BLIND REVERBERATION TIME ESTIMATION USING A CONVOLUTIONAL NEURAL NETWORK

Hannes Gamper and Ivan J. Tashev

Audio and Acoustics Research Group
Microsoft Research
Redmond, WA, USA

ABSTRACT

The reverberation time of an acoustic environment is a useful parameter for applications including source localisation, speech recognition and mixed reality. However, estimating the reverberation time blindly and on the fly remains a challenge. Here we propose formulating the estimation as a regression problem and using a convolutional neural network (CNN) to estimate the reverberation time directly from a four second long single-channel recording of reverberant speech in noise. Evaluation on the ACE Challenge data corpus suggests that the proposed method is computationally efficient and outperforms state-of-the-art methods.

***Index Terms*—** T60, energy decay rate, deep neural networks

1. INTRODUCTION

The reverberation time is one of the most important parameters describing an environment's acoustic behaviour. It is typically defined as the time, T_{60}, it takes for the acoustic impulse response (AIR) energy to decay by 60 dB. Besides its perceptual relevance [1, 2, 3], the T_{60} is important in practical applications, including mixed reality and voice-controlled systems, as it affects the performance of sound source localisation [4] and speech recognition systems [5, 6].

While well-established methods exist to determine the T_{60} from an AIR [7, 8], the AIR itself is typically not available in practical scenarios. Instead, the T_{60} has to be inferred directly from signals present in the acoustic environment, e.g., speech captured by a user's device. This can be challenging, especially if the recorded signals stem from unknown sources and are corrupted by ambient noise. In 2015 the ACE Challenge workshop was held to address the question of blindly estimating the T_{60} from speech signals recorded in reverberant, noisy environments [9]. The challenge resulted in a number of state-of-the-art T_{60} estimation methods, including classic signal processing as well as machine learning approaches.

Prego et al. contributed the method with the best performance in terms of the Pearson correlation coefficient between estimated and ground-truth T_{60} [10, 11]. The method relies on estimating the signal-to-noise ratio (SNR) from silence at the beginning of a sample and applying a noise reduction technique before estimating the T_{60} from features in the Short Time Fourier Transform (STFT) domain.

Faraji et al. propose fitting a first-order infinite impulse response (IIR) model to reverberant speech, showing a relation between the IIR pole and the T_{60} [12]. While they report that their approach outperforms one of the baseline methods in the ACE Challenge [13], it is not clear how it would compare to state-of-the-art approaches. Lee and Chang employ a fully-connected neural network with three hidden layers to estimate the T_{60} of speech samples convolved with synthetic room impulse responses and additive babble noise at 20 dB SNR [14]. They propose using a Mel-frequency spectrogram rather than STFT features to reduce computational complexity. However, experimental results are only reported for synthetic data, making it difficult to assess how well the method would generalise to real recordings of noisy, reverberant speech. Senoussaoui et al. propose an approach that fuses short- and long-term speech features to estimate T_{60} from speech in noisy environments [15]. They show that their method outperforms one of the ACE Challenge baseline methods [16]. However, no comparison to state-of-the art methods is reported. Lee and Chang propose using a deep neural network to estimate reverberation time from multi-channel recordings [17]. However, the method is only evaluated on simulated AIRs. The number, variety, and complexity of the previously proposed algorithms are an indication of the difficulty of the T_{60} estimation problem.

Here we propose a single-channel T_{60} estimation approach that is conceptually straightforward and computationally efficient, and we evaluate it on the ACE Challenge corpus, allowing direct comparison with state-of-the-art methods. Recently, convolutional neural networks have been applied successfully in many areas, including image classification [18], classic speech and audio problems [19, 4], as well as perceptual modelling [20]. They can be advantageous over standard feedforward networks as they have fewer free parameters and are thus easier to train [18]. We hypothesise that learning and combining convolution kernels is a well-suited approach for extracting features related to the T_{60} from reverberant speech, as it enables the network to learn representations that rely both on local spectral and temporal features (e.g., short, band-limited decay slopes) and their combinations at higher abstraction levels.

2. REGRESSION LEARNING USING A NEURAL NETWORK

Machine learning has rapidly advanced the state-of-the-art in areas including speech recognition [21] and image classification [22]. In the audio domain, neural networks have been used successfully to estimate the T_{60} [23], the direction of arrival of a sound source [4], and the polar angle of a binaural signal [20], by formulating the goal of estimating a continuous variable as a classification problem. Here we propose formulating the T_{60} estimation as a regression problem. This has three advantages over using a classification approach:

- the ground truth T_{60} data do not need to be quantised;
- we utilise a loss function that directly minimises the estimation error, rather than the classification error, potentially leading to better estimation performance [24];
- the model directly outputs a continuous-valued T_{60} estimate.

978-1-5386-8151-0/18/$31.00 ©2018 IEEE

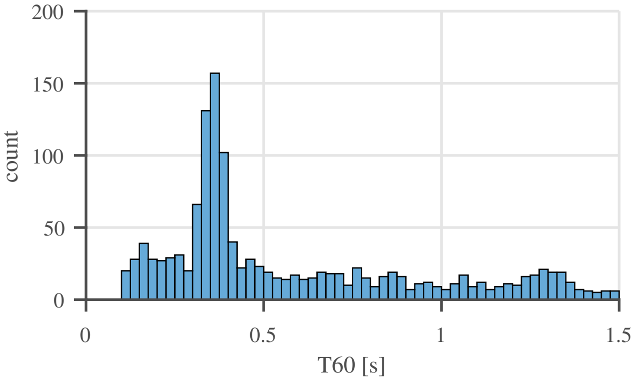

Fig. 1. Histogram of T_{60} values in training data.

Type	data	# samples
training	1325 synthetic and measured IRs	34489
validation	ACE development set [9]	1986
testing	ACE evaluation set [9]	16080

Table 1. Training, validation, and test sets.

We use a convolutional neural network combined with a fully connected layer with a single output node that directly estimates T_{60}. As the training loss function we use the squared error between the T_{60} estimate and the ground truth value. Unlike loss functions used for classification tasks, e.g., the cross-entropy loss, that do not encode class order or distance, the squared error is a distance metric. To minimise the training loss the network is forced to learn a representation that minimises the distance between samples with similar T_{60}. We hypothesise that this leads to a more robust model and better estimation performance than approaches based on classification.

3. PROPOSED APPROACH

The proposed approach for estimating T_{60} blindly from speech relies on a deep convolutional network trained with a large number of noisy, reverberant speech samples with known T_{60} values.

3.1. Training data generation

Obtaining training data of sufficient quality and quantity is crucial for the performance of a deep neural network. To generate noisy, reverberant speech samples we follow the specifications for noise types and SNR levels outlined in the ACE Challenge [9]. To ensure that the trained model does not overfit the training data or hone in on artifacts stemming from the data synthesis process, it is important to carefully separate training and test data. A typical setup splits all available data into three separate sets: a training set for training the network; a validation set to monitor whether the network is overfitting during training; and a test set to evaluate the final trained model.

The data corpus used for the ACE Challenge contains separate development and evaluation sets [9]. Here we use the ACE development set for model validation, and the ACE evaluation set for model testing. To create training data, we generated 670 synthetic AIRs using the image source method for shoebox environments of varying sizes and with varying absorption coefficients [25]. For each simulated shoebox, one randomly placed source and five randomly placed receivers were simulated. The synthetic AIRs were combined with 655 measured multi-channel AIRs from internal and publicly available databases, including the Openair database [26], the RWCP database [27], the REVERB Challenge dataset [5], the EchoThief database [28], and datasets available in the SOFA format [29], yielding a total of 1325 AIRs with T_{60} values between 0.1 and 1.5 s. The ground truth T_{60} values were estimated using a method proposed by Karjalainen et al. [8]. A histogram of the T_{60} values of all AIRs used to generate training data is shown in Figure 1. As can be seen, the distribution is not uniform, and it is not clear whether this distribution is representative of the T_{60} values encountered in real environments. For methods to address class imbalance with convolutional neural networks the reader is referred to the work by Buda et al. [30]. No such methods were considered in the present work.

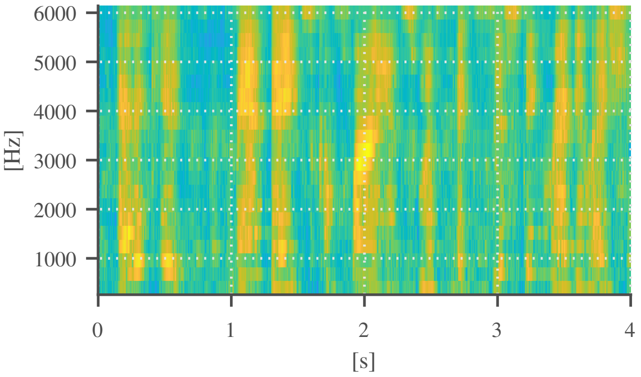

Fig. 2. Pre-processed input sample (21 × 1996), for $\mathrm{T}_{60} = 0.206$ s.

Speech samples were selected randomly from the TIMIT database [31] and an internal corpus of close-mic recordings. After discarding samples with low SNR or other artifacts, this resulted in a set of 903 English utterances by female and male speakers.

For the purposes of this work we only considered the noise types contained in the ACE corpus: “ambient”, “fan”, and “babble” [9]. Due to this limitation, it is not clear how the proposed approach would behave in the case of noise types not seen during training. To simulate ambient and fan noise, we extracted the magnitude spectra of random 10 s long segments of the corresponding noise recordings in the ACE development set and shaped white Gaussian noise with those spectra. 22 anechoic sound samples (foot steps, coughing, office equipment, etc.) were randomly added to the noise samples to simulate non-stationary background noise. We then convolved these noise samples with the multi-channel AIRs in our training set to simulate decorrelated ambient and fan noise recordings. To simulate babble noise, we convolved random speech samples from our speech corpus with the multi-channel AIRs and added them to the simulated ambient noise samples.

The training samples were created by convolving random speech samples with the training AIRs and adding the synthetic noise samples to yield SNRs of 0, 10, and 20 dB, using the tools provided with the ACE corpus [9]. This resulted in a total of 23 850 reverberant, noisy training utterances with durations between 4 and 10 s. Table 1 summarises the data sets used in this work.

3.2. Data preprocessing

While neural network architectures exist that consume raw audio [19], and the first layers of deep convolutional neural networks have been shown to learn pre-processing filters directly from the

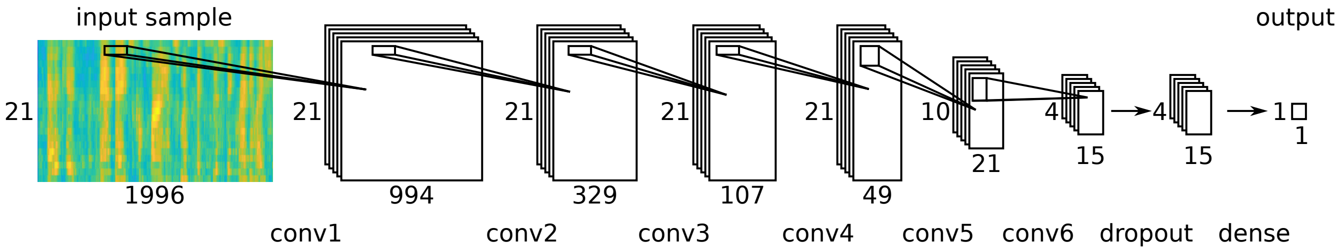

Fig. 3. Block diagram of CNN architecture.

data [18], for T_{60} estimation an intuitive first step seems to be to apply a transform to the input signal that reveals spectro-temporal features. The best-performing algorithms in the ACE Challenge transformed the input signal into the STFT domain before further processing [11]. When using convolutional neural networks, the dimensions of the input data affect the number of trainable parameters and thus the model complexity. Highly complex models are difficult to train and may result in overfitting [18]. To keep model complexity low while preserving signal information we deemed relevant for the T_{60} estimation problem we chose a transform with low spectral and high temporal resolution.

The input samples were split into chunks of four seconds with 0.5 seconds overlap. To remove parts with little or no audio activity, chunks whose RMS level was more than 20 dB lower than the RMS level of the entire sample were discarded. This yielded a total of 34 489 training samples. The level of each chunk was normalised using an A-weighting filter. Each chunk was then processed through a gammatone ERB filterbank [32] with 21 frequency bands from 400 Hz to 6 kHz. Temporal features were obtained in each band by taking the log of the energy summed in frames of 64 samples with an overlap of 32 samples, resulting in a feature matrix of size 21×1996 for each input chunk. Finally, we performed spectral whitening by subtracting the median value from each row of the feature matrix and normalised the features to have approximately zero mean and a standard deviation of one [18]. An example of the resulting pre-processed input feature matrix is shown in Figure 2.

3.3. Network architecture

The proposed network consists of six convolutional layers with a rectified linear unit (ReLU) activation function, each followed by a batch normalisation layer [33]. A single 50% drop-out layer is added to prevent overfitting [18]. The final layer consists of a fully connected layer with a linear activation function and a single output node. The output node produces a T_{60} estimate for every input feature matrix, i.e., one estimate for every four seconds of audio input. A block diagram of the proposed architecture is shown in Figure 3. The parameters of the convolutional layers are summarised in Table 2. As can be seen, the filters in the first four layers contain only a single row, i.e., they extract mostly temporal information. Spectral features are combined only in the last two layers as well as the final fully connected layer. The whole network contains a total of 2541 trainable parameters. The network seems to be of sufficiently low complexity to minimise overfitting the training data, while having enough capacity to capture the relations between spectro-temporal features and the T_{60}. Removing layers or complexity seemed to negatively affect performance, while increasing the number of trainable parameters seemed to either have only a marginal effect on performance or lead to overfitting.

	conv1	conv2	conv3	conv4	conv5	conv6
size	1×10	1×10	1×11	1×11	3×8	4×7
stride	(1, 2)	(1, 3)	(1, 3)	(1, 2)	(2, 2)	(2, 1)
# filters	5	5	5	5	5	5

Table 2. Specifications of the convolutional layers. The total number of trainable parameters in the whole network is 2541.

4. EXPERIMENTAL EVALUATION

The convolutional neural network was implemented using the Microsoft Cognitive Toolkit (CNTK) [34] and trained using the data generated as described in Section 3.1. Training was performed on two GPUs using stochastic optimisation [35] of a squared error loss function, over 1500 epochs. The final trained model was tested using the evaluation set of the ACE Challenge corpus [9].

To evaluate the performance of the proposed method the same criteria as for the ACE Challenge are used [9]:

- the estimation bias, calculated as the mean of the estimation error;
- the mean squared error (MSE);
- the Pearson correlation coefficient, ρ, between estimated and ground truth T_{60} values.

The ACE Challenge results also include the real-time factor (RTF) for each algorithm, calculated as the ratio between compute time and sample duration [11]. However, as the compute time is dependent on the specific computer hardware used for evaluation, the RTF of the proposed method and those reported in the ACE Challenge are not directly comparable. For reference, the RTF of the proposed method was estimated at about 0.05, i.e., 20 times faster than real-time, with about 95% of the compute time spent pre-processing speech files in Matlab. The actual T_{60} estimation using the pre-processed input samples and running on a GPU clocked in at a RTF of about 0.002, i.e., 500 times faster than real-time, due to CNTK being highly optimised [34]. While this result is not directly comparable to RTF values reported for the algorithms in the ACE Challenge, it serves as an indication of the computational efficiency of the proposed method.

Table 3 shows a comparison of the proposed method and two benchmark algorithms from the ACE Challenge: the best-performing machine-learning based algorithm [23], as well as the best-performing algorithm overall [10]. As can be seen, the proposed method outperforms both benchmark algorithms on all metrics, achieving the lowest bias and MSE as well as the highest Pearson correlation coefficient, ρ. It should be noted that the proposed method operates on input chunks with a fixed length of four seconds, i.e., shorter utterances are zero-padded while longer

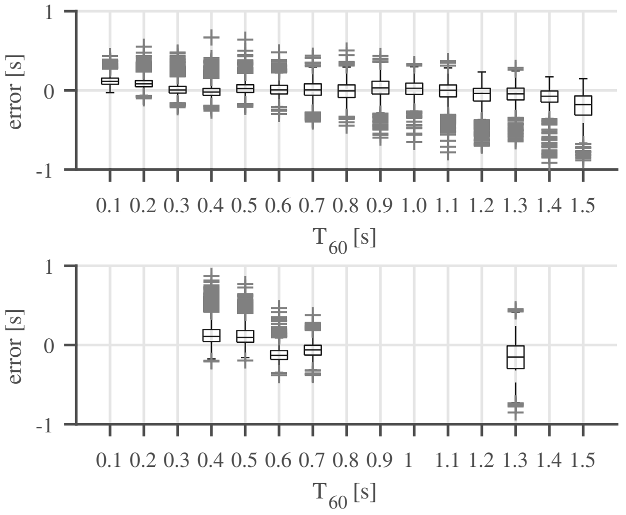

Fig. 4. Estimation error for training set (top) and evaluation set (bottom). For better visibility, the results are binned by T_{60} with a resolution of 0.1 s.

utterances result in multiple T_{60} estimates. Although measures were taken to prevent overfitting, the performance of the model on the training set was substantially better than on the ACE Challenge evaluation set, with a bias of 0.0055, a MSE of 0.0125, and a Pearson correlation coefficient of 0.953. This discrepancy between training and test performance illustrates the importance of strictly separating training and test sets when evaluating a machine learning model, especially when using small and/or synthetic data sets, as is quite common in the audio domain. Figure 4 shows the error performance for the training set (top) and the ACE Challenge evaluation set (bottom). The estimation error seems to increase towards higher T_{60} values. This is expected, as estimating a long T_{60} presumably requires a long input sample that is sufficiently sparse to observe long energy decays. Figure 5 shows confusion matrices for the training set (left) and the ACE Challenge evaluation set (right). As can be seen, the estimation error is distributed around the true T_{60} value, indicating that the CNN successfully learned a representation of the underlying regression problem. SNR did not seem to have a major effect on performance for the noise levels and types tested here.

5. CONCLUSION AND FUTURE WORK

Blind T_{60} estimation from noisy, reverberant speech remains a challenging problem. We show that the estimation can be modelled as a regression problem and implemented with a convolutional neural network (CNN). After training the model using over 30 000 input samples containing varying levels of ambient noise and reverberation and taking measures to combat overfitting, the proposed method is shown to outperform state-of-the-art algorithms on the ACE Challenge evaluation corpus for T_{60} estimation [9, 11]. Due to the highly optimised implementation of the model [34], the proposed estimation algorithm is shown to be computationally efficient, running significantly faster than real-time.

While these results are encouraging and prove the potential of the proposed approach, future work is needed to expand the training set, e.g., by collecting more data or using data augmentation [18]. This would potentially allow training a more complex network architecture with higher capacity and better performance. Furthermore, the model should be evaluated on a broader set of test cases to verify that the method generalises to unseen scenarios and noise types.

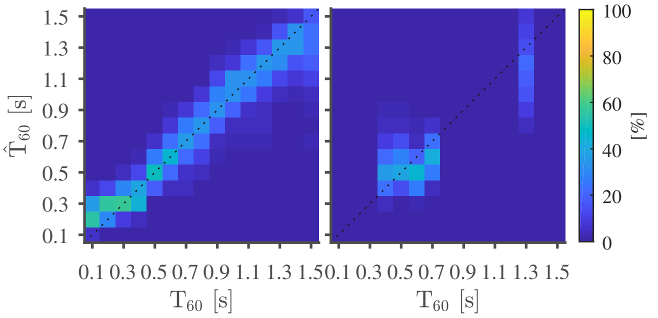

Fig. 5. Confusion matrices of ground truth T_{60} and estimates $\hat{T}_{60}$ for training set (left) and evaluation set (right). For better visibility, the results are binned by T_{60} with a resolution of 0.1 s.

Method	Bias	MSE	ρ
MLP [23]	−0.0967	0.104	0.48
QA Reverb [10]	−0.068	0.0648	0.778
CNN (proposed)	**0.0304**	**0.0384**	**0.836**

Table 3. Experimental results for the best-performing machine-learning based method [23] and the best method overall [10] in the ACE Challenge compared to the proposed approach, all evaluated on the ACE Challenge data corpus.

6. REFERENCES

[1] A. K. Nábělek and P. K. Robinson, "Monaural and binaural speech perception in reverberation for listeners of various ages," *J. Acoust. Soc. Am.*, vol. 71, no. 5, pp. 1242–1248, 1982.

[2] T. I. Niaounakis and W. J. Davies, "Perception of reverberation time in small listening rooms," *J. Audio Eng. Soc*, vol. 50, no. 5, pp. 343–350, 2002.

[3] H. A. Javed, B. Cauchi, S. Doclo, P. A. Naylor, and S. Goetze, "Measuring, modelling and predicting perceived reverberation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, March 2017, pp. 381–385.

[4] S. Chakrabarty and E. A. P. Habets, "Broadband DOA estimation using convolutional neural networks trained with noise signals," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct 2017.

[5] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2013, pp. 1–4.

[6] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015, pp. 5014–5018.

[7] M. Schroeder, “New method of measuring reverberation time,” *The Journal of the Acoustical Society of America*, vol. 37, no. 6, pp. 1187–1188, 1965.

[8] M. Karjalainen, P. Antsalo, A. Mäkivirta, T. Peltonen, and V. Välimäki, “Estimation of modal decay parameters from noisy response measurements,” *J. Audio Eng. Soc*, vol. 50, no. 11, pp. 867, 2002.

[9] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, “The ace challengecorpus description and performance evaluation,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.

[10] T. d. M. Prego, A. A. de Lima, R. Zambrano-López, and S. L. Netto, “Blind estimators for reverberation time and direct-to-reverberant energy ratio using subband speech decomposition,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.

[11] J. Eaton, N. D. Gaubitch, A. H. Moore, P. A. Naylor, J. Eaton, N. D. Gaubitch, A. H. Moore, P. A. Naylor, N. D. Gaubitch, J. Eaton, et al., “Estimation of room acoustic parameters: The ace challenge,” *IEEE/ACM Trans. Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 10, pp. 1681–1693, 2016.

[12] N. Faraji, S. M. Ahadi, and H. Sheikhzadeh, “Reverberation time estimation based on a model for the power spectral density of reverberant speech,” in *Proc. European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 1453–1457.

[13] H. Löllmann, E. Yilmaz, M. Jeub, and P. Vary, “An improved algorithm for blind reverberation time estimation,” in *Proceedings of International Workshop on Acoustic Echo and Noise Control (IWAENC)*, 2010, pp. 1–4.

[14] M. Lee and J. H. Chang, “Blind estimation of reverberation time using deep neural network,” in *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, Sept 2016, pp. 308–311.

[15] M. Senoussaoui, J. F. Santos, and T. H. Falk, “Speech temporal dynamics fusion approaches for noise-robust reverberation time estimation,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5545–5549.

[16] J. Eaton, N. D. Gaubitch, and P. A. Naylor, “Noise-robust reverberation time estimation using spectral decay distributions with reduced computational cost,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 161–165.

[17] M. Lee and J.-H. Chang, “Deep neural network based blind estimation of reverberation time based on multi-channel microphones,” *Acta Acustica united with Acustica*, vol. 104, no. 3, pp. 486–495, 2018.

[18] 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.

[19] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *CoRR*, vol. abs/1609.03499, 2016.

[20] E. Thuillier, H. Gamper, and I. Tashev, “Spatial audio feature discovery with convolutional neural networks,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, Alberta, Canada, 2018, IEEE, pp. 1–5.

[21] 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.

[22] D. Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3642–3649.

[23] F. Xiong, S. Goetze, and B. T. Meyer, “Joint estimation of reverberation time and direct-to-reverberation ratio from speech using auditory-inspired features,” in *Proc. ACE Challenge Workshop, a satellite of IEEE WASPAA*, New Paltz, NY, USA, 2015.

[24] Y. Xie, F. Xing, X. Kong, H. Su, and L. Yang, “Beyond classification: Structured regression for robust cell detection using convolutional neural network,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham, 2015, pp. 358–365, Springer International Publishing.

[25] A. Politis, *Microphone array processing for parametric spatial audio techniques*, G5 artikkelivitskirja, Aalto University, 2016.

[26] D. T. Murphy and S. Shelley, “Openair: An interactive auralization web resource and database,” in *Proc. 129th Audio Engineering Society Convention*. Audio Engineering Society, 2010.

[27] S. Nakamura, K. Hiyane, F. Asano, and T. Nishiura, “Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition,” in *Proc. International Conference on Language Resources and Evaluation*, Athens, Greece, 2000.

[28] “EchoThief Impulse Response Library,” http://www.echothief.com/, Online; accessed May 2018.

[29] “SOFA general purpose database,” https://www.sofaconventions.org/mediawiki/index.php/Files, Online; accessed May 2018.

[30] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *CoRR*, vol. abs/1710.05381, 2017.

[31] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, “Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1,” *NASA STI/Recon technical report n*, vol. 93, 1993.

[32] D. M. Brookes, “VOICEBOX: Speech processing toolbox for MATLAB,” http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, 2009, Online; accessed May 2018.

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

[34] F. Seide and A. Agarwal, “CNTK: Microsoft’s open-source deep-learning toolkit,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2135–2135.

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