INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Online Blind Reverberation Time Estimation Using CRNNs

Shuwen Deng, Wolfgang Mack and Emanuël A. P. Habets

International Audio Laboratories Erlangen (a joint institution of the Friedrich-Alexander-University Erlangen-Nuremberg (FAU) and Fraunhofer IIS), Germany

{wolfgang.mack,emanuel.habets}@audiolabs-erlangen.de
shuwen.deng@fau.de

Abstract

The reverberation time, T_{60}, is an important acoustic parameter in speech and acoustic signal processing. Often, the T_{60} is unknown and blind estimation from a single-channel measurement is required. State-of-the-art T_{60} estimation is achieved by a convolutional neural network (CNN) which maps a feature representation of the speech to the T_{60}. The temporal input length of the CNN is fixed. Time-varying scenarios, e.g., robot audition, require continuous T_{60} estimation in an online fashion, which is computationally heavy using the CNN. We propose to use a convolutional recurrent neural network (CRNN) for blind T_{60} estimation as it combines the parametric efficiency of CNNs with the online estimation of recurrent neural networks and, in contrast to CNNs, can process time-sequences of variable length. We evaluated the proposed CRNN on the *Acoustic Characterization of Environments Challenge* dataset for different input lengths. Our proposed method outperforms the state-of-the-art CNN approach even for shorter inputs at the cost of more trainable parameters.

Index Terms: acoustic parameter, online, reverberation time (T60) estimation, CRNN, deep learning, ACE challenge

1. Introduction

Acoustic parameters are useful to characterize the acoustic properties of enclosed spaces. The reverberation time, T_{60}, is one of the most important parameters to quantify the severity of reverberation. Many speech processing algorithms, e.g., in speech enhancement [1], or recognition [2], rely on it to mitigate the effects of reverberation or improve speech recognition accuracy. The T_{60} is defined as the time required for a 60 dB decay in sound energy after switching off a sound source in steady-state [3]. The energy decay characterized by the T_{60} is also observable in the energy decay curve (EDC) [4] of room impulse responses (RIRs) of the respective environment. Given access to RIRs, models can be fitted to the EDC to extract the T_{60}. Karjalainen et al. [5] proposed a nonlinear optimization strategy, where a parametric exponential decay model that incorporates a stationary noise floor is fitted to the decay envelope of the RIR. Nonlinear least-squares optimization is employed for the decay envelope fitting to search for an accurate and robust estimation of the decay rate.

In practice, the RIRs are typically unavailable, and blind estimation remains a challenging task. Consequently, it is desired to estimate the T_{60} blindly from audio recordings. Additive noise, thereby, further complicates T_{60} estimation [6]. In 2015, the *Acoustic Characterization of Environments* (ACE) challenge was held to find the most promising blind T_{60} estimation methods and to provide real noisy reverberant speech data for evaluation purposes (ACE Eval) [7]. The best single-channel T_{60} estimators were based on traditional signal processing approaches [8, 9]. Prego et al. [8] robustly estimated the T_{60} by statistically analyzing the signal decay in different frequency bands obtained by a frame-based energy decay function. Löllmann et al. [9] proposed a statistical model-based method for blind T_{60} estimation by using maximum-likelihood (ML) optimization to determine the most likely decay rate from the signal. Other approaches analyze decay rate distributions using model-based [10] or data-driven methods [11] to extract the T_{60}.

Recently, also deep learning-based methods have been proposed for blind T_{60} estimation achieving state-of-the-art performance on ACE Eval. Lee and Chang [12] used a feedforward neural network (FNN) to learn a mapping from input features to the T_{60} through multiple nonlinear hidden layers. The input feature representation is composed of reverberant speech decay rates in the short-time Fourier transform (STFT) domain and its distribution for each frequency bin. For their approach, a relatively long input signal is required to make the extracted decay rates reliable, and only one estimate can be obtained at the end, which makes it difficult to handle time-varying acoustic scenarios. Gamper et al. [13] proposed a convolutional neural network (CNN) to estimate the T_{60} directly from a four-second long recording of reverberant speech. In contrast to [12], extraction of features is outsourced to the CNN. Experimental results show that it achieves better performance and has higher computational efficiency than other methods in the ACE challenge. However, [13] operates on a fixed-length input to output one single predicted T_{60} value. In applications where the T_{60} changes over time, e.g., robot audition, audio augmented reality, and speech dereverberation for hearing aids, a real-time online system is desired to estimate the T_{60} continuously.

To handle time-varying scenarios, we propose to use a convolutional recurrent neural network (CRNN) to estimate the T_{60} blindly from reverberant noisy speech. The CNN layers ensure parameter efficiency, whereas the recurrent layers enable online T_{60} estimation for variable input lengths. The remainder is structured as follows. In Section 2, we introduce a signal model followed by a review of [13]. In Section 3, we elaborate on the proposed T_{60} estimation with a CRNN. The dataset generation and an overview of the datasets are given in Section 4. In Section 5, we describe experiments and evaluation results based on ACE Eval including a comparison to [8], [13] and [14].

2. Problem Formulation

We assume a noisy reverberant speech signal in time-domain, $y[t]$, with discrete-time index, t, captured by a single microphone in a room. The signal $y[t]$ is constructed by convolving a source speech signal $s[t]$ with a RIR $h[t]$ and adding additional noise $v[t]$, expressed as

$$y[t] = h[t] * s[t] + v[t] = \sum_{t'} h[t']s[t-t'] + v[t]. \qquad (1)$$

http://dx.doi.org/10.21437/Interspeech.2020-2156

The objective is to estimate the reverberation time directly from $y[t]$ without further information about h or s.

For this task, Gamper et al. [13] proposed a CNN to estimate the T_{60} from a four-second long signal. The architecture consists of six convolutional layers with rectified linear unit (ReLU) activation functions and one linear fully-connected layer, as shown in Figure 1. Each convolution layer is followed by a batch normalization layer (not shown in Figure 1). A dropout layer is added before the dense layer to prevent overfitting. The input of the CNN is the gammatone transformation of $y[t]$, denoted by $Y[k, n]$ with frequency index k and time frame index n. The gammatone transform with low spectral resolution reduces the model complexity while it is able to show the relevant information of the task [13]. The CNN is trained to minimize the mean-squared-error (MSE) between the estimated $\hat{T}_{60}$ and the oracle T_{60},

$$\text{MSE} = \frac{1}{T}\sum_{i=1}^{T}(\hat{T}_{60_i} - T_{60_i})^2, \qquad (2)$$

where i represents the sample index and T the batch size during training. The CNN has a low computational complexity and outperforms the competitors in the ACE challenge. The required fixed-length inputs, however, prevent estimates from shorter inputs and avoid usage of the complete temporal context assuming longer inputs are given. In addition, the architecture is not designed to estimate time-variant T_{60} values due to its single output. In the following, we present a method that addresses both issues.

3. Proposed Method

We propose to use a CRNN to perform online blind T_{60} estimation. Figure 2 depicts the proposed CRNN architecture. The CRNN consists of six convolutional layers with ReLU activation followed by a long short-term memory (LSTM) [15], a max-pooling layer, a dropout layer, and a time-distributed fully-connected layer.

For the convolutional layers, we adapt the framework from [13] and employ different temporal stride sizes in layers 2 and 3. Reducing the stride sizes of these two layers from (1, 3) to (1, 2) has a marginal effect on the estimation accuracy while allowing for shorter inputs to be fed into the CRNN. The DNN parameters are summarized in Table 1. The hidden layer size of the LSTM is 20, which is chosen according to the input size fed into the LSTM. A max-pooling layer is subsequently applied to reduce the parameters before the dense layer. The pooling size is set to 2, as proposed in [16]. The dropout rate after the max-pooling layer is 0.4. In contrast to [13], we add an additional ReLU activation function at the output, as the T_{60} values are supposed to be positive. The CRNN contains 5611 trainable parameters compared to 2541 parameters in [13].

The input to $\hat{T}_{60}$ mapping of the proposed CRNN is shown in Figure 2. A sliding window of length L with a specific hop-size R is applied to the input, which is subsequently fed into the CRNN. For each time frame, the CRNN yields one T_{60} estimate based on the information of the current windowed input and past information obtained from the recurrent LSTM structure. The minimum input length (corresponding to the window length) that can be fed into the CRNN depends on the kernel and stride sizes of the CNN part of the CRNN. The hop-size R determines the update time for $\hat{T}_{60}$.

Our proposed method combines parameter efficient feature extraction as in [13] with the sequence processing capabilities of LSTMs to achieve low complexity online T_{60} estimation. We propose to optimize the CRNN simultaneously for different input lengths with

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

Table 1: *Specifications of convolutional layers in the CRNN architecture.*

$$\text{MSE} = \frac{1}{TN}\sum_{i=1}^{T}\sum_{n=1}^{N}(\hat{T}_{60_{i,n}} - T_{60_i})^2, \qquad (3)$$

where n is the time step index of the LSTM output, and N is the total number of time steps per speech sample. For evaluation, we use the most recent estimate, i.e., $\hat{T}_{60}[N]$. With (3), in contrast to (2), the CRNN is trained simultaneously for input lengths ranging from L/f_s to $L/f_s + (N-1)\cdot R/f_s$ seconds with sampling frequency f_s.

4. Data Sets

The samples in the experiment are considered to be composed of reverberant speech and noise. For training data generation, we artificially generate a large number of labeled training and validation data samples by simulating the reverberation process with synthetic RIRs. The test set is the ACE challenge evaluation (ACE Eval) dataset, which allows direct comparison with state-of-the-art methods reported in [7] and [13].

4.1. Data Preprocessing

The reverberant noisy speech samples are resampled to a sampling frequency of 16 kHz and truncated to four-seconds. The input level is first processed by an A-weighting filter. The signal is then converted by the gammatone filterbank with 21 audible frequency bands that span the range 400 Hz to 6 kHz. The energy-per-band is calculated using 64 sample long windows with 32 samples hop-length. Finally, we subtracted the median value from each gammatone frequency band and standardized the complete input feature matrix for each sample to obtain an approximately zero mean and a standard deviation of one. The size of the pre-processed feature matrix is 21 × 1999. The input feature representation and the selection of parameters are as in [13].

4.2. Data Generation

The ACE Eval dataset is generated by using the software and the collection of real recorded speech files provided by the ACE challenge [7]. The EVAL dataset includes noisy reverberant speech files from 5 different rooms, with two microphone positions per room. The noise is of type ambient, fan, or babble and is mixed with the speech with a signal to noise ratio (SNR) of 0, 10, and 20 dB. The test set includes the ACE Eval single microphone configuration data and the first channel of the Eval multi-microphone configuration data. For evaluation, we only selected speech longer or equal to 4 s. For training and validation set generation, we convolve the artificial RIRs described in Section 4.3 with speech samples from the LibriSpeech Corpus [17] from training and validation set, respectively. We simulate

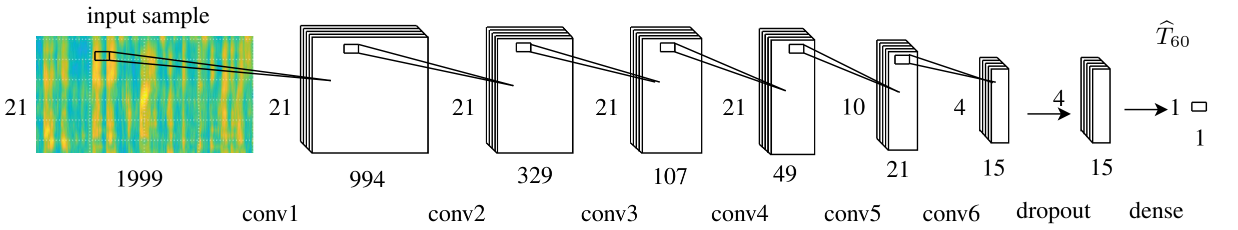

Figure 1: *Block diagram of the baseline CNN architecture [13]. A gammatone representation of the input is mapped to the reverberation time.*

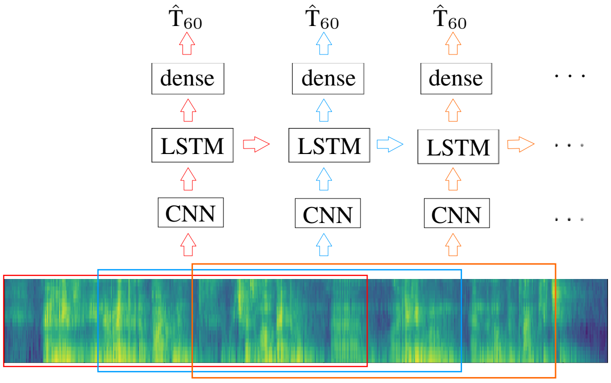

Figure 2: *The proposed CRNN architecture. A gammatone representation of the input is mapped to the reverberation time.*

Dataset	RIRs	# Speech Samples
training	synthetic	35000
validation	synthetic	2000
ACE Eval [7]	measured	16200

Table 2: *An overview of training, validation, and test sets.*

the noise similar to the noise in the ACE challenge as [13]. An overview of the parameters for the training, validation, and test set is given in Table 2.

4.3. Room Impulse Response Generation

For the training and validation set, RIRs are generated with the RIR generator [18], which is based on the source image-method [19]. Subsequently, we give an overview of the acoustic parameters for the RIR generator. We simulated RIRs for seven rooms as specified by the ACE challenge [7]. We define the source-microphone distance set $\mathcal{D}$ as $\{0.7$ m, 1 m, 1.3 m, 1.7 m, 2 m, 2.5 m, 3 m, 3.5 m$\}$. For each distance in $\mathcal{D}$, 10 source-microphone positions are generated, yielding a total of $7 \cdot 8 \cdot 10 = 560$ source-microphone positions. To provide a wide range of T_{60} values, the reverberation time set is defined from 0.3 s to 1.5 s, with an increment of 0.1 s.

To generate one RIR, the acoustic parameters are randomly sampled from the T_{60} range and the source-microphone positions. For each reverberant speech sample, a new RIR is generated. The oracle reverberation time T_{60} is calculated from each RIR with the method by Karjalainen et al. [5]. This is necessary due to inaccuracies between the input and the obtained reverberation time when using the RIR generator.

Method	Bias (s)	MSE (s)	ρ
MLP [14]	-.0967	.104	.48
QA Reverb [8]	-.068	.0648	.778
CNN [13]	**.0304**	.0384	.836
CNN [13] ($\star$)	.1163	.0375	.900
CRNN (proposed $\star$)	-.0488	**.0206**	**.917**

Table 3: *Experimental results on ACE Eval for blind T_{60} estimation of the proposed CRNN for four-second long inputs, high-performance algorithms [8] [14] in the ACE challenge and the baseline method [13]. Results obtained from our experiments are marked with $\star$. The other results are from the respective papers for recapitulation.*

5. Performance Evaluation

We trained two DNNs, the proposed CRNN and the CNN from [13]. We mark the DNNs we trained with a $\star$. For training, we used the Adam optimizer [20], and a learning rate of 0.001. We trained 100 epochs and selected the model with the lowest validation loss for evaluation. The performance of the proposed method is evaluated using the evaluation metrics proposed in [7], the bias, MSE, Pearson correlation coefficient ρ and, the real-time factor (RTF).

5.1. ACE Evaluation

The ACE challenge provides bias, MSE and, ρ results of signal processing and machine learning-based methods for blind T_{60} estimation on the ACE Eval dataset. For recapitulation, we report these results and also the results from [13] in Table 3. In addition, we reimplemented and trained the CNN proposed in [13] using our simulated training set, and report these results and our CRNN results for blind T_{60} estimation also in Table 3. For the CRNN, the MSE is calculated as in (2), using the last time-step output of the LSTM as $\hat{T}_{60}$.

In Table 3, the reimplemented CNN achieves similar results compared to those reported in [13] in terms of MSE. Our proposed CRNN outperforms all state-of-the-art methods in terms of MSE and ρ, and has a comparable bias as [13]. The highest Pearson correlation coefficient $\rho = 0.917$ indicates the high prediction accuracy of our proposed method. A boxplot of the estimation error over the T_{60} of the CRNN is depicted in Figure 3. As can be seen, the median values are close to 0 s and the percentiles are within ± 0.25 s. Better performance can be observed in the range of lower T_{60} values similar as in [13]. We assume that higher T_{60} values require more context to observe long energy decays for an accurate estimation.

The RTF is computed by averaging the required DNN processing time of 20000 input samples of four seconds on a CPU

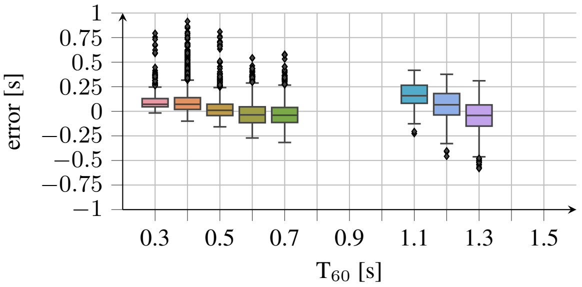

Figure 3: *Estimation errors on ACE Eval by the proposed CRNN. For each box, the notch inside the box is the median, the edges are the 25th and 75th percentiles, the whiskers extending above and below show the extreme values, and the outliers are plotted individually.*

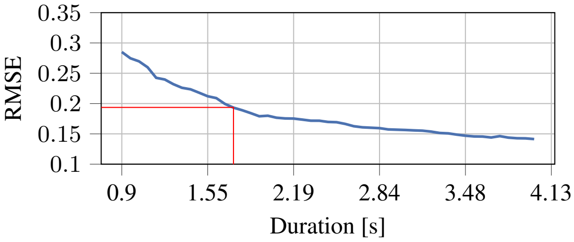

Figure 4: *RMSE of the CRNN on ACE Eval over different input duration lengths.*

with an Intel Core i7 processor. The RTF of the CRNN is approximately 110e-5 and 34e-5 for the CNN. Note that the CRNN yields 49 T_{60} estimates for a four-second input, whereas the CNN just yields a single estimate.

5.2. Effect of Input Duration on Estimation Accuracy

In this section, we investigate the performance of the CRNN as a function of the sample length. Figure 4 and Figure 5 show the root-mean-squared-error (RMSE) of the CRNN over the input sample duration. The window length L/f_s is approximately 0.9 s and the hop-size of subsequent context window R/f_s is about 0.0646 s. The time interval ranges from 0.9 to 4 s of sample length corresponding to $N = 49$ estimates $\hat{T}_{60}$. In Figure 4, the prediction accuracy of T_{60} values significantly depends on the length of the input sample although trained for all input sample lengths. The CRNN performs better for longer sample lengths. For the shortest input sample length of 0.9 s, the RMSE of $\approx$ 0.285 s is approximately twice the RMSE at 4 s. The red lines in Figure 4 and 5 mark the equal performance point (MSE of 0.0375 s, RMSE of 0.1936 s) of our proposed CRNN and [13]. In Figure 4, the required input sample length for equal performance is approximately 1.8 s. For longer input lengths, the CRNN outperforms [13]. In Figure 5, we analyze the effect of the input sample length on different T_{60} ranges. For that purpose, we divide the T_{60} in groups ranging from 0.3 s $\leq$ T_{60} $<$ 0.5 s, 0.5 s $\leq$ T_{60} $\leq$ 0.7 s, and 1.1 s $\leq$ T_{60} $\leq$ 1.3 s. Comparable to the results from Figure 4, more temporal context helps to reduce estimation errors. The required sample length for comparable performance to [13] does not depend on different T_{60} ranges. For a short context, the estimation in Figure 5 seems to be biased towards the medium T_{60} range as the respective RMSE is smaller compared to low and high T_{60}s. With increasing context lengths, the T_{60} estimates improve. High T_{60} values appear to be harder to estimate as the respective RMSE is higher at 4 s compared to the RMSE for the low and medium T_{60} ranges, similar to the results in Figure 3. Note that there is a non-linear relation between the decay and the T_{60}. Consequently, we get larger changes in the decay at lower T_{60} compared to those at higher T_{60}. Also, extracting information about a strong decay instead of a weak decay is less prone to noise. In line with our experiments, higher T_{60} are harder to estimate.

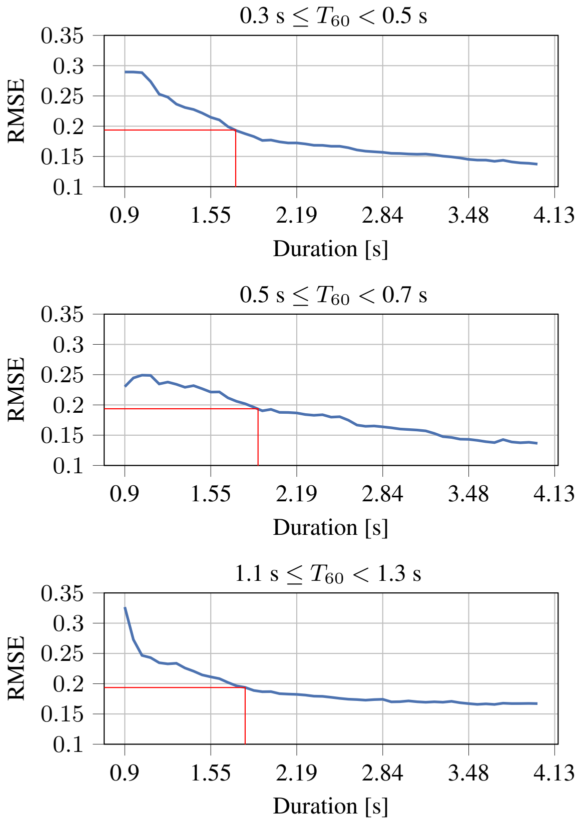

Figure 5: *RMSE of the CRNN on ACE Eval over different input duration lengths for 0.3 s $\leq T_{60} <$ 0.5 s (top), 0.5 s $\leq T_{60} \leq$ 0.7 s (middle), and 1.1 s $\leq T_{60} \leq$ 1.3 s (bottom).*

6. Conclusion

We presented a CRNN to achieve single-channel online blind T_{60} estimation, which can adapt to changing acoustic conditions. The proposed method extends the flexibility of DNNs for T_{60} estimation from fixed-length to variable-length inputs and achieves online T_{60} estimation. Our proposed method outperformed other state-of-the-art methods on ACE Eval in terms of Pearson correlation coefficient and MSE, even with shorter input lengths. Equi-performance to a CNN baseline with 4 s inputs was reached by our CRNN after 1.8 s at the cost of a higher computational complexity.

7. References

[1] E. A. P. Habets, "Single- and multi-microphone speech dereverberation using spectral enhancement," Ph.D. dissertation, Technische Universiteit Eindhoven, The Netherlands, 2007. [Online]. Available: http://alexandria.tue.nl/extra2/200710970.pdf

[2] T. Fukumori, M. Nakayama, T. Nishiura, and Y. Yamashita, "Estimation of speech recognition performance in noisy and reverberant environments using PESQ score and acoustic parameters," in *Proc. of the Asia-Pacific Sig. and Inf. Proc. Assoc. (APSIPA)*, 2013, pp. 1–4.

[3] M. R. Schroeder, "Reverberation time: Definition, theory, and measurement," *J. Acoustic. Soc. Am.*, vol. 39, no. 6, pp. 1230–1230, 1966.

[4] ——, "New method of measuring reverberation time," *J. Acoustic. Soc. Am.*, vol. 37, no. 6, pp. 1187–1188, 1965.

[5] P. Antsalo, A. Makivirta, V. Valimaki, T. Peltonen, and M. Karjalainen, "Estimation of modal decay parameters from noisy response measurements," *J. Audio Eng. Soc.*, vol. 50, no. 11, pp. 867–878, 2002.

[6] N. D. Gaubitch, H. W. Löllmann, M. Jeub, T. H. Falk, P. A. Naylor, P. Vary, and M. Brookes, "Performance comparison of algorithms for blind reverberation time estimation from speech," in *Proc. Intl. Workshop Acoust. Signal Enhancement (IWAENC)*, 2012, pp. 1–4.

[7] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor, "Estimation of room acoustic parameters: The ACE challenge," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 10, pp. 1681–1693, 2016.

[8] 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 Appl. Signal Process. Audio Acoust. (WASPAA)*, 2015.

[9] H. Löllmann, A. Brendel, P. Vary, and W. Kellermann, "Single-channel maximum-likelihood T60 estimation exploiting subband information," in *Proc. ACE Challenge Workshop*, 2015.

[10] J. Y. Wen, E. A. P. Habets, and P. A. Naylor, "Blind estimation of reverberation time based on the distribution of signal decay rates," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 329–332.

[11] R. Talmon and E. A. P. Habets, "Blind reverberation time estimation by intrinsic modeling of reverberant speech," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 156–160.

[12] M. Lee and J.-H. Chang, "Blind estimation of reverberation time using deep neural network," in *Proc. Intl Conf. on Network Infr. and Dig. Content (IC-NIDC)*, 2016, pp. 308–311.

[13] H. Gamper and I. J. Tashev, "Blind reverberation time estimation using a convolutional neural network," in *Proc. Intl. Workshop Acoust. Signal Enhancement (IWAENC)*, 2018, pp. 136–140.

[14] 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. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, 2015.

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

[16] F.-R. Stöter, S. Chakrabarty, B. Edler, and E. A. P. Habets, "CountNet: Estimating the number of concurrent speakers using supervised learning," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 2, pp. 268–282, 2019.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[18] E. A. P. Habets, *Room impulse response (RIR) generator. [Online]*, 2008, available: http://github.com/ehabets/RIR-Generator.

[19] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, 1979.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Intl. Conf. on Learn. Repr. (ICLR)*, 2015.