



Effects of Skip Connections in CNN-Based Architectures for Speech Enhancement

Nengheng Zheng¹ · Yupeng Shi¹ · Weicong Rong¹ · Yuyong Kang¹

Received: 19 February 2019 / Revised: 5 December 2019 / Accepted: 8 January 2020 / Published online: 15 May 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Eliminating the negative effect of adverse environmental noise has been an intriguing and challenging task for speech technology. Neural networks (NNs)-based denoising techniques have achieved favorable performance in recent years. In particular, adding skip connections to NNs has been demonstrated to significantly improve the performance of NNs-based speech enhancement systems. However, in most of the studies, the adding of skip connections was kind of tricks of the trade and lack of sufficient analyses, quantitatively and/or qualitatively, on the underlying principle. This paper presents a denoising architecture of Convolutional Neural Network (CNN) with skip connections for speech enhancement. Particularly, to investigate the inherent mechanism of NNs with skip connections in learning the noise properties, CNN with different skip connection schemes are constructed and a set of denoising experiments, in which statistically different noises being tested, are presented to evaluate the performance of the denoising architectures. Results show that CNNs with skip connections provide better denoising ability than the baseline, i.e., the basic CNN, for both stationary and nonstationary noises. In particular, benefit by adding more sophisticated skip connections is more significant for nonstationary noises than stationary noises, which implies that the complex properties of noise can be learned by CNN with more skip connections.

Keywords Speech enhancement · Convolutional neural network · Skip connections · Stationary noises · Nonstationary noises

1 Introduction

Speech enhancement, aiming at eliminating the negative effect of environmental noises from the target speech, is a fundamental task for speech applications including Man-Machine speech interaction, hearing assistive devices, etc. [1–3]. Although microphone array-based speech enhancement techniques have been widely implemented in practical speech systems [1], monaural speech enhancement is still an intriguing

and challenging problem. Numerous signal processing techniques have been developed for the task over the past decades. For example, spectral subtraction and Wiener filtering are two of the most fundamental unsupervised noise suppression algorithms and have been studied extensively for various practical problems [4]. More sophisticated algorithms including the minimum mean squared error (MMSE) estimation, optimally modified log-spectral amplitude (OM-LSA) speech estimator, subspace projection, to name only a few, have been presented in literature [4, 5]. However, due to the diversity of noise conditions in real-world applications, these methods suffer from the inability to estimate and reduce the nonstationary noises effectively.

In recent years, neural networks (NNs)-based deep learning systems, e.g., the deep neural network (DNN), the convolutional neural network (CNN), etc., have been widely implemented for automatic speech recognition (ASR) [6–9] and the recognition performance of such systems significantly outperform the conventional Hidden Markov Models (HMMs)-based systems [10]. The success of NNs in ASR has inspired the study of NNs-based speech enhancement. Fu et al. proposed an SNR aware CNN model for speech denoising [11]. A CNN was trained to estimate the SNR of the noisy speech, with which a specific SNR-

✉ Yupeng Shi
2172262986@email.szu.edu.cn

Nengheng Zheng
nhzheng@szu.edu.cn

Weicong Rong
2172262944@email.szu.edu.cn

Yuyong Kang
1810262077@email.szu.edu.cn

¹ Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University, Shenzhen, Guangdong, China

dependent CNN is selected (from a pool of pre-trained SNR-dependent CNNs) to denoise the current input speech. In [12], denoising autoencoders (DAEs) were constructed with fully connected convolutional topology. Results showed that better performance can be achieved by the mapping-based DAs than that by the explicit masking-based ones. In [13], a convolutional network architecture with multiple skip connections was proposed. The authors modified the convolutional Encoder-Decoder (CED) network proposed in [14] by removing max-pooling layers and adding skip connections to the convolutional layers. The results showed that the denoising performance of the proposed architecture, even with very small network size, was better than that of the Feedforward Neural Network (FNN) and the Recurrent Neural Network (RNN) [15]. More recently, sophisticated CNN-based architectures with skip connections were adopted for monaural speech enhancement [16, 17]. So-called generative adversarial networks (GANs) [18, 19], in which the generator and the discriminator are all composed of CNN structures, were adopted for speech enhancement with remarkable success in both waveform and spectral domains.

The effectiveness of adopting skip connections to CNN was first investigated in image recognition [20, 21]. It was shown that the features passed by skip connections carried more useful details for the CNN to recover an image. Similar results were obtained in speech enhancements, e.g., [13, 22], where skip connections force the NN to learn the noise properties. Generally, the statistical properties of speech are much more complex than that of noise. Therefore, learning the noise properties could achieve better performance than learning the speech ones, as demonstrated by [13]. In [13], skip connections were added between the convolutional layers and their corresponding deconvolutional layers, which equivalent to subtract the NN output (i.e., the noise signal) from the noisy input for the denoised speech. In [22], a denoising architecture was constructed with a deep convolutional network topology, where skip connections were added between network inputs and outputs to learn *a posterior* SNR. As a result, the estimated SNR is multiplied with the noisy input to obtain the denoised speech.

Although the effectiveness of adding skip connections to NNs for speech enhancement have been demonstrated, there is still lack of sufficient analyses, quantitatively and/or qualitatively, on the underlying principle. This study aims at investigating the inherent mechanism of the skip connections in learning the noise properties in the CNN-based speech enhancement. We first elaborate that, in NN-based speech enhancement using spectral magnitudes as input features (as in [13]), adding skip connections forces the networks to estimate the residual signal (i.e., the additive noise rather the speech) which is similar to the noise estimation in spectral subtraction; On the other hand, in NN-based speech enhancement using log-spectral magnitudes as network inputs (as in [22]), skip connections enable NN to learn the signal to noise ratio (SNR), which can be equivalent to *a posterior* SNR as computed in Wiener filtering. Then, NNs with

different skipping schemes for speech enhancement are presented. Specifically, two skipping schemes, with the baseline CNN [22] for comparison, are implemented. To evaluate the effectiveness of different systems, a set of speech enhancement experiments are conducted. In particular, statistically different noises are adopted to study the performance of the skipping in CNN in learning the noise properties. Results show that the CNNs with skip connections provide better denoising ability than the baseline for both stationary and nonstationary noises. In particular, benefit from adding more sophisticated skip connections is more significant for dealing with nonstationary noises than for stationary noises, which implies that the complex properties of noise can be learned by CNN with more skip connections.

The rest of this paper is organized as follows. Section 2 describes the fundamentals of NN-based noise reduction. The three systems adopted in this study and their mathematical principles from signal processing point of view. Section 3 presents the experimental setup including data preprocessing, the neural networks training parameters and post processing of the results for performance evaluation. Section 4 gives results and analyses. The final section concludes this work.

2 System Descriptions

2.1 NN-Based Speech Enhancement

Most of the traditional monaural speech enhancements can be regarded as an explicit masking-based noise suppressing. The denoised output is obtained from the noisy input via specific operations, e.g., multiplying the noisy spectra by an SNR-dependent gain (*a posterior* SNR) to suppress the noise components in Wiener filtering. On the other hand, many NNs-based speech enhancements adopt a mapping scheme [11–13]. As demonstrated in [12], such a mapping-based method achieved significant improvement over the explicit ideal ratio masking one on single-channel speech enhancement.

Assuming that there exists a linear/nonlinear mapping function \mathcal{F} which the NNs are trained to learn [23, 24]. In CNN-based speech enhancement, the networks fed with the noisy speech segments are trained to estimate the denoised ones, and the well-trained CNN is generally a kind of nonlinear mapping function. A structure of mapping-based speech enhancement using NNs is illustrated in Fig. 1.

Let y_t , x_t and n_t denote noisy speech, clean speech and the additive noise, respectively, where $t = 1 : T$ and T is the number of time frames, we have the signal model and the mapping relationship as follows

$$y_t = x_t + n_t \quad (1)$$

$$\hat{x}_t = \hat{\mathcal{F}}(y_t) \quad (2)$$

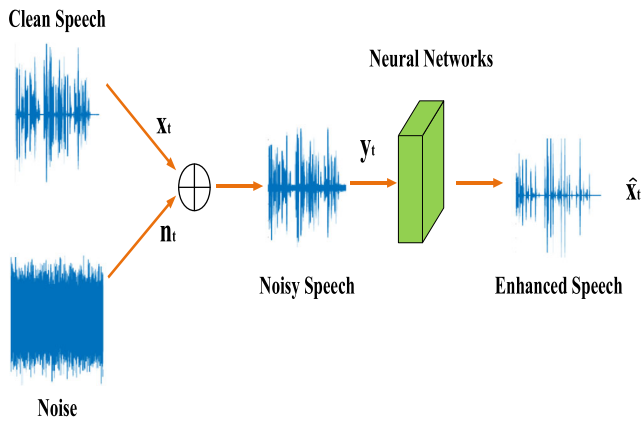


Fig. 1 Diagram of mapping-based speech enhancement framework with NNs.

where $\hat{\mathcal{F}}$ is an approximation of \mathcal{F} learned by the NNs and \hat{x}_t is the estimated speech. Generally, the NNs are trained with the objective function of minimizing the mean squared error (MSE) between x_t and \hat{x}_t , i.e.,

$$\epsilon = \min \sum_{t=1}^T \left\| \hat{x}_t - x_t \right\|_2^2 \tag{3}$$

In this study, the input features to the NNs are the log-spectral magnitudes of speech. In particular, the t^{th} frame of clean speech is estimated from its splicing adjacent 7 frames of the noisy input, i.e.,

$$\hat{X}_{t,f} = \hat{\mathcal{F}}(Y_{t-4 \dots t+2}, f) \tag{4}$$

where subscript $f = 1:F$ and F denotes the total number of frequency representation index. The human auditory system is less sensitive to the noise near the high-energy regions of spectrum (the formant peaks) [25]. Based on the masking effect of the auditory system, a weighting factor $W_{t,f}$ is applied in ℓ_2 norm of $|\hat{X}_{t,f} - X_{t,f}|$ to compute the loss function for CNN, i.e.,

$$W_{t,f} = \frac{A(z)}{A\left(\frac{z}{\gamma}\right)} = \frac{1 - \sum_{i=1}^p a_i z^{-i}}{1 - \sum_{i=1}^p a_i \gamma^i z^{-i}} \tag{5}$$

$$\epsilon = \min \sum_{t=1}^T \sum_{f=1}^F \left\| \hat{X}_{t,f} - X_{t,f} \right\| + \log(|W_{t,f}| + 10^{-10}) \tag{6}$$

In (5), $A(z)$ is the linear prediction (LP) polynomial and a_i are the LP coefficients, γ is the parameter to control the energy of the error near the formant peaks and p is the prediction order. In the experiments, γ and p are set to be 0.9 and 10, respectively.

In general, the curve of $W_{t,f}$ is inversely proportional to the envelope of log-spectral magnitudes $X_{t,f}$. It serves as a penalty factor to reduce the contribution of the error near the formant peaks while placing more emphasis on the spectral valleys.

2.2 CNN with Skipping Architectures

Figure 2 gives different CNN architectures. Figure 2(a) shows a very deep CNN architecture as presented in [26]. The network consists of repetitions of very deep convolutional layers and max-pooling layers. Figure 2(b)–(d) are three CNN-based denoise architectures presented in this study, noted as CNN0, CNN1, and CNN2, respectively. As illustrated, CNN0 is modified from the one in Fig. 2(a) but with significantly reduced complexity, i.e., max-pooling layers are removed and the depth of each convolutional layer is reduced and variable to stack encoding and the corresponding decoding layers (three Encoder-Decoder frameworks). By reducing the network complexity, the new CNN structure is expected to achieve faster convergence (i.e., reducing the processing delays) without significant loss in denoising performance.

Upon CNN0, two skip connection schemes, as proposed in [20, 21], respectively, are adopted to generate CNN1 and CNN2. In CNN1, as shown in Fig. 2(c), a single skip connection from the network input to the output of the last convolutional layer is added to CNN0. By adding the raw input features to the output of the last convolutional layer, the system indirectly forces the CNN to learn the residual error between the noisy speech and the clean speech by back-propagation.

Given the features being the spectral magnitudes, as in [13], this method works in an end-to-end mapping from the noisy features to the clean features, i.e.,

$$X = Y + R \tag{7}$$

where Y, R, X denotes the spectral magnitudes of the network input, the output from the last convolutional layer, and the estimated speech, respectively. This mapping is similar to the spectral subtraction method, i.e.,

$$X = Y - N \tag{8}$$

where N denotes the noise. That is, the CNN tends to learn (the output from the last convolutional layer, R) the noise components (in negative).

On the other hand, if the features are set as log-spectral magnitudes, as adopted in this study, Eq. (7) should be modified as

$$\log(X) = \log(Y) + \log(H) = \log(Y \times H) \tag{9}$$

where $\log(H)$ denotes the output from the last convolutional layer, and H is equivalent to a posterior SNR estimated in Wiener filtering.

In CNN2, multiple skip connections are added upon CNN1. As illustrated in Fig. 2(d), besides the single skip connection as in CNN1, multiple skip connections are added to each Encoder-Decoder (ED), respectively. Given the log-spectral magnitudes being the network features, skip

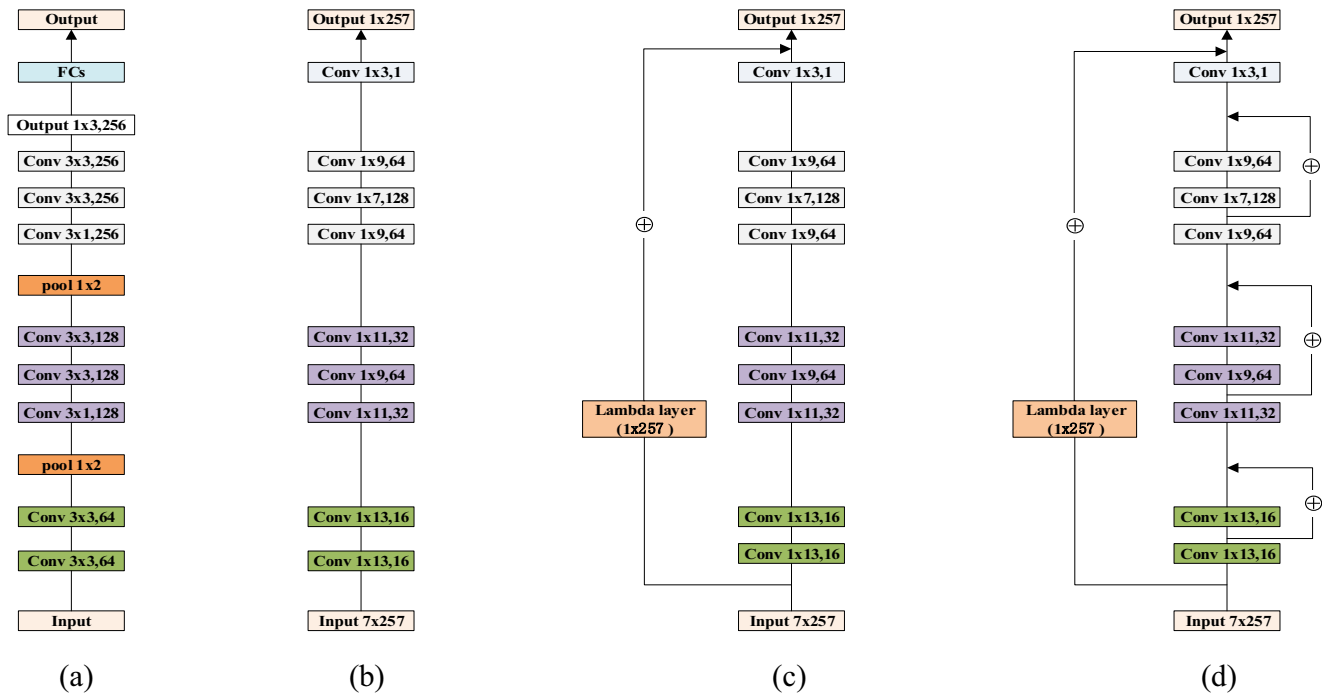


Fig. 2 Diagram of a very deep CNN (a), the proposed basic CNN architecture (b) and CNN with different skip connections (c-d).

connections in CNN2 enable the CNN to estimate *a posterior* SNR in each ED, and the denoised speech in each ED can be obtained by multiplying *the posterior* SNR with the noisy input. This process can be regarded as a Wiener filtering composed of multiple filters.

As shown in Fig. 2(b)-(d), considering the correlation among the splicing adjacent frames, multiple adjacent frames are stacked to feed into the CNNs (CNN0-CNN2) in order to estimate the current target frame. Besides, for single skip connection, a convolutional layer (or a Lambda layer) without activation functions is applied to resize the input frames (seven log-spectral magnitude frames in this study) for the target noisy features which subtract the NN output (i.e., the noise signal) for the denoised speech.

2.3 CNN-Based Speech Enhancement System

Figure 3 gives the diagram of the CNN-based speech enhancement system. A mapping-based CNN model is trained in the training stage with the training data, i.e., pairs of the log-spectral magnitude features extracted from the noisy speech and the corresponding clean speech. The choice of log-spectral magnitude mapping method is motivated by the favorable performance of mapping-based NN model for speech enhancement and dereverberation as suggested in [12, 27].

In the enhancement stage, the trained CNN-based speech enhancement system is fed with the log-spectral magnitude features of testing noisy speech so as to generate the enhanced log-spectral magnitude features. The phase information extracted from testing noisy speech, inverse fast Fourier

transform (IFFT) and overlap-add method are employed to synthesize the enhanced time-domain signal.

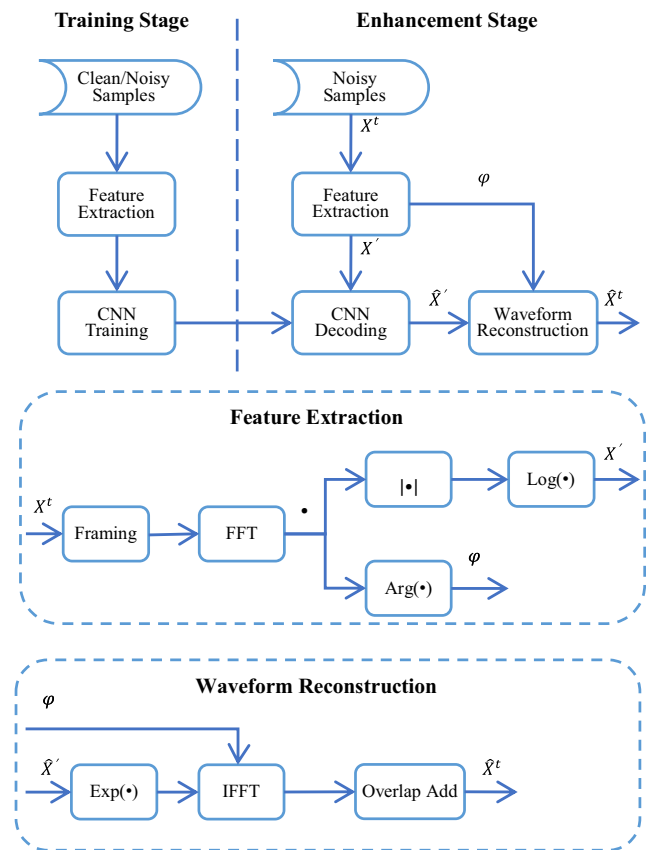


Fig. 3 Diagram of CNN-based speech enhancement system.

3 Experimental Setup

3.1 Data Preprocessing

A set of speech enhancement experiments are carried out to evaluate the performance of different CNN-based speech enhancement systems described in Section 2. In particular, both stationary and nonstationary noises are adopted for evaluation such that the effects of different skip connections could be analyzed comparatively.

Two open access databases, i.e., THCHS-30 [28] and ESC-50 [29], are adopted in the experiments. THCHS-30 is a Mandarin speech database containing a training set (10,000 utterances), a development set (893 utterances) and a test set (2495 utterances). Each utterance contains a Mandarin sentence with sampling rate of 16 kHz. In this experiment, the training data consists of 1000 utterances randomly pick out from the training set, the validation data consists of 100 utterances randomly pick out from the development set, and the test data consists of all utterances from the test set.

ESC-50 consists of 2000 environmental audio recordings. It is adopted to generate the noisy data for the experiments. The recordings were originally sampled at 44.1 kHz and are down-sampled to 16 kHz in this study.

To evaluate the systems performance with stationary vs. nonstationary noises, we compute the stationarity of all the noise recordings in ESC-50 and select 10 stationary noises and 10 nonstationary noises based on their stationarity. To do so, the augmented Dickey–Fuller test (ADF test) [30] is applied to compute the stationarity of the noise data. The critical value [31] and p value [32] tests are then implemented. In specific, the statistic results, critical values and p -values of each noise can be computed through ADF test. According to ADF test principle, those noises with test statistic less than critical values or p -values less than a threshold (a value approaching zero) can be considered as stationary, while those with statistic results greater than critical values or the p -values greater than the threshold are nonstationary. In this study, more stringent test conditions are set to test the stationarity of the noise, i.e., only those noises with test statistic less than critical values as well as p -values less than a threshold (0.05) are labeled as stationary, while those with statistic results greater than critical values and p -values greater than 0.05 are labeled as nonstationary. A Python module (namely `adfuller` in `statsmodels`) [33] is utilized for implementing ADF test. As listed in Table 1, 10 types of stationary noises with lowest p -values and 10 types of nonstationary noises with highest p -values are selected as stationary and nonstationary noise dataset for the experiments.

To generate the noisy data, 8 of 10 noise types are selected randomly from both stationary and nonstationary noises as given in Table 1. That is, there are 16 types of noises serving as the seen noise dataset for training, validating and testing the denoising systems. The rest 4 types of noises (2 stationary and 2

nonstationary, as listed in bold text in Table 1) are served as the unseen noise dataset for testing. As for SNR, seven seen levels (−6 dB to 12 dB with step at 3 dB) and two unseen levels (−5 dB and 5 dB) are adopted to generate noisy speech. As demonstrated in Table 2, there are totally 16,000 noisy utterances (i.e., 1000 clean utterances \times 16 noises \times 1 SNR randomly selected from the seven SNRs) in the training set, 1600 noisy utterances (100 clean utterances \times 16 noises \times 1 SNR randomly selected) in the validating set, and 2495 noisy utterances (2495 clean utterances \times 1 noise randomly selected from the seen noises \times 1 SNR randomly selected) in the test set. To evaluate the generalization of the systems, there are another 2495 utterances (2495 clean utterances \times 1 noise randomly selected from the four unseen noises \times 1 SNR randomly selected from the two unseen SNRs) in the test dataset.

To further investigate the effects of skip connections in the CNN-based denoising systems for stationary and nonstationary noises, another four test sets (Set 1–4 in Table 2), each containing 2495 noisy utterances, are generated. Specifically, each of the sets are generated with seen stationary noises, seen nonstationary noises, unseen stationary noises and unseen nonstationary noises, respectively.

3.2 Feature Extraction

Each time-domain signal is segmented into a sequence of frames by multiplying a Hamming window function (32 ms frame length and 24 ms overlapping). To each frame, 512-point FFT is implemented and the log-spectral magnitudes of the first 257 frequency components are calculated to compose a vector of 257 components. To a specific frame, a 7×257 -dimensional feature matrix is generated as the input to each network system, as described in Section 2.3. Before being fed into the networks, the input features are normalized to the normal distribution.

3.3 Network Setting

Layer weights and bias for the CNN-based speech enhancement systems are initialized as in [34], which will be updated using the back-propagation algorithm [35]. Activation functions used in the experiments are all “ReLU”. All networks are trained for 86 epochs with gradient descent optimization using Adam optimizer with a batch size of 512. The learning rate is initialized to be 0.0015 with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. During the whole training stage, the learning rate is utilized to control the gradient rate of the loss function optimized by the optimizer. To smooth the curve of the loss function defined by Eq. (6), the learning rate decreases in exponential decay rate (initialized to 0.9) per epoch. Besides, batch normalization and ℓ_2 regularization ($\lambda = 10^{-4}$) are applied to each fully connected and convolutional layer in order to accelerate the training and to avoid overfitting. In the experiments, only those models with the best evaluation results for the validating dataset are used in the test stage.

Table 1 The information of selected 20 noise types and the corresponding p-values.

stationarity	type	p-values	stationarity	type	p-values
Nonstationary	Insects (flying)_1	0.747	Stationary	Insects (flying)_4	1.914e-23
	Crickets_1	0.654		Water drops	3.611e-19
	Engine	0.653		Chirping birds_2	1.288e-18
	Crickets_2	0.646		Clapping_1	3.358e-16
	Insects (flying)_2	0.612		Clapping_2	4.474e-16
	Crackling fire	0.562		Clapping_3	3.916e-14
	Chirping birds_1	0.548		Chirping birds_3	4.017e-14
	Insects (flying)_3	0.546		Insects (flying)_5	9.309e-14
	Clock tick	0.526		Clock alarm	9.963e-13
	Brushing teeth	0.523		Sea waves	2.698e-12

3.4 Evaluating Metrics

Three objective metrics, i.e., the perceptual evaluation of speech quality (PESQ) [36], the short-time objective intelligibility (STOI) [37], and the logarithmic spectral distance (LSD) [38], are adopted for performance evaluation.

PESQ, recommended by ITU-T as Standard P.862, is an objective speech quality assessment. To compute PESQ, a linear combination of disturbance parameters is used to predict subjective mean opinion score (MOS). Generally, the partial compensation in PESQ is implemented to avoid the need for using a large number of parameters to predict quality. Two parameters, one symmetric disturbance (d_{SYM}) and asymmetric disturbance (d_{ASYM}), are combined to predict the speech quality. The predictor processing in PESQ is as follows

$$PESQ = 4.5 - 0.1d_{SYM} - 0.0309d_{ASYM} \tag{10}$$

where d_{SYM} and d_{ASYM} are calculated by the disturbance processing model in PESQ [36]. In general, the higher PESQ score, the better perceptual speech quality. For normal

Table 2 Details about generating the training, validating and test data in the experiments.

	Train	Val	Test					
			Seen	Unseen	Set1	Set2	Set3	Set4
Clean speech	1000	100	2495	2495	2495	2495	2495	2495
Noise	16	16	1(16)	1(4)	1(8)	1(2)	1(8)	1(2)
SNR levels	1(7)	1(7)	1(7)	1(2)	1(7)	1(2)	1(7)	1(2)
Noisy speech	16,000	1600	2495	2495	2495	2495	2495	2495

the contents 1(n) means that only 1 type of data selected from the total n types. Set1: seen stationary noise and seen SNR; Set2: unseen stationary noise and unseen SNR; Set3: seen nonstationary noise and seen SNR; Set4: unseen nonstationary noise and unseen SNR. In specific, the information of both stationary and nonstationary noises used in Set 1–4 are listed in Table 1

subjective test material, the values lie between 1.0 (bad) and 4.5 (no distortion). PESQ may fall below 1 in extremely high distortion conditions, but this is very uncommon.

STOI is a short-time objective intelligibility measure. As in [37], the short-time (386 ms) temporal envelopes of the clean speech and the estimated speech are utilized to calculate STOI scores, i.e.,

$$STOI = \frac{1}{J, M} \sum_{j, m} \frac{(x_{j, m} - \mu_x)^T (x_{j, m} - \mu_x)}{\|x_{j, m} - \mu_x\| \|x_{j, m} - \mu_x\|} \tag{11}$$

where x_j, m and $x_{j, m}$ denote the short-time temporal envelope of the clean speech and the estimated speech, where j and m is the time and frequency indexes; μ denotes the sample average of the corresponding vector (x_j or \bar{x}_j); J and M are the total number of frames and frequency bins. In general, STOI varies from 0 (worst) to 1 (no distortion) and higher STOI illustrates better perceptual intelligibility for speech signal.

LSD measures the logarithmic spectral distance (averaged over all frames) between two speech samples (usually, a clean one and its distorted version), i.e.,

$$LSD(x, \hat{x}) = \frac{1}{T} \sum_{t=1}^T \sqrt{\frac{1}{F} \sum_{f=1}^F (x_{t, f} - \hat{x}_{t, f})^2} \tag{12}$$

Table 3 Mean scores for PESQ, LSD and STOI obtained from different speech enhancement systems.

	PESQ		LSD		STOI	
	seen	unseen	seen	unseen	seen	unseen
Noisy	2.32	2.24	1.67	1.86	0.86	0.85
Wiener	2.50	2.48	1.57	1.57	0.85	0.84
CNN0	2.71	2.47	0.99	1.16	0.86	0.85
CNN1	3.04	2.71	0.93	1.13	0.90	0.86
CNN2	3.22	2.76	0.87	1.10	0.92	0.87

Table 4 Noise type dependent systems performance (Z-score) and performance gaps between SN and NSN with seen noise.

	Noisy	CNN0		CNN1		CNN2		
		Z-s	Z-s	Gap	Z-s	Gap	Z-s	Gap
PESQ	SN	3.89	5.37	+	5.60	+	5.95	-
	NSN	3.95	5.29	0.08	5.57	0.03	5.96	0.01
LSD	SN	5.16	3.69	-	3.43	-	3.35	-
	NSN	4.85	3.81	0.12	3.50	0.07	3.38	0.03
STOI	SN	4.58	5.34	+	5.68	+	5.83	+
	NSN	4.82	5.28	0.06	5.66	0.02	5.83	0

Table 5 Noise type dependent systems performance (Z-score) and performance gaps between SN and NSN with unseen noise.

	Noisy	CNN0		CNN1		CNN2		
		Z-s	Z-s	Gap	Z-s	Gap	Z-s	Gap
PESQ	SN	3.85	4.95	+	5.15	+	5.32	+
	NSN	3.89	4.90	0.05	5.13	0.02	5.31	0.01
LSD	SN	5.12	4.02	-	3.84	-	3.67	+
	NSN	4.95	4.11	0.09	3.85	0.01	3.65	0.02
STOI	SN	4.36	5.02	+	5.17	+	5.35	-
	NSN	4.45	4.96	0.06	5.15	0.02	5.36	0.01

where x, \hat{x} denote the target speech and the enhanced speech, respectively. T and F are the number of time frames and the frequency bins, respectively. Lower LSD tells less signal distortions. $LSD = 0$ refers to clean speech.

3.5 Post-Processing of the Results

It is a general knowledge that dealing with nonstationary noise is much more difficult than dealing with stationary noise for a speech enhancement system. A score normalization method, i.e., the Z-score [39], is adopted for post-processing the raw results, such that the ability of different skipping schemes on learning the noise properties under different noise conditions can be compared fairly. Z-score is a common method used for

data preprocessing and have been adopted in many NNs-based classification systems [40, 41]. To each of the three objective metrics abovementioned, Z-scores are computed from the corresponding raw scores as follows. Firstly, compute a cumulative percentile corresponding to every element of raw scores; Then, according to a normative table of the normal distribution with zero mean and standard deviation of 1, the percentiles computed in previous step is converted to their corresponding Z-scores. For convenience, a fixed bias of 5 was added to the Z-scores in this study.

4 Results and Discussion

4.1 Overall Performances

As described in Section 2.2, the proposed CNN-with-skipping denoising system can be regarded as a mapping-based version of the Wiener filtering. Therefore, although the main objective of this study is to investigate the contributions of adding different skipping in CNN for speech enhancement, it is worthwhile to include Wiener filtering for comparison in and only in this subsection. Specifically, the parametric Wiener filtering [42] where parameters α and β both set as 1 is adopted in the experiment.

Table 3 gives mean scores of PESQ, LSD and STOI for noisy speech (Noisy) and speech outputs from Wiener filtering (Wiener), CNN without skipping (CNN0), and CNN with two different skipping (CNN1, CNN2). The ‘seen’ and ‘unseen’ stand for the scores obtained from test data with seen and unseen noises as listed in ‘Seen’ and ‘Unseen’ test set of Table 2. For each seen/unseen condition, overall performance, i.e., mean scores over all noise types (both nonstationary and stationary listed in Table 1) are computed. As illustrated, CNN-based systems achieve better performances than Wiener filtering for all metrics and in all conditions, except that CNN0 is comparable to Wiener filtering for PESQ in unseen data and STOI in both seen and unseen data. It is obvious that adding skip connections to CNN can significantly improve the denoising performance and best results can be obtained by CNN2.

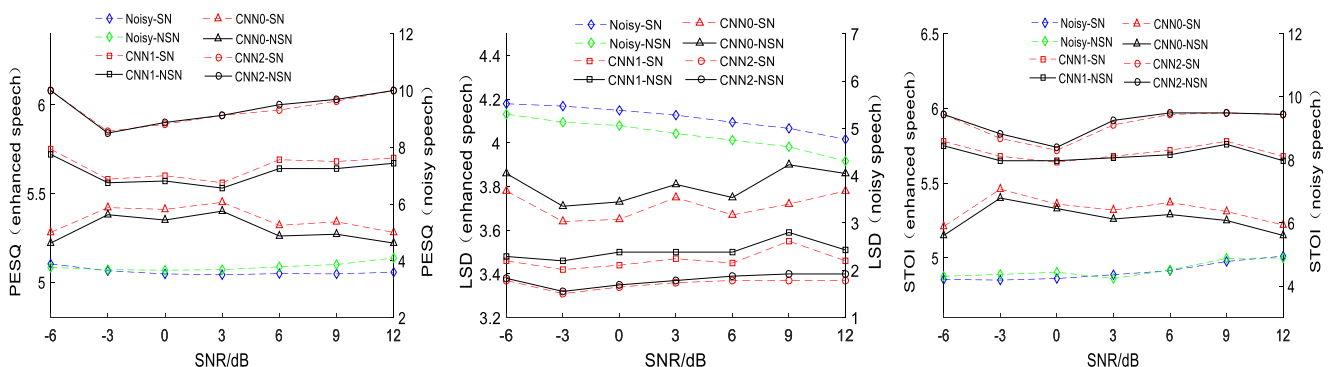


Fig. 4 SNR and noise type dependent performances for different denoising systems under seen noise conditions.

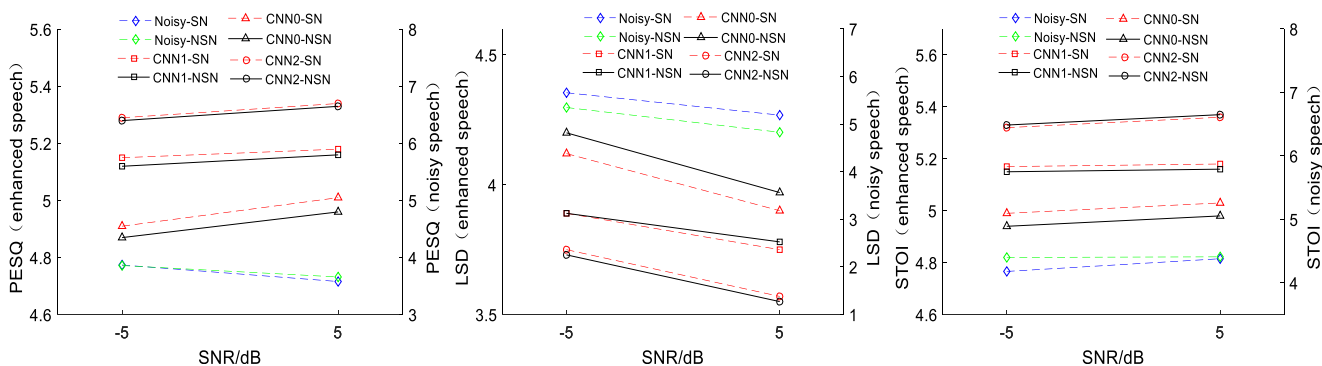


Fig. 5 SNR and noise type dependent performances for different denoising systems under unseen noise conditions.

4.2 Noise-Dependent (Stationary Vs. Nonstationary) Performances

To investigate how the skip connections can help CNN in learning the noise properties, noise type dependent, i.e., stationary noise (SN) vs. nonstationary noise (NSN) performances for the three CNN-based systems are given in Table 4 for seen noises (i.e., Set 1 & Set 3 in Table 2) and in Table 5 for unseen noises (Set 2 & Set 4 in Table 2). As described in Section 3.4, Z-scores (Z-s) are computed from the raw scores for fair comparison. To each system, the performance gap between SN and NSN is also given in the tables.

As illustrated in Table 4, under seen noise conditions, performances increase as the number of skip connections increases (i.e., from CNN0 to CNN1 and to CNN2) for all metrics in both SN and NSN. Meanwhile, a gradually narrowing gap can be achieved for all metrics with increasing skip connections. In general, it is well recognized that tackling nonstationary noise is a relatively harder task than dealing with stationary noise in speech enhancement. Nevertheless, the diminishment in the performance gap tells that adding more skip connections offers more significant improvement for NSN than that for SN. That is, benefit by adding more sophisticated skip connections is more significant for dealing with nonstationary noises than that for stationary noises, which implies that the complex properties of noise can be learned by CNN with more skip connections.

Under unseen noise conditions, as illustrated in Table 5, similar performances as that under seen noise conditions can be obtained. In particular, better performance can be achieved in NSN than in SN for LSD and STOI with CNN2. The results tell that the gain brought by skip connections for nonstationary noises does not degrade at the unseen noises.

To further elaborate the performances, SNR- and noise-dependent performances of different systems are given in Fig. 4 (seen data) and Fig. 5 (unseen data). Similar performances to that demonstrated in Tables 4 and 5 can be observed from the two figures. In each specific SNR condition and noise type, improved performances can be obtained when more sophisticated skip connections are added to the CNNs.

Furthermore, for CNN0, there are significant performance gaps between SN and NSN at each specific SNR conditions. For CNN2, however, performances on NSN approach to, or even exceed in some particular cases, that on SN.

Table 6 illustrates how the skip connections can help CNN to learn the properties of noises with different complexity (in the sense of non-stationarity) from Table 1. As shown, the overall performances for noises with descending *p*-values (i.e., increasing stationarity) among three CNN-based systems are demonstrated. All three systems perform better when processing more stationary noises. In specific, the gaps between two adjacent columns are reducing with more skip connections in CNN, which tells that more skip connections enable the network to learn more complex properties of noise with increasing non-stationarity. From signal processing point of view, as elaborated in Section 2.2, the skip connections enable the corresponding ED in the basic CNN to estimate the posterior SNR, which is then multiplied to the noisy input of each ED to obtain the enhanced features. More skip connections added to different EDs can be regarded as the cascading of multiple filters, with which better performance could be achieved.

Table 6 Mean Z-scores for PESQ, LSD and STOI obtained from different systems dealing with noise with descending *p*-values (increasing stationarity).

metrics	systems	p-values			
		0.747	0.526	2.698e-12	1.914e-23
PESQ	CNN0	5.21	5.30	5.35	5.45
	CNN1	5.54	5.58	5.58	5.65
	CNN2	5.95	5.97	5.94	5.97
LSD	CNN0	3.88	3.80	3.71	3.64
	CNN1	3.53	3.48	3.44	3.41
	CNN2	3.41	3.36	3.36	3.34
STOI	CNN0	5.25	5.29	5.33	5.38
	CNN1	5.66	5.68	5.67	5.70
	CNN2	5.82	5.84	5.82	5.85

5 Conclusions

In this study, CNN-based denoising architectures with different skip connections were proposed and investigated for speech enhancements. A set of experiments were carried out to evaluate the denoising performances of the proposed architectures and, in particular, to investigate the inherent mechanism of the skip connections in learning the noise properties. Results showed that the proposed systems significantly outperformed a parametric Wiener filtering denoising algorithm in both ‘seen’ and ‘unseen’ test data conditions. Furthermore, results showed that adding skip connections to CNNs can significantly improve the denoising performance. In particular, for nonstationary noises with statistically more complicated properties, adding more skip connections to the CNNs might effectively reduce the performance gap between stationary noises and nonstationary noises.

Acknowledgements This work is jointly supported by Guangdong Key R&D Project (Grant No. 2018B030338001), NSF of China (Grant No. 61771320) and Shenzhen Science & Innovation Funds (Grant No. JCYJ 20170302145906843).

References

- Wang, D. L., & Chen, J. T. (2018). Supervised speech separation based on deep learning: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 26(10), 1702–1726.
- Virtanen, T., Singh, R., & Raj, B. (2012). Techniques for noise robustness in automatic speech recognition. John Wiley & Sons.
- Bolner, F., Goehring, T., Monaghan, J., Dijk, B. V., Wouters, J., & Bleeck, S. (2016). Speech enhancement based on neural networks applied to cochlear implant coding strategies. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6520–6524).
- Loizou, P. C. (2013). *Speech enhancement: Theory and practice*. CRC Press.
- Wang, Q., Du, J., Dai, L. R., & Lee, C. H. (2018). A multiobjective learning and ensembling approach to high performance speech enhancement with compact neural network architectures. *IEEE Transactions on Audio, Speech, and Language Processing*, 26(7), 1185–1197.
- Dahl, G., Yu, D., Deng, L., & Acero, A. (2011). Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4688–4691).
- Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1), 30–42.
- Abdel-Hamid, O., Mohamed, A.-R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(10), 1533–1545.
- Seltzer, M. L., Yu, D., & Wang, Y. (2013). An investigation of deep neural networks for noise robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 7398–7402).
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257–286.
- Fu, S.-W., Tsao, Y., & Lu, X. (2016). SNR-aware convolutional neural network modeling for speech enhancement. In *Seventeenth Proceedings of the Annual Conference of the International Speech Communication Association, San Francisco, CA, USA*, (pp. 8–12).
- Kounovsky, T., & Malek, J. (2017). Single channel speech enhancement using convolutional neural network. *Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*.
- Park, S. R., & Lee, J. W. (2017). A fully convolutional neural network for speech enhancement. In *the Eighteenth Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12), 3371–3408.
- Osako, K., Singh, R., & Raj, B. (2015). Complex recurrent neural networks for denoising speech signals. In *Applications of Signal Processing to Audio and Acoustics*.
- Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer (pp. 234–241).
- Pascual, S., Bonafonte, A., & Serra, J. (2017). SEGAN: speech enhancement generative adversarial network. In *the Eighteenth Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- Michelsanti, D., & Tan, Z.-H. (2017). Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification. In *the Eighteenth Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.
- Mao, X. J., Shen, C. H., & Yang, Y. B. (2016). Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. *arXiv preprint arXiv:1603.09056*.
- Shi, Y. P., Rong, W. C., & Zheng, N. N. (2018). Speech enhancement using convolutional neural network with skip connections. In *the 11th international symposium on Chinese spoken language processing (ISCSLP)*.
- Lim, J. S., & Oppenheim, A. V. (1978). All-pole modeling of degraded speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(3), 197–210.
- Ram, R., & Mohanty, M. N. (2018). The use of deep learning in speech enhancement. In *Proceedings of the First International Conference on Information Technology and Knowledge Management*, 14, 107–111.
- Hu, Y., & Loizou, P. C. (2003). A perceptually motivated approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, 11(5), 457–465.
- Qian, Y., Bi, M., Tan, T., & Yu, K. (2016). Very deep convolutional neural networks for noise robust speech recognition. *IEEE Transactions, Audio, Speech, and Language Processing*, 24(12), 2263–2276.
- Zhao, Y., Wang, Z.-Q., & Wang, D. L. (2019). Two-stage deep learning for noisy-reverberant speech enhancement. *IEEE Transactions on Audio, Speech, Language Processing*, 27(1), 53–62.
- Wang, D., & Zhang, X. W. (2015). THCHS-30: A free Chinese speech corpus. *arXiv preprint arXiv: 1512.01882v2*.
- Piczak, K. J. (2015). ESC: Dataset for environmental sound classification. In *Proceedings of the ACM International Conference on Multimedia* (pp. 1015–1018).

30. Said, S. E., & Dickey, D. A. (1984). Testing for unit roots in autoregressive moving average models of unknown order. *Biometrika*, 71(3), 599–607.
31. MacKinnon, J. G. (2010). Critical values for Cointegration tests. *Queen's Economics Department Working Paper*, 1227.
32. MacKinnon, J. G. (1994). Approximate asymptotic distribution functions for unit-root and cointegration tests. *Journal of Business and Economic Statistics*, 12, 167–176.
33. Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*.
34. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *the thirteenth International Conference on Artificial Intelligence and Statistics*, 9, 249–256.
35. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.
36. Rix, A. W., Beerends, J. G., Hollier, M. P., & Hekstra, A. P. (2001). Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 2, 749–752.
37. Taal, C. H., Hendriks, R. C., Heusdens, R., & Jensen, J. (2010). A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4214–4217).
38. Gray, A., & Markel, J. (1976). Distance measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(5), 380–391.
39. Hegde, V., & Pallavi, M. S. (2015). Descriptive analytical approach to analyze the student performance by comparative study using Z score factor through R language. In *IEEE International Conference on Computational Intelligence and Computing Research (ICIC)* (pp. 10–12).
40. Abdullah, N., Rashid, N. E. A., Khan Z. I., & Musirin, I. (2015). Analysis of different Z-score data to the neural network for automatic FSR vehicle classification. In *the Third IET International Radar Conference*.
41. Chittineni, S., & Bhogapathi, R. B. (2012). A study on the behavior of a neural network for grouping the data. In *the International Journal of Computer Science Issues (IJCSI)*, 9(1), 228–234.
42. Lim, J. S., & Oppenheim, A. V. (1979). Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, 67(12), 1586–1604.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Nengheng Zheng received the B.S. degree in electronic engineering and the M.S. degree in acoustics from Nanjing University, Nanjing, China, in 1997 and 2002, respectively, and the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong, Hong Kong SAR of China, in 2006. He is currently an Associate Professor at the College of Electronics and Information Engineering, Shenzhen University, China. From 2014 to

2015, he was a visiting scholar at the School of Electrical Engineering and

Telecommunications, University of New South Wales, Australia. His research focuses on speech and audio signal processing for human and machine perceptions.



Yupeng Shi received the B.S. degree in communication engineering from Shenzhen University, Shenzhen, China, in 2015. He now is a postgraduate at the College of Electronics and Information Engineering in Shenzhen University, China. His current research interests include speech enhancement/dereverberation based on deep learning and signal processing algorithms for cochlear implants applications.



Weicong Rong received the B.S. degree in communication engineering from Dongguan University of Technology, China, in 2016. He now is a M.S. student at the College of Electronics and Information Engineering, Shenzhen University, China. His research focuses on signal processing and automatic speech recognition based on deep learning.



Yuyong Kang received the B.S. degree in electrical information engineering from Shenyang Aerospace University, China, in 2018. He now is a M.S. student at the College of Electronics and Information Engineering in Shenzhen University, China. His research focuses on speech enhancement and dereverberation for cochlear implants using deep learning.