.0%	34.2%	36.7%	35.1%	31.5%	16.3%
	0.96	1.01	0.91	0.96	0.92	1.05
8 frames	32.5%	42.0%	31.0%	32.6%	27.6%	14.6%
	1.21	0.95	1.02	0.96	1.00	1.06
10 frames	36.9%	43.2%	29.6%	23.6%	23.3%	15.8%
	0.96	0.96	1.00	1.07	1.04	1.09
12 frames	37.4%	37.5%	34.6%	21.7%	16.8%	12.6%
	0.97	0.95	1.04	1.04	1.10	1.09

Table 2: Residential area environment: F-score (%) and Error Rate (ratio) for different values of sparsity regularizer λ and temporal context (number of concatenated frames) using CSNMF

λ	0	0.1	0.2	0.3	0.5	1
1 frame	8.3%	8.0%	7.1%	7.5%	**11.7%**	11.4%
	5.72	3.18	1.73	1.79	**1.25**	1.32
4 frames	6.0%	7.4%	6.8%	10.0%	7.9%	11.6%
	3.13	1.83	1.36	1.41	1.35	1.35
6 frames	8.4%	9.3%	6.3%	6.7%	9.3%	10.1%
	1.77	1.58	1.40	1.51	1.48	1.65
8 frames	7.9%	10.0%	7.9%	6.5%	6.6%	7.6%
	1.62	1.48	1.39	1.40	1.60	1.82
10 frames	4.9%	6.6%	7.7%	8.6%	6.8%	7.0%
	1.52	1.55	1.45	1.56	1.74	1.94
12 frames	6.6%	6.4%	5.8%	6.9%	5.1%	10.0%
	1.51	1.37	1.52	1.72	1.78	2.03

Table 3: Home environment: F-score (%) and Error Rate (ratio) for different values of sparsity regularizer λ and temporal context (number of concatenated frames) using CSNMF

Table 4 shows the influence of the temporal context for the second proposed method, i.e. MRF classification. The best F-score and Error Rate for residential area environment is achieved for 2D time-frequency patches of 4 concatenated frames. For home environment the best result is achieved for 1 frame only.

3.2.3. Method comparison

The results on the development dataset for each context for the baseline, our Coupled Sparse Non-negative Matrix Factorization

	Residential area		Home	
	F	ER	F	ER
1 frame	36.8%	0.84	10.3%	**0.97**
4 frames	44.7%	**0.83**	14.9%	0.98
6 frames	**45.0%**	0.85	16.8%	0.98
8 frames	44.2%	0.84	**17.4%**	0.98
10 frames	44.1%	0.84	17.3%	0.99
12 frames	43.7%	0.83	16.2%	0.99

Table 4: F-score and Error Rate for different temporal context (number of concatenated frames) using Multi-class Random Forest (MRF) for home and residential area environments

(CSNMF) approach and our Multi-class Random Forest (MRF) approach in terms of F-score and Error Rate are shown in Table 5. As a baseline we chose the system provided for the DCASE2016 Challenge, which is based on MFCC acoustic features and GMM classifier [23]. On the development set, the MRF classifier outperforms the baseline in residential area environment and achieves comparable average results. CSNMF achieves similar F-score as the baseline but with a higher Error Rate. The results on the evaluation dataset are shown in Table 6. MRF outperformed the baseline and CNMF in both F-score and Error Rate.

	Residential area		Home		Average	
System	F	ER	F	ER	F	ER
baseline	31.5%	0.86	15.9%	**0.96**	23.7%	**0.91**
CSNMF	35.8%	0.86	11.7%	1.25	23.8%	1.06
MRF	**44.7%**	**0.83**	**17.4%**	0.98	**31.1%**	**0.91**

Table 5: Development dataset: Results for each context for the baseline system, Coupled Sparse Non-negative Matrix Factorization (CSNMF) approach and Multi-class Random Forest (MRF) approach in terms of F-score and Error Rate

Evaluation dataset		
System	F	ER
baseline	34.3%	0.88
CSNMF	29.2%	1.07
MRF	**44.1%**	**0.82**

Table 6: Evaluation dataset: Average results for the baseline system, Coupled Sparse Non-negative Matrix Factorization (CSNMF) approach and Multi-class Random Forest (MRF) approach in terms of F-score and Error Rate

4. DISCUSSION

We observe much lower performance of each of the compared methods on the home environment of the development dataset. This may be due to higher level of polyphony.

We have investigated the influence of the temporal context, i.e. number of concatenated frames, on the performance of the systems. Taking CSNMF method into consideration, introducing 2D spectral patches was beneficial for the residential area acoustic scene, however, it did not improve the performance for the home acoustic scene. That may be due to the fact that in the home environment we experience more impact sounds, such as "object impact", "cupboard", "object snapping". Therefore, longer temporal context may blur the information contained in a single frame. On the contrary, residential area contains many sounds with long characteristics, such as "car passing by", "bird singing", which are at the same time the most frequent ones in the dataset.

We also investigated the influence of inducing explicit sparsity on matrix factorization, which has lead to better performance. Higher sparsity needs to be imposed for shorter temporal context.

An interesting discussion is brought up by analysing the results on the residential area environment of the evaluation dataset. As seen in Table 7, the algorithms, especially MRF, in fact recognize two classes very well, i.e. "bird singing" and "car passing by", appearing much more often in the dataset than the others. Nevertheless, the average F-score of MRF is 9.9 percentage points higher than the baseline. Such a result shows the necessity of either evaluating the algorithms on a balanced dataset, where the events are distributed more evenly or using class-wise metrics to compare the performance of the algorithms.

		MRF		CSNMF	
Event label	Nref	F	ER	F	ER
(object) banging	11	0.0%	1.00	0.0%	1.27
bird singing	413	54.7%	1.15	56.3%	1.54
car passing by	213	68.6%	0.65	16.1%	0.98
children shouting	15	0.0%	1.00	0.0%	1.13
people speaking	57	0.0%	1.14	16.3%	2.16
people walking	146	0.0%	1.00	10.1%	1.46
wind blowing	48	0.0%	1.00	4.1%	0.98

Table 7: Residential area environment: F-score, Error Rate and number of references (Nref) in the dataset for each class using Multi-class Random Forest (MRF) and Coupled Sparse NMF (CSNMF)

5. CONCLUSION

In this paper we presented two methods for polyphonic AED in real environments. Both of the presented approaches achieve similar segment-wise F-score and Error Rate to the DCASE2016 baseline system on the TUT Sound Events 2016 dataset. The proposed MRF classification of spectral patches outperformed significantly the baseline on evaluation dataset despite its obvious drawbacks, such as incapability of recognizing a combination of sounds that was not present in the training dataset. Therefore, in the future we will investigate the possibilities of generating new combinations of sounds by allowing for multi-label classification. Moreover, spectro-temporal patches retrieved by shingling are a straightforward way for modelling temporal context, which did not improve detection on home environment. Therefore, we will investigate more complex time-frequency structure descriptors, such as the scattering transform. Additionally, we will investigate multi-resolution approaches to model both short and long acoustic events.

6. REFERENCES

[1] P. V. Hengel and J. Anemüller, "Audio event detection for in-home care," in *Proceedings of International Conference on Acoustics (NAG-DAGA)*, 2009, pp. 618–620.

[2] J. Kotus, K. Lopatka, and A. Czyzewski, "Detection and localization of selected acoustic events in acoustic field for smart surveillance applications," *Multimedia Tools and Applications*, vol. 68, no. 1, pp. 5–21, 2014.

[3] M. Bugalho, J. Portêlo, I. Trancoso, T. Pellegrini, and A. Abad, "Detecting audio events for semantic video search," in *Proceedings of the 10th International Conference of the International Speech Communication Association (Interspeech 2009)*, 2009, pp. 1151–1154.

[4] F. Meucci, L. Pierucci, E. Del Re, L. Lastrucci, and P. Desii, "A real-time siren detector to improve safety of guide in traffic environment," *Proceedings of the 16th European Signal Processing Conference (EUSIPCO 2008)*, 2008.

[5] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real-life recordings," in *Proceedings of the 18th European Signal Processing Conference (EUSIPCO 2010)*, 2010, pp. 1267–1271.

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

[7] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2011)*, pp. 69–72, 2011.

[8] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, 2013.

[9] A. Diment, T. Heittola, and T. Virtanen, "Sound event detection for office live and office synthetic AASP challenge," in *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (D-CASE 2013)*, 2013, extended abstract. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OL/DHV.pdf

[10] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization." *Nature*, vol. 401, no. 6755, pp. 788–91, 1999.

[11] D. Giannoulis, D. Stowell, E. Benetos, M. Rossignol, M. Lagrange, and M. D. Plumbley, "A database and challenge for acoustic scene classification and event detection," in *Proceedings of the 21st European Signal Processing Conference (EUSIPCO 2013)*, vol. 47, 2013, pp. 552–567.

[12] A. Dessein, A. Cont, and G. Lemaitre, *Real-time detection of overlapping sound events with non-negative matrix factorization*, F. Nielsen and R. Bhatia, Eds., 2012.

[13] D. Stowell, D. Giannoulis, and E. Benetos, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17 (10), pp. 1733–1746, 2015.

[14] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, H. Van, K. Arenberg, T. M. Kempen, and K. U. Leuven, "An exemplar-based NMF approach to audio event detection," in *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (D-CASE 2013)*, 2013, extended abstract. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OL/GVV.pdf

[15] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, 2013, pp. 8677–8681.

[16] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2016)*, 2016, pp. 6450–6454.

[17] O. Dikmen and A. Mesaros, "Sound event detection using non-negative dictionaries learned from annotated overlapping events," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2013)*, 2013, pp. 5–8.

[18] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, 2015, pp. 151–155.

[19] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015)*, 2015, pp. 2551–2555.

[20] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, 2016, pp. 6440–6444.

[21] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[22] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, 2015, pp. 171–175.

[23] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24rd European Signal Processing Conference 2016 (EUSIPCO 2016)*, 2016.

[24] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference (SciPy 2015)*, K. Huff and J. Bergstra, Eds., 2015, pp. 18–25.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, 1937.

[27] J. Eggert and E. Körner, "Sparse coding and NMF," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, no. 4, 2004, pp. 2529–2533.

DCASE 2016 ACOUSTIC SCENE CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

Michele Valenti[*1], *Aleksandr Diment*[2], *Giambattista Parascandolo*[2], *Stefano Squartini*[1], *Tuomas Virtanen*[2]

[1]Università Politecnica delle Marche, Department of Information Engineering, Ancona, Italy

[2]Tampere University of Technology, Department of Signal Processing, Tampere, Finland

ABSTRACT

This workshop paper presents our contribution for the acoustic scene classification (ASC) task proposed for the "detection and classification of acoustic scenes and events" (DCASE) 2016 challenge. We propose the use of a convolutional neural network trained to classify short sequences of audio, represented by their log-mel spectrogram. We also propose a training method that can be used when the system validation performance saturates as the training proceeds. The system is evaluated on the public ASC development dataset provided for the DCASE 2016 challenge. The best accuracy score obtained by our system on a four-fold cross-validation setup is 79.0% which constitutes a 8.8% relative improvement with respect to the baseline system.

Index Terms— Acoustic scene classification, convolutional neural networks, DCASE, computational audio processing

1. INTRODUCTION

When we talk of ASC we refer to the capability of a human or an artificial system to understand an *audio context*, either from an on-line stream or from a recording. "Context" or "scene" are concepts that humans commonly use to identify a particular acoustic environment, *i.e.* the ensamble of background noises and sound events that we associate to a specific audio scenario, like a restaurant or a park. For humans this may look like a simple task: complex calculations that our brain is able to perform and our extensive life experiences allow us to easily associate these ensembles of sounds to specific scenes. However, this task is not trivial for artificial systems. The interest in computational ASC lies in its many possible applications, like context-aware computation [1], intelligent wearable interfaces [2] and mobile robot navigation [3]. In the field of machine learning different models and audio feature representations have been proposed to deal with this task, especially in the last few years, thanks to the contributions to the previous DCASE challenge in 2013 [4]. Some examples of classifiers used in the previous challenge are Gaussian mixture models (GMMs) [5], support vector machines [6] and tree bagger classifiers [7].

Nowadays, application of convolutional neural networks (CNNs) for audio-related tasks is becoming more and more widespread, for example in speech recognition [8], environmental sound classification [9] and robust audio event recognition [10]. To the best of our knowledge this is the first work introducing a CNN-based classifier specifically designed for ASC.

Our system is designed to output class prediction scores for short audio sequences. During training a recently-proposed regularization technique is used, *i.e.* batch normalization. In addition, we propose a training procedure that will allow the classifier to achieve a good generalizing performance on the development dataset. Hence our intent is to propose a system capable of improving the baseline system, represented by a GMM [11] classifier, and to give a novel contribution for future development in the use of neural networks for the ASC task.

In Section 2 a brief background about CNNs is given. Then, a detailed description of the proposed system is reported in Section 3. Finally, results for the proposed model and comparisons between different configurations are presented in Section 4 and our conclusions are reported in Section 5.

2. CONVOLUTIONAL NEURAL NETWORKS

In a CNN inputs are processed by small computational units (neurons) organized in a layered structure. The most remarkable feature that makes CNNs a particular subset of feed-forward neural networks is the presence of convolutional layers. Convolutional layers are characterized by neurons (called kernels) that perform subsequent non-linear filtering operations along small context windows of the input, *i.e.* their receptive fields (RFs). This localized filtering is a feature known as local connectivity and it represents one key characteristic for obtaining invariance against input pattern shifts. Parameters defining the RF are its width (L), height (H), depth (D) and stride. The area ($L \times H$) defines the dimension of the kernel's context window and the stride defines how much this window will slide between two filtering operations. Both area and stride are free parameters, whereas the depth is the same as the input's. For example, if the input is a three-channel (RGB) picture the network will be fed with a tensor with depth $D = 3$. In an audio analysis scenario the input can be a spectrogram, represented by a bi-dimensional matrix, which has unitary depth dimension ($D = 1$). Moreover, if the convolutional layer is acting on the output of another convolutional layer, then D will be equal to the number of kernels of the former layer.

Non-linear filtering consists of two steps. In the first place an output is calculated through a linear combination of each pixel currently seen by the RF. Then, this output is fed into a non-linear function, *i.e.* the activation function. Activation functions that are typically used are the sigmoid, the hyperbolic tangent or the rectifier function. Finally, subsequent kernel outputs are collected in matrices that are called feature maps.

Pooling layers are usually placed after each convolutional layer to reduce each feature map dimensions and to enhance the network invariance to input pattern shifts. The most common pooling layer is formed by filters that operate with non-overlapping windows by extracting the highest value from the area, *i.e.* max-pooling. Therefore, the stacking of multiple convolutional and pooling layers will make

[*]This work has been done during an internship at Tampere University of Technology.

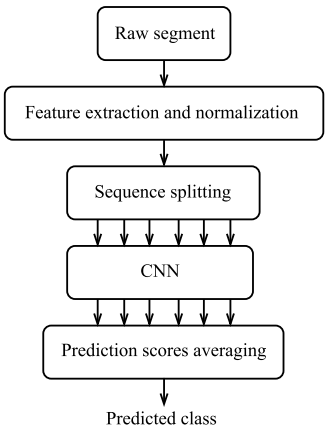


Figure 1: Block diagram of the proposed method.

the network able to extract features with a gradual increment of the input overview. The output layer is usually a fully-connected softmax layer. So, if we define y_i as the output of neuron i in the last layer we will have:

$$y_i = \text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{N}(\exp(x_j))}, \quad (1)$$

where N is the number of possible classes, x_i the input to the non-linearity and y_i the prediction score for the input sequence to belong to the i^{th} class. Hence the overall output is a vector $\mathbf{y}$ containing all network's prediction scores associated to each class. If we let y_j to be the highest of these scores, the predicted class for the input sequence will be the j^{th} class. In order to optimize the network parameters a comparison between the prediction vector $\mathbf{y}$ and a target vector is performed in terms of a loss function. Typically used loss functions are the mean squared error or the categorical cross-entropy.

3. PROPOSED METHOD

In this section we describe our method and the architecture chosen for the proposed system. The main steps are represented as a block diagram in Figure 1.

3.1. Feature representation and preprocessing

The feature representation we choose for our system is the log-mel spectrogram. To calculate it we apply a short-time Fourier transform (STFT) over windows of 40 ms of audio with 50% overlap and Hamming windowing. We then square the absolute value of each bin and apply a 60-band mel-scale filter bank. Finally, the logarithmic conversion of the mel energies is computed. The whole feature extraction process has been implemented in Python using the librosa [12] library.

After the extraction process we normalize each bin by subtracting its mean and dividing by its standard deviation, both calculated on the whole training set of each fold. We then split normalized spectrograms into shorter spectrograms, which we will call sequences hereafter. Tests with different sequence lengths are reported in Section 4. Unlike frames used for the STFT, we choose sequences to be non-overlapping. At the end of this process the input to the CNN is a matrix which can be treated as a mono-channel image.

3.2. Proposed architecture

The proposed model consists of a deep CNN and it is represented in Figure 2. Parameters we report here are chosen as a result of experiments aimed to test different kernel numbers and different RF areas.

The first layer performs a convolution over the input spectrogram with 128 kernels characterized by 5×5 RFs and unitary depth and stride in both dimensions. The obtained feature maps are then sub-sampled with a max-pooling layer which operates over 5×5 non-overlapping squares. The second convolutional layer is the same as the first one, with the exception that more kernels (256) are used in order to grant higher level representation. The second and last sub-sampling is then performed aiming to the "destruction" of the time axis. Therefore, we use a max-pooling layer which operates over the entire sequence length and, on the frequency axis, only over four non-overlapping frequency bands. The activation function used for kernels in both convolutional layers is the rectifier function, therefore kernels are usually called rectifier linear units (ReLUs) [13]. Finally, since the classification involves 15 different classes, the last is a softmax layer composed of 15 fully-connected neurons.

The classification for the whole segment is obtained by averaging all prediction scores obtained for its sequences. Recalling the notation introduced in Section 2, the CNN output $\mathbf{y}^{(i)}$ is now a vector containing all class-wise prediction scores for the i^{th} sequence. Then, the predicted class c^* for the whole segment is calculated as:

$$c^* = \arg\max_c \left[\frac{1}{M} \sum_{i=1}^{M} y_c^{(i)} \right], \quad (2)$$

where M is the number of sequences into which the segment is split and $y_c^{(i)}$ is the c^{th} entry of $\mathbf{y}^{(i)}$. In other words, the predicted class is the position of the maximum entry y_{c^*} in the vector given by the average of all prediction vectors output for each sequence.

The system is implemented with the Keras library (vers. 1.0.4) [14] for Python and its training is performed with an Nvidia Tesla K80 GPU showing an average training time of 50 s per epoch. The loss function we use for training is the categorical cross-entropy and the optimization algorithm we choose for its minimization is the adaptive momentum (adam) [15]. Basing on preliminary experiments we propose to use the optimizer default parameter configuration.

3.3. Regularization and model training

Batch normalization, introduced in [16], is a technique that addresses the issue described by Shimoidara *et al.* [17], known as internal covariate shift. We can look at batch normalization as an intermediate layer placed after each of the two convolutional layers in order to whiten the output of such layers. As showed in [16] and in our preliminary experiments, this practice can drastically reduce the training convergence time. A batch normalization layer applies a linear transformation $\text{BN}_{\beta,\gamma}$ to its input x as follows:

$$\text{BN}_{\beta,\gamma}(x) = \frac{\gamma}{\sqrt{\text{Var}[x] + \epsilon}} \cdot x + \left(\beta - \frac{\gamma \cdot \text{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}} \right), \quad (3)$$

where $\text{E}[x]$ and $\text{Var}[x]$ are the mean and the variance of the input to the batch normalization layer, calculated for a batch of samples. Moreover, γ and β represent the transformation parameters that will be learned during training. We use batch normalization to normalize kernel outputs, therefore we have one γ and one β for each kernel.

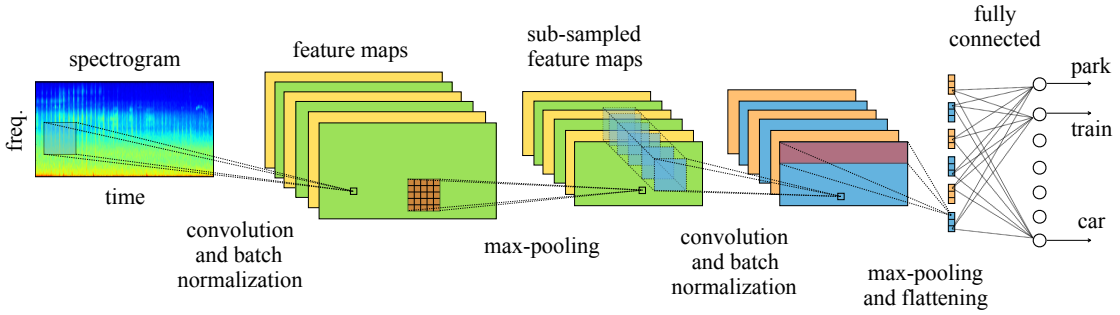


Figure 2: Block scheme of the used convolutional neural network. Max-pooling windows are represented in red, kernel RFs are in light blue.

By using batch normalization we increase the model complexity, but preliminary experiments showed better performance and a drastic reduction of the number of epochs needed for training convergence.

The proposed training method consists of two phases. The first, called *non-full training*, starts with a splitting of the whole training data into two subsets: one for training and one for validation. Every epoch we collect training spectrograms into class-wise feature lists so to randomly shuffle and time-shift them before the sequence splitting. This is done in order to increase the input variability, hence showing the network always slightly different sequence spectrograms. Then, every five epochs we check the segment-wise performance on both training and validation sets according to the metrics described in Section 4. After the check, we save the network parameters if the segment-wise validation score has improved. Finally, we stop the training if no improvement is recorded after 100 epochs. With this setup it is possible to notice that the segment-wise validation performance is prone to saturate. This means that the score starts to oscillate around a fixed, stable value. When this happens we say that the system has converged, therefore it is possible to proceed to the second phase, which we call *full training*. In this phase we decide to re-train the network on all the training data for a fixed number of epochs. This number is chosen by looking at the convergence time of segment-wise validation accuracies during the non-full training. A model trained this way will reach a convergence state without excessive overfit and making the best of all the available training data. This fact may turn out to be particularly desirable when dealing with small datasets, such as in this case.

4. EVALUATION

4.1. Dataset and metrics

For our evaluation we use the DCASE 2016 [11] development dataset as provided at the beginning of the challenge, when mobile phone interferences were not annotated. We do not consider this a problem, since only less than the 1% of the total audio duration is affected. The dataset consists of 1170 audio segments of thirty seconds, equally distributed between 15 different classes, and of text files including both the annotated ground truth and the recommended data subdivision. The 15 classes are: beach, bus, café/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train and tram.

Groups of segments have been obtained by splitting a longer audio file recorded in a single location. Due to this fact it is important not to train and evaluate the network performance on segments coming from the same location, since this would falsify the generalization score. Because of this, we decide to use the training and test subsets recommended in the dataset annotations. The train/test split gives us approximatively 880 training segments for each fold, some folds having fewer segments. This is due to the fact that different long recordings have been split into a variable number of segments, ranging from three to ten. This means that the distribution of per-location segments is not uniform. Similarly to what was done for the recommended sets, for the training/validation split we decide how many locations to use for validation and then pick all segments coming from those locations.

The model is evaluated according to a four-fold cross-validation scheme. For the evaluation of each fold, per-class accuracies are initially calculated on the test set on a segment-wise level. These accuracies are obtained by dividing the number of correctly classified segments by the total number of segments belonging to the class. Accuracies for each fold are then obtained by averaging all the 15 per-class accuracies. Finally, the overall accuracy is calculated by averaging the four per-fold accuracies.

4.2. Classification accuracy and sequence length

Our ability to distinguish different scenes from acoustic information is influenced by the length of the sequence we can listen to. Hence, the main focus of this section is a comparison between accuracies obtained with different sequence lengths. The average convergence time we estimate, hence the chosen full training period, is 200 epochs. For full training configurations we compute mean accuracies over four experiments involving different random weights initializations. Results we report in Table 1 apparently highlight that medium-length sequences, like three or five seconds, perform better than extremely short or long sequences. The best accuracy is achieved by the three-second configuration, with an average accuracy of 75.9% with the non-full training configuration. The accuracy rises to 79.0% if full training is performed.

A deeper insight into which classes are mostly misclassified can be obtained by looking at the confusion matrix in Figure 3, which we obtained by grouping results of all folds. What emerges is that some classes — *e.g.* “park” and “residential area”, or “bus” and “train” — are often confused by the system. This may indicate that our model is relying more on the background noise of the sequence rather then on acoustic event occurrences. We believe that this can also explain why classes with very similar background noises are confused even when 30-second sequences are used. Due to results in Table 1, we choose three seconds as the sequence length used for the challenge evaluation results. For the final training we use the whole development dataset and we estimate the new convergence time to be 400 epochs. With this configuration our model reaches a 86.2% accuracy score, therefore ranking the sixth place out 57 systems submitted in the first task of

Table 1: Accuracy comparison for the proposed model trained with different sequence lengths (seq. len.) and training modes ("non-full" and "full"). Standard deviations refer to full training accuracies.

seq. len. (s)	*accuracy* (%)		
	non-full	*full*	± *s.d.*
0.5	68.2	75.4	0.75
1.5	74.0	78.4	1.19
3	75.9	**79.0**	0.68
5	74.1	78.3	0.86
10	71.5	77.3	0.88
30	74.0	75.6	0.44

Table 2: Accuracy comparison for different systems and training modes ("non-full" and "full"). Neural architectures are identified by their number of hidden layers.

system	*seq. len.* (s)	*accuracy* (%)	
		non-full	*full*
two-layer MLP (log-mel)	-	66.6	69.3
one-layer CNN (log-mel)	3	70.3	74.8
two-layer CNN (log-mel)	3	75.9	**79.0**
two-layer CNN (MFCC)	5	67.7	72.6
baseline GMM (MFCC)	-	-	72.6

the DCASE 2016 challenge.

4.3. Comparison of other systems

Here we report new comparisons with other systems that have been tested during our development. All parameter configurations are chosen in order to give each system a representation capacity that can be comparable with the proposed network's.

The first system we introduce is a multi-layer perceptron (MLP) with two hidden layers. The input layer consists of 900 nodes to which we apply log-mel features for a context window of 15 frames. We then stack two hidden layers with 512 ReLU units (both batch-normalized) and an output layer with 15 softmax neurons. The difference from the proposed network is that the output is now the class associated to the central frame of the context window, which is sliding with unitary stride along the spectrogram. Segment-wise classification is performed as described in Eq. (2), but now M represents the number of frames in a segment. With the non-full training setup the proposed model achieves a 66.6% accuracy with a convergence time of 100 epochs. A full training for 100 epochs lets the model achieve a 69.3% accuracy.

The second model we use for the comparison consists of a CNN whose input is a three-second sequence. The input is processed by only one hidden convolutional layer with 2048 ReLU kernels whose RFs cover all the mel-energy bands and a context window of 15 frames. In this very first step the frequency dimension is narrowed, hence the first layer outputs are matrices with unitary height. Then, after a batch normalization layer, we stack a max-pooling layer to shrink also the time dimension, hence obtaining a single number for each feature map. This structure is very similar to a MLP, since each kernel looks at all the mel-energy bands of a single context window at the same time. The key difference is represented by the max-pooling, which, similarly to the proposed model's, emphasizes the occurrence of a particular input pattern no matter where it appears in the sequence. This network achieves a 70.3% accuracy with the non-full training setup, which rises to 74.8% if a full training is performed for 100 epochs.

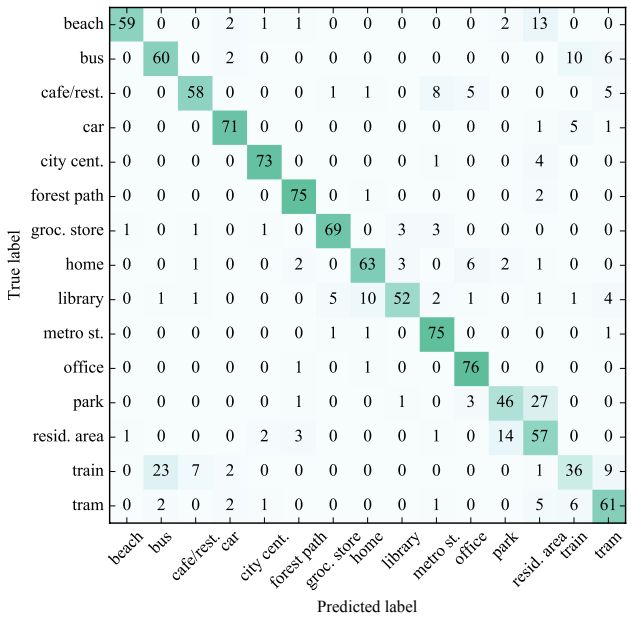


Figure 3: Confusion matrix for the proposed CNN evaluated on the four folds. Three-second sequences are used.

The last result we report intends to compare our architecture and the baseline system when both are trained on mel-frequency cepstral coefficients (MFCCs). Details about the feature parameters are reported in [11]. The baseline system trains 15 different mixtures of 16 Gaussians in order to model each of the 15 classes. Classification is performed by comparing each mixture to the test audio file in terms of log-probabilities of the data under each model. The most similar mixture gives the chosen class. We choose a sequence length of five seconds for this comparison. Our model reaches a 67.7% overall accuracy with the non-full training setup and the average convergence time on the validation data is 100 epochs. The performance reaches a 72.6% overall accuracy with a full training setup, which equals the baseline accuracy.

5. CONCLUSIONS

Our work proposes one out of many possible ways of approaching acoustic scene classification with CNNs. In doing so we reach a 79.0% accuracy on the DCASE 2016 development dataset, demonstrating that a two-layered convolutional network can achieve higher accuracies if compared to a two-layer MLP (9.7% more), a one-layer CNN (4.2% more) and a GMM-MFCC system (6.4% more). We observed also that, under particular circumstances, training the network without monitoring its generalization performance can lead to a relevant accuracy improvement. This is true especially when the lack of training data is a narrow bottleneck for the network generalizing performance.

6. ACKNOWLEDGMENTS

The authors wish to acknowledge CSC — IT Center for Science, Finland, for generous computational resources.

7. REFERENCES

[1] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. IEEE, 1994, pp. 85–90.

[2] Y. Xu, W. J. Li, and K. K. Lee, *Intelligent wearable interfaces*. John Wiley & Sons, 2008.

[3] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. J. Mataric, "Where am I? Scene recognition for mobile robots using audio features," in *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 885–888.

[4] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[5] M. Chum, A. Habshush, A. Rahman, and C. Sang, "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.

[6] J. T. Geiger, B. Schuller, and G. Rigoll, "Large-scale audio feature extraction and SVM for acoustic scene classification," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

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

[8] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition." in *INTERSPEECH*, 2013, pp. 3366–3370.

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

[10] H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," *arXiv preprint arXiv:1604.06338*, 2016.

[11] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," *In 24th Acoustic Scene Classification Workshop 2016 European Signal Processing Conference (EUSIPCO)*, 2016.

[12] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015.

[13] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[14] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[17] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

ABROA : AUDIO-BASED ROOM-OCCUPANCY ANALYSIS USING GAUSSIAN MIXTURES AND HIDDEN MARKOV MODELS

Rafael Valle

UC Berkeley, Center for New Music and Audio Technologies (CNMAT),
Berkeley, California, 94709, USA
rafaelvalle@berkeley.edu

ABSTRACT

This paper outlines preliminary steps towards the development of an audio-based room-occupancy analysis model. Our approach borrows from speech recognition tradition and is based on Gaussian Mixtures and Hidden Markov Models. We analyse possible challenges encountered in the development of such a model, and offer several solutions including feature design and prediction strategies. We provide results obtained from experiments with audio data from a retail store in Palo Alto, California. Model assessment is done via *leave-two-out* Bootstrap and model convergence achieves good accuracy, thus representing a contribution to multimodal people counting algorithms.

Index Terms— Acoustic Traffic Monitoring, Audio Forensics, Retail Analytics

1. INTRODUCTION

Information about the occupancy of a certain location is relevant for several applications, specially in surveillance tasks and staff management. For example, occupancy can be used to detect intruders in a house, or to generate optimal employee schedules according to shopper traffic. The existing systems for occupancy detection rely on multimodal systems, including video, Wifi, Bluetooth and, to a much lesser extent, audio. Conversely, the use of audio in occupancy estimation empowers the development of a model, not dependent on speaker separation, that is robust to issues that are common in computer vision and systems that rely on tracking electronic devices[1], such as occlusion and people clusters[1].

1.1. Related work

The occupancy estimation literature can be subdivided in invasive and non-invasive strategies. Invasive strategies [2, 3, 4] use various devices, e.g. smartphones and ultrasonic transmitters, to project sound, e.g. sinusoids and chirps, onto the environment and use the environment's response to the projected sound to estimate activity in the space; Non-invasive strategies [5, 6, 7] rely on detecting speech sounds in the environment to estimate occupancy. In addition to potentially disturbing humans and animals, invasive strategies require the expensive task of deploying devices, e.g. 891 mobile devices to cover 600 square meters and less than 20 people [6], in the location and badly suffer from the addition of non-human objects and subjects to the space being analyzed. Non-invasive strategies that rely on speech only to estimate occupancy will disregard people who are in a space but not talking, and badly suffer from situations in which speech diarization is not possible. In addition, most of these systems are only able to handle small groups of people.

The limitations presented above and the lack of a standard technique or key paper on the topic of audio-based room-occupancy analysis confirm the need for the development of such a technique. There is no annotated dataset for audio-based room-occupancy analysis and, therefore, the acquisition of data remains a blatant challenge. Although [8] describes the layout of a rather promising prototype to estimate the occupancy of rooms and buildings based on audio, they provide no information on experiments nor results describing the efficiency of their system. In this paper we describe a system that is not invasive, suited for large groups of people, based on real data and computationally inexpensive.

2. METHODOLOGY

2.1. Dataset and ground truth

The dataset used in this research is comprised of proprietary audio recordings made with a smartphone placed in a retail store located in the United States. The smartphone's microphone was aimed towards the inside of the store and placed at the store's main and single entrance, at approximately 5 meters from the floor. The recordings took place during open hours (10h, 22h). The ground truth data is divided into 15-minute slices and it provides the cumulative occupancy at the

This research was supported in part by the TerraSwarm Research Center, one of six centers supported by the STAR net phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

[1] An increasingly hard task with the randomization of MAC addresses and privacy laws

end of each 15-minute time window. The ground truth was obtained from video data submitted to Amazon's Mechanical Turk. Figure 2 illustrates the occupancy for that specific week.

Figure 1: Occupancy

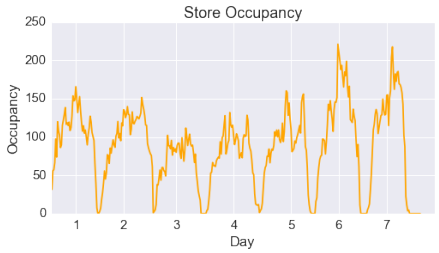


Figure 2: Store occupancy for the first week of April.

2.2. Room-occupancy Analysis

Several challenges are present in the development of an audio-based room-occupancy analysis system. In our context, the ground truth only provides information about the aggregated occupancy at the end of each 15-minute interval. This provides a challenge to feature selection, that is, selecting the audio slice that best represents the ground truth. In addition, since there's one dependent variable for each 15-minute interval, regression models would require the design of summary statistics of the audio data to be used as the independent variable, thus extremely reducing the amount of information retrieved from each training sample. During evaluation we considered generalized linear models (GLM). Surprisingle, the mean error of the best linear model, Poisson Regression, was 33% worse than the GMM-HMM model described in this paper.

2.2.1. Audio Features

In our experiments we performed cross-validation on the training set using lasso regression models with the following features and linear regression models using all posible combination of amplitude (median, mean, standard deviation), spectral (centroid, spread, skewness, kurtosis, slope) and mfcc (raw, 1st delta, 2nd delta) features.

We observed the p-values, 5% significance, of the regression models and concluded that the MFCC features contributed the most to prediction accuracy on the training set. Given this conclusion, our room-occupancy analysis algorithm uses the well-known Mel-Frequency Cepstral Coefficients[9] (MFCC). A total of 20 MFCCs (computed with a FFT Size of 4096 samples, Hop Size of 1024 samples and audio sample rate is 11050 hz) along with their first (20 features) and second (20 features) deltas[10] assembling a feature vector with 60 dimensions total. Given that sounds produced by humans are rarely stationary, the delta features provide valuable information.

2.2.2. Window selection

Our labeled data provides the cumulative occupancy at the end of a 15-minute time window. This represents two challenges: first, it is necessary to find out what temporal slice of the audio data should be used; second, the length of this slice must be chosen such that it maximizes the model's performance. Window size is chosen under the one-standard-error rule using 50 iterations of *leave-two-out* Bootstrap with window sizes in the interval [30, 260] seconds, with a 10 seconds step size and starting at the end and increasing towards the beginning of the audio file.

2.2.3. GMM-HMM model

We propose a solution that references the speech recognition literature[11] and combines Gaussian Mixtures with Hidden Markov Models. The GMM is appropriate, for it provides a better categorization of the distribution of the audio features and a reliable estimate of the likelihood function, $p(X|\lambda)$, where $X = x_1, x_2, \ldots, x_T$ is a sequence of feature vectors (MFCCs and deltas in our case), x_t is a feature vector indexed at discrete time $t \in [1, 2, ..., T]$, and λ represents some model. As described in[12], for a D-dimensional feature vector x (60-dimensional in our case), the mixture density used for the likelihood of data x given model λ is defined as:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i p_i(x) \tag{1}$$

This density is a weighted linear combination of M unimodal Gaussian densities, $p_i(x)$, with parameters μ_i ($D \times 1$ mean vector) and Σ_i ($D \times D$ covariance matrix):

$$p_i(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\} \tag{2}$$

Under the assumptions in[12], our model only uses the diagonal covariance matrix and the maximum likelihood model parameters are estimated using the well-known iterative expectation-maximization (EM) algorithm[11]. Traditionally, the feature vectors of X are assumed independent and the log-likelihood for some sequence of feature vectors X is computed as:

$$LL(X|\lambda) = \sum_{t=1}^{T} log(p(x_t|\lambda)) \tag{3}$$

We bin our occupancy data by taking the integer square root of occupancy values, thus circumscribing the problem of creating one GMM per occupancy value. The lowest occupancy

is 0 and the maximum is 221, thus producing 15 occupancy bins. For each occupancy bin and its respective audio data, one bin-dependent GMM is trained with the MFCC features described above and using the Bayesian Information Criterion (BIC) for model selection over the number of components in the set $C = \{2, 3, 4, 5, 6, 7, 8, 16, 32\}$ on a test set.

In addition to the GMM, a HMM[13] can be used to compute $P(O|\lambda)$, that is the probability of the observation sequence $O = o_1, o_2, \ldots, o_T$, i.e. occupancy sequence, given model λ. As described in[14], the probability of observations O for a fixed state sequence $Q = q_1, q_2, \ldots, q_T$, $P(O|Q, \lambda)$ is:

$$\prod_{t=1}^{T} P(o_t|q_t, \lambda) = b_{q_1}(o_1)b_{q_2}(o_2)...b_{q_T}(o_T) \quad (4)$$

where b is an array storing the emission probabilities at time t given model (bin-dependent GMM in our case). The probability of the state sequence Q is given by:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T} \quad (5)$$

where π is an array storing the initial probabilities and a is an array storing the transition probability from state q_t to state q_{t+1}. We can calculate the probability of the observations given the model as:

$$P(O|\lambda) = \sum_{Q} P(O|Q, \lambda)P(Q|\lambda) \quad (6)$$

Decoding of the hidden state is computed using the Viterbi[15] algorithm to find the best path (single best state sequence) for an observation sequence. We define the probability of the best state path for the partial observation sequence as:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1 q_2, \cdots, q_t = s_1, o_1, o_2, \cdots, o_t|\lambda) \quad (7)$$

Figure 3 illustrates our system and its use of the Hidden Markov Model and the Viterbi algorithm. Each circle represents the log-likelihood score of each feature vector given each binned GMM. The lines represent the transition probabilities and the highlighted states and bold lines show the state path that maximizes the log-likelihood.

3. EXPERIMENTS AND RESULTS

Several techniques were used for prediction, starting with using the log-likelihoods of each feature vector given each GMM and ending with a HMM.

3.1. Prediction with GMMs

Following the bin-dependent GMM training, two prediction strategies are chosen, including aggregating the posterior probabilities of each GMM and majority voting.

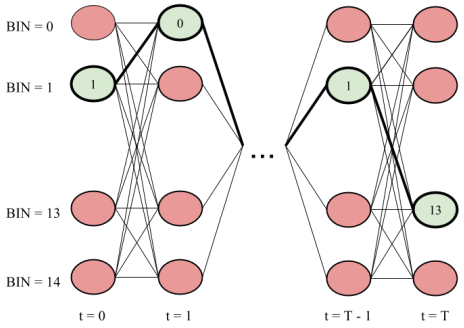


Figure 3: HMM and Viterbi illustration

3.1.1. *Posterior Probabilities Aggregation (PPA)*

This procedure consists of aggregating the posterior probabilities of each state by computing the sum of bin occupancy predictions weighted by their posterior probabilities. Let x be an audio feature vector and Λ be a set of 15 bin-dependent GMMs. From Bayes theorem:

$$P(x|\Lambda_i) = \frac{P(\Lambda_i|x)P(x)}{P(\Lambda_i)} \quad (8)$$

Assuming that $P(\Lambda)$ is uniform and knowing that $P(x)$ is similar for all models, we conclude that $P(\Lambda_i|x) \propto P(x|\Lambda_i)$. Finally, we define the estimated occupancy bin $(\hat{B})$[2] for feature vector x at time t and models Λ as:

$$\hat{B}(x_t) = \sum_{i=0}^{|\Lambda|-1} i e^{LL(x_t|\Lambda_i)} \quad (9)$$

This will produce underpredictions because the maximum value mapped to a bin is always larger than the bin's maximum predicted value, e.g. bin 2 maximum mapped and predicted values are 6 and 4 respectively.

3.1.2. *Majority Voting (MJ)*

This procedure consist of calculating the log-likelihood, $LL(X, \lambda)$ of the audio features given each bin-dependent GMM and finally selecting the bin that more often has the highest log-likelihood score.

Using the GMM only approach and these techniques, informally speaking the prediction results circulate around the correct prediction value. However, the results for both techniques show jumps in occupancy prediction that are rather unlikely because the model ignores transition probabilities between states. Therefore, we decided to address this problem by using a HMM and the Viterbi algorithm.

[2]We use log-sum-exp to prevent computational underflow

3.2. Prediction with HMM and Viterbi

This strategy divides prediction into four stages.The first computes the log-likelihood of the audio features given each bin-dependent GMM; the second applies the Viterbi algorithm to the computed log-likelihoods to obtain the best path for a specific time window. The third predicts the occupancy bin using posterior probabilities aggregation or majority voting; finally, the inverse of the square root of the predicted bin is calculated, thus translating the prediction back into the linear domain. For the HMM, the initial probabilities π are uniform (1/15), the emission probabilities b are computed using the GMMs and the transition probabilities a are derived using heuristics.

3.3. Model Assessment and Selection

We used *leave-two-out* Bootstrap[16] for model assessment and selection, as it is a suitable technique for our small dataset of 386 samples. The Bootstrap technique was used to estimate the best window size, within the range [30, 260] seconds, and the most accurate prediction strategy based on the GMM-HMM model. For each window size and prediction strategy, a total of 50 bootstrap iterations were performed and the prediction errors were computed.

Figure 4 shows bootstrapped root mean squared errors (RMSE) for all window sizes and using the MJ and PPA techniques. Accuracy increases, specially in PPA, almost linearly and proportionally to the window size.

Figure 4: RMSE using GMM-HMM and Poisson Regression(GLM). The PPA technique converges around 210 seconds and considerably outperforms other techniques.

Model and window selection is performed using the one-standard-error rule and analysis of the violin plot provided in Figure 5. The best predictor uses the PPA technique with a window size of 210 seconds. Figure 6 shows room-occupancy predictions with a window of 210 seconds and both prediction techniques.

4. CONCLUSION AND FUTURE WORK

We have described an algorithm for audio-based room-occupancy analysis that relies on Gaussian Mixtures and Hidden Markov Models. Our algorithm has advantages over other algorithms for audio-based occupancy analysis:

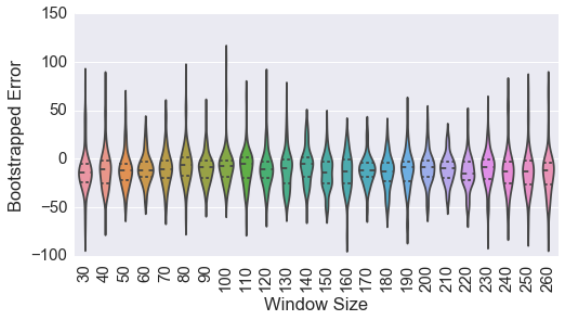


Figure 5: Bootstrapped error using GMM-HMM and PPA.

It is not invasive our algorithm does not require projecting audio into an environment, thus not causing disturbance to humans or animals present.

It handles large groups our algorithm is capable of handling occupancy prediction up to 200 people and with capability of expansion.

It is inexpensive prediction is computationally cheap and can be easily done on a smart phone, thus preventing privacy issues from sending audio via network.

We analyzed different types of prediction techniques and concluded that the GMM-HMM posterior probabilities aggregation is the most accurate approach. The algorithm performed considerably well in retail store environments with occupancy up to 200 people.

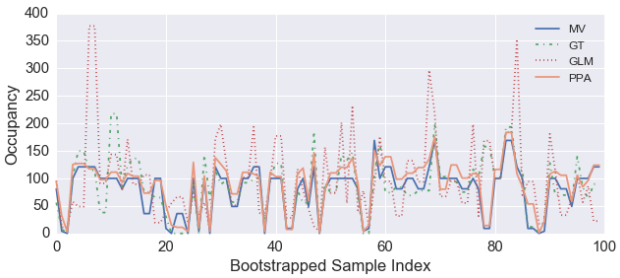


Figure 6: Ground Truth (GT) and predictions using the above described techniques. Poisson Regression (GLM) performs considerably worse than the GMM-HMM models.

The results from the current work validate the GMM-HMM model despite the expectation of better results with regression models. For future work, we plan to compute the transition probabilities from data, add a Voice Activity Detection pre-processing step to the pipeline, denoise the audio data and perform comparative analysis with the results obtained in this paper.

5. REFERENCES

[1] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer vision and image understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[2] S. Srinivasan, A. Pandharipande, and D. Caicedo, "Presence detection using wideband audio-ultrasound sensor," *Electronics Letters*, vol. 48, no. 25, pp. 1577–1578, 2012.

[3] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: unsupervised speaker count with smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 43–52.

[4] O. Shih and A. Rowe, "Occupancy estimation using ultrasonic chirps," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, 2015, pp. 149–158.

[5] M. Khan, H. Hossain, and N. Roy, "Infrastructure-less occupancy detection and semantic localization in smart environments," in *Proceedings of the 12th international conference on mobile and ubiquitous systems: computing, networking and services*, 2015.

[6] P. G. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L.-S. Peh, "Low cost crowd counting using audio tones," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 155–168.

[7] S. Stillman and I. Essa, "Towards reliable multimodal sensing in aware environments," in *Proceedings of the 2001 workshop on Perceptive user interfaces*. ACM, 2001, pp. 1–6.

[8] S. Uziel, T. Elste, W. Kattanek, D. Hollosi, S. Gerlach, and S. Goetze, "Networked embedded acoustic processing system for smart building applications," in *Design and Architectures for Signal and Image Processing (DASIP), 2013 Conference on*. IEEE, 2013, pp. 349–350.

[9] B. Gold, N. Morgan, and D. Ellis, *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons, 2011.

[10] S. Furui, "Speaker-independent isolated word recognition based on emphasized spectral dynamics," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, vol. 11. IEEE, 1986, pp. 1991–1994.

[11] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[12] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[13] L. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[14] P. Blunsom, "Hidden markov models," *Lecture notes, August*, vol. 15, pp. 18–19, 2004.

[15] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[16] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009.

HIERARCHICAL LEARNING FOR DNN-BASED ACOUSTIC SCENE CLASSIFICATION

Yong Xu, Qiang Huang, Wenwu Wang, Mark D. Plumbley

Centre for Vision, Speech and Signal Processing, University of Surrey, UK
{yong.xu, q.huang, w.wang, m.plumbley}@surrey.ac.uk

ABSTRACT

In this paper, we present a deep neural network (DNN)-based acoustic scene classification framework. Two hierarchical learning methods are proposed to improve the DNN baseline performance by incorporating the hierarchical taxonomy information of environmental sounds. Firstly, the parameters of the DNN are initialized by the proposed hierarchical pre-training. Multi-level objective function is then adopted to add more constraint on the cross-entropy based loss function. A series of experiments were conducted on the Task1 of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2016 challenge. The final DNN-based system achieved a 22.9% relative improvement on average scene classification error as compared with the Gaussian Mixture Model (GMM)-based benchmark system across four standard folds.

Index Terms— Acoustic scene classification, deep neural network, hierarchical pre-training, multi-level objective function

1. INTRODUCTION

In recent years, much research effort has been attracted for making sense of everyday or environmental sounds. It focuses on how to convert audio (non-speech and non-music) recordings into understandable and actionable information: specifically how to allow people to search, browse and interact with sounds. Some specific tasks were investigated in recent years, including acoustic scene classification (ASC) [1], sound event detection (SED) [2, 3] and domestic audio tagging. ASC aims to associate a semantic label to an audio segment that identifies the sound environment where it has been produced [1]. The goal of SED is to detect the sound events that are present within an audio signal, estimate their start and end times, and give a class label to each of the events. For audio tagging, there is no information about sound event onset or offset, only labels. This paper will focus on the ASC task.

The ASC problem was first proposed by Sawhney and Maes [4]. Recently, more related work was conducted during the IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events [5, 6, 7]. Mel Frequency Cepstrum Coefficients (MFCCs) were used as the audio feature by most of the submitted systems. GMMs, Support Vector Machines (SVMs) or hidden Markov models (HMMs) were commonly used classifier [6, 8, 9]. Other methods, such as non-negative matrix factorization (NMF) approaches can also be used to extract an intermediate representation prior to classification [10].

Recently, deep learning methods have obtained great successes in speech, image and video fields [11, 12, 13, 14] since Hinton and Salakhutdinov [15] showed the insights in using a greedy layer-wise unsupervised learning procedure to train a deep model in 2006. Deep learning methods were also investigated for acoustic scene classification tasks in [16, 17, 18]. In [16], a series of experimental investigations on the DNN structure, including the number of hidden layers and input frame expansion, were presented. It also demonstrated that DNN can yield better results than GMM and SVM. Convolutional neural networks (CNNs) which are the variant of DNNs have been also adopted for environmental sound classification in [17].

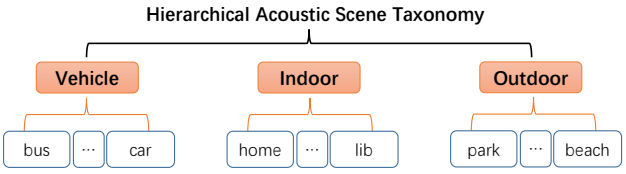


Figure 1: Example of a hierarchical acoustic scene taxonomy.

There has also been research about the taxonomy of the environmental sounds [19, 20]. The taxonomy of environmental sounds indicates that hierarchical categories information exists in sound classes. For example, environmental sounds can be coarsely classified into *indoor*, *outdoor* and *vehicle* in Fig. 1, and these are the high-level scene classes. Meanwhile, corresponding branches denote the low-level scene classes.

In this paper, we propose a hierarchical learning method incorporating the acoustic scene taxonomy information for ASC in a DNN-based framework. Two approaches are presented to utilize the acoustic scene taxonomy information. Firstly, a high-level DNN is discriminatively trained to predict three high-level classes, namely *vehicle*, *indoor* and *outdoor*. Then the trained DNN is used to initialize the low-level DNN except for the top classification layer to learn the more difficult low-level scene classes, namely *bus*, *home*, *park*, etc. This learning process is named as **hierarchical pre-training**, which follows the common "easiest thing first hardest second" learning experience of human [21]. Hierarchical pre-training is a supervised process which is different from the common Restricted Boltzmann machine (RBM) based unsupervised pre-training [15]. The second idea is based on a proposed **multi-level objective function**, which means the DNN not only predicts the target low-level scene classes, but also predicts the three high-level scene classes as the auxiliary task. It is actually a multi-task learning [22] which has been demonstrated to be effective in recent DNN-based speech enhancement [23] and speech recognition [24].

The rest of the paper is organized as follows. In Section 2 we describe the DNN-based acoustic scene classification baseline system. Hierarchical pre-training and multi-level objective function are presented in Section 3. In Section 4, we present a series of experi-

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of the UK under the grant EP/N014111/1.

ments to assess the system performance. Finally we summarize our findings in Section 5.

2. DNN-BASED ACOUSTIC SCENE CLASSIFICATION

DNN is a non-linear multi-layer model with powerful capability to extract robust feature related to a specific classification [13] or regression [12] task. ASC is a typical classification problem where a specific scene label should be assigned to an audio segment.

2.1. DNN baseline

A basic DNN consists of a number of different layers stacked together in a deep architecture: an input layer, several hidden layers and an output layer. More precisely, when the goal is to classify an audio feature $\mathbf{x}$ among N acoustic scene classes, a DNN estimates the posteriors p_j, $j \in \{1, ..., J\}$, of each class given the input feature $\mathbf{x}$. The input $\mathbf{x}$ which is fed into DNN represents the contextual audio feature, such as 11 consecutive frames centered at the current frame. Such contextual information was shown to improve the prediction performance in DNN-based speech enhancement or speech recognition [12, 13]. The activation functions used in each hidden unit of the hidden layers are non-linear sigmoid or Rectified Linear Units (ReLUs) [25] function. The ReLU, which is adopted in this work, has several advantages over the sigmoid: faster computation and more efficient gradient propagation and it is defined below:

$$f(y) = \max(0, y) \tag{1}$$

where y is the output of the hidden unit before activated by ReLU. The output is computed via the softmax nonlinearity to force the target label to have the maximum posterior while competing with other non-targets. The objective is to minimize the cross entropy between the predictions of DNN $\mathbf{p} = [p_1, ..., p_J]^T$ and the target probabilities $\mathbf{d} = [d_1, ..., d_J]^T$. The loss function is defined as follows:

$$L = -\sum_{j=1}^{J} d_j \cdot \log(p_j) \tag{2}$$

The classical back-propagation (BP) algorithm [13] can be used to update the weights and bias of DNN based on the calculated error.

2.2. Dropout for the over-fitting problem

Deep learning architectures have a natural tendency to over-fitting especially when there is a little training data. Dropout is a simple but effective way to alleviate this problem [25]. In each training iteration, the feature value of every input unit and the activation of every hidden unit are randomly removed with a predefined probability (e.g., ρ). These random perturbations effectively prevent the DNN from learning spurious dependencies. At the decoding stage, the DNN discounts all of the weights involved in the dropout training by $(1 - \rho)$, regarded as a model averaging process [26].

For the acoustic scene classification task, the testing audio segment could be totally different from the used training audio segments due to the presence of background noise. Thus Dropout should be adopted to improve its robustness to generalize to variants of testing segments.

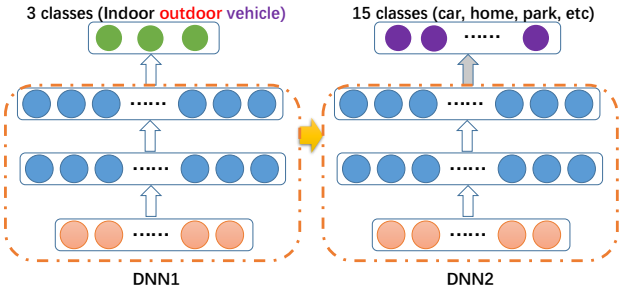


Figure 2: Proposed hierarchical pre-training.

2.3. Decision maker based on average confidence

ASC aims to assign a single semantic label to an audio segment. Majority voting is often used to make a global decision across all of the single audio frames in this segment [1]. Here we proposed to use a more precise decision making scheme:

$$\hat{c} = \max_j \left(\frac{1}{T} \sum_{t=1}^{T} p_{t,j} \right) \tag{3}$$

where T is the total number of frames belonging to the current testing audio segment, $\hat{c}$ denotes the predicted global scene label based on the average confidence across the whole frames, and $p_{t,j}$ represents the estimated DNN posterior at the t-th frame for class j.

3. PROPOSED HIERARCHAL LEARNING FOR ASC

In this section, we present two novel methods: hierarchical pre-training and multi-level objective function incorporating the scene taxonomy information for DNN-based ASC.

3.1. Hierarchical pre-training

Pre-training is crucial to avoid the algorithm getting stuck in a local optimum for training a deep model especially when the training data is not sufficient. The two most notable pre-training methods are the RBMs [15] based and stacked auto-encoders [27] based greedy layer-wise algorithms. They are both unsupervised while the proposed hierarchical pre-training is supervised. In the acoustic scene taxonomy research [20], the acoustic scenes are naturally categorized into hierarchical classes. Fig. 2 shows how the proposed DNN-based method incorporates the hierarchical taxonomy information. The hierarchical pre-training consists of two steps. Firstly, the DNN1 was trained to predict the three high-level acoustic scene classes, namely indoor, outdoor and vehicle. DNN2 was then trained to estimate the posterior of the 15 target low-level acoustic scene classes with the initialized weights from DNN1. Note that the classification layer of DNN2 was initialized with random weights because this top layer is different from DNN1. It is easier for DNN to learn the three coarsely classified high-level classes than the 15 target classes. However, the DNN2 can be better fine-tuned based on DNN1. It follows the common sense of human learning process: easiest things first, hardiest second. The experience of learning easier things could benefit the learning for harder things.

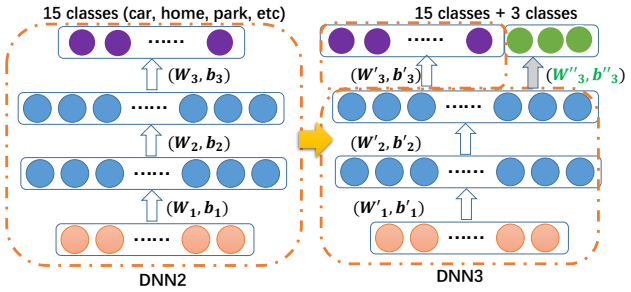


Figure 3: Proposed multi-level objective function based on the well trained DNN2. **W** and **b** denote the weights and bias, respectively.

3.2. Multi-level objective function

Multi-task learning [22] is successfully adopted in DNN-based speech enhancement [23] and DNN-based speech recognition [24]. The auxiliary target was demonstrated beneficial for the primary target. Inspired by this, a multi-level objective function is proposed to incorporate the hierarchical acoustic scene taxonomy information into the integrated objective function.

Fig. 3 shows the proposed multi-level objective function based on the well trained DNN2. The main difference between DNN2 and DNN3 is that an additional softmax layer is designed to describe the three high-level classes (indoor, outdoor and vehicle). **W** and **b** denote the weights and bias, respectively. $\mathbf{W}'$ and $\mathbf{b}'$ of DNN3 were both initialized by **W** and **b** of DNN2. The additional softmax layer $(\mathbf{W}'', \mathbf{b}'')$ was randomly initialized. With this modification, the cross entropy based loss function should be changed to contain two parts as follows:

$$L_{1:N} = -\alpha \sum_{t=1}^{N} \sum_{j=1}^{J} d_{t,j} \log(p_{t,j}) - (1-\alpha) \sum_{t=1}^{N} \sum_{k=1}^{K} d_{t,k} \log(p_{t,k}) \quad (4)$$

where N is the mini-batch size, $d_{t,j}$ denotes the target probability at the t-th frame for the j-th low-level scene class, $d_{t,k}$ denotes the DNN predicted posterior at the t-th frame for the k-th high-level scene class. α is the weighting factor to tune the error contribution from the above two parts. J and K represent the 15 low-level classes and the three high-level classes, respectively.

Hence, the proposed multi-level objective function is another idea to utilize the hierarchical scene taxonomy information besides the proposed pre-training in Sec. 3.1.

4. EXPERIMENTAL SETUP AND RESULTS

The proposed methods were evaluated on the Task1 of DCASE 2016 challenge. There are 15 acoustic scenes for this task [1]. Three high-level scene classes are also indicated. For all of the acoustic scenes, each of the recordings was captured in a different location: different streets, different parks and different homes. Recordings were made using a Soundman OKM II Klassik/studio A3, electret binaural microphone and a Roland Edirol R-09 wave recorder using 44.1 kHz sampling rate and 24 bit resolution. The recordings are down-sampled into 16 kHz in this paper. The microphones are specifically made to look like headphones, being worn in the ears. As an effect of this, the recorded audio is very similar to the sound that reaches the human auditory system of the person wearing the equipment.

The dataset consists of two subsets: a development dataset and an evaluation dataset. In this paper, only the development dataset is used for evaluation because the labels of the evaluation dataset have not been released. The development dataset contains 1170 segments in total with 30 seconds length for each. A cross-validation setup with four folds is provided for the development dataset. The scoring of acoustic scene classification will be based on classification accuracy. Each segment is considered as an independent test sample. Confusion matrix among various acoustic scene classes would also be presented.

The official baseline system is based on the MFCC acoustic features and GMM classifier. The system learns one acoustic model per acoustic scene class, and performs the classification with maximum likelihood classification scheme. The length of each frame is 40 ms with 50% hop size. The acoustic features include 20-dimension MFCC static coefficients (0th coefficient included), delta coefficients and acceleration coefficients.

For the DNN method, 11 frames of Mel-filter bank features with 40 channels were used as the input. Two hidden layers with 500 ReLU hidden units for each layer were adopted for DNN. The learning rate was 0.005. The momentum was set to 0.9. Weight cost was not used. The dropout value for the input layer was 0.1 while 0.3 for hidden layers. α in Eq. 4 was 0.6. NVIDIA-Tesla-M2090 GPU was used to train the DNN models. The output unit number for DNN1, DNN2 and DNN3 were 3, 15 and 18, respectively.

4.1. Evaluations for the proposed methods

As shown in Fig. 2, DNN1 should be trained as the pre-trained model for DNN2. Table 1 gives the frame-wise accuracy (%) for the three high-level scene classes on four cross-validation (CV) folds using DNN1. All of the related CV audio segments were excluded from the training sampels. An average of frame-level 90% accuracy can be obtained for the classification of three high-level acoustic scene classes, namely indoor, outdoor and vehicle. Therefore, DNN can easily deal with this learning. It would offer a good starting optimization point for the fine-tuning of DNN2 with the initialized weights from DNN1. Then the DNN2 was trained to predict the 15 target acoustic scene classes based on the well trained DNN1. Table 2 presented the overall comparison of acoustic scene accuracy (%) on different CV folds among the DCASE2016 official GMM baseline, the DNN baseline improved by dropout, the DNN2 with the hierarchical pre-training based on DNN baseline, and the DNN3 optimized by the proposed multi-level objective function based on

System	Fold 1	Fold 2	Fold 3	Fold 4	Average
DNN1	93%	90%	89%	91%	90%

Table 1: Frame-wise accuracy (%) for three high-level scene classes on different cross-validation (CV) folds using DNN1. All of the related CV audio segments were excluded from the training samples.

[1] C1: Lakeside beach (outdoor); C2: Bus, traveling by bus in the city (vehicle); C3: Cafe / Restaurant, small cafe/restaurant (indoor); C4: Car, driving or traveling as a passenger (vehicle); C5: City center (outdoor); C6: Forest path (outdoor); C7: Grocery store, medium size grocery store (indoor); C8: Home (indoor); C9: Library (indoor); C10: Metro station (indoor); C11: Office, multiple persons, typical work day (indoor); C12: Urban park (outdoor); C13: Residential area (outdoor); C14: Train (traveling, vehicle); C15: Tram (traveling, vehicle).

Systems	Fold 1 ACC (%)	Fold 2 ACC (%)	Fold 3 ACC (%)	Fold 4 ACC (%)	Average ACC (%)
GMM-baseline	72.50	66.80	70.10	75.70	71.28
DNN-baseline (+dropout)	79.62	67.24	75.84	78.08	75.19
DNN2 (+hierarchical pre-training)	80.69	71.72	77.52	78.77	77.17
DNN3 (++multi-level objective func)	**81.38**	**72.41**	**77.85**	**79.79**	**77.86**

Table 2: The overall comparison of acoustic scene accuracy (%) on different cross-validation (CV) folds among the DCASE2016 official GMM baseline, the DNN baseline improved by dropout, the DNN2 with the hierarchical pre-training based on DNN baseline, and the DNN3 optimized by the proposed multi-level objective function based on DNN2. All of the related CV audio segments were excluded from the training samples.

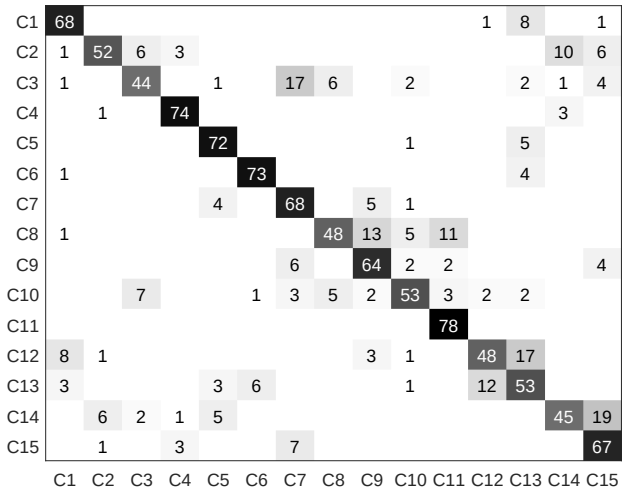


Figure 4: The confusion matrix among 15 acoustic scene classes by comparing the DNN predicted class with the target class on all of the four folds. $Cx, x \in \{1, ..., 15\}$ represent the indices of the 15 classes which were defined in footnote 1.

DNN2. All of the related CV audio segments were excluded from the training samples. The DNN baseline improved by dropout outperformed the provided GMM-MFCC baseline at all folds. The acoustic scene accuracy was increased from 71.28% to 75.19% on average. It also should be noted that the DNN is just slightly better than GMM on Fold 2 where the performance is the lowest. However, with the proposed hierarchical pre-training, its accuracy was significantly improved from 67.24% to 71.72% on Fold 2. Therefore, it demonstrates that the proposed hierarchical pre-training is important in challenging scene classification situations. DNN2 obtains an 8% relative improvement compared with the DNN baseline from 75.19% to 77.17%.

The DNN3 optimized by the proposed multi-level objective function gives further improvement. The final average acoustic scene accuracy was increased to 77.86%. It indicates that the additional constraint imposed in Eq. 4 can benefit the primary target. Finally, the proposed DNN system offers 22.9% and 10.8% relative improvements compared with the GMM-MFCC baseline and the DNN baseline, respectively. Note that the DNN baseline is a strong system since it is optimized by dropout training.

4.2. Further discussions

Fig. 4 presents the confusion matrix among 15 acoustic scene classes by comparing the DNN predicted class with the target class on all of the four folds. $Cx, x \in \{1, ..., 15\}$ represents the indices of the 15 classes which were defined in footnote 1. Observed from this confusion matrix, one phenomenon is that *park* ($C12$) easily gets confused by *residential area* ($C13$), and vice versa. It could be explained that similar acoustic events happened in both acoustic environments, like the *bird singing* and *car passing-by*. Another interesting case is that *grocery store* ($C7$) tends to be mis-recognized as *restaurant* ($C3$) due to the common human speech events. This might suggest that the presence of common human speech needs to be reduced in the audio segments before the acoustic scene classification is conducted. *Tram* ($C15$) also has the tendency to be incorrectly identified as *Train* ($C14$). Table 3 presents the final accuracy for the evaluation set. The final DNN model was trained with the whole 1170 segments of the development set. Note that the proposed method can only get 73.3% accuracy if the DNN was trained on Fold1 only (as given on DCASE2016 Task1 website).

System	Proposed method	DCASE2016 Baseline
acc	80.5%	77.2%

Table 3: Accuracy (%) for the final evaluation set.

5. CONCLUSIONS

In this paper, we have studied how to incorporate the taxonomy information into deep learning framework, and developed two DNN-based hierarchical learning methods for the acoustic scene classification task. The first novel method, called hierarchical pre-training which is a supervised learning process, can help the second DNN to get a better initialized weights based on the learning experience from the three high-level coarsely classified classes. It can achieve an 8% relative improvement compared with the DNN baseline improved by Dropout. The second proposed approach was the multi-level objective function which was inspired by the multi-task learning. It can help improve the prediction accuracy of the primary 15 target low-level classes by adding additional estimation of the three high-level classes in the DNN output, which was also regarded as imposing more constraint on the cross-entropy loss function. This idea can further improve the scene classification performance. Finally, the proposed DNN system has obtained 22.9% and 10.8% relative improvements over the GMM-MFCC baseline and the well trained DNN baseline, respectively.

6. REFERENCES

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

[2] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *IEEE 18th European Signal Processing Conference*, 2010, pp. 1267–1271.

[3] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1543–1551, 2010.

[4] N. Sawhney and P. Maes, "Situational awareness from environmental sounds," *URL: http://web. media. mit. edu/˜nitin/papers/Env_Snds/EnvSnds. html*, 1997.

[5] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. Plumbley, "IEEE AASP challenge: Detection and classification of acoustic scenes and events," Queen Mary University of London, Tech. Rep., 2013.

[6] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013, pp. 1–4.

[7] 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, 2015.

[8] J. T. Geiger, B. Schuller, and G. Rigoll, "Large-scale audio feature extraction and svm for acoustic scene classification," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013, pp. 1–4.

[9] K. Lee, Z. Hyung, and J. Nam, "Acoustic scene classification using sparse feature learning and event-based pooling," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013, pp. 1–4.

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

[11] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[12] ——, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] Y. Petetin, C. Laroche, and A. Mayoue, "Deep neural networks for audio scene recognition," in *IEEE 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 125–129.

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

[18] M. Ravanelli, B. Elizalde, K. Ni, and G. Friedland, "Audio concept classification with hierarchical deep neural networks," in *2014 Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 606–610.

[19] M. Niessen, C. Cance, and D. Dubois, "Categories for soundscape: Toward a hybrid classification," in *Inter-Noise and Noise-Con Congress and Conference Proceedings*, vol. 2010, no. 5, 2010, pp. 5816–5829.

[20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 1041–1044.

[21] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1721–1728.

[22] R. Camana, "Multitask learning: A knowledge-based source of inductive bias," in *Proceedings of the Tenth International Conference on Machine Learning*, 1993, pp. 41–48.

[23] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," in *Sixteenth Annual Conference of the International Speech Communication Association INTERSPEECH*, 2015.

[24] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, "Rapid adaptation for deep neural networks through multi-task learning," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[25] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8609–8613.

[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[27] 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," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

FULLY DNN-BASED MULTI-LABEL REGRESSION FOR AUDIO TAGGING

Yong Xu, Qiang Huang*, Wenwu Wang, Philip J. B. Jackson, Mark D. Plumbley*

Centre for Vision, Speech and Signal Processing, University of Surrey, UK
{yong.xu, q.huang, w.wang, p.jackson, m.plumbley}@surrey.ac.uk

ABSTRACT

Acoustic event detection for content analysis in most cases relies on a lot of labeled data. However, manually annotating data is a time-consuming task, which thus makes few annotated resources available so far. Unlike audio event detection, automatic audio tagging, a multi-label acoustic event classification task, only relies on weakly labeled data. This is highly desirable to some practical applications using audio analysis. In this paper we propose to use a fully deep neural network (DNN) framework to handle the multi-label classification task in a regression way. Considering that only chunk-level rather than frame-level labels are available, the whole or almost whole frames of the chunk were fed into the DNN to perform a multi-label regression for the expected tags. The fully DNN, which is regarded as an encoding function, can well map the audio features sequence to a multi-tag vector. A deep pyramid structure was also designed to extract more robust high-level features related to the target tags. Further improved methods were adopted, such as the Dropout and background noise aware training, to enhance its generalization capability for new audio recordings in mismatched environments. Compared with the conventional Gaussian Mixture Model (GMM) and support vector machine (SVM) methods, the proposed fully DNN-based method could well utilize the long-term temporal information with the whole chunk as the input. The results show that our approach obtained a 15% relative improvement compared with the official GMM-based method of DCASE 2016 challenge.

Index Terms— Audio tagging, deep neural networks, multi-label regression, dropout, DCASE 2016

1. INTRODUCTION

Due to the use of smart mobile devices in recent years, huge amounts of multimedia data are generated and uploaded to the internet everyday. These data, such as music, field sounds, broadcast news, and television shows, contain sounds from a wide variety of sources. The need for analyzing these sounds has been now increased as it is useful, e.g., for automatic tagging in audio indexing, automatic sound analysis for audio segmentation or audio context classification. Although supervised approaches have proved to be effective in many applications, their effectiveness relies heavily on the quantity and quality of the training data. Moreover, manually labeling a large amount of data is very time-consuming. To handle this problem, two types of methods have been developed. One is to convert low-level acoustic features into "bag of audio words" using unsupervised learning methods [1, 2, 3, 4, 5]. The second type of methods is based on only weakly labeled data [6], e.g. audio tagging. It is clear that tagging audio chunks needs much less time compared to precisely locating event boundaries within recordings. This will certainly improve tractability of obtaining manual annotations for large databases. In this paper, we will focus on the audio tagging task.

To overcome the lack of annotated training data, Multiple Instance Learning (MIL) is proposed in [7] as a variation of supervised learning for problems with incomplete knowledge about labels of training examples. It aims to classify sets of instances instead of recognizing single instances. Following this work, Andrews *et al.* [8] proposed a new formulation of MIL as a maximum margin problem, which had led to some further work [9, 10, 11, 12, 13] in audio and video processing using weakly labeled data. Mandel and Ellis in [9] used clip-level tags to derive tags at the track, album, and artist granularities by formulating a number of music information related multiple-instance learning tasks and evaluated two MIL based algorithms on them. In [14], Phan *et al.* used event-driven MIL to learn the key evidences for event detection. Recently, [6] also presented a SVM based MIL system for audio tagging and event detection. GMM, as a common model, was used as the official baseline method in DCASE 2016 for audio tagging. More details can be found in [15].

Although the methods mentioned above have led to some useful results in detection and analysis of audio data, most of them ignored possible relationships of any contextual information and only focused on training the model for each single event class independently. To better use the data with weak labels, our work will utilize the whole or almost whole frames of the observed chunk as the input of a fully deep neural network to make a mapping from an audio feature sequence to a multi-tag vector.

Recently, deep learning technologies have obtained great successes in speech, image and video fields [16, 17, 18, 19] since Hinton and Salakhutdinov showed the insights using a greedy layer-wise unsupervised learning procedure to train a deep model in 2006 [20]. The deep learning methods were also investigated for related tasks, like acoustic scene classification [21] and acoustic event detection [22]. And better performance could be obtained in these tasks. For music tagging task, [23, 24] have also demonstrated the superiority of deep learning methods. However, to the best of our knowledge, the deep learning based methods have not been used for environmental audio tagging which is a newly proposed task in DCASE 2016 challenge based on the CHiME-home dataset [25]. For the audio tagging task, only the chunk-level instead of frame-level labels were available. Furthermore, multiple instances could happen simultaneously, for example, the *child speech* could exist with *TV sound* for several seconds. Hence, a good way is to feed the DNN with the whole frames of the chunk to predict the multiple tags in the output.

In this paper, we propose a fully DNN-based method, which can

*This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of the UK under the grant EP/N014111/1.

†The first author and the second author have equal contribution for this paper.

well utilize the long-term temporary information, to map the whole sequence of audio features into a multi-tag vector. The fully neural network structure was also successfully used in image segmentation [26]. To get a better prediction of the tags, a deep pyramid structure is designed with gradually shrinked size of layers. This deep pyramid structure can reduce the non-correlated interferences in the whole audio features while focusing on extracting the robust high-level features related to the target tags. Dropout [27] and background noise aware training [28] are adopted to further improve the tagging performance in the DNN-based framework.

The rest of the paper is organized as follows. In section 2, we will introduce the related work using GMM and SVM based MIL in detail, and depict our DNN based framework in section 3. The data description and experimental setup will be given in section 4. We will show the related results and discussions in section 5, and finally draw a conclusion in section 6.

2. RELATED WORK

Two baseline methods compared in our work are briefly summarized below.

2.1. Audio Tagging using Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are a commonly used generative classifier. A GMM is parametrized in $\Theta = \{\omega_m, \mu_m, \Sigma_m\}, m = \{1, \cdots, M\}$, where M is the number of mixtures and w_m is the weight of the m-th mixture component.

To implement multi-label classification with simple event tags, a binary classifier is built associating with each audio event class in the training step. For a specific event class, all audio frames in an audio chunk labeled with this event are categorized into a positive class, whereas the remaining features are categorized into a negative class. On the classification stage, given an audio chunk C_i, the likelihoods of each audio frame $x_{ij}, (j \in \{1 \cdots L_{C_i}\})$ are calculated for the two class models, respectively. Given audio event class k and chunk C_i, the classification score $S_{C_{ik}}$ is obtained as log-likelihood ratio:

$$S_{C_{ik}} = \sum_j log(f(x_{ij}, \Theta_{pos})) - \sum_j log(f(x_{ij}, \Theta_{neg})) \quad (1)$$

2.2. Audio Tagging using Multiple Instance SVM

Multiple instance learning is described in terms of bags $\mathbf{B}$. The jth instance in the ith bag, B_i, is defined as x_{ij} where $j \in I = \{1 \cdots l_i\}$, and l_i is the number of instances in B_i. B_i's label is $Y_i \in \{-1, 1\}$. If $Y_i = -1$, then $x_{ij} = -1$ for all j. If $Y_i = 1$, then at least one instance $x_{ij} \in B_i$ is a positive example of the underlying concept [8].

As MI-SVM is the bag-level MIL support vector machine to maximize the bag margin, we define the functional margin of a bag with respect to a hyper-plane as:

$$\gamma_i = Y_i \max_{j \in I}(\langle \mathbf{w}, \mathbf{x}_{ij} \rangle + b) \quad (2)$$

Using the above notion, MI-SVM can be defined as:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}^2\| + A\sum_i \xi_i \quad (3)$$
$$\text{subject to:} \quad \forall_i : \gamma_i \geq 1 - \xi_i, \;\; \xi_i \geq 0$$

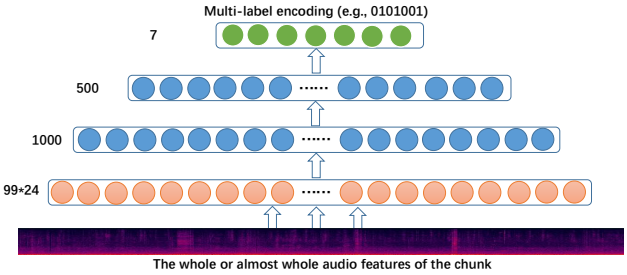


Figure 1: Fully DNN-based audio tagging framework using the deep pyramid structure.

where $\mathbf{w}$ is weight vector, b is bias, ξ is margin violation, and A is a regularization parameter.

Classification with MI-SVM proceeds in two steps. In the first step, $\mathbf{x}_i$ is initialized as the centroid for every positive bag B_i as follows

$$\bar{\mathbf{x}}_i = \sum_{j \in I} \mathbf{x}_{ij}/l_i \quad (4)$$

The second step is an iterative procedure in order to optimize the parameters.

Firstly, $\mathbf{w}$ and b are computed for the data set with positive samples $\{x_I : Y_i = 1\}$.

Secondly, we compute

$$f_{ij} = \langle \mathbf{w}, \mathbf{x}_{ij} \rangle + b, \qquad \mathbf{x}_{ij} \in \mathbf{B}_i$$

Thirdly, we change $\bar{\mathbf{x}}_i$ by

$$\bar{\mathbf{x}}_i = \mathbf{x}_j$$
$$j = \arg\max_{j \in I} f_{ij}, \forall I, Y_I = 1$$

The iteration in this step will stop when there is no change of $\bar{\mathbf{x}}_i$, The optimized parameters will be used for test.

3. PROPOSED FULLY DNN-BASED AUDIO TAGGING

DNN is a non-linear multi-layer model for extracting robust features related to a specific classification [18] or regression [17] task. The objective of the audio tagging task is to perform multi-label classification on audio chunks (i.e. assign zero or more labels to each audio chunk of a length e.g. four seconds in our experiments). This chunk only has utterance-level labels without frame-level labels. Multiple events happen at many particular frames. Hence, the common frame-level cross entropy based loss function can not be adopted. We propose a method to encode the whole or almost whole chunk.

3.1. Fully DNN-based multi-label regression using sequence to sequence mapping

Fig. 1 shows the proposed fully DNN-based audio tagging framework using the deep pyramid structure. With the proposed framework, the whole or almost whole audio features of the chunk are encoded into a vector with values $\{0, 1\}$ in a regression way. Sigmoid was used as the activation function of the output layer to learn the presence probability of certain events. Minimum mean squared error (MMSE) was adopted as the objective function. A stochastic gradient descent algorithm is performed in mini-batches with mul-

tiple epochs to improve learning convergence as follows,

$$Er = \frac{1}{N}\sum_{n=1}^{N} \|\hat{\mathbf{X}}_n(\mathbf{Y}_{n-\tau}^{n+\tau}, \mathbf{W}, \mathbf{b}) - \mathbf{X}_n\|_2^2 \quad (5)$$

where Er is the mean squared error, $\hat{\mathbf{X}}_n(\mathbf{Y}_{n-\tau}^{n+\tau}, \mathbf{W}, \mathbf{b})$ and $\mathbf{X}_n$ denote the estimated and reference tag vector at sample index n, respectively, with N representing the mini-batch size, $\mathbf{Y}_{n-\tau}^{n+\tau}$ being the input audio feature vector where the window size of context is $2*\tau+1$. It should be noted that the input window size should cover the whole or almost whole of the chunk considering that the reference tags are in chunk-level rather than frame-level labels. However, slightly relaxing the window size without covering all of the chunk frames could increase the total training samples for DNN. It can improve the performance in our experiments. $(\mathbf{W}, \mathbf{b})$ denoting the weight and bias parameters to be learned. The updated estimate of $\mathbf{W}^\ell$ and $\mathbf{b}^\ell$ in the ℓ-th layer, with a learning rate λ, can be computed iteratively as follows:

$$(\mathbf{W}^\ell, \mathbf{b}^\ell) \leftarrow (\mathbf{W}^\ell, \mathbf{b}^\ell) - \lambda \frac{\partial Er}{\partial(\mathbf{W}^\ell, \mathbf{b}^\ell)},\ 1 \le \ell \le L+1 \quad (6)$$

where L denotes the total number of hidden layers and $L+1$ represents the output layer.

During the learning process where the DNN can be regarded as an encoding function, the audio tags are automatically predicted. Hence the multi-label regression rather than classification can be conducted. Two additional methods are given below to improve the DNN-based audio tagging performance.

3.2. Dropout for the over-fitting problem

Deep learning architectures have a natural tendency towards overfitting especially when there is little training data. This audio tagging task only has about four hours training data with imbalanced training data distribution for each type of tag. Dropout is a simple but effective way to alleviate this problem [27]. In each training iteration, the feature value of every input unit and the activation of every hidden unit are randomly removed with a predefined probability (e.g., ρ). These random perturbations effectively prevent the DNN from learning spurious dependencies. At the decoding stage, the DNN discounts all of the weights involved in the dropout training by $(1-\rho)$, regarded as a model averaging process [29].

A mismatch problem may also exist in this task, and testing audio segments could be totally different from existed training audio segments due to the presence of lots of background noise. Thus Dropout should be adopted to improve its robustness to generalize to variation in testing segments.

3.3. Background noise aware training

Different types of background noise in different recording environments could lead to the mismatch problem between the testing chunks and the training chunks. To alleviate this, we propose a simple background noise aware training (or background noise adaptation method). To enable this noise awareness, the DNN is fed with the primary audio features augmented with an estimate of the background noise. In this way, the DNN can use additional on-line background noise information to better predict the expected tags. The background noise is estimated as follows:

$$\mathbf{V}_n = [\mathbf{Y}_{n-\tau}, ..., \mathbf{Y}_{n-1}, \mathbf{Y}_n, \mathbf{Y}_{n+1}, ..., \mathbf{Y}_{n+\tau}, \hat{\mathbf{Z}}_n] \quad (7)$$

$$\hat{\mathbf{Z}}_n = \frac{1}{T}\sum_{t=1}^{T} \mathbf{Y}_t \quad (8)$$

where the background noise $\hat{\mathbf{Z}}_n$ is fixed over the utterance and estimated using the first T frames. Although this noise estimator is simple, a similar idea was shown to be effective in DNN-based speech enhancement [17, 28].

4. EXPERIMENTAL SETUP AND RESULTS

4.1. DCASE2016 data set for audio tagging

The data that we used for evaluation is the dataset of Task4 of DCASE 2016 [15]. The audio recordings are made in a domestic environment. The audio data are provided as 4-second chunks at two sampling rates (48kHz and 16kHz) with the 48kHz data in stereo and with the 16kHz data in mono. The 16kHz recordings were obtained by downsampling the right-hand channel of the 48kHz recordings. Each audio file corresponds to a single chunk [15].

For each chunk, multi-label annotations were first obtained from each of the 3 annotators. The annotations are based on a set of 7 label classes. A detailed description of the annotation procedure is provided in [25]. To reduce uncertainty of the test data, the evaluation is based on those chunks where 2 or more annotators agreed about label presence across label classes. Moreover, with the aim of approximating typical recording capabilities of commodity hardware, only the monophonic audio data sampled at 16kHz are used for test.

4.2. Experimental Setup

In our experiments, following the original configuration of Task4 of DCASE 2016 [15], we use the same five folds as the evaluation set from the given development dataset, and use the remain of the audio recordings for training.

We pre-process each audio chunk by segmenting them using a (80ms) sliding window with a 40ms hop size, and converting each segment into 24-D MFCCs. For each 4-second chunk, 99 frames of MFCCs are obtained. A 91-frame expansion as the input instead of the total frames were found better because this relaxed input scheme can increase the total training samples. Hence the input size of DNN was 2208 with 91-frame MFCCs and also the appended noise vector. One hidden layer with 1000 units and the second hidden layer with 500 units were used to construct a pyramid structure. Seven sigmoid outputs were adopted to predict the seven tags. The learning rate was 0.005. Momentum was set to be 0.9. The dropout rates for input layer and hidden layer were 0.1 and 0.2, respectively. The mini-batch size was 3. T in Equation 8 was 6. It should be noted that the remaining 2432 chunks without 'strongly agreement' labels in the development dataset were also added into the DNN training considering that DNN has a better fault-tolerant capability. Meanwhile, these 2432 chunks without 'strongly agreement' labels were also added into the training data for GMM and SVM training.

For a comparison, we also ran two baselines using GMMs and the MI-SVM mentioned in Section 2. For the GMM based method, the number of mixture components is 8. Since the GMM based baseline focuses on computing frame-level likelihoods and MI-SVM prefers to instance-level scores, the sliding window and hop size set for the two baselines are different. The GMM based baseline uses a 20ms sliding window with 10ms hop size, while the sliding window and hop size for MI-SVM are set to be 400ms

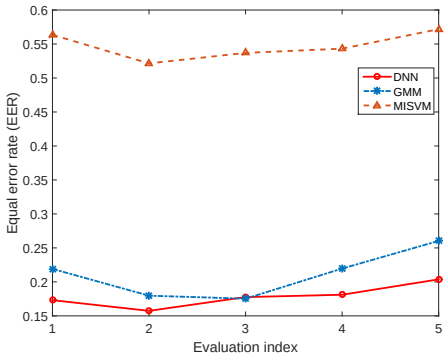


Figure 2: Equal error rates obtained using the proposed fully DNN based approach and the two baselines, namely GMM and MI-SVM across five evaluations folds of the development set.

and 200ms, respectively. To handle audio tagging with MI-SVM, each audio recording will be viewed as a bag and its shorter segments obtained by a sliding window can be treated as an instance. To accelerate computation, we use linear function kernel in our experiments.

To evaluate the effectiveness of our approach, as compared with the two baselines, we use equal error rate (EER) as a metric. EER is defined as the point of the graph of false negative rate (FNR) versus false positive rate (FPR) [30]

$$FNR = \frac{\#false\ negative}{\#positive}$$

$$FPR = \frac{\#false\ positive}{\#negative}$$

EERs are computed individually for each evaluation, and we then average the obtained EERs across the five evaluations to get the final performance.

5. RESULTS AND DISCUSSIONS

Figure 2 shows the results obtained using our approach and two baselines. Fully DNN-based approach outperforms the two baselines across the five-fold evaluations. This may be because of the following two main reasons: First, our proposed approach can well utilize the long-term temporary information instead of treating those information independently. Second, it can map the whole audio features sequence into a multi-tag vector by working as an encoding function. However, GMM and SVM based methods build the models only on single instances. The contextual information and the potential relationship among different tags were not well utilized.

The GMM based method yields a close performance to the proposed method only in the third evaluation. We find that two of the audio event classes, namely adult male's speech (label 'm') and other identifiable sounds (label 'o'), are well identified in this fold evaluation. This case is probably because the acoustic characteristics and their variations of the two event classes in the evaluation data can match with the trained models. The use of MI-SVM does not yield competitive performances in comparison with our proposed approach and the GMM-based baseline. This is because MI-SVM, actually working as a discriminative learning, is more sensitive to the quantity and quality of the used training data. Furthermore, MI-SVM does not use the long contextual information.

Table 1: Average EER among the proposed fully DNN method, GMM and MI-SVM methods, for each event across five-fold evaluations of the development set.

Various tag	Proposed DNN	GMM	MI-SVM
b	0.0868	**0.0755**	0.1672
c	**0.1686**	0.2107	0.6466
f	**0.2409**	0.3037	0.7626
m	**0.1943**	0.2847	0.7046
o	**0.2867**	0.2903	0.7303
p	**0.2197**	0.2613	0.6724
v	0.0530	**0.0484**	0.1481
Average	**0.1785**	0.21	0.5474

For a further comparison, Table 1 shows the detailed performances obtained using our approach and the two baselines on each audio tag. We can easily find that the use of the fully-DNN based approach yields great improvements over the two baselines across all of the seven audio tags. Compared with the GMM method, the proposed fully DNN method could get similar performance on tag 'b' and 'v', but it can significantly outperform the competing counterparts on some difficult tags. On average, the proposed DNN method could get a relative 15% improvement by contrasting with the GMM baseline.

System	Proposed method	DCASE2016 Baseline
EER	19.0%	20.9%

Table 2: EER (%) for the final evaluation set.

Table 2 presents the final EER for the evaluation set. The final DNN model was trained with the whole segments of the development set. Note that the proposed method achieve only 19.5% EER if the DNN was trained on Fold1 only (as on the DCASE2016 Task4 website).

6. CONCLUSIONS

In this paper we have presented to use a fully-DNN based approach to handle audio tagging with weak labels, in the sense that only the chunk-level instead of the frame-level labels are available. This fully DNN is regarded as an encoding function to map the audio features sequence to a multi-tag vector in a regression way. To extract robust high-level features, a deep pyramid structure was designed to reduce most of the non-correlated interfering features while keeping the highly related features. The dropout and background noise aware training methods were adopted to further improve its generalization capacity for new recordings in unseen environments. We tested our approach on the dataset of the Task4 of the DCASE 2016 challenge, and obtained significant improvements over two baselines, namely GMM and MI-SVM. Compared with the official GMM-based baseline system given in the DCASE 2016 challenge, the proposed DNN system could reduce the EER from 0.21 to 0.1785 on average. For the future work, we will use fully convolutional neural network (CNN) to extract more robust high-level features for the audio tagging task.

7. REFERENCES

[1] G. Chen and B. Han, "Improve k-means clustering for audio data by exploring a reasonable sampling rate," in *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, 2010, pp. 1639–1642.

[2] M. Riley, E. Heinen, and J. Ghosh, "A text retrieval approach to contnent-based audio retrieval," in *Proceedings of International Conference on Music Information Retrieval*, 2008, pp. 1639–1642.

[3] X. Shao, C. Xu, and M. Kankanhalli, "Unsupervised classification of music genre using hidden markov model," in *Proceedings of International Conference on Multimedia and Expo*, 2004, pp. 2023–2026.

[4] R. Cai, L. Lu, and A. Hanjalic, "Unsupervised content discovery in composite audio," in *Proceedings of International Conference on Multimeida*, 2005, pp. 628–637.

[5] T. Sainath, D. Kanevsky, and G. Ivengar, "Unsupervised audio segmentation using extended Baum-Welch transformations," in *Proceedings of International Conference on Acoustic, Speech and Signal Processing*, 2007, pp. 2009–2012.

[6] A. Kumar and B. Raj, "Audio event detection using weakly labeled data," *CoRR*, vol. abs/1605.02401, 2016. [Online]. Available: http://arxiv.org/abs/1605.02401

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[8] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proceedings of Advances in Neural Information Processing Systems*, 2003, pp. 557–584.

[9] M. Mandel and D. Ellis, "Multiple-instance learning for music information retrieval," in *The International Society of Music Information Retrieval*, 2008, pp. 577–582.

[10] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, and R. Raich, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *Journal of Acoustic Society of America*, vol. 131, pp. 4640–4650, 2012.

[11] P. Carbonetto, G. Dorko, C. Schmid, H. Kuck, and N. D. Freitas, "Learning to recognize objects with little supervision," *International Journal of Computer Vision*, vol. 77, pp. 219–237, May 2008.

[12] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1931–1947, 2006.

[13] A. Ulges, C. Schulze, and T. M. Breuel, "Multiple instance learning from weakly labeled videos," in *Proceedings of the Workshop on Cross-Media Information Analysis, Extraction and Management*, 2008, pp. 17–24.

[14] S. Phan, D. D. Le, and S. Satoh, "Multimedia event detection using event-driven multiple instance learning," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1255–1258.

[15] http://www.cs.tut.fi/sgn/arg/dcase2016/task-audio-tagging.

[16] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[17] ——, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[18] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[21] Y. Petetin, C. Laroche, and A. Mayoue, "Deep neural networks for audio scene recognition," in *23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 125–129.

[22] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[23] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *ICASSP*, 2014, pp. 6964–6968.

[24] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *arXiv preprint arXiv:1606.00298*, 2016.

[25] P. Foster, S. Sigtia, S. Krstulovic, J. Barker, and M. Plumbley, "CHiME-home: A dataset for sound source recognition in a domestic environment," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015, pp. 1–5.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[27] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),*, 2013, pp. 8609–8613.

[28] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks." in *INTERSPEECH*, 2014, pp. 2670–2674.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[30] K. P. Murohy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

GATED RECURRENT NETWORKS APPLIED TO ACOUSTIC SCENE CLASSIFICATION AND ACOUSTIC EVENT DETECTION

Matthias Zöhrer and Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria
matthias.zoehrer@tugraz.at, pernkopf@tugraz.at

ABSTRACT

We present two resource efficient frameworks for acoustic scene classification and acoustic event detection. In particular, we combine gated recurrent neural networks (GRNNs) and linear discriminant analysis (LDA) for efficiently classifying environmental sound scenes of the IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE2016). Our system reaches an overall accuracy of 79.1% on DCASE 2016 task 1 development data, resulting in a relative improvement of 8.34% compared to the baseline GMM system. By applying GRNNs on DCASE2016 real event detection data using a MSE objective, we obtain a segment-based error rate (ER) score of 0.73 – which is a relative improvement of 19.8% compared to the baseline GMM system. We further investigate semi-supervised learning applied to acoustic scene analysis. In particular, we evaluate the effects of a hybrid, i.e. generative-discriminative, objective function.

Index Terms— Acoustic Scene Labeling, Gated Recurrent Networks, Deep Linear Discriminant Analysis, Semi-Supervised Learning

1. INTRODUCTION

In acoustic scene classification the acoustic environment is labeled. Many different features, representing the scene, and models have been suggested in a recent acoustic scene classification challenge, summarized in [1]. One of the most popular baseline models are Gaussian mixture models (GMMs) [2] or hidden Markov models (HMMs) [3, 4] using mel-frequency cepstral coefficients (MFCCs). Interestingly, various deep architectures have not been applied in [1]. Recent work however, shows that deep neural networks (DNNs) boost the classification accuracy when applied to audio data [5]. In particular, Cakir et al. [6, 7] proposed a DNN architecture for acoustic scene classification. In [8], long-short-term memory networks (LSTMs), i.e. DNNs capable of modeling temporal dependencies, were applied to acoustic keyword spotting. Performance in recognition comes at the expense of computational complexity and the size of labeled data available. LSTMs have a relatively high model complexity. Furthermore, parameter tuning for LSTMs is not always simple.

This work was supported by the Austrian Science Fund (FWF) under the project number P27803-N15 and the K-Project ASD. The K-Project ASD is funded in the context of COMET Competence Centers for Excellent Technologies by BMVIT, BMWFJ, Styrian Business Promotion Agency (SFG), the Province of Styria - Government of Styria and the Technology Agency of the City of Vienna (ZIT). The program COMET is conducted by Austrian Research Promotion Agency (FFG). Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

Due to the great success of deep recurrent networks for sequence modeling [9, 10], we advocate gated recurrent neural networks (GRNNs) [11, 12, 13] for acoustic scene and event classification. GRNNs are a temporal deep neural network with reduced computational complexity compared to LSTMs. We evaluate GRNNs on environmental sound scenes of the IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE2016) [14]. GRNNs proof themselves in practice through fast and stable convergence rates. We obtain an overall accuracy of 79.1% on development data, i.e. a relative improvement of 8.34% compared to the baseline GMM system, using GRNNs and linear discriminant analysis (LDA). Furthermore, we used GRNNs for acoustic event detection, i.e. task 3 in DCASE2016. For this task we obtain a segment-based error rate (ER) of 0.82 and 0.63 for the scene categories *home* and *residential area*, respectively.

This work is structured as follows: Firstly, we introduce GRNNs in Section 3. In Section 4 we discuss various regularizers for GRNNs. Finally, we show experimental results for the challenge data in Section 7 and draw a conclusion in Section 8, respectively.

2. ACOUSTIC ANALYSIS FRAMEWORKS

2.1. Scene Classification Framework

Figure 1 shows the processing pipeline of our acoustic scene analysis framework. We extract sequences of features frames x_f, where $x_f \in \mathbb{R}^D$. In particular, we derive *MFCCs* or *log-magnitude spectrograms*, given the raw audio data x_t. We feed frequency domain features into the GRNN and estimate a class label for every frame f. Finally, we compute a histogram over all classified frames of the audio segment, where the maximum value determines the final scene class.

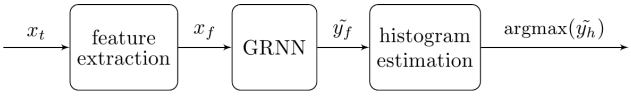


Figure 1: DCASE2016 task 1: GRNN scene classification system.

2.2. Event Detection Framework

Figure 2 shows the processing pipeline of our acoustic event detection framework. Similar as above, we extract sequences of feature frames x_f of *MFCCs* or *log-magnitude spectrograms*, given the raw audio data x_t. These feature frames are processed by a GRNN and

class labels are determined by applying individual thresholds on the real-valued output of the GRNN. Similar as in [14], we post-process the events by detecting contiguous regions neglecting events smaller than 0.1 seconds as well as ignoring consecutive events with a gap smaller than 0.1 seconds.

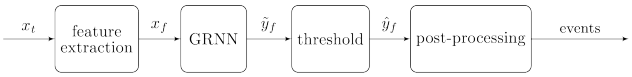


Figure 2: DCASE2016 task 1: GRNN event detection system.

3. DISCRIMINATIVE GRNNS

GRNNs are recurrent neural networks (RNNs) using blocks of gated recurrent units. GRNNs are a simple alternative to LSTMs, reaching comparable performance, but having fewer parameters. They only use *reset*- and *update*-gates. These switches couple static and temporal information allowing the network to learn temporal information. In particular, the *update*-gate z decides to re-new the current state of the model, whenever an important event is happening, i.e. some relevant information is fed into the model at step f. The *reset*-gate r is able to delete the current state of the model, allowing the network to forget the previously computed information. Figure 3 shows the corresponding flow diagram of a GRNN layer, respectively. It gives a visual interpretation how the *update*- and *reset*-gates, i.e. z and r, govern the information in the network.

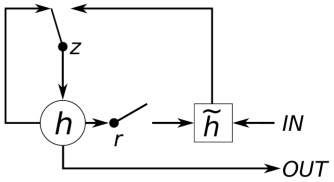


Figure 3: Flow graph of one GRNN layer [12].

The Equations (1-4) model the network behavior, mathematically. Starting at the output state h_f^l of layer l, the network uses the *update*-state z_f^l to compute a linear interpolation between past state h_{f-1}^l and current information $\tilde{h}_f^l$ in (1). In particular, the *update*-state z_f^l decides how much the unit updates its content; z_f^l is computed as sigmoid function of input x_f^l and the past hidden state h_{f-1}^l in Equation (2). The weights and bias terms in the model are denoted as W and b, respectively.

$$h_f^l = (1 - z_f^l)h_{f-1}^l + z_f^l \tilde{h}_f^l \quad (1)$$

$$z_f^l = \sigma(W_z^l x_f^l + W_{hz}^l h_{f-1}^l + b_z^l) \quad (2)$$

$$\tilde{h}_f^l = g(W_x^l x_f^l + W_{hh}^l (r_f^l \cdot h_{f-1}^l) + b_h^l) \quad (3)$$

$$r_f^l = \sigma(W_r^l x_f^l + W_{hr}^l h_{f-1}^l + b_r^l) \quad (4)$$

The state $\tilde{h}_f^l$ of the network is computed by applying a non-linear function g to the affine transformed input and previous hidden state h_{f-1}^l in (3). This is similar to *vanilla* RNNs. However, an additional *reset*-state, i.e. r_f^l, is introduced in GRNNs. In particular, an element-wise multiplication is applied between r_f^l and h_{f-1}^l. In (4), the reset state is computed based on the current input frame x_f and the provided hidden state h_{f-1}^l. Multiple GRNN layers can be stacked, forming a deep neural network.

4. DISCRIMINATIVE-GENERATIVE GRNNS

Recent advances in the field of semi-supervised learning combines discriminative learning objectives with generative cost terms [15, 16, 17, 18]. In particular, by modeling the data frame x_f using unlabeled examples, discriminative training objectives are regularized to prevent overfitting. These so called *hybrid* architectures outperform pure discriminative models if little labeled information is available. In order to exploit this regularization constraint, a reconstruction $\tilde{x}_f$ of the input frame x_f is computed by routing the network's output $\tilde{y}_f$ back to the bottom layer. This is done in any auto-encoder network by default [19], but could be also achieved via a separate *decoder*-network, visualized in Figure 4.

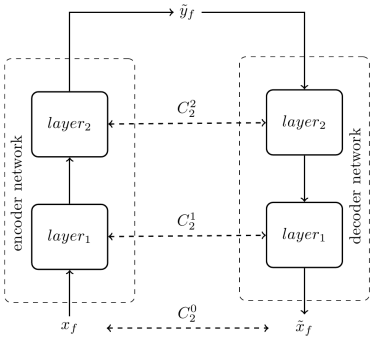


Figure 4: Flow graph of 2-layer hybrid GRNN network.

Following the idea of [15], we add an additional GRNN *decoder* network to the model. In particular, we use a noisy version of the input, i.e. $x_f + \mathcal{N}(\mu{=}0, \sigma{=}1)$, compute the output activation and feed the network's output $\tilde{y}_f$ back into the *decoder* network, which passes the information layer-by-layer down to the bottom and compute a reconstruction $\tilde{x}_f$ of the input. Next, the MSE between every hidden states h^l of a clean encoder and noisy decoder is computed. Adding this generative regularization term to the network's objective leads to the following hybrid objective function:

$$C = \underbrace{C_1(\tilde{y}_f, y_f)}_{\textit{discriminative}} + \lambda \cdot \underbrace{\frac{1}{L}\sum_{l=0}^{L} C_2^l(h_e^l, h_d^l)}_{\textit{generative}}, \quad (5)$$

where C_1 and C_2 are specific cost functions, such as the MSE criteria. The variable $\tilde{y}_f$ is the network's output and y_f is the current target label. The states h_e^l denote the hidden states of the *encoder* and h_d^l the hidden states of the *decoder*, respectively. The variable λ determines the tradeoff between the generative and discriminative objective.

5. VIRTUAL ADVERSARIAL TRAINING

Virtual adversarial training (VAT) [20, 17, 21] regularizes discriminative learners by generating *adversarial* training examples. Given

a clean training example x_f, an input noise pattern $\tilde{n}$ is generated by maximizing the KL-divergence between $P(y_f|x_f)$ and $P(y_f|x_f + n)$ using a softmax output layer, where the noise n is limited to $||n||_2 < \epsilon$, i.e. to the sphere of radius ϵ located around x_f. This means that the perturbed $x_f + \tilde{n}$ maximally changes the KL divergence between the posterior distributions, i.e the virtual adversarial example is most sensitive with respect to the KL-divergence. The newly obtained adversarial sample $\tilde{x}_f = x_f + \tilde{n}$ is used as additional training example. VAT can be used as a semi-supervised learning criteria. In this case a contractive cost term is applied on unlabeled data, scaled by a parameter λ. Further details can be found in [20].

6. DEEP LINEAR DISCRIMINANT ANALYSIS

Deep linear discriminant analysis (DLDA) [22] combines neural networks with the linear discriminant analysis (LDA). LDA is a discriminative learning criterion minimizing the inner class variance and maximizing the between class variance. Due to the lack of representational power the LDA criterion is usually not applied to high dimensional data. However, if combined with a non-linear system acting as a frontend, the LDA boosts classification performance to a certain extend. Following [22], we aim to maximize the eigenvalues v_i of the generalized eigenvalue problem:

$$S_b e_i = v_i(S_w + \lambda I)e_i, \tag{6}$$

where I is the identity matrix, S_b is the between class scatter matrix and S_w is the within class scatter matrix extracted from the network's output given the target labels, respectively. Details are in [22]. The eigenvalues $\{v_i, ..., v_k\}$ reflect the separation in the corresponding eigenvector space $\{e_1, ..., e_k\}$. In [22], they propose to optimize the smallest of all $C - 1$ eigenvalues. This leads to the following discriminative optimization criterion:

$$\underset{\theta}{\operatorname{argmax}} \frac{1}{k} \sum_{i=1}^{k} v_i, \tag{7}$$

where $\{v_1, ..., v_k\} = \{v_i | v_i < \min\{v_1, ..., v_{C-1}\} + \epsilon\}$, and ϵ acts as a threshold pushing variance to all $C - 1$ feature dimensions. This prevents the network from maximizing the distance to classes where good separation have already been found, and forces the model to concentrate on potentially non-separated examples instead. The cost function is differentiable, therefore, any neural network trainable with backpropagation can act as a frontend. The parameters θ are the network's weights and bias, respectively.

7. EXPERIMENTS

7.1. Experimental Setup: Acoustic Scene Classification

We pre-processed all DCASE2016 utterances with a STFT using a hamming window with window-size 40ms and 50% overlap. Next, MFCCs including Δ- and Δ^2-features were computed. All features were normalized to zero-mean unit variance using the training corpus. For the experiments we used either MFCCs + Δ + Δ^2 features, resulting in a 60-bin vector per frame as in [14], or raw 1025-bin log magnitude spectrograms. In order to guarantee a stable stochastic optimization, all observations need to be randomized. We implemented a variant of *on-the-fly* shuffling proposed in [23]. In particular, we processed batches of 500 randomly indexed, time-aligned utterance-chunks, cropped to a fixed length of 100 frames, in each optimization step. By doing so, we ensure proper randomization, preserving the sequential ordering of each utterance. The final classification score was obtained by computing a majority vote over all classified frames of the acoustic scene signal.

We put much effort in designing a solid machine learning framework which is also runnable on an embedded system. Therefore, we did not make use of ensemble or boosting methods [24], which usually, increases the classification performance. We built a single multi-label classification system instead. In particular, we used 3-layer GRNNs initialized with orthogonal weights [25] and rectifier activation functions. A linear output gate was used as a top layer. All networks have 200 neurons per layer. ADAM [26] was used for optimizing either the MSE or LDA objective.

7.2. Experimental Database: Acoustic Scene Classification

The DCASE2016 task 1 scene dataset is divided into a training and test set consisting of 1170 and 290 scene recordings, respectively. K-fold cross-validation was used for training all networks. In particular, we split the training corpus into 4 folds including 880 training utterances and 290 validation scenes, respectively. We report the average classification accuracy for all 4-folds of the training set. The labels of the test set are not published yet. More details about the data and the evaluation setup are in [14].

7.3. Experimental Results: Acoustic Scene Classification

Table 1 shows the overall scores of the DCASE2016 task 1, i.e. acoustic scene classification. We compared different feature representation using a 3-layer GRNN. Feeding raw spectrograms into the network slightly improves the classification performance, compared to MFCCs. This is consistent with the findings of [27].

Model	Features	Objective	Accuracy
baseline	MFCC	MLE	72.5%
GRNN	MFCC	MSE	74.0%
GRNN	spectrogram	MSE	76.1%
GRNN	MFCC	LDA	78.2%
GRNN	spectrogram	LDA	**79.1**%

Table 1: DCASE2016 task 1: Comparing MSE and LDA objectives using a 3-layer GRNN on different input feature representations.

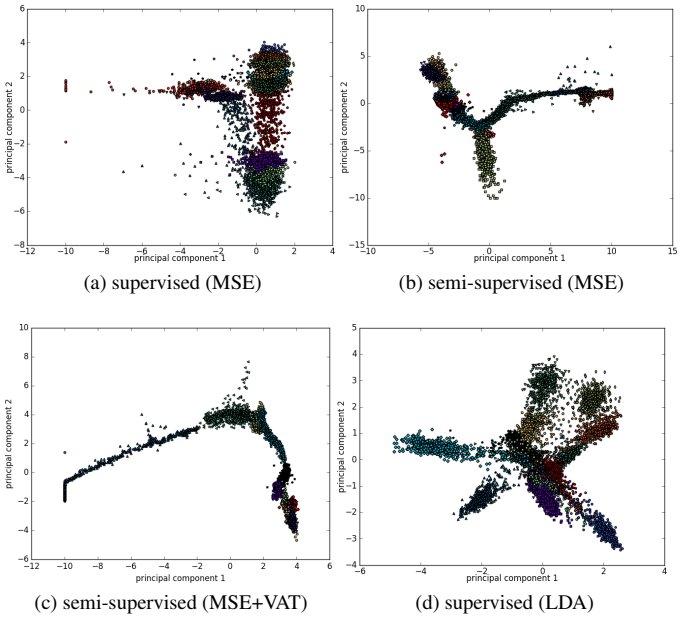
The use of a temporal model boosts recognition results in general. Most interestingly, the use of the LDA criterion achieved the best overall result, i.e. 79.1%, which leads to a relative improvement of 8.34% compared to the GMM baseline.

Table 2 shows the overall accuracy of GRNNs using a VAT regularized objective including the evaluation and test data as a semi-supervised data set. Furthermore, results for semi-supervised discriminative-generative GRNNs using the MSE objective are reported. VAT slightly improves the classification performance when using MFCC features. However, the result is slightly worse compared to the LDA criterion. The use of an additional generative cost function slightly improves the results. In particular, the *hybrid* learning criterion, i.e. Equation 5, achieves a relative improvement of 3.8% compared to the baseline system. However, VAT still

Model	Regularizer	Features	Objective	Accuracy
GRNN	VAT	MFCC	MSE	**77.8**%
GRNN	VAT	spectrogram	MSE	77.4%
GRNN	MSE (Eq. 5)	MFCC	MSE	75.4%
GRNN	MSE (Eq. 5)	spectrogram	MSE	**76.7**%

Table 2: DCASE2016 task 1: Semi-supervised training with a 3-layer GRNN using a VAT regularizer ($\lambda = 0.1, \epsilon = 0.25, I_p = 1$) and hybrid MSE objective (λ =1e-4).

outperformed the *hybrid* MSE objective.



(a) supervised (MSE) (b) semi-supervised (MSE)

(c) semi-supervised (MSE+VAT) (d) supervised (LDA)

Figure 5: Juxtaposition of supervised and semi-supervised training using a 3-layer GRNN and a subset of DCASE2016 scene spectrograms. Figure 5a-5d show the 1^{st} and 2^{nd} principal component of the activations generated from the last hidden layer using different optimization criteria.

Visual interpretations of the hidden activations provide some insights into neural networks, which are treated as *blackbox* models. Figure 5 shows the first two principal components of the activations the last hidden layer of a 3-layer GRNN, using a subset of DCASE2016 task 1 data. Starting with a pure discriminative learning criterion in Figure 5a, we see that a non-regularized MSE objective produces slightly overlapping clusters. By adding a *generative* cost function, i.e. *hybrid* MSE optimization criterion in (5), as well as a VAT regularizer (see Section 5) the inner class variance is lowered, improving the overall class margins in the end. In both, MSE and VAT objectives, the between class variance is not maximized. In Figure 5d however, we clearly see that the LDA criterion in (6) produces more separated class projections. In this case, the within class variance is minimized, whereas the between class variance is maximized.

7.4. Experimental Database: Acoustic Event Detection

The DCASE2016 task 3 acoustic event dataset is divided into a training and test set containing 22 recordings. The dataset has two scene categories, i.e. *home* and *residential area*. The *home* training corpus contains 7 event classes with 563 events, whereas the *residential area* training corpus contains 11 event classes including 906 events. Similar as in Section 7.2 K-fold cross validation, using 4 folds, was applied. More details about the data and the evaluation setup are in [14].

7.5. Experimental Setup: Acoustic Event Detection

We applied the same pre-processing routines, i.e. STFT calculation, MFCC and log-magnitude spectrogram extraction, as in Section 7.1 using data of the DCASE2016 sound event detection in real life audio challenge (task 3). Regarding the training procedure, we extended the *on-the-fly* shuffling routine in two ways: We drop frames with a probability of 50% and use smaller permuted sequence batches. By doing so, we increase the data size by introducing slight permutations and variations of the training sequences. Frames with multiple event labels were removed in the training corpus, forcing the model to extract class specific features. Apart from that, an additional *blank* label was introduced. The model sizes and configuration parameters are kept the same as in Section 7.1.

7.6. Experimental Results: Acoustic Event Detection

Model	Features	Objective	ER	F [%]
baseline	MFCC	MLE	0.90	37.3
GRNN	MFCC	MSE	0.74	42.3
GRNN	spectrogram	MSE	**0.73**	**47.6**

Table 3: DCASE2016 task 3: Classification results GRNNs using MFCCs or log-magnitude spectrograms and a MSE objective.

Model	Acoustic Scene	Segment-based		Event-based	
		ER	F [%]	ER	F [%]
GRNN	home	0.82	37.3	1.55	2.9
GRNN	residential area	0.63	57.9	4.64	0.9
GRNN	average	0.73	47.6	3.9	1.9

Table 4: DCASE2016 task 3: Detailed classification results with a 3-layer GRNN using a MSE objective and log-magnitude spectrograms.

Table 3 shows the results of the DCASE2016 task3 real audio event detection task using GRNNs. We did not apply a LDA due to overlapping events in the test set. GRNNs trained on log-magnitude spectrograms achieved an overall segment-based error rate (ER) of 0.73 and an F-score of 47.6%. This results in a relative improvement of 19.8% and 51.1% compared to the baseline GMM model for the ER- and F-scores, respectively. The error measures are specified in detail in [14]. In Table 4 detailed segment- and event-based results for both, *home* and *residential area* are reported.

8. CONCLUSION

We applied gated recurrent neural networks (GRNNs) to acoustic scene and event classification. In particular, we trained a 3-layer GRNN on environmental sounds of the IEEE Detection and Classification of Acoustic Scenes and Events challenge (DCASE2016) (task 1 and task 3). The use of virtual adversarial training (VAT) slightly improves the model performance using MFCC features. For scene classification, models trained with a deep linear discriminant objective (LDA) using log-magnitude spectrogram representations outperformed VAT regularized networks. For acoustic event detection we use a multi-label GRNN. For both tasks we outperform the GMM baseline system significantly.

9. REFERENCES

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

[2] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *18th European Signal Processing Conference*, 2010, pp. 1267–1271.

[3] J. Keshet and S. Bengio, "Discriminative keyword spotting," in *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*. Wiley Publishing, 2009, pp. 173–194.

[4] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4087–4091.

[5] M. Zhrer, R. Peharz, and F. Pernkopf, "Representation learning for single-channel source separation and bandwidth extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2398–2409, 2015.

[6] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Multi-label vs. combined single-label sound event detection with deep neural networks," in *23rd European Signal Processing Conference 2015 (EUSIPCO 2015)*, 2015.

[7] E. Cakir, T. Heittola, H.Huttunen, and T.Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[8] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5236–5240.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3104–3112.

[10] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 273–278.

[11] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014.

[12] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *CoRR*, vol. abs/1502.02367, 2015.

[14] 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, Hungary, 2016.

[15] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 3546–3554.

[16] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3581–3589.

[17] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial autoencoders," *CoRR*, vol. abs/1511.05644, 2015.

[18] M. Zöhrer and F. Pernkopf, "General stochastic networks for classification," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2015–2023.

[19] 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.

[20] T. Miyato, S. Shin-ichi Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing by virtual adversarial examples." *CoRR*, vol. abs/1507.00677, 2015.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[22] M. Dorfer, R. Kelz, and G. Widmer, "Deep linear discriminant analysis," *International Conference of Learning Representations (ICLR)*, vol. abs/1511.04707, 2015.

[23] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[24] H. Drucker, R. Schapire, and P. Simard, "Improving performance in neural networks using a boosting algorithm," in *Advances in Neural Information Processing Systems*, 1993, pp. 42–49.

[25] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *International Conference of Learning Representations (ICLR)*, 2014.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[27] M. Espi, M. Fujimoto, and T. Nakatani, "Acoustic event detection in speech overlapping scenarios based on high-resolution spectral input and deep learning." *IEICE Transactions on Information and Systems E98D*, pp. 1799–1807, 2015.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3807-0