INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

End-to-End Speaker Identification in Noisy and Reverberant Environments Using Raw Waveform Convolutional Neural Networks

Daniele Salvati, Carlo Drioli, Gian Luca Foresti

Department of Mathematics, Computer Science and Physics, University of Udine
{daniele.salvati, carlo.drioli, gianluca.foresti}@uniud.it

Abstract

Convolutional neural network (CNN) models are being investigated extensively in the field of speech and speaker recognition, and are rapidly gaining appreciation due to their performance robustness and effective training strategies. Recently, they are also providing interesting results in end-to-end configurations using directly raw waveforms for classification, with the drawback however of being more sensible on the amount of training data. We present a raw waveform (RW) end-to-end computational scheme for speaker identification based on CNNs with noise and reverberation data augmentation (DA). The CNN is designed for a frame-to-frame analysis to handle variable-length signals. We analyze the identification performance with simulated experiments in noisy and reverberation conditions comparing the proposed RW-CNN with the mel-frequency cepstral coefficients (MFCCs) features. The results show that the method offers robustness to adverse conditions. The RW-CNN outperforms the MFCC-CNN in noise conditions, and they have similar performance in reverberant environments.

Index Terms: speaker identification, convolutional neural network, end-to-end configuration, raw waveform, data augmentation, noise, reverberation.

1. Introduction

Since many decades, machine learning and neural network models have been successfully employed in a wide range of speech and audio processing applications, such as automatic speech recognition (ASR) [1], audio forensic [2], music information retrieval [3], sound classification [4], array signal processing [5]. Moreover, since the new computational and performance advances brought by the recent developments in the field of deep neural networks (DNNs) research, the use of this particular computation model is now being investigated in a variety of acoustic and speech oriented applications. Among these, a consistent number of researches has recently focused on speaker localization [6, 7], speech enhancement [8], speaker/speech recognition [9, 10, 11, 12].

Recently, there have been various interests to directly use the raw waveform (RW) to learn feature representation with DNNs [13, 14, 15, 16]. In [13], it is shown that the RW and mel-frequency cepstral coefficients (MFCCs) with convolutional neural network (CNN) have comparable performance for the estimation of phoneme class conditional probabilities. The ASR performance with RW and DNN has been analyzed in [14] with promising results that however present a slight worse gap with the MFCC features. In [15], it is shown that RW matches the ASR performance of log-mel filterbank energies using convolutional long short-term memory DNNs. The RW-DNN has been demonstrated to outperform a DNN that uses log-mel filterbank magnitude features using a multichannel system for ASR [16].

In general, the machine learning is more sensible to the amount of training data when the RW are directly used as input in the DNN. For example, the reverberation can significantly affect the waveform degrading the classification performance. While deep neural networks in speech recognition systems may lead to superior performances over other traditional schemes, this comes at the cost of higher requirements in terms of training datasets, since deep architectures require a large speech database for training. Among the possible solutions, data augmentation (DA), a widespread strategy to increase the quantity of training data in small datasets, has been shown to be beneficial for neural network training as it both reduces overfitting and increases robustness against noise [17, 18, 19]. This approach can be exploited on acoustic data by a variety of techniques. In [19], DA for the acoustic model training of an ASR system is achieved by operating perturbations on traditional parameters computed by an acoustic front-end. In [18], DNN-ASR is also addressed by training data augmentation, relying in this case also on vocal tract normalization and vocal tract length perturbation. In [17], the authors discuss different audio-level speech manipulations obtained by directly processing the raw acoustic signal, including speech waveform perturbation that results into a shift of the spectrum in the mel spectrogram.

In this paper, we discuss the performance of a RW-based CNN model training with data augmentation strategy based on noise and reverberation. The technique is investigated in an end-to-end computational scheme for speaker identification, in which the raw audio signal is used as the input of the network, without any further acoustic front-end processing. While the use of noise is a consolidate practice in the DA for DNN, to the best of our knowledge, no systematic study has been conducted on the reverberation effect with a RW-CNN approach. Specifically, we demonstrate that the DA related to reverberation in the RW-CNN is independent of the room size and of the relative position between microphone and source. Hence, the RW-CNN performance is primarily affected by late reverberation, and the DA is effective by taking into account the reverberation time. We present a one-dimensional CNN scheme based on frame-to-frame analysis to handle variable-length signals. We design accurately the CNN model by analyzing the convolutional and pooling effects on raw signals. This analysis leads to set the same convolutional filter size in all layers. We investigate the identification performance of the model in noisy and reverberant environments with simulated experiments, comparing the proposed RW-CNN performance with the MFCC-CNN.

2. Raw Waveform CNN Acoustic Model

Let $s_n(t)$ denote the signal generated by a nonstationary speaker source n ($n = 1, 2, \ldots, N$, where n identifies the speaker of a dataset contains N speakers) at time t. The out-

http://dx.doi.org/10.21437/Interspeech.2019-2403

put signal model of the microphone is given by

$$x(t) = (h * s_n)(t) + v(t), \tag{1}$$

where $h(t)$ is the impulse responses from the source to the sensor, and $v(t)$ is an additive noise that is assumed to be spatially white Gaussian.

We aim at designing a nonlinear function $F(\cdot, \mathbf{\Theta})$ using a CNN ($\mathbf{\Theta}$ are the learned parameters during the training), which maps the input raw waveform $\mathbf{x}(t)$ of the n-the speaker to the output prediction class n

$$n(t) = F(\mathbf{x}(t), \mathbf{\Theta}). \tag{2}$$

The input signal is a vector of length L

$$\mathbf{x}(t) = [x(t - L + 1), x(t - L + 2), \ldots, x(t)]. \tag{3}$$

The overall structure of the one-dimensional convolution CNN network is made of several convolution layers, followed by fully-connected layers and a classification layer. The data undergoes a filtering and activation detection step operated through the one-dimensional convolutional, as

$$\mathbf{h}^l = \sigma(\mathbf{w}^l * \mathbf{h}^{l-1} + b^l), \tag{4}$$

where $\mathbf{h}^l$ and $\mathbf{h}^{l-1}$ are feature maps in two consecutive layers, $\mathbf{w}^l$ is a trained kernel, b^l is a bias parameter, $\sigma(\cdot)$ is the activation function, and * denotes convolution. The rectified linear unit (ReLU) [20] is a common operation for generating the output of the convolutional layer. It computes the function $f(x) = \max(0, x)$. The bias guarantees that every node has a trainable constant value. The kernels are computed through a stochastic gradient descent method [21], which minimizes a loss function measuring the discrepancy between the CNN predictions and the targets. The loss function for classification is the cross entropy [22].

The output of the convolutional layers is then used as the input of one or more fully connected layer, in which each neuron is connected to all neurons of the previous layer. A fully connected layer multiplies the input by a weight matrix and then adds a bias vector

$$\mathbf{h}^l_{\text{FC}} = \sigma(\mathbf{W}^l \mathbf{h}^{l-1}_{\text{FC}} + \mathbf{b}^l). \tag{5}$$

To operate a dimensionality reduction and yield more robust features, a pooling layer following the ReLU layer is typically used with an averaging or maximizing operation with respect to the dimension of the feature [23]. The pooling operation performs down-sampling and consists in dividing the input into pooling regions, computing the average or the maximum of each region. Another operation that is commonly used in CNNs to prevent overfitting is the dropout [24], which randomly sets input elements to zero with a given probability. To speed up the training of CNNs and reduce the sensitivity to network initialization, the batch normalization is used to normalize the data across a mini-batch, back-propagating the gradients through the normalization parameters [25]. The activation function used in classification layer is the softmax function [26].

The features extraction operation from the raw waveform consists mainly of two processes: convolution and max (or average) pooling. It is interesting to underline the effect of these operations to a time-domain audio signal. First, the convolution theorem states that the Fourier transform of a convolution of two signals is the point-wise product of their Fourier transforms. The convolution, which is typically computed with small kernel size, computes a filter transformation of the input with low frequency resolution. For example, assuming a sampling rate of 16 kHz and a kernel size of 32, the frequency resolution will be 500 Hz. In this case, through the optimization algorithm, the kernel learns to emphasize the input data on 16 sub-bands with resolution 500 Hz. The max pooling performs down-sampling. Considering an operation with size 2 and stride 2, an input of length L will be $L/2$ after the pooling. Basically, a sub-sampling is computed reducing the sampling frequency by half. If the sampling rate is 16 kHz, it will be 8000 kHz after the pooling. We can note that this operation introduces aliasing in the transformed signal. The frequency components between 4000 and 8000 Hz are not lost, but they are projected in the spectrum as aliasing frequency [27]. After successive convolutional and pooling layers, the audio signal is again filtered and down-sampled. This results in an effective features extraction operation. The convolution emphasizes frequency components of the input with few sub-bands, and the pooling reduces the input size without loss of information since high frequency components are projected as aliasing frequencies in the spectrum. A small filtered version of the audio input will be more robust for the neural network processing if it represents the input with essential characteristics. For this reason, it is particularity important to appropriately set the kernel size taking into account that the pooling operation has to reduce and to filter the input size preserving prominent frequency components.

3. Speaker Identification

We propose a network scheme to handle variable-length signals by designing the CNN input using a short signal frame L. This strategy makes the network flexible for the speaker segment to analyze due to the short frame setting.

The speaker identification based on the raw waveform CNN acoustic model is computed using a segment of the signal composed of B frames of length L. The sequence of input vectors is $\mathbf{x}(t + bR)$, $b = 0, 1, \ldots, B - 1$, where R is the overlap step. Each input vector $\mathbf{x}(t + bR)$ of size L is processed by the CNN, which estimates B predictions for the identification.

The speaker identification is calculated as

$$\hat{n} = \underset{n}{\text{argmax}} \Big[\sum_{b=0}^{B-1} p_n(t + bR) \Big], \tag{6}$$

where $p_n(t + bR)$ is the prediction output of the input vector $\mathbf{x}(t + bR)$ for the classification of the n-th speaker. The B outputs whose sum correspond to the largest value provides the speaker class predicted for the speech input signal.

The noise and reverberation at the microphone strongly affects the RW-CNN model. We adopt a DA strategy in the training phase that takes into account the signal-to-noise ratio (SNR) level and the reverberation time (RT_{60}). Different SNR levels are obtained by adding mutually independent white Gaussian noise to the training signals. The reverberation effects are obtained with an image-source model [28]. In general, the acoustic impulse response is composed of the direct-path signal, the early reflections and the late reverberation. Early reflections have strong and distinct peaks providing spatial information, while the late reverberation has low intensity and high density of reflections providing room information, and no longer depends on source position [29]. We will show in the experiment section that the RW-CNN is primarily depended on late reverberation, and the DA is effective computed by considering only the reverberation time, since the position between source and

microphone does not affect significantly the identification performance.

Given a training data segment $\mathbf{s}_n$, the DA training segment is based on the following signal transformation

$$\mathbf{s}_n^{\text{DA}} = \Big[\frac{\mathbf{s}_n}{\max|\mathbf{s}_n|}, \frac{\mathbf{s}_n^{\text{SNR}^1}}{\max|\mathbf{s}_n^{\text{SNR}^1}|}, \frac{\mathbf{s}_n^{\text{SNR}^2}}{\max|\mathbf{s}_n^{\text{SNR}^2}|} \cdots, \frac{\mathbf{s}_n^{\text{SNR}^V}}{\max|\mathbf{s}_n^{\text{SNR}^V}|} \\ \frac{\mathbf{s}_n^{\text{RT}_{60}^1}}{\max|\mathbf{s}_n^{\text{RT}_{60}^1}|}, \frac{\mathbf{s}_n^{\text{RT}_{60}^2}}{\max|\mathbf{s}_n^{\text{RT}_{60}^2}|} \cdots, \frac{\mathbf{s}_n^{\text{RT}_{60}^I}}{\max|\mathbf{s}_n^{\text{RT}_{60}^I}|} \Big], \tag{7}$$

where $\mathbf{s}_n^{\text{SNR}^v}$ is a noisy version of $\mathbf{s}_n$ with a specific SNR, V is number of SNR levels, $\mathbf{s}_n^{\text{RT}_{60}^i}$ is a reverberant version of $\mathbf{s}_n$ with a specific RT_{60}, and I is number of RT_{60}. Note that we consider noisy and reverberant conditions as independent in this work. However, both noisy and reverberant signals can be used in (7).

4. CNN Architecture

In this section, the architectures of the proposed CNN will be described in detail. The network consists of 5 one-dimensional convolutional layers, 3 fully connected layers, and a classification layer with softmax function. We carefully tune the kernel size and the number of filters of convolutional layers, the max pooling operation, and the output of intermediate fully connected layers to obtain an optimization performance. After each convolutional layer, the batch normalization and the activation with the ReLU are computed. Then a max pooling layer operates a dimensionality reduction. Each kernel of the convolutional layers has dimension 1×16 with stride of 1 adding zero padding to have the same size of the output as the input. In the first convolutional layer, the number of filters is 32, and it is doubled for each subsequent convolutional layer. The max pooling layers have dimensions 1×2 with stride 2. To enhance nonlinearity and to reduce overfitting, 3 fully connected layers are used with two dropout layers between them. The dropout layer is set with a probability of 0.5. The first and the second fully connect layers have 512 neurons. The last fully connect layer has N neurons. The network is thus composed by 29 layers (1 input, 5 convolutional, 5 max pooling, 5 batch normalization, 7 ReLU, 3 fully connect, 2 dropout, 1 classification).

In this study, we use a length frame L of 1024 samples (64 ms) with a sampling rate of 16 kHz. The size of convolutional kernels is the same for all layers. This setting allows the increment of the filter resolution at each next convolutional layer due to the down-sampling operated by the max pooling. A kernel of size 16 corresponds to a filtering operation with frequency resolution of 1000 Hz in the first convolutional layer, of 500 Hz in the second convolutional layer, and so on. The last convolutional layer has a frequency resolution of 31.25 Hz. The size of the feature maps is hence 32 samples with 512 filters. Table 1 shows the architecture of the proposed CNN.

The training of the CNN is computed through a stochastic gradient descent method [21], which minimizes a cross entropy loss function measuring the discrepancy between the CNN prediction and the target. The learning rate is set to 0.01 using a mini-batch size of 128. The number of epochs is 100.

5. Simulations

The speaker identification performance is illustrated through a set of simulated experiments. The simulations in noisy conditions were conducted with different SNR levels, obtained by adding mutually independent white Gaussian noise. The experiments in reverberant conditions was simulated with an improved image-source model [30]. The source speech signals used to generate noisy and reverberant speech were taken from the TSP speech database [31]. The TSP speech database consists of 1378 utterances spoken by 23 speakers (12 females, 11 males). Each utterance has a length of about 2 s. The speech was recorded in an acoustic anechoic room. The dataset partitioning is a 70-30 split of the number of segments in training and test subsets. The training and the test subsets consist of 889 and 389 utterances, respectively. The identification performance was computed for each utterance with an overlap step of $R = 512$ samples. The number of blocks B for the test subset was in the range $[52, 103]$. Each utterance was normalized (peak normalization) before passing to the CNN.

The DA training was conducted on clean signals, on two SNR level corrupted signals ($V = 2$), and on three reverberation signals ($I = 3$). The considered SNRs were 0 dB and 10 dB. The reverberation was computed with a simulated room of 5 m $\times$ 4 m $\times$ 3 m. The position of the microphone was $(2, 3, 1)$ m and the position of the source was $(4, 1, 2)$ m. The considered RTs_{60} were 0.1 s, 0.3 s, and 0.5 s.

We compared the performance of the speaker identification based on the RW-CNN and on the MFCC-CNN, training with

Table 1: *The architecture of the proposed CNN.*

l	Layer	Description	Output Size
1	Input	raw waveform	1024×1
2	Convolution	1×16, 32 filters	1024×32
3	Batch normalization		1024×32
4	ReLU		1024×32
5	Max pooling	1×2, stride 2	512×32
6	Convolution	1×32, 64 filters	512×64
7	Batch normalization		512×64
8	ReLU		512×64
9	Max pooling	1×2, stride 2	256×64
10	Convolution	1×16, 128 filters	256×128
11	Batch normalization		256×128
12	ReLU		256×128
13	Max pooling	1×2, stride 2	128×128
14	Convolution	1×16, 256 filters	128×256
15	Batch normalization		128×256
16	ReLU		128×256
17	Max pooling	1×2, stride 2	64×256
18	Convolution	1×16, 512 filters	64×512
19	Batch normalization		64×512
20	ReLU		64×512
21	Max pooling	1×2, stride 2	64×512
22	Fully connected	output size: 512	1×512
23	ReLU		1×512
24	Dropout	probability: 0.5	1×512
25	Fully connected	output size: 512	1×512
26	ReLU		1×512
27	Dropout	probability: 0.5	1×512
28	Fully connected	output size: N	$1 \times N$
29	Classification	predicted class probabilities	$1 \times N$

Table 2: *The identification performance IA (%) in noisy conditions.*

SNR (dB)	30	25	20	15	10	5	0	-5
MFCC-CNN	100	99.49	98.71	96.92	90.49	80.72	61.18	31.62
RW-CNN	100	100	100	100	100	99.49	93.32	44.99

Table 3: *The identification performance IA (%) in reverberant conditions (room of 7 m × 6 m × 4 m).*

RT_{60} (s)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
MFCC-CNN	97.43	96.66	96.14	96.14	96.14	95.37	94.60	94.09
RW-CNN	97.43	95.37	94.09	93.32	92.29	92.03	90.97	90.46

Table 4: *The identification performance IA (%) in reverberant conditions (room of 12 m × 7 m × 3.5 m).*

RT_{60} (s)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
MFCC-CNN	97.94	96.92	95.63	95.12	95.12	94.09	93.57	93.32
RW-CNN	97.94	95.12	94.09	93.32	92.03	91.77	91.52	91.52

the same DA dataset. MFCCs vectors of length 21 were used as input (the zero-th order coefficient is excluded). The MFCCs are calculated with a frame of 1024 samples (the same used for the RW input) and they are normalized with zero mean and standard deviation equal to 1. The same CNN structure for the raw waveform was used. In this case, the max-pooling operations were removed, and the filter sizes for the 5 convolutional layers were: 7, 5, 5, 3, 3. Performance is reported in terms of the percentage of identification accuracy (IA).

First, a simulation in noisy conditions was conducted. Table 2 reports the results at variation of the SNR level. We can observe the robustness to noise of the RW-CNN, while MFCC-CNN degrades the performance when the SNR level decreases. The RW-CNN has a good performance until an SNR of 0 dB (the DA training was performed with SNR of 0 dB and 10 dB), and the IA becomes poor for an SNR of -5 dB.

Next, an evaluation in reverberant conditions was performed using two room setups different from the training. Tables 3 and 4 show the IA performance for a room 7 m × 6 m × 4 m (the same of the training) and of 12 m × 7 m × 3.5 m, respectively. The SNR was 30 dB. The source and the microphone positions were randomly chosen for each utterance by considering a minimum distance from the wall of 0.5 m and a minimum distance between source and microphone of 1 m. The identification was hence computed on 389 source-microphone positions. In this case, the MFCC-CNN and RW-CNN has a comparable performance with a slightly better identification rate for MFCC when the reverberation time increases. We can underline that the identification of the RW-CNN has an excellent generalization with different rooms and relative position between microphone and source. Hence, the RW-CNN results dependent primarily on the late reverberation and the reverberation time. We can also note that the RW-CNN performance does not degrade for RT_{60} greater than 0.5 s (the maximum reverberation time in the DA dataset).

In Table 5, we can see the RW-CNN performance using a training dataset with only clean signals and with clean-noise signals. We can observe, the advantage of the DA using noise in a low SNR condition, which is however ineffective in reverberation conditions. The effect of late reverberation has to take into account as we have seen in Tables 3 and 4.

Finally, the frame IA (FIA) was reported in Table 6. The FIA is the percentage of identification accuracy by considering each frame individually. The capability of the RW-CNN to correctly identify the speaker is widely greater for clean and noisy signals with respect to MFCC features. In the reverberation case, the RW and MFCC have similar identification performance.

To conclude, the proposed CNN model that uses raw waveforms in the identification task is rather promising since it provides robustness to noise and the DA reverberation results robust to room size variations and to the early reverberation due to different positions of the microphone and the source. We foresee the integration of the proposed technique with a multichannel sensor array [32] that can be used to enhance the acoustic front in order to attenuate point-source interferences and to address multi-source cases.

Table 5: *The identification performance IA (%) of RW-CNN without DA (clean training) and with noise DA (clean-noise training).*

	Clean	Noise (SNR=5 dB)	Reverb (RT_{60}=0.5 s)
RW-CNN (clean tr.)	100	42.42	36.25
RW-CNN (clean-noise tr.)	100	99.49	31.11

Table 6: *The frame identification performance FIA (%).*

	Clean	Noise (SNR=5 dB)	Reverb (RT_{60}=0.5 s)
MFCC-CNN	45.72	16.85	37.32
RW-CNN	64.01	46.24	38.51

6. Conclusions

In this paper, we presented an end-to-end raw waveform CNN acoustic model for the speaker identification based on a data augmentation with noise and reverberation. The CNN architecture is designed to operate in a frame-by-frame analysis to handle variable-length signals. We have shown that the DA reverberation is independent of the relative position between microphone and source and of the room dimension, and it is primarily related to late reverberation by considering the reverberation time. We have demonstrated that the RW-CNN is more robust to noise if compared to the MFCC-CNN trained with the same DA dataset, and they have similar performance in reverberant environments. Future works include the integration of the technique in a multichannel sensor array system.

7. References

[1] J. P. Campbell Jr., "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[2] X. Lin, J. Liu, and X. Kang, "Audio recapture detection with convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1480–1487, 2016.

[3] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2016.

[4] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[5] D. Salvati, C. Drioli, and G. L. Foresti, "A weighted MVDR beamformer based on SVM learning for sound source localization," *Pattern Recognition Letters*, vol. 84, pp. 15–21, 2016.

[6] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 405–409.

[7] D. Salvati, C. Drioli, and G. L. Foresti, "Exploiting CNNs for improving acoustic source localization in noisy and reverberant conditions," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 103–116, 2018.

[8] N. Zheng and X. Zhang, "Phase-aware speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 63–76, 2019.

[9] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[10] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, 2014.

[11] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2203–2213, 2014.

[12] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[13] D. Palaz, R. Collobert, and M. Magimai-Doss, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks," in *Proceedings of the Conference of the International Speech Communication Association*, 2013.

[14] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Proceedings of the Conference of the International Speech Communication Association*, 2014.

[15] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proceedings of the Conference of the International Speech Communication Association*, 2015.

[16] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4624–4628.

[17] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proceedings of the Conference of the International Speech Communication Association*, 2015.

[18] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, 2015.

[19] I. Rebai, Y. BenAyed, W. Mahdi, and J.-P. Lorré, "Improving speech recognition using data augmentation and acoustic model fusion," *Procedia Computer Science*, vol. 112, pp. 316–322, 2017.

[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the International Conference on Machine Learning*, 2010, pp. 807–814.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams., "Learning internal representations by error propagation." in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*. MIT Press, 1986, pp. 318–362.

[22] P. T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

[23] M. Ranzato, F. J. Huang, Y. L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

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

[26] J. S. Bridle, *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*. Springer, 1990, pp. 227–236.

[27] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[28] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[29] H. Kuttruff, *Room Acoustics*. Spon Press, 2009.

[30] E. Lehmann and A. Johansson, "Prediction of energy decay in room impulse responses simulated with an image-source model," *Journal of the Acoustical Society of America*, vol. 124, no. 1, pp. 269–277, 2008.

[31] P. Kabal, "TSP speech database," McGill University, Montreal, Quebec, Tech. Rep., 2002.

[32] D. Salvati, C. Drioli, and G. L. Foresti, "Joint identification and localization of a speaker in adverse conditions using a microphone array," in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 21–25.