2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2015, BOSTON, USA

ENVIRONMENTAL SOUND CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

Karol J. Piczak

Institute of Electronic Systems
Warsaw University of Technology

ABSTRACT

This paper evaluates the potential of convolutional neural networks in classifying short audio clips of environmental sounds. A deep model consisting of 2 convolutional layers with max-pooling and 2 fully connected layers is trained on a low level representation of audio data (segmented spectrograms) with deltas. The accuracy of the network is evaluated on 3 public datasets of environmental and urban recordings. The model outperforms baseline implementations relying on mel-frequency cepstral coefficients and achieves results comparable to other state-of-the-art approaches.

Index Terms— environmental sound, convolutional neural networks, classification

1. INTRODUCTION

Convolutional neural networks date back as far as to the 1980s [1–3], yet only recently have they been adopted as a method of choice for various object classification tasks. The work of Krizhevsky et al. [4] marked a turning point in large scale visual recognition [5] in 2012. Since then, by replacing techniques relying on manually engineered features, convolutional neural networks allowed for significant progress in numerous pattern recognition tasks, including classification of traffic signs [6], house numbers [7, 8] and handwritten digits [9], pedestrian detection [10], and electron microscopy image processing [11].

Although primarily used in visual recognition contexts, convolutional architectures have been also successfully applied in speech [12–18] and music analysis [19, 20]. These efforts have shown that approaches taking advantage of data locality can provide viable solutions to problems encountered in other domains. A thorough coverage of current deep learning methods and their applications in various contexts, including speech and audio processing, can be found in recently published books discussing this topic [21–23].

At the same time, classification of environmental sounds (everyday audio events that do not consist of music or speech data and are often more diverse and chaotic in their structure) is still predominantly based on applying general classifiers (Gaussian mixture models, support vector machines, hidden Markov models) to manually extracted features, such as mel-frequency cepstral coefficients. Recent reviews pertaining to this subject [24, 25] present a detailed analysis of the most common approaches. The introduction of deep learning techniques in this context has slowly begun in the last two years. However, these efforts are still mostly limited to analyzing highly pre-processed acoustic features [26–30].

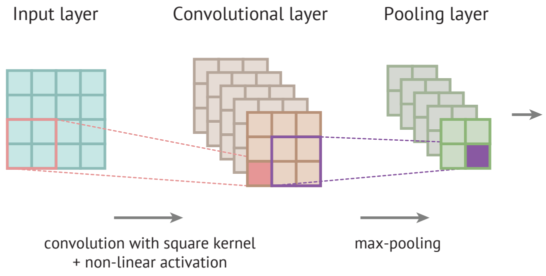


Fig. 1. A schematic visualization of a typical convolution-pooling operation performed on the input data.

This contrast in the development of different fields raises a research question which has not yet been widely addressed. *Can convolutional neural networks be effectively used in classifying environmental and urban sound sources*? The goal of this paper is to provide an answer to this question.

2. CONVOLUTIONAL NEURAL NETWORKS

In essence, convolutional neural networks are a simple extension of the multilayer perceptron model. However, their architectural differences have significant practical consequences.

2.1. Layer architecture

A typical convolutional neural network consists of a number of different layers stacked together in a deep architecture: an input layer, a group of convolutional and pooling layers (which can be combined in various ways), a limited number of fully connected hidden layers, and an output (loss) layer. The

978-1-4673-7454-5/15/$31.00 ©2015 IEEE

actual difference, when compared to the multilayer perceptron, lies in the introduction of a combination of convolution and pooling operations (depicted in Figure 1).

A convolutional layer introduces a special way of organizing hidden units which aims to take advantage of the local structure present in the two-dimensional input data (mostly, but not limited to, images). Each hidden unit, instead of being connected to all the inputs coming from the previous layer, is limited to processing only a tiny part of the whole input space (e.g. small 3×3 blocks of pixels), called its *receptive field*. The weights of such a hidden unit create a *convolutional kernel* (*filter*) which is applied to (tiled over) the whole input space, resulting in a *feature map*. This way, one set of weights can be reused for the whole input space. This is based on the premise that locally useful features will be also useful in other places of the input space - a mechanism which not only vastly reduces the number of parameters to estimate, but improves robustness to translational shifts of the data. A typical convolutional layer will consist of numerous *filters* (*feature maps*).

Further dimensionality reduction can be achieved through *pooling* layers, which merge adjacent cells of a feature map. The most common pooling operations performed are taking the *max* (winner takes all) or *mean* of the input cells. This downsampling further improves invariance to translations.

2.2. Rectified Linear Units

Traditionally, logistic sigmoid and hyperbolic tangent have been used as typical non-linear activation functions in a multilayer perceptron setup. Recent implementations of deep architectures have unequivocally replaced them with alternative solutions. One of the most common is the application of *Rectified Linear Units* (ReLUs), which use the following activation function:

$$f(x) = max(0, x) \tag{1}$$

ReLUs have several advantages over traditional units: faster computation and more efficient gradient propagation (they do not saturate as is the case with sigmoid units), biological plausibility (one-sidedness) and sparse activation structure [31], while still retaining sufficient discriminatory properties despite their simplicity. One of their drawbacks is that depending on the state of the random weight initialization, multiple units may prematurely fall into the “dead zone” - outputting a constant gradient of zero. For this reason, alternatives with a non-zero slope such as *Leaky Rectified Linear Units* have been suggested [32], and empirical results seem to confirm their usefulness [33].

2.3. Dropout learning

Deep neural architectures have a natural tendency to overfitting. Even in convolutional neural networks, where the quantity of parameters is reduced through weight sharing, the number of estimated values is most of the times bigger than the number of training cases by an order of magnitude. This can result in poor out-of-sample generalization.

One way to tackle this problem was introduced in the form of *dropout* learning [34]. The concept is quite simple, yet highly effective. In each training iteration every hidden unit is randomly removed with a predefined probability (originally 50%), and the learning procedure continues normally. These random perturbations effectively prevent the network from learning spurious dependencies, and creating complex co-adaptations between hidden units. This way big groups of neurons become helpful not only in the context of other neurons. Architecture averaging introduced by dropout tries to ensure that each hidden unit learns feature representations that are generally favorable in producing the correct classification answer.

3. SOUND CLASSIFICATION

3.1. Datasets

One of the main problems with training deep neural architectures in a supervised manner is the amount of computational effort and labeled data required for efficient learning. While the former is in some part addressed on a universal basis by hardware advances and general-purpose GPU computing, the latter is very domain-dependent.

Unfortunately, publicly available datasets of environmental recordings are still very limited - both in number and in size[1]. This is quite understandable, considering the high cost of manual annotation. Although the situation gradually improves with the introduction of new collections of recordings [35, 36], it is still one of the major hindrances to the development of new data-intensive approaches in this field. This is especially important, since the performance of supervised deep models is strongly influenced by the size of the dataset available for learning. Therefore, the original research problem has to be extended in this context - *can convolutional neural networks be effectively used for environmental sound classification with limited amount of training data*?

To answer this question, three publicly available datasets were selected for evaluation of the models: *ESC-50* [36], *ESC-10* [36] and *UrbanSound8K* [35].

The *ESC-50* dataset is a collection of 2000 short (5 seconds) environmental recordings comprising 50 equally balanced classes of sound events in 5 major groups (animals, natural soundscapes and water sounds, human non-speech sounds, interior/domestic sounds, and exterior/urban noises) prearranged into 5 folds for comparable cross-validation. It is a demanding compilation with baseline approaches (classification with random forest ensemble using features derived from mel-frequency cepstral coefficients and zero-crossing rate) achieving a mean accuracy of 44%, and recognition

[1] A list of datasets in this field is currently maintained by Toni Heittola at: http://www.cs.tut.fi/~heittolt/datasets [*last accessed: July 30, 2015*].

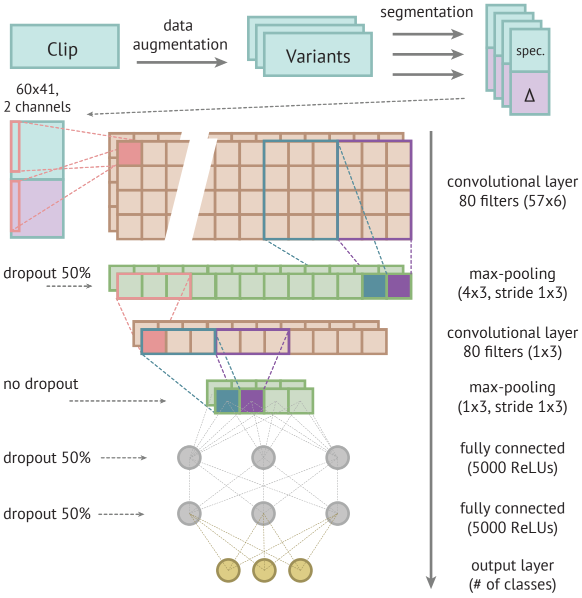


Fig. 2. Model architecture for the short segment variation.

accuracy of untrained human participants at approximately 81% [36].

ESC-10 is a less complex standardized subset of 10 classes (400 recordings) selected from the *ESC-50* dataset (*dog bark*, *rain*, *sea waves*, *baby cry*, *clock tick*, *person sneeze*, *helicopter*, *chainsaw*, *rooster*, *fire crackling*). Using the same implementations of the baseline classifier yields an accuracy of 73% (5 fold cross-validation), with respective human benchmark of 96% [36].

UrbanSound8K is a collection of 8732 short (less than 4 seconds) excerpts of various urban sound sources (*air conditioner*, *car horn*, *playing children*, *dog bark*, *drilling*, *engine idling*, *gun shot*, *jackhammer*, *siren*, *street music*) prearranged into 10 folds. Based on this dataset, the work of Salamon and Bello [37] compares a baseline system with unsupervised feature learning performed on patches of PCA-whitened log-scaled mel-spectrograms. The average classification accuracy obtained is respectively 68% for the baseline and 73.6% for the best performing variant of the evaluated system.

3.2. Experiment setup[2]

Training a convolutional neural network involves a lot of decisions that have to be made regarding both the architecture (format of the input data, number and size of layers, amount of spatial pooling, filter dimensions) and learning hyperparameters (learning rate, momentum, batch size, dropout probability, amount of regularization applied). This selection process is still mostly based on heuristics, especially when entering uncharted territories of new applications.

Due to the time required for training a complete model, exhaustive evaluation of all potential combinations was infeasible. Therefore, the selection of the most promising model had to be based on limited validation (single fold) performed for most significant factors (number of layers/filters, filter shape, learning rate, dropout probability). The final system that was evaluated in detail can be described through the following process depicted in Figure 2:

- The *ESC-50* and *ESC-10* training datasets were augmented by applying random time delays to the original recordings. Additionally, class-dependent time-stretching and pitch-shifting was performed on the *ESC-10* training set. 10 augmentations were created for each clip of the *ESC-10* dataset and 4 variants for the *ESC-50*. Simple augmentation techniques proved to be unsatisfactory for the *UrbanSound8K* dataset given the considerable increase in training time they generated and negligible impact on model accuracy.
- Log-scaled mel-spectrograms were extracted from all recordings (resampled to 22050 Hz and normalized) with window size of 1024, hop length of 512 and 60 mel-bands, using the *librosa* implementation[3].
- As learning on whole clips was too limiting on the number of examples available for training, the spectrograms were split into 50% overlapping segments of 41 frames (short variant, segments of approx. 950 ms) or 101 frames with 90% overlap (long variant, approx. 2.3 s), discarding silent segments in the process.
- Segments (e.g. 60 rows/bands × 41 columns/frames) were provided together with their deltas (computed with default *librosa* settings) as a 2-channel input to the network.
- The first convolutional ReLU layer consisted of 80 filters of rectangular shape (57×6 size, 1×1 stride) allowing for slight frequency invariance. Max-pooling was applied with a pool shape of 4×3 and stride of 1×3. A small selection of learned filters is presented in Figure 3.

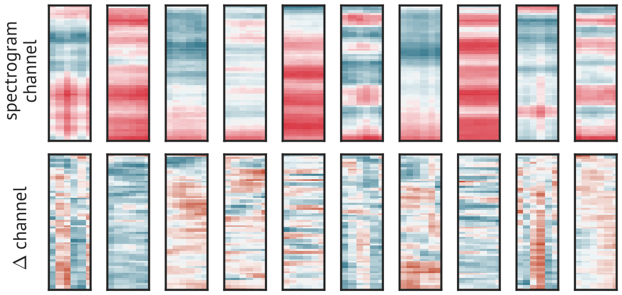


Fig. 3. A selection of filters learned by the first convolutional layer.

[2] Source code for study replication is available as an IPython notebook at: https://github.com/karoldvl/paper-2015-esc-convnet.

[3] *librosa: v0.3.1* library by B. McFee et al., DOI: 10.5281/zenodo.12714

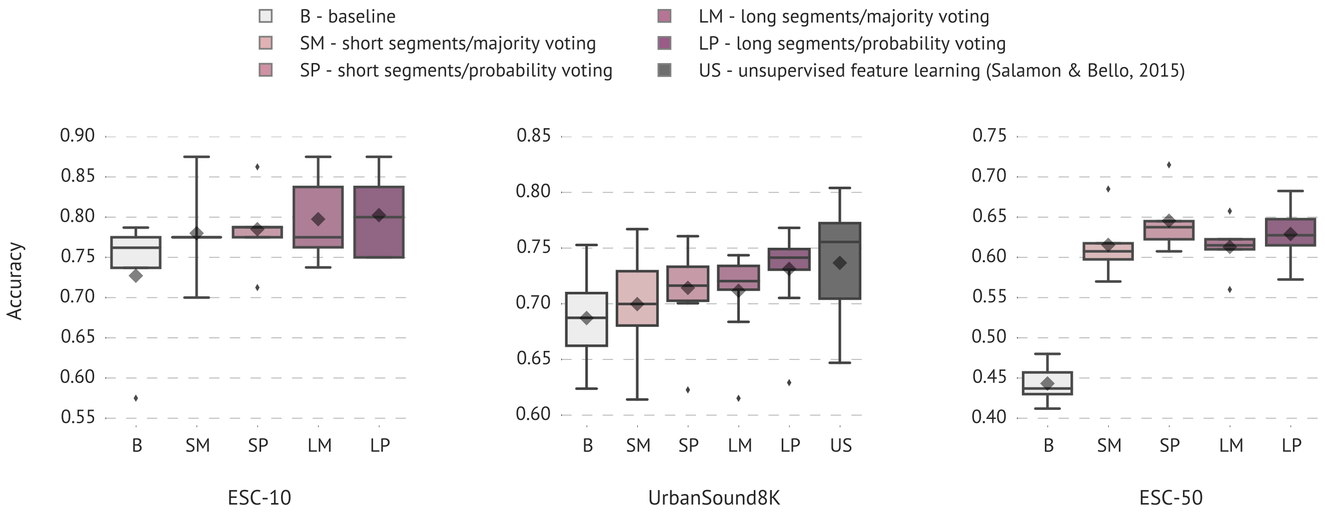


Fig. 4. Comparison of classification accuracy on the evaluated datasets.

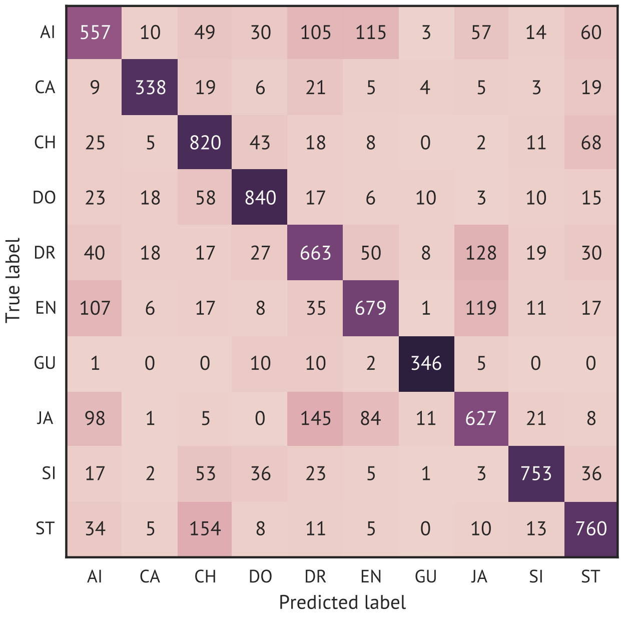
- A second convolutional ReLU layer consisted of 80 filters (1×3 size, 1×1 stride) with max-pooling (1×3 pool size, 1×3 pool stride).
- Further processing was applied through two fully connected hidden layers of 5000 ReLUs each and a softmax output layer.
- Training was performed using *pylearn2* [38] implementation of mini-batch (stochastic) gradient descent with even shuffled sequential batches (batch size of 1000), Nesterov momentum of 0.9 [39], 0.001 L2 weight decay for each layer and 0.5 dropout probability for fully connected layers and the first convolutional layer.
- Training procedure was stopped after 300 epochs for the short segment variant (learning rate of 0.002) and 150 epochs for the long variant (learning rate 0.01).
- Final predictions for a clip were generated using either a majority-voting scheme or by taking into account the probabilities predicted for each segment.

3.3. Results

The model was evaluated in a 5-fold (*ESC-10* and *ESC-50*) and 10-fold (*UrbanSound8K*) cross-validation regime with a single training fold used as an intermittent validation set. Figure 4 presents classification accuracy achieved by variants of the evaluated model and baseline implementations, with mean accuracy across folds indicated by diamond marks.

In all cases models based on a convolutional neural network performed better than respective implementations using manually-engineered features, especially when classifying audio events from more varied categories of the *ESC-50* dataset (baseline accuracy: 44%, best network: 64.5%). Models working on longer time scales also seemed to offer slight improvements over variants operating on shorter segments. Unfortunately, further extensions of segment length would most probably be counter-balanced by diminishing returns due to overfitting, as they drastically reduce the effective number of training examples. The probability-voting scheme proved to be universally favorable as compared to a majority-voting setup.

However, owing to the fact that the presented baselines should be treated as most common approaches and not fine-tuned state-of-the-art techniques, the real potential of convo-



	AI	CA	CH	DO	DR	EN	GU	JA	SI	ST
AI	557	10	49	30	105	115	3	57	14	60
CA	9	338	19	6	21	5	4	5	3	19
CH	25	5	820	43	18	8	0	2	11	68
DO	23	18	58	840	17	6	10	3	10	15
DR	40	18	17	27	663	50	8	128	19	30
EN	107	6	17	8	35	679	1	119	11	17
GU	1	0	0	10	10	2	346	5	0	0
JA	98	1	5	0	145	84	11	627	21	8
SI	17	2	53	36	23	5	1	3	753	36
ST	34	5	154	8	11	5	0	10	13	760

Fig. 5. Confusion matrix for the (LP) model evaluated on the *UrbanSound8K* dataset. Classes: air conditioner (AI), car horn (CA), children playing (CH), dog bark (DO), drilling (DR), engine idling (EN), gun shot (GU), jackhammer (JA), siren (SI), street music (ST). See [37] for comparison.

lutional neural networks would still require further evaluation. Nonetheless, a convolutional model using long segments and probability voting achieves results comparable to other best performing models on the *UrbanSound8K* dataset (LP - 73.1%, US - 73.7%) with smaller dispersion.

The model's confusion matrix for the *UrbanSound8K* dataset (Fig. 5), when compared with its analogue for the unsupervised technique presented by Salamon & Bello [37], shows some complementary characteristics. The convolutional network performs considerably better in recognizing specific classes (*air conditioner*, *car horn*, *playing children*, *dog bark*) while at the same time performing relatively poor for sounds with short-scale temporal structure (*drilling*, *engine idling*, *jackhammer*). This could indicate that ensemble averaging of different approaches (both convolutional and non-convolutional) may yield even more efficient systems.

4. SUMMARY

The goal of this paper was to evaluate whether convolutional neural networks can be successfully applied to environmental sound classification tasks, especially considering the limited nature of datasets available in this field.

It seems that they are indeed a viable solution to this problem. Conducted experiments show that a convolutional model outperforms common approaches based on manually-engineered features and achieves a similar level as other feature learning methods. Although, taking into consideration much longer training times, the result is far from groundbreaking, it shows that convolutional neural networks can be effectively applied in environmental sound classification tasks even with limited datasets and simple data augmentation. What is more, it is quite likely that a considerable increase in the size of the available dataset would vastly improve the performance of trained models, as the gap to human accuracy is still profound.

One of the possible questions open for future inquiry is whether convolutional neural networks could be effectively used in ensembles with other less complex models, as they seem to concentrate on distinct aspects of sound events.

Acknowledgments

I would like to thank Justin Salamon for providing details on the results obtained on the *UrbanSound8K* dataset [37].

5. REFERENCES

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[2] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

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

[4] 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.

[5] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *arXiv preprint arXiv:1409.0575*, 2014.

[6] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[7] I. J. Goodfellow et al., "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *arXiv preprint arXiv:1312.6082*, 2013.

[8] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*. IEEE, 2012, pp. 3288–3291.

[9] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3642–3649.

[10] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 3626–3633.

[11] D. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems*, 2012, pp. 2843–2851.

[12] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 1096–1104.

[13] L. Deng, O. Abdel-Hamid, and D. Yu, "A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6669–6673.

[14] O. Abdel-Hamid et al., "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 10, pp. 1533–1545, 2014.

[15] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8614–8618.

[16] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4277–4280.

[17] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in *Proceedings of INTERSPEECH*, 2013, pp. 3366–3370.

[18] L. Deng et al., "Recent advances in deep learning for speech research at Microsoft," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8604–8608.

[19] S. Dieleman, P. Brakel, and B. Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *Proceedings of the 12th International Society for Music Information Retrieval (ISMIR) conference*, 2011, pp. 669–674.

[20] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems*, 2013, pp. 2643–2651.

[21] Y. Bengio, I. J. Goodfellow, and A. Courville, *Deep Learning*, 2015, Book in preparation for MIT Press.

[22] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, Springer, 2014.

[23] L. Deng and D. Yu, *Deep Learning: Methods and Applications*, NOW Publishers, 2014.

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

[25] S. Chachada and C.-C. J. Kuo, "Environmental sound recognition: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 3, pp. e14, 2014.

[26] M. Asgari, I. Shafran, and A. Bayestehtashk, "Inferring social contexts from audio recordings using deep neural networks," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2014.

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

[28] M. Ravanelli, B. Elizalde, K. Ni, and G. Friedland, "Audio concept classification with hierarchical deep neural networks," in *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 606–610.

[29] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*. IEEE, 2014, pp. 506–510.

[30] L. Xue and F. Su, "Auditory scene classification with deep belief network," in *MultiMedia Modeling*. Springer, 2015, pp. 348–359.

[31] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP*, 2011, vol. 15, pp. 315–323.

[32] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[33] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[34] G. E. Hinton et al., "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[35] 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*. ACM, 2014, pp. 1041–1044.

[36] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2015, in press.

[37] 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.

[38] I. J. Goodfellow et al., "Pylearn2: a machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013.

[39] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8624–8628.