Convolutional Recurrent Neural Network Based on Short-Time Discrete Cosine Transform for Monaural Speech Enhancement

Jinzuo Guo[1(✉)], Yi Zhou[1], Hongqing Liu[1], and Yongbao Ma[2]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
s200131221@stu.cqupt.edu.cn
[2] Suresense Technology, Chongqing 400065, China

Abstract. Speech enhancement algorithms based on deep learning have greatly improved speech's perceptual quality and intelligibility. Complex-valued neural networks, such as deep complex convolution recurrent network (DCCRN), make full use of audio signal phase information and achieve superior performance, but complex-valued operations increase the computational complexity. Inspired by the deep cosine transform convolutional recurrent network (DCTCRN) model, in this paper real-valued discrete cosine transform is used instead of complex-valued Fourier transform. Besides, the ideal cosine mask is employed as the training target, and the real-valued convolutional recurrent network (CRNN) is used to enhance the speech while reducing algorithm complexity. Meanwhile, the frequency-time-LSTM (F-T-LSTM) module is used for better temporal modeling and the convolutional skip connections module is introduced between the encoders and the decoders to integrate the information between features. Moreover, the improved scale-invariant source-to-noise ratio (SI-SNR) is taken as the loss function which enables the model to focus more on the part of signal variation and thus obtain better noise suppression performance. With only 1.31M parameters, the proposed method can achieve noise suppression performance that exceeds DCCRN and DCTCRN.

Keywords: Speech enhancement · Deep learning · Convolutional recurrent neural network · Discrete cosine transform

1 Introduction

Speech enhancement refers to the extraction of the purest possible target speech from noisy speech. It belongs to an important branch of audio front-end processing. Traditional single-channel speech enhancement algorithms include spectral-subtractive algorithms [1], minimum mean square error estimation [2], and wiener filtering [3]. These algorithms have fast calculations and require low-performance

F. Gao et al. (Eds.): ChinaCom 2022, LNICST 500, pp. 163–173, 2023.
https://doi.org/10.1007/978-3-031-34790-0_13

hardware, but their robustness is poor, and they usually can not handle non-stationary noise well. Deep learning-based methods treat speech enhancement as a supervised learning problem and use neural networks' powerful nonlinear fitting ability to remove non-stationary noise in complex acoustic environments. These methods can be divided into two main categories: mapping and mask-based methods. The mapping methods learn the mapping relationship between the noisy speech and the clean speech by training a neural network model. The mask-based methods estimate a mask to classify noise and clean speech signals and then obtain the enhanced speech signal by weighting it with the noisy signal. Common masks include the ideal binary mask (IBM) [4], the ideal ratio mask (IRM) [5], the phase sensitive mask (PSM) [6] and the complex ratio mask (CRM) [7], which show better performance than direct spectral mapping.

The end-to-end model is a typical time-domain method by inputting the original speech signal and directly outputting the final enhanced speech signal, for example, Conv-tasnet [8], which belongs to the encoder-decoder framework. It extracts features from the speech waveform by a 1-D convolutional neural network (Conv1d) in the encoding stage and then passes them through the temporal convolutional network (TCN) as the enhancement module. Finally, the speech is reconstructed by a 1-D convolutional transpose neural network (ConvTranspose1d) in the decoding stage. Although the performance is excellent, a large number of Conv1d layers used to obtain a suitable receptive field lead to large latency and computational complexity, which limits its application in the low-latency domain.

Another popular method is to convert the speech signal to the time-frequency domain by short-time fast Fourier transform (STFT) and then estimate the amplitude spectrum of the original signal from the amplitude spectrum of the noisy signal, and finally combine it with the phase information of the noisy signal to obtain the enhanced speech signal [9]. It focuses on signal amplitude and neglects the phase information which greatly limits the performance of the model. The deep complex convolution recurrent network DCCRN [10] model combines the advantages of deep complex u-net (DCUNET) [11] and convolutional recurrent network (CRNN) [12] using the complex-valued network to estimate CRM and reconstruct the magnitude and phase of speech by simultaneously augmenting the real and imaginary components of the spectrogram of the speech signal. It has shown excellent performance in the 2020 Deep Noise Suppression (DNS) challenge. However, complex-valued networks increase computational complexity. To address this issue, the recently proposed deep cosine transform convolutional recurrent network DCTCRN [13] model, which uses discrete cosine variation as input, is trained using a real-valued network, which reduces the number of parameters and improves the performance compared with DCCRN.

In this paper, we make full use of the advantages of DCCRN and DCTCRN. Firstly, discrete cosine transforms are used in preparing input features. Secondly, the F-T-LSTM [14] is used for temporal modeling, and the convolutional module is employed to integrate inter-feature information in the skip connections part [15]. Lastly, we modify the scale-invariant source-to-noise ratio (SI-SNR) as a loss function. Experimental results show that the proposed model with fewer

parameters outperforms those of both DCCRN and DCTCRN in terms of objective metrics.

2 Proposed Model

2.1 System Architecture

CRNN is a single-channel real-time speech enhancement model proposed by Ke Tan in 2018 [12], which is an encoder-decoder architecture with a two-layer LSTM as the processor. We have made several revisions to the CRNN model: 1. Introduce the F-T-LSTM module to temporal modeling [14]. 2. Add the convolutional module in the skip connections section between the encoders and the decoders [15]. 3. Optimize SI-SNR as loss function. The structural diagram of the proposed model is shown in Fig. 1.

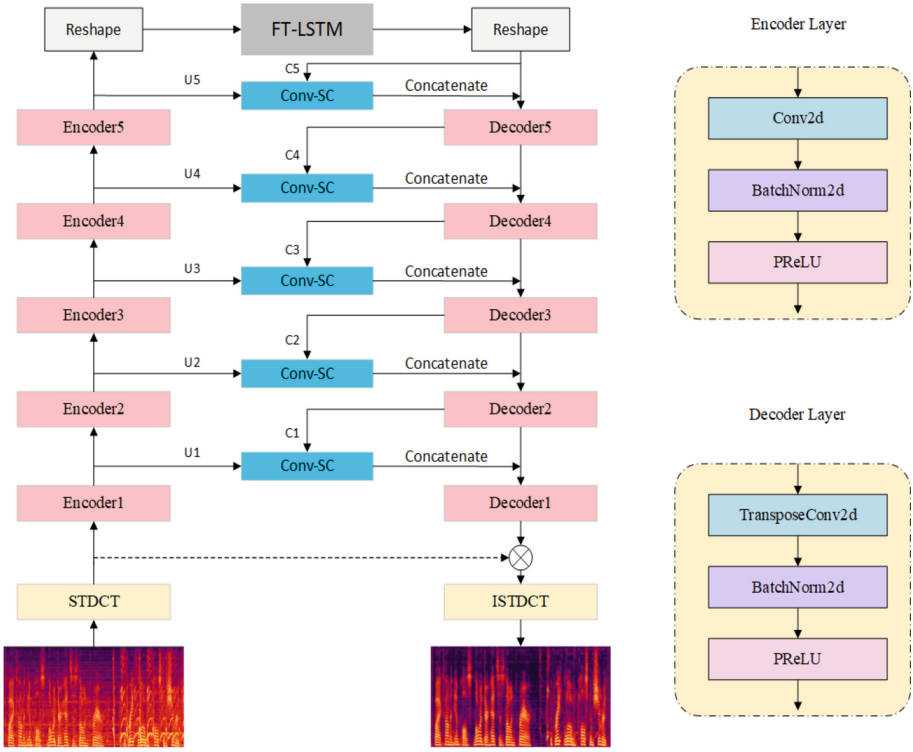

Fig. 1. The proposed network structure

2.2 F-T-LSTM Block

The frequency-time-LSTM (F-T-LSTM) module first scans the frequency bands to generate a summary of the spectral information, then uses the output layer activations as the input to a traditional time-LSTM (T-LSTM) for time scale summarization. It can be described as follows:

$$\begin{aligned} \text{F-LSTM: } & \begin{cases} U_f = \text{BLSTM}(E[i,:,:]), i = 1 \cdots F \\ O_f = U_f + E \end{cases} \\ \text{T-LSTM: } & \begin{cases} U_t = \text{LSTM}(O_f[:,i,:]), i = 1 \cdots T \\ O_t = U_t + O_f \end{cases} \end{aligned}, \tag{1}$$

where $E \in R^{F \times T \times C}$ denotes the output of the encoder. Send E to BLSTM to obtain U_f, then add a residual connection to obtain O_f (the output of the frequency-LSTM) as the input of the T-LSTM to do time scale analysis. T-LSTM also adds the residual connection, O_t denotes the output of the T-LSTM module, which is subsequently fed into the decoder.

Compared with the traditional LSTM, the F-T-LSTM module can achieve better noise reduction by scanning and aggregating the correlation information among the frequency points, in addition to the long time memory of the timing information.

2.3 Convolutional Skip Connections Block

CRNN introduces a skip connections module between encoder-decoder to avoid gradient vanishment. The typical approach is to concatenate the encoder output and the last decoder output as the next decoder layer input [12], but this may not be a favorable approach. Therefore, the convolutional skip connections module is introduced between the encoders and the decoders, which uses Conv2d blocks to extract correlation information between features to speed up the gradient flow, the structure is shown in Fig. 2.

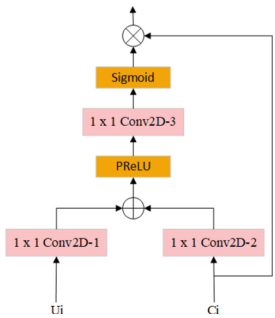

Fig. 2. Convolutional skip connections block

U_i is the output of the encoder layers and C_i is the output of F-T-LSTM or the decoder layers. There are two 1×1 Conv2D layers with output channels being twice that of the input channels, mapping U_i and C_i to a high-dimensional space for information integration, with corresponding weights W_C and W_U, respectively. The high-dimensional space feature layer output can be described as:

$$A_i = PReLU\left(W_U \otimes U_i + W_C \otimes C_i\right) \tag{2}$$

where U_i and C_i represent the ith layer of the encoder and the decoder, respectively. $PReLU$ is Parametric Rectified Linear Unit (PReLU, the range is -inf to inf). The output of the convolutional skip connections block is

$$B_i = \sigma\left(W_f \otimes A_i\right) \cdot C_i \tag{3}$$

where W_f represents a 1×1 Conv2D layer with output channels being half of the input channels. σ is the sigmoid function, and the range is 0 to 1.

2.4 Input Feature

DCT is a transform related to DFT but uses only real numbers [13]. In addition to the general orthogonal transform properties, the basis vectors of the transform array of DCT can well describe the correlation characteristics of speech and image signals. Therefore, DCT is considered as a quasi-optimal transform in transforming speech signals and image signals. DCT is defined as:

$$F(u) = c(u) \sum_{n=0}^{N-1} f(n) \cos\left[\frac{\pi u(2n+1)}{2\,N}\right], u = 0, 1, \ldots, N-1, \tag{4}$$

and the inverse DCT is defined as:

$$f(n) = \sum_{n=0}^{N-1} c(u) F(u) \cos\left[\frac{\pi u(2n+1)}{2\,N}\right], n = 0, 1, \ldots, N-1 \tag{5}$$

where $c(u)$ is a compensation factor that allows the DCT transformation matrix to be orthogonal.

$$c(u) = \begin{cases} \sqrt{\dfrac{1}{N}}, \; u = 0, \\ \sqrt{\dfrac{2}{N}}, \; u = 1, 2, \ldots, N-1. \end{cases} \tag{6}$$

where $f(n)$ is the original signal. $F(u)$ is the DCT-transformed coefficient, and N is the number of points of the original signal.

2.5 Training Target

The training target is an ideal cosine mask (ICM) optimized by signal approximation (SA). The ICM can be defined as:

$$ICM_{t,f} = \frac{S_{t,f}}{Y_{t,f}} \tag{7}$$

where $S_{t,f}$ and $Y_{t,f}$ denote the DCT coefficients of the clean speech and the noisy speech in a particular T-F unit respectively.

2.6 Loss Fuction

The loss function is based on SI-SNR, which is an important metric of speech quality and defined as:

$$\begin{cases} s_{\text{target}} = \frac{<\widehat{s}, s> \cdot s}{\|s\|_2^2}, \\ e_{\text{noise}} = \widehat{s} - s, \\ SI - SNR = 10 * \log_{10} \left(\frac{\|s_{\text{target}}\|_2^2}{\|e_{\text{noise}}\|_2^2} \right). \end{cases} \tag{8}$$

where s and $\widehat{s}$ are the clean and estimated time-domain speech data, respectively. $< ., . >$ denotes the dot product between two vectors, and $\|.\|_2$ is the Euclidean norm (L2 norm).

Improved SI-SNR uses the noisy signal and clean signal to calculate the value of SI-SNR and then uses the enhanced signal and clean signal to calculate the value of SI-SNR. The final result is the subtraction of the above two values. The advantage of the improved SI-SNR is it enables the model to focus more on the part of the signal variation, and our experiments prove that the improved SI-SNR works better than the SI-SNR in noise suppression tasks. The improved SI-SNR is defined as:

$$\text{SI} - \text{SNR}_{\text{i}} = \text{SI} - \text{SNR}_{(S,\widehat{S})} - \text{SI} - \text{SNR}_{(S,Y)} \tag{9}$$

where $\text{SI} - \text{SNR}_{(S,\widehat{S})}$ and $\text{SI} - \text{SNR}_{(S,Y)}$ represent the SI-SNR score of the enhanced and clean signal and the SI-SNR score of the noisy and clean signal, respectively.

3 Experiment

3.1 Datasets

In our experiments, we evaluate the proposed models on two datasets.

3.1.1 Dataset 1 (DNS 2020): The first dataset is generated based on the Interspeech 2020 DNS Challenge dataset [16], all the waveforms are sampled at 16kHz. The DNS Challenge clean speech dataset was derived from the public audiobook dataset Librivox1. It has a recording of volunteers reading over 10,000 public domain audiobooks in different languages, most of which are in English. It contains over 500 h of speech from 2150 speakers. And the noise dataset consists of a 180-hour noise set which includes 150 classes and 65,000 noise clips, which were selected from Audioset2 and Freesound3. We randomly select speech clips and noise clips to create a 500-hour noisy training set, with a signal-to-noise ratio being set at -10db to 20db. Each selected audio clip is set to 10 s. We estimated the proposed model with the DNS-2020 synthetic no reverb test set.

3.1.2 Dataset 2 (Noisy Speech Database): The clean data in the second dataset are from [17], which is widely used in speech enhancement research. This clean set is obtained from sentence recordings of various text passages and 30 English speakers were selected from the Voice Bank corpus [18], including males and females with various accents. 28 and 2 speakers were assigned to the training and test sets, respectively. The noise data are obtained from NoiseX-92 [19], which contains 15 types of noise such as White noise, Pink noise, HF channel noise, Speech babble, factory floor noise, etc. We use the above clean speech and noise to synthesize a 50 h training set with a signal-to-noise ratio (SNR) of 0 db-20 db, 40% of the data set without reverberation, and 60% of the data set with reverberation (T60 from 0.3 s to 1.3 s). Room impulse response (RIR) is randomly-selected from the DNS RIR dataset. To verify the noise suppression performance of the model under different SNR and reverberation or non-reverberation. We generate two test sets: reverberant and non-reverberant test sets, and the SNR are set to (0 db,5 db,10 db,15 db,20 db).

3.2 Training Setup and Baselines

The baseline structure is shown in Fig. 1, with the difference that instead of introducing the convolutional module in the skip connections section between the encoders and the decoders, simple stacking is used. The details of the setup are as follows: In the DCT transform, the window function is the periodic Hanning window, the window length and frameshift are 32 ms and 8 ms, and the DCT length is 512 points. The optimizer is Adam gradient, with an initialized learning rate of 1e–3, and it will decay 0.5 when the validation loss goes up. The model is selected by early stopping. The loss function is SI-SNR or Improved SI-SNR. We compare the proposed model with DTLN, DCCRN-E and DCTCRN, and their detail settings are as follows.

- DTLN: The window length and hop sizes are 32 ms and 8 ms, and the FFT length is 512. The number of each LSTM nodes is set to 128. During training, 25% of dropout is applied between the LSTM layers. The 1D-Conv Layer to create the learned feature representation has 256 fifilters.

- DCCRN-E: the window length and hop sizes are 25 ms and 6.25 ms, and the FFT length is 512. The number of channels for the DCCRN-E is {32, 64, 128, 128, 256, 256}. The kernel size and stride are set to (5,2) and (2,1). The number of two-layer LSTM nodes is set to 256. There is a 1024*256 fully connected layer after the LSTM. In the encoder module, pad one zero-frame in front of the time dimension at each convolutional encoder layer. In the decoder module, look ahead with one frame in each convolutional layer.
- DCTCRN: the window length and hop sizes are 32 ms and 8 ms, and the DCT length is 512. The number of channels for the DCTCRN is {8, 16, 32, 64, 128, 128, 256}. The kernel size and stride are set to (5,2) and (2,1). The number of two-layer LSTM nodes is set to 256. In the encoder module, pad one zero-frame in front of the time dimension at each Conv2ds. In the decoder module, remove the last time frame at each transpose convolutional decoder.
- Baseline: the window length and hop sizes are 32 ms and 8 ms, and the DCT length is 512. The number of channels for the baseline is {16, 32, 64, 128, 128}. The kernel size and stride are set to (5,2) and (2,1). The number of F-T-LSTM nodes is set to 128. As with DCCRN-E, pad one zero-frame in front of the time dimension at each encoder and look ahead a frame at each decoder, totally $5 \times 8 = 40$ ms, confined with the DNS challenge limit—40 ms.

3.3 Evaluation Results and Discussions

The perceptual evaluation of speech quality (PESQ) [20] is employed to verify the noise reduction performance of DTLN, DCTCRN, DCCRN-E, and our model on dataset 1. We conduct ablation experiments to verify the performance of each module. Our proposed model achieves the highest PESQ scores among all models, which are shown in Table 1. In addition, we use flops-counter. Pytorch to compute the MACs and parameters of the models. Our model only has 1/3 parameters and 60% GMACs when compared with DCCRN, but the PESQ score on the DNS-2020 synthetic no reverb test set is 0.18 higher. Compared with DCTCRN, although our model is more computationally intensive, it can achieve a better noise reduction effect with fewer parameters.

Table 1. Various models' PESQ on DNS-2020 synthetic no reverb test set

Modle	Para.(M)	GMacs	look ahead(ms)	PESQ
Noisy	–	0	0	2.45
DTLN	1.0	1.58	0	3.04
DCTCRN	2.86	2.69	0	3.24
DCCRN-E	3.98	10.1	37.5	3.26
Baseline	1.08	5.12	40	3.39
+Convolutional SC	1.31	6.06	40	3.43
+Improved SI-SNR	1.31	6.06	40	**3.44**

To verify the noise reduction performance of the models at each dB and with or without reverberation. The PESQ and STOI [21] scores of the models in the test set 2 are tested. Table 2 and Table 3 show the objective results on the test set without reverberation, and Table 4 and Table 5 show the results under reverberant conditions, respectively (In the table, PROPOSED stands for Baseline + Convolutional SC + Improved SI-SNR). In each case, the best result is highlighted by a boldface number.

Table 2. Various models' PESQ on the non-reverberation dataset 2

test SNR	0 dB	5 dB	10 dB	15 dB	20 dB	Avg.
Noisy	1.559	1.876	2.222	2.560	2.870	2.217
DCTCRN	2.443	2.482	3.148	3.403	3.617	3.018
DCCRN-E	2.542	2.907	3.207	3.460	**3.704**	3.164
PROPOSED	**2.622**	**2.954**	**3.236**	**3.476**	3.684	**3.194**

Table 3. Various models' STOI(IN%) on the non-reverberation dataset 2

test SNR	0 dB	5 dB	10 dB	15 dB	20 dB	Avg.
Noisy	73.58	82.01	88.68	93.24	96.06	86.71
DCTCRN	83.62	90.13	93.81	96.05	97.52	92.22
DCCRN-E	84.97	90.71	94.01	96.11	97.54	92.66
PROPOSED	**85.65**	**91.03**	**94.22**	**96.24**	**97.65**	**92.95**

From the results of the non-reverberant set, it can be found that the PESQ score of DCCRN-E is slightly higher than our model at 20 dB. In all other cases, our model outperforms DCCRN-E and DCTCRN in both PESQ and STOI. Our model achieves state-of-the-art performance, with DCCRN-E being the second and DCTCRN being the worst. As can be seen from Tables 2 and 3, our model outperforms DCCRN-E at low SNRs and is similar to that of DCCRN-E at high SNRs.

Table 4. Various models' PESQ on the reverberation dataset 2

test SNR	0 dB	5 dB	10 dB	15 dB	20 dB	Avg.
Noisy	1.687	1.980	2.299	2.628	2.911	2.301
DCTCRN	2.412	2.804	3.134	3.408	3.626	3.076
DCCRN-E	2.433	2.801	3.128	3.416	3.653	3.086
PROPOSED	**2.511**	**2.905**	**3.235**	**3.507**	**3.724**	**3.176**

Table 5. Various models' STOI(IN%) on the reverberation dataset 2

test SNR	0 dB	5 dB	10 dB	15 dB	20 dB	Avg.
Noisy	72.38	82.95	90.57	95.37	97.86	87.82
DCTCRN	82.78	90.68	95.11	97.52	98.73	92.96
DCCRN-E	83.35	90.96	95.22	97.57	98.80	93.18
PROPOSED	**84.14**	**91.38**	**95.43**	**97.69**	**98.84**	**93.49**

On the reverberation test set, our model gets the best results among all conditions. DCTCRN and DCCRN-E yield similar PESQ and STOI scores, while our model performs much better than DCTCRN and DCCRN-E. Unlike the non-reverberant case, our model is much better than DCCRN-E in all dB conditions, and the results indicate that it is more promising for denoising with reverberation.

3.4 Conclusions

In this work, we propose a DCT-based real-valued CRNN for single-channel speech enhancement. We introduce the F-T-LSTM module and the convolutional skip connections module on the original CRNN and improve the loss function SI-SNR. Experimental results show that our model has only 1/3 parameters and 60% computational effort of DCCRN-E, but it outperforms both DCCRN-E and DCTCRN. In addition, our model has excellent noise suppression performance in the reverberation case.

References

1. Boll, S.: Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust. Speech Sig. Process. **27**(2), 113–120 (1979)
2. Hendriks, R.C., Heusdens, R., Jensen, J.: MMSE based noise PSD tracking with low complexity. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4266–4269. IEEE (2010)
3. Abd El-Fattah, M., Dessouky, M.I., Diab, S., Abd El-Samie, F.: Speech enhancement using an adaptive wiener filtering approach. Prog. Electromagnet. Res. M **4**, 167–184 (2008)
4. Hu, G., Wang, D.: Speech segregation based on pitch tracking and amplitude modulation. In: Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575), pp. 79–82. IEEE (2001)
5. Srinivasan, S., Roman, N., Wang, D.: Binary and ratio time-frequency masks for robust speech recognition. Speech Commun. **48**(11), 1486–1501 (2006)
6. Wang, X., Bao, C.: Mask estimation incorporating phase-sensitive information for speech enhancement. Appl. Acoust. **156**, 101–112 (2019)
7. Williamson, D.S., Wang, Y., Wang, D.: Complex ratio masking for monaural speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(3), 483–492 (2015)

8. Luo, Y., Mesgarani, N.: Conv-TasNet: surpassing ideal time-frequency magnitude masking for speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(8), 1256–1266 (2019)
9. Xu, Y., Du, J., Dai, L.R., Lee, C.H.: An experimental study on speech enhancement based on deep neural networks. IEEE Sig. Process. Lett. **21**(1), 65–68 (2013)
10. Hu, Y., et al.: DCCRN: deep complex convolution recurrent network for phase-aware speech enhancement. arXiv preprint arXiv:2008.00264 (2020)
11. Choi, H.S., Kim, J.H., Huh, J., Kim, A., Ha, J.W., Lee, K.: Phase-aware speech enhancement with deep complex U-Net. In: International Conference on Learning Representations (2018)
12. Tan, K., Wang, D.: A convolutional recurrent neural network for real-time speech enhancement. In: Interspeech. vol. 2018, pp. 3229–3233 (2018)
13. Li, Q., Gao, F., Guan, H., Ma, K.: Real-time monaural speech enhancement with short-time discrete cosine transform. arXiv preprint arXiv:2102.04629 (2021)
14. Li, J., Mohamed, A., Zweig, G., Gong, Y.: LSTM time and frequency recurrence for automatic speech recognition. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 187–191. IEEE (2015)
15. Zhou, L., Gao, Y., Wang, Z., Li, J., Zhang, W.: Complex spectral mapping with attention based convolution recurrent neural network for speech enhancement. arXiv preprint arXiv:2104.05267 (2021)
16. Reddy, C.K., et al.: ICASSP 2021 deep noise suppression challenge. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6623–6627. IEEE (2021)
17. Valentini-Botinhao, C., et al.: Noisy speech database for training speech enhancement algorithms and TTS models (2017)
18. Veaux, C., Yamagishi, J., King, S.: The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In: 2013 International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), pp. 1–4. IEEE (2013)
19. Varga, A.: The NOISEX-92 study on the effect of additive noise on automatic speech recognition. ICAL Report, DRA Speech Research Unit (1992)
20. Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P.: Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221). vol. 2, pp. 749–752. IEEE (2001)
21. Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J.: A short-time objective intelligibility measure for time-frequency weighted noisy speech. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4214–4217. IEEE (2010)