INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition

Naoya Takahashi[1], *Michael Gygli*[2] *Beat Pfister*[2], *Luc Van Gool*[2]

[1]Sony Corporation, Japan
[2]Dept. Information Technology and Electrical Engineering, ETH Zurich, Switzerland
NaoyaA.Takahashi@jp.sony.com, {gygli, vangool}@vision.ee.ethz.ch, pfister@tik.ee.ethz.ch

Abstract

We propose a novel method for Acoustic Event Recognition (AER). In contrast to speech, sounds coming from acoustic events may be produced by a wide variety of sources. Furthermore, distinguishing them often requires analyzing an extended time period due to the lack of a clear sub-word unit. In order to incorporate the long-time frequency structure for AER, we introduce a convolutional neural network (CNN) with a large input field. In contrast to previous works, this enables to train audio event detection end-to-end. Our architecture is inspired by the success of VGGNet [1] and uses small, 3×3 convolutions, but more depth than previous methods in AER. In order to prevent over-fitting and to take full advantage of the modeling capabilities of our network, we further propose a novel data augmentation method to introduce data variation. Experimental results show that our CNN significantly outperforms state of the art methods including Bag of Audio Words (BoAW) and classical CNNs, achieving a 16% absolute improvement.
Index Terms: convolutional neural networks, data augmentation, large input field, acoustic event recognition.

1. Introduction

Scenes typically contain many sound sources. While speech is arguably one of the most important types, non-speech sounds such as music or laughter provide important information as well. In most conversations no mention is made of the environment, like its location or people and objects present. Automatic speech recognition (ASR) could benefit from having such contextual knowledge though [2]. Knowing the type of non-speech sounds improves the performance of source separation and speech enhancement [3]. Furthermore, multi-media tasks such as video classification [4] and video summarization [5] have been shown to improve when including audio information. Acoustic EventRecognition (AER) is attracting more and more attention also due to new applications incl. surveillance [6, 7, 8], multimedia content retrieval [9] and audio segmentation [10, 11].

Traditional methods for AER apply techniques from ASR directly. For instance, Mel Frequency Cepstral Coefficients (MFCC) were modeled with Gaussian Mixture Models (GMM) or Support Vector Machines (SVM) [12, 13, 14, 15]. Yet, applying standard ASR approaches leads to inferior performance due to differences between speech and non-speech signals. Thus, more discriminative features were developed. Most were hand-crafted and derived from low-level descriptors such as MFCC [16, 17], filter banks [18, 19] or time-frequency descriptors [20]. These descriptors are frame-by-frame representations (typically frame length is in the order of ms) and are usually modeled by GMMs to deal with the sounds of entire acoustic events that normally last seconds at least. Another common method to aggregate frame level descriptors is the Bag of Audio Words (BoAW) approach, followed by an SVM [17, 21, 22, 23]. These models however discard the temporal order of the frame level features, causing considerable information loss. Moreover, methods based on hand-crafted features optimize the feature extraction process and the classification process separately, rather than learning end-to-end.

Recently Deep Neural Networks (DNNs) have been very successful at many tasks, including ASR [24, 25], image classification [26, 1], and visual object detection [27]. One advantage of DNNs is their capability to jointly learn feature representations and appropriate classifiers. Supported by large amounts of training data, more recently, deeper architectures further pushed the state-of-the art for several competitions in computer vision [1]. In comparison, few AER methods rely on DNNs. One reason is the lack of large, publicly available datasets. In [28, 29], DNNs were built on top of MFCCs. Miquel *et al.* [30] utilize a Convolutional Neural Network (CNN) [31] to extract features from spectrograms. These networks are still relatively shallow (e.g. 3 layers). Furthermore, the networks take only a few frames as input and the complete acoustic events are modeled by Hidden Markov Models (HMM) or simply by calculating the mean of the network outputs, which is too simple to model complicated acoustic event structures.

In this work, we introduce novel network architectures with up to 9 layers and a large input field. The large input field allows the networks to directly model entire audio events and be trained end-to-end, as depicted in *Fig. 1*. Our network architecture is inspired by VGG Net [1] which obtained second place in the ImageNet 2014 competition and was successfully applied for ASRs [32]. The main idea of VGG Net is to replace large (typically 9×9) convolutional kernels by a stack of 3×3 kernels without pooling between these layers. Advantages of this architecture are (1) additional non-linearity hence more expressive power, and (2) a reduced number of parameters (i.e. one 9×9 convolution layer with C channel has $9^2C^2 = 81C^2$ weights while three-layer 3×3 convolution stack has $3(3^2C^2) = 27C^2$ weights). Our first goal is to adapt the VGG Net architecture to AER. In order to train our network we further propose a novel data augmentation method, especially effective for AER. For our experiments, we created a new dataset harvested from the Freesound repository [33] and conducted acoustic event classification. Experimental results show that our deeper CNN significantly outperforms several baseline techniques, including state-of-the-art methods based on BoAW and classical DNNs. We further show that the proposed data augmentation method improves the performance by more than 12%.

http://dx.doi.org/10.21437/Interspeech.2016-805

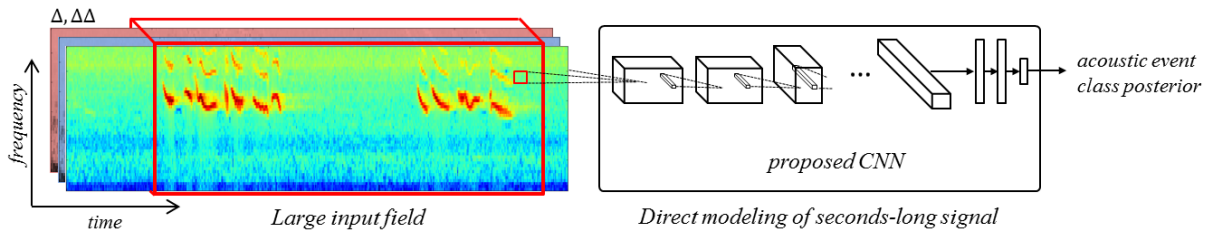

Figure 1: *Our deeper CNN models several seconds of acoustic event sound directly and outputs the posterior probability of classes.*

Table 1: *The architecture of our deeper CNNs. Unless mentioned explicitly convolution layers have 3×3 kernels.*

	Baseline		Proposed CNN	
#Fmap	DNN	Classic CNN	A	B
64		conv5×5 (3,64) pool 1×3 conv5×5(64,64)	conv(3,64) conv(64,64) pool 1×2	conv(3,64) conv(64,64) pool 1×2
128			conv(64,128) conv(128,128) pool 2×2	conv(64,128) conv(128,128) pool 2×2
256				conv(128,256) conv(128,256) pool 2×1
FC	FC4096 FC2048 FC2048 FC28	FC1024 FC1024 FC28	FC1024 FC1024 FC28	FC2048 FC2048 FC28
	softmax			
#param	258×10^6	284×10^6	233×10^6	257×10^6

2. Architectural and Training Novelties

2.1. Convolutional Network Architecture

We propose two CNN architectures, adapted to AER, as outlined in *Table 1*. Architecture A has 4 convolutional and 3 fully connected layers, while Architecture B has 9 weight layers: 6 convolutional and 3 fully connected. In this table, the convolutional layers are described as conv(input feature maps, output feature maps). All convolutional layers have 3×3 kernels, thus henceforth kernel size is omitted. The convolution stride is fixed to 1. The max-pooling layers are indicated as $time \times frequency$ in Table 1. They have a stride equal to the pool size. All hidden layers except the last fully-connected layer are equipped with the Rectified Linear Unit (ReLU) nonlinearity. In contrast to [1], we do not apply zero padding before convolution since the output size of the last pooling layer is still large enough in our case. The networks were trained by minimizing the cross entropy loss L with l_1 regularization using back propagation:

$$\arg\min_W \sum_i L(X_i, Y_i, W) + \lambda \|W\|_1 \qquad (1)$$

where X_i, Y_i, W are the ith input, label and network parameters, respectively. λ is a constant which is set to 10^{-6} in this work.

2.2. Large input field

In ASR, few-frames descriptors are typically concatenated and modeled by GMM or DNN [24, 25]. This is reasonable since they aim to model sub-word units like phonemes which typically last less than a few hundreds of ms. The sequence of sub-word units is typically modeled by a HMM. Most works in AER also follow similar frameworks, where signals lasting from tens to hundreds of ms are modeled first. These small input field representations are then aggregated to model longer signals by HMM, GMM [10, 12, 30, 9, 34] or a combination of BoAW and SVM [21, 22, 23]. Yet, unlike speech signals, non-speech signals are much more diverse even within a category and it is not clear that a sub-word approach is suitable for AER. Hence, we design a network architecture that directly models the entire acoustic event, based on a single input of multiple seconds. This also enables the networks to optimize the parameter end-to-end.

2.3. Data Augmentation

Since the proposed CNN architectures have many hidden layers and a large input field, the number of parameters is high, as shown in the last row of *Table 1*. A large number of training data is vital to train such networks. Jaitly *et al.* [35] showed that the data augmentation based on Vocal Tract Length Perturbation (VTLP) is effective to improve ASR performance. VTLP attempts to alter the vocal tract length during extraction of descriptors, such as a log filter bank, and perturbs the data in a certain non-linear way.

In order to introduce more data variation, we propose a different augmentation technique. For most sounds coming with an acoustic event, mixed sounds from the same class also belong to that class, except when the class is differentiated by the number of sound sources. For example, when mixing two different ocean surf sounds, or of breaking glass, or birds tweeting, the result still belongs to the same class. Considering this property we produce augmented sounds by randomly mixing two sounds of a class, with randomly selected timings. In addition to mixing sounds, we further perturb the sound by moderately modifying frequency characteristics of each source sound by boosting/attenuating a particular frequency band to introduce further varieties while keeping the sound recognizable. An augmented data sample s_{aug} is generated from source signals for the same class as the one both s_1 and s_2 belong to, as follows:

$$s_{aug} = \alpha\Phi(s_1(t), \psi_1) + (1-\alpha)\Phi(s_2(t-\beta T), \psi_2) \qquad (2)$$

where $\alpha, \beta \in [0, 1)$ are uniformly distributed random values, T is the maximum delay and $\Phi(\cdot, \psi)$ is an equalizing function parametrized by ψ. In this work, we used a second order parametric equalizer parametrized by $\psi = (f_0, g, Q)$ where $f_0 \in [100, 6000]$ is the center frequency, $g \in [-8, 8]$ is a gain and $Q \in [1, 9]$ is a Q-factor. An arbitrary number of such synthetic samples can be obtained by randomly selecting parameters α, β, ψ for each data augmentation. We refer to this approach as Equalized Mixture Data Augmentation (EMDA).

2.4. Multiple Instance Learning

Since we used web data to build our dataset (see *Sec. 3.1*), the training data is expected to be noisy and to contain outliers. In

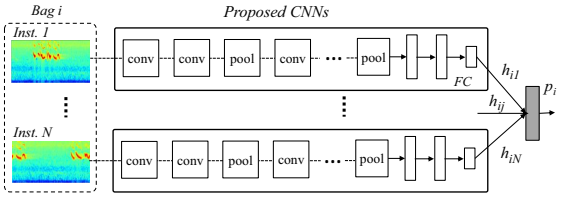

Figure 2: *Architecture of our deeper CNN model adapted to MIL. The softmax layer is replaced with the aggregation layer.*

order to alleviate the negative effects of outliers, we also employed multiple instance learning (MIL) [36, 37]. In MIL, data is organized as bags $\{X_i\}$ and within each bag there are a number of instances $\{x_{ij}\}$. Labels $\{Y_i\}$ are provided only at the bag level, while labels of instances $\{y_{ij}\}$ are unknown. A positive bag means that at least one instance in the bag is positive, while a negative bag means that all instances in the bag are negative. We adapted our CNN architecture for MIL as shown in *Fig. 2*. N instances $\{x_1, \cdots, x_N\}$ in a bag are fed to a replicated CNN which shares parameters. The last softmax layer is replaced with an aggregation layer where the outputs from each network $h = \{h_{ij}\} \in R^{M \times N}$ are aggregated. Here, M is the number of classes. The distribution of class of bag p_i is calculated as $p_i = f(h_{i1}, h_{i2}, \cdots, h_{iN})$ where $f()$ is an aggregation function. In this work, we investigate 2 aggregation functions: max aggregation

$$p_i = \frac{exp(\hat{h_i})}{\sum_i exp(\hat{h_i})} \tag{3}$$

$$\hat{h_i} = \max_j(h_{ij}) \tag{4}$$

and Noisy OR aggregation [38],

$$p_i = 1 - \prod_j (1 - p_{ij}) \tag{5}$$

$$p_{ij} = \frac{exp(h_{ij})}{\sum_j exp(h_{ij})}. \tag{6}$$

Since it is unknown which sample is an outlier, we can not be sure that a bag has at least one positive instance. However, the probability that all instances in a bag are negative exponentially decreases with N, thus the assumption becomes very realistic.

3. Experiments

3.1. Dataset

The proposed methods are evaluated on a novel acoustic event classification database [1] harvested from Freesound [33], which is a repository of audio samples uploaded by users. The database consists of 28 events as described in *Table 2*. Note that since the sounds in the repository are tagged in free-form style and the words used vary a lot, the harvested sounds contain irrelevant sounds. For instance, a sound tagged 'cat' sometime does not contain a real cat meow, but instead a musical sound produced by a synthesizer. Furthermore sounds were recorded with various devices under various conditions (e.g. some sounds are very noisy and in others the acoustic event occurs during a short time interval among longer silences). This makes our database more challenging than previous datasets such as [39].

[1] The dataset is available at https://data.vision.ee.ethz.ch/cvl/ae_dataset

Table 2: *The statistics of the dataset.*

Class	Total minutes	# clip	Class	Total minutes	# clip
Acoustic guitar	23.4	190	Hammer	42.5	240
Airplane	37.9	198	Helicopter	22.1	111
Applause	41.6	278	Knock	10.4	108
Bird	46.3	265	Laughter	24.7	201
Car	38.5	231	Mouse click	14.6	96
Cat	21.3	164	Ocean surf	42	218
Child	19.5	115	Rustle	22.8	184
Church bell	11.8	71	Scream	5.3	59
Crowd	64.6	328	Speech	18.3	279
Dog barking	9.2	113	Squeak	19.8	173
Engine	47.8	263	Tone	14.1	155
Fireworks	43	271	Violin	16.1	162
Footstep	70.3	378	Water tap	30.2	208
Glass breaking	4.3	86	Whistle	6	78
			Total	768.4	5223

In order to reduce the noisiness of the data, we first normalized the harvested sounds and eliminated silent parts. If a sound was longer than 12 sec, we split the sound in pieces so that split sounds were less than 12 sec. All audio samples were converted to 16 kHz sampling rate, 16 bits/sample, mono channel. Similar to [34], the data was randomly split into training set (75%) and test set (25%). Only the test set was manually checked and irrelevant sounds not containing the target acoustic event, were omitted. The data augmentation was applied only to the training set.

3.2. Implementation details

Through all experiments, 49 band log-filter banks, log-energy and their delta and delta-delta were used as a low-level descriptor using 25 ms frames with 10 ms shift, except for the BoAW baseline described in *Sec. 3.3.1*. Input patch length was set to 400 frames (i.e. 4 sec). The effects of this length were further investigated in *Sec. 3.3.2*. During training, we randomly crop 4 sec for each sample. The networks were trained using mini-batch gradient descent based on back propagation with momentum. We applied dropout [40] to each fully-connected layer with keeping probability 0.5. The batch size was set to 128, the momentum to 0.9. For data augmentation we used VTLP and the proposed EMDA. The number of augmented samples is balanced for each class. During testing, 4 sec patches with 50% shift were extracted and used as input to the Neural Networks. The class with the highest probability was considered the detected class. The models were implemented using the Lasagne library [41].

3.3. Experimental Results and Discussions

3.3.1. State-of-the-art comparison

In our first set of experiments we compared our proposed deeper CNN architectures to three different state-of-the-art baselines, namely, BoAW [17], HMM+DNN/CNN as in [29], and classical DNN/CNN with large input field.

BoAW We used MFCC with delta and delta-delta as a low-level descriptor. K-means clustering was applied to generate an audio word code book with 1000 centers. We evaluated both SVM with a χ^2 kernel and a 4 layer DNN as a classifier. The layer sizes of the DNN classifier were (1024, 256, 128, 28).

DNN/CNN+HMM We evaluate the DNN-HMM system. The neural network architectures are described in the left 2 columns in Table 1. Both DNN and CNN models are trained

Table 3: *Accuracy of the deeper CNN and baseline methods, trained with and without data augmentation (%).*

Method	Data augmentation without	with
BoAW+SVM	74.7	79.6
BoAW+DNN	76.1	80.6
DNN+HMM	54.6	75.6
CNN+HMM	67.4	86.1
DNN+Large input	62.0	77.8
CNN+Large input	77.6	90.9
A	77.9	91.7
B	80.3	**92.8**

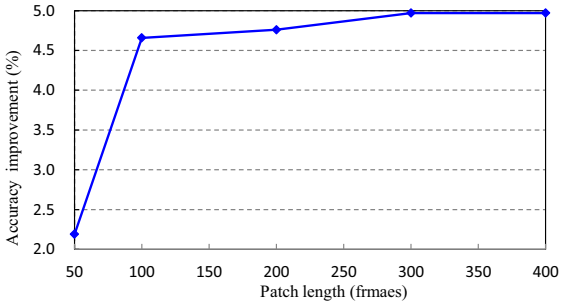

Figure 3: *Performance of our network for different input patch lengths. The plot shows the increase over using a CNN+HMM with a small input field of* 30 *frames.*

to estimate HMM state posteriors. The HMM topology consists of one state per acoustic event, and an ergodic architecture in which all states have a self-transition and equal transitions to all other states, as in [30]. The input patch length for CNN, DNN is 30 frames with 50% shift.

DNN/CNN+Large input field In order to evaluate the effect of using the proposed CNN architectures, we also evaluated the baseline DNN/CNN architectures with the same large input field, namely, 400 frame patches.

The classification accuracies of these systems trained with and without data augmentation are shown in Table 3. Even without data augmentation, the proposed CNN architectures outperform all previous methods. Furthermore, the performance is significantly improved by applying data augmentation, achieving 12.5% improvement for the B architecture. The best result was obtained by the B architecture with data augmentation. It is important to note that the B architecture outperforms classical DNN/CNN even though it has less parameters as shown in *Table 1*. This result supports the efficiency of deeper CNNs with small kernels for modelling large input fields.

3.3.2. Effectiveness of large input field

Our second set of experiments focuses on input field size. We tested our CNN with different patch size 50, 100, 200, 300, 400 frames (i.e. from 0.5 to 4 sec). The B architecture was used for this experiment. As a baseline we evaluated the CNN+HNN system described in *Sec. 3.3.1* but using our architecture B, rather than a classical CNN. The performance improvement over the baseline is shown in *Fig. 3*. The result shows that larger input fields improve the performance. Especially the performance with patch length less than 1 sec sharply drops. This proves that modeling long signals directly with deeper CNN is superior to handling long sequences with HMMs.

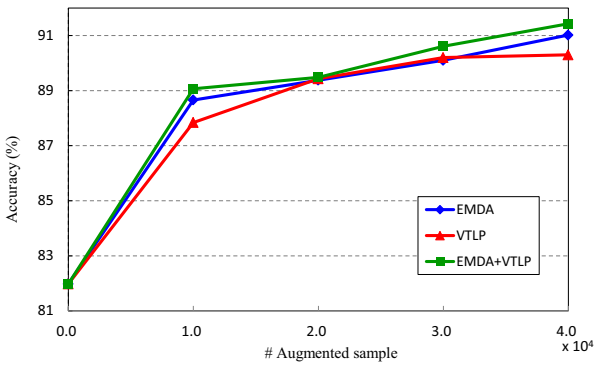

Figure 4: *Effects of different data augmentation methods with varying amounts of augmented data.*

3.3.3. Effectiveness of data augmentation

We verified the effectiveness of our EMDA data augmentation method in more detail. We evaluated 3 types of data augmentation: EMDA only, VTLP only, and a mixture of EMDA and VTLP (50%, 50%) with different numbers of augmented sample 10k, 20k, 30k, 40k. Fig. 4 shows that using both EDMA and VTLP always outperforms EDMA or VTLP only. This shows that EDMA and VTLP perturbs original data and create new samples in a different way, providing more effective variation of data and helping to train the network to learn a more robust and general model from limited amount of data.

3.3.4. Effects of Multiple Instance Learning

Finally, the A and B architectures with a large input field were adapted to MIL to handle the noise in the database. The number of parameters were identical since both max and Noisy OR aggregation methods are parameter free. The number of instances in a bag was set to 2. We randomly picked 2 instances from the same class during each epoch of the training. Table 4 shows that MIL didn't improve performance. However, MIL with a medium size input field (i.e. 2 sec) performs as good as or even slightly better than single instance learning with a large input field. This is perhaps due to the fact that the MIL took the same size input length (2 sec $\times 2$ instances $=$ 4 sec), while it had less parameter. Thus it managed to learn a more robust model.

Table 4: *Accuracy of MIL and normal training (%).*

Architecture	Single instance	MIL Noisy OR	MIL Max	MIL Max (2sec)
A	91.7	90.4	92.6	**92.9**
B	92.8	91.3	92.4	92.8

4. Conclusions

We proposed new CNN architectures and showed that they allow to learn a model for AER end-to-end, by directly modeling a several seconds long signal. We further proposed a method for data augmentation that prevents over-fitting and leads to superior performance even when training data is fairly limited. Experimental results shows that proposed methods significantly outperforms state of the arts. We further validated the effectiveness of deeper architectures, large input fields and data augmentation one by one. Future work will be directed towards applying the proposed AER to different applications such as video segmentation and summarization.

5. References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[2] S. Araki, T. Hori, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, M. Delcroix, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato, "Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 151–152, 2011.

[3] A. Ozerov, A. Liutkus, and R. Badeau, "Informed source separation: Source coding meets source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASSPA)*, vol. 2, 2011, pp. 8–11.

[4] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. ICMR*, 2011.

[5] Y. Li and B. Mérialdo, "Video summarization based on balanced AV-MMR," in *Proc. International Conference on Multimedia Modeling*, 2012, pp. 1–6.

[6] R. Radhakrishnan, A. Divakaran, and P. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2005, pp. 158–161.

[7] W. Choi, J. Rho, D. K. Han, and H. Ko, "Selective background adaptation based abnormal acoustic event recognition for audio surveillance," in *IEEE Conference on Advanced Video and Signal-Based Surveillance, AVSS*, 2012, pp. 118–123.

[8] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS*, 2007, pp. 21–26.

[9] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, "Audio keywords generation for sports video analysis," *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 4, no. 2, pp. 1–23, 2008.

[10] 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.

[11] T. Zhang and C.-C. J.Kuo, "Audio content analysis for online audiovisual data segmentation and classification," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 4, pp. 441–457, 2001.

[12] 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, 2006.

[13] Z. Huang, Y. C. Cheng, K. Li, V. Hautamäki, and C. H. Lee, "A blind segmentation approach to acoustic event detection based on I-vector," in *Proc. INTERSPEECH*, 2013, pp. 2282–2286.

[14] K. Lee and D. P. W. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1406–1416, 2010.

[15] A. Temko, E. Monte, and C. Nadeu, "Comparison of sequence discriminant support vector machines for acoustic event classification," in *Proc. ICASSP*, vol. 5, 2006, pp. 721–724.

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

[17] S. Pancoast and M. Akbacak, "Bag-of-Audio-Words Approach for Multimedia Event Classification," in *Proc. INTERSPEECH*, Portland, OR, USA, 2012.

[18] W. Choi, S. Park, D. K. Han, and H. Ko, "Acoustic event recognition using dominant spectral basis vectors," in *Proc. INTERSPEECH*, 2015, pp. 2002–2006.

[19] J. Beltrán, E. Chávez, and J. Favela, "Scalable identification of mixed environmental sounds, recorded from heterogeneous sources," *Pattern Recognition Letters*, vol. 68, pp. 153–160, 2015.

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

[21] X. Lu, P. Shen, Y. Tsao, C. Hori, and H. Kawai, "Sparse representation with temporal max-smoothing for acoustic event detection," in *Proc. INTERSPEECH*, 2015, pp. 1176–1180.

[22] H. Lim, M. J. Kim, and H. Kim, "Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation," in *Proc. INTERSPEECH*, 2015, pp. 3325–3329.

[23] F. Metze, S. Rawat, and Y. Wang, "Improved audio features for large-scale multimedia event detection," in *Proc. ICME*, 2014, pp. 1–6.

[24] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, 2012.

[25] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for Speech Recognition," in *ICASSP*, 2012, pp. 4277–4280.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2014, pp. 580–587.

[28] Z. Kons, O. Toledo-Ronen, and M. Carmel, "Audio Event Classification Using Deep Neural Networks," in *Proc. INTERSPEECH*, 2013, pp. 1482–1486.

[29] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," *Proc. EUSIPCO*, no. 1-5 Sept. 2014, pp. 506–510, 2014.

[30] 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. 26, pp. 1–12, 2015.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," in *Proc. of the IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324.

[32] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. ICASSP*, 2016, pp. 4955–4959.

[33] "freesound," http://www.freesound.org/.

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

[35] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," *ICML*, vol. 28, 2013.

[36] J. Wu, Yinan Yu, Chang Huang, and Kai Yu, "Deep multiple instance learning for image classification and auto-annotation," in *Proc. CVPR*, 2015, pp. 3460–3469.

[37] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Proc. NIPS*, vol. 74, no. 10, 2005, pp. 1769–1775.

[38] H. David, "A tractable inference algorithm for diagnosing multiple diseases." in *Proc. UAI*, 1989, pp. 163–171.

[39] S. Nakamura, K. Hiyane, and F. Asano, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *International Conference on Language Resources & Evaluation*, 2000, pp. 2–5.

[40] 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*, 2012.

[41] E. Battenberg, S. Dieleman, D. Nouri, E. Olson, C. Raffel, J. Schlüter, S. K. Sønderby, D. Maturana, M. Thoma, and et al., "Lasagne: First release." http://dx.doi.org/10.5281/zenodo.27878, Aug. 2015.