CASCADE OF LSTM-RES-UNET-BASED ENHANCEMENT AND BEAMFORMER FOR TARGET SPEECH EXTRACTION ON WEARABLE GLASSES

Jun Hyung Kim, Bon Hyeok Ku, Seok Hyun Kim, Jae Hyun Ko, Ui Hyeop Shin

Sogang University

ABSTRACT

The goal of the Speech enhancement for augmented reality (SPEAR) challenge is to obtain the best possible enhancement of the target binaural signal with head-worn microphone arrays where positional information is available to the algorithm. To solve this problem, we adopt low-latency DNN based speech enhancement based on LSTM-ResUNet, superdirective beamformer, and TRU-Net.

Index Terms— Frame-online speech enhancement, microphone array processing, deep learning, signal processing, beamforming.

1. INTRODUCTION

Array microphone-based speech signal processing are applied in many fields. In hearing aids, speech enhancement and separation is important tasks. Processing the hearing aid signal, such as a head-worn microphone array, is difficult because the relative positions of the array and sound sources change rapidly.

SPEAR Challenge provides head-worn microphone array signals to separate and enhance each target speaker [1]. The data consists of conversations between 3 to 6 people in a typical noisy restaurant. Real recorded data of SPEAR Challenge is from the EasyCom database [2]. Simulation data of SPEAR Challenge is generated using Tascar [3]. The goal of this challenge is to maximize the enhancement of the target signal from the array's binaural microphone. The enhanced target signal is evaluated in the Signal to Noise Ratio (SNR), Speech Intelligibility (SI), and Speech Quality (SQ) categories.

To achieve the goal of SPEAR challenge, we adopt low-latency DNN-based speech enhancement, superdirective beamformer followed by post-filtering. The DNN-based speech enhancement model is U-Net [4] based speech enhancement model using IRM [5] with 40ms algorithms latency. The superdirective beamformer generates a diffuse covariance matrix from microphone positions and separate speech signals. TRU-Net [6] is a voice enhancement model that separates frequency and time axis calculations from the 2D-CNN-based U-Net model and makes it lightweight and operates online.

2. MODEL DESCRIPTION

2.1. Problem Definition

The physical model of given input signal can be formulated in the Short-Time Fourier Transform (STFT) domain as

$$\begin{aligned}\mathbf{Y}(t,f) &= \sum_{n=1}^{N} \mathbf{X}_n(t,f) + \mathbf{V}(t,f) \\ &= \sum_{n=1}^{N} [\mathbf{S}_n(t,f) + \mathbf{H}_n(t,f)] + \mathbf{V}(t,f),\end{aligned} \tag{1}$$

where $\mathbf{Y}(t,f)$, $\mathbf{X}(t,f)$, $\mathbf{V}(t,f)$, $\mathbf{S}(t,f)$, and $\mathbf{H}(t,f)$ denote the STFT vectors of the mixture, reverberant target speech, reverberant noise, target speech and early reflection of target speech, late reflection of target speech at time t and frequency f for each target speaker n among N people, respectively.

And the target speech to be extracted can be formulated as

$$\widehat{\mathbf{S}}_n(t,f) = f(\mathbf{X}, \theta_n, \phi_n), \tag{2}$$

where $\widehat{\mathbf{S}}_n(t,f)$ is estimated speech of target speaker n using given azimuth θ_n and elevation ϕ_n.

2.2. Primary DNN-based speech enhancement

2.2.1. Network architecture

Inspired by LSTM-ResUNet[7], we adopt a low-latency speech enhancement model. To prevent distortion in phase components for better linear spatial filter performance, we perform primary DNN-based single-channel speech enhancement on the magnitude domain. Speech enhancement was performed on each channel separately with the same parameters.

As illustrated in Fig.1, the primary model is U-Net with long short-term memory(LSTM) bottleneck. And the structures of each block are described in Fig. 2. Each encoder contains two-dimensional(2D) convolution, batch normalization (BN), parametric ReLU (PReLU), and residual block. Channels, kernel size, and stride of five encoders are 30,(3,1),and (2,1) respectively. Likewise, each decoder consists of 2D

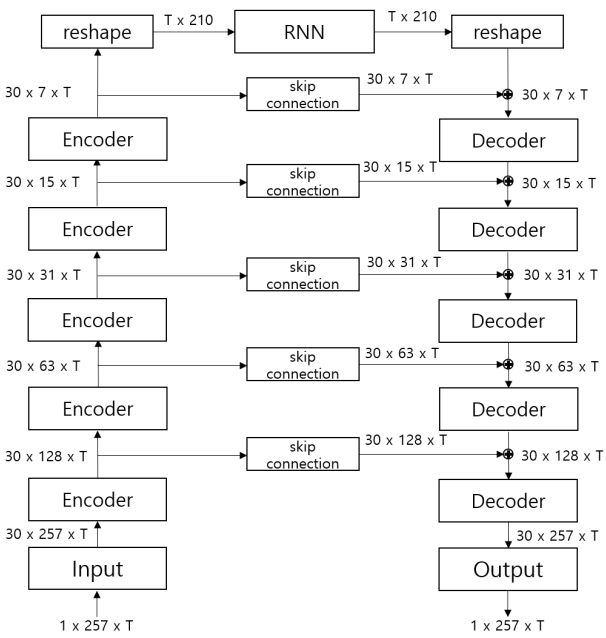

Fig. 1: Network architecture of Primary speech enhancement Model.

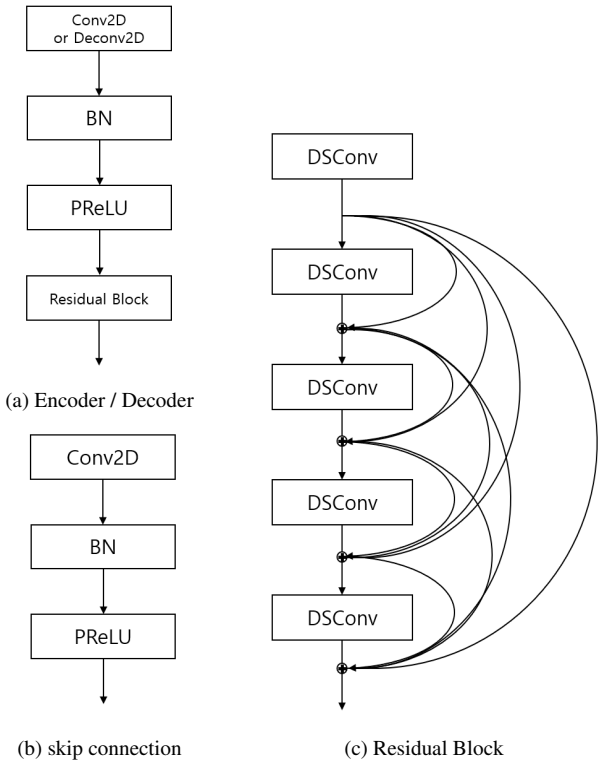

Fig. 2: Block structures.

deconvolution, batch normalization (BN), parametric ReLU (PReLU), and residual block. Channels, stride, and padding of five decoders are 30,(2,1), (1,0) respectively. Kernel sizes of each decoder are (4,1), (4,1),(4,1), (5,1), and (4,1). The input layer consists of 2D convolution, LeakyReLU, and BN followd by Layer Normalization. The output Layer contains 2D deconvolution and sigmoid activation.

2.2.2. Residual block

Each residual block in the encoder and decoder contains five depthwise separable 2D convolutions(DSConv). The kernel size of depthwise convolution is (3,3) which has 8ms look ahead. The dilation rates are 1, 2, 4, 8, and 16, respectively.

2.2.3. Algorithmic latency

Because most component of human speech resides under 8kHz, we resampled the input audio of the primary model as 16kHz. A 512-point Discrete Fourier Transform(DFT) is applied to extract 257-dimensional STFT coefficients at each frame. The window size of the model is 32ms with 8ms look ahead, which makes algorithmic latency 40ms.

2.2.4. Experiment

SPEAR challenge dataset is used as training and validation. Each channel of data is separately used as input and reference. We adopt the loss function form of Mean Square Error(MSE). The Adam algorithm is used to optimize the models.

2.3. Superdirective beamformer

The superdirective beamformer is one of the fixed beamformers that obtains the maximum gain against diffuse noise. The baseline for the SPEAR challenge generates an isotropic diffuse covariance matrix from acoustic transfer functions for superdirective beamformer. We generate the spherically isotropic noise field from microphone positions [2]. Since microphones 5 and 6 are on each ear, we use channels 1,2,3, and 4 to avoid a mismatch of microphone position.

The diffuse covariance matrix is calculated as

$$[\mathbf{R}(\omega)]_{i,j} = \frac{\sin(\omega d_{i,j})}{\omega d_{i,j}}, \tag{3}$$

where ω is angular frequency and $d_{i,j}$ is the delay between i-th and j-th microphones. Therefore, the beamformer weight of the direction azimuth θ and elevation ϕ can be obtained by

$$\mathbf{w}(\omega,\theta,\phi) = \frac{(\mathbf{R}(\omega)+\delta\mathbf{I})^{-1}\,\bar{\mathbf{h}}(\omega,\theta,\phi)}{\bar{\mathbf{h}}^{H}(\omega,\theta,\phi)\,(\mathbf{R}(\omega)+\delta\mathbf{I})^{-1}\,\bar{\mathbf{h}}(\omega,\theta,\phi)}, \tag{4}$$

where

$$\bar{\mathbf{h}}(\omega,\theta,\phi) = \mathbf{h}(\omega,\theta,\phi)/h_{ref}(\omega,\theta,\phi). \tag{5}$$

$\bar{\mathbf{h}}(\omega, \theta, \phi)$ is normalized steering vector and δ is diagonal loading parameter. $\mathbf{h}(\omega, \theta, \phi)$ is the acoustic transfer function of each channel and $h_{ref}(\omega, \theta, \phi)$ is the acoustic transfer function of reference channel for each binaural channel.

STFT parameters are the same as in Section 2.1. The input data is downsampled as 16kHz. The 512-point DFT is applied. The window size of the model is 32ms with 8ms look ahead, which makes algorithmic latency 40ms.

2.4. Post DNN based speech enhancement

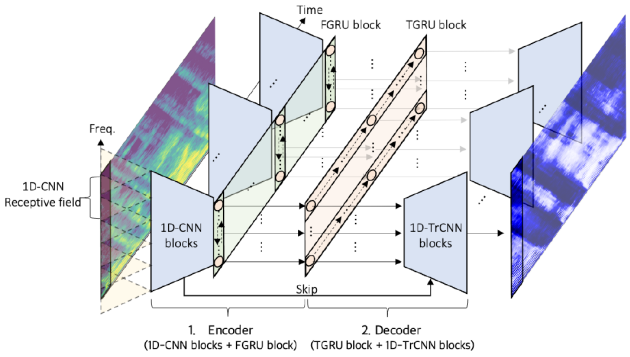

Fig. 3: Network architecture of TRU-Net.

2.4.1. *Network architecture*

Inspired by TRU-Net, we adopt a low-latency speech enhancement model. The architecture is designed to enable efficient decoupling of the frequency-axis and time-axis computations, which makes the network fast enough to process a single frame in real-time.

TRU-Net is based on U-Net architecture, except that the convolution kernel does not span the time-axis. Therefore, it can be considered a frequency-axis U-Net with 1D Convolutional Neural Networks (CNNs) and recurrent neural networks in the bottleneck layer. The encoder is composed of 1D Convolutional Neural Network (1DCNN) blocks and a Frequency-axis Gated Recurrent Unit (FGRU) block. Each 1D-CNN block is a sequence of pointwise convolution and depthwise convolution similar to [8], except the first layer, which uses the standard convolution operation without a preceding pointwise convolution. To spare the network size, we use six 1D-CNN blocks, which downsample the frequency-axis size from 256 to 16 using strided convolutions. This results in a small receptive field which may be detrimental to the network performance. To increase the receptive field, we use a bi-directional GRU layer [9] along the frequency-axis instead of stacking more 1D-CNN blocks. That is, the sequence of 16 vectors from 1D-CNN blocks is passed into the bi-directional GRU(FGRU) to increase the receptive field and share the information along the frequency-axis. Pointwise convolution, batch normalization, and rectified linear unit (ReLU) are used after the FGRU layer, composing an FGRU block. We used 64 hidden dimensions for each forward and backward FGRU cell.

The decoder is composed of a Time-axis Gated Recurrent Unit (TGRU) block and 1D Transposed Convolutional Neural Network (1D-TrCNN) blocks. The output of the encoder is passed into a unidirectional GRU layer(TGRU layer) to aggregate the information along the timeaxis. While one can apply different GRU cells to each frequency-axis index of the encoder output, we shared the same cell on each frequency-axis index to save the number of parameters. A pointwise convolution, BN, and ReLU follow the TGRU layer, composing a TGRU block. We used 128 hidden dimensions for the TGRU cell. Finally, 1D-TrCNN blocks are used to upsample the output from the TGRU block to the original spectrogram size. The 1D-TrCNN block takes two inputs - 1. a previous layer output, 2. a skipped tensor from the encoder at the same hierarchy - and upsamples them as follows. First, the two inputs are concatenated and projected to a smaller channel size (256 -> 64) using a pointwise convolution. Then, 1D transposed convolution is used to upsample the compressed information. This procedure saves both the number of parameters and computation compared to the usual U-Net implementation where the two inputs are concatenated and upsampled immediately using the transposed convolution operation. Every convolution operation used in the encoder and decoder is followed by BN and ReLU.

Channels, kernel size, and stride of six 1D-CNN in encoders are (64,5,2), (128,3,1), (128,5,2), (128,3,1), (128,5,2), and (128,3,2), respectively. Likewise, six 1D-TrCNN in decoders are (128,3,2), (64,5,2), (64,3,1), (64,5,2), (64,3,1), and (10,5,2), respectively. Note that the pointwise convolution operations share the same output channel configuration with the exception that kernel size and strides are both 1. The overview of TRU-Net and the number of parameters used for 1D-CNN blocks, FGRU blocks, TGRU blocks, and 1DTrCNN blocks are shown in Fig. 3.

2.4.2. *TRUNet preprocess*

As inputs of TRUNet, STFT magnitude, phase and PCEN were used. Per-channel energy normalization (PCEN) [10] combines both dynamic range compression and automatic gain control, which reduce the variance of foreground loudness and suppress background noise when applied to a spectrogram [11]. PCEN is also suitable for online inference scenarios as it includes a temporal integration step, which is essentially a first-order infinite impulse response filter that depends solely on a previous input frame. In this work, we employ the trainable version of PCEN.

2.4.3. *Algorithmic latency*

For the same reason as the previous DNN, the input model was resampled at 16kHz and a 512-point DFT was used. The

method	STOI	PESQ-NB	SegSNR	SI-SDR
noisy	0.524	1.512	-6.645	-15.711
baseline	0.639	1.834	-4.920	-16.984
DNN_1	0.678	2.069	-2.946	-16.747
DNN_1 + beamforming	0.651	1.982	-2.959	-18.009
DNN_1 + beamforming + DNN_2	0.626	1.835	-1.615	-17.631

Table 1: Validation results for noisy input, baseline with superdirective beamformer, and proposed methods

window size of the model is 32ms with 8ms look ahead, which makes algorithmic latency 40ms.

2.4.4. Experiment

The SPEAR Challenge dataset was fine-tuned to the model pre-trained with the ICASSP 2022 deep noise suppression challenge dataset [12]. Each channel of data is separately used as input and reference. We adopt the loss function that sum of CosSDRLoss and SpectrogramLoss. The Adam algorithm is used to optimize the models.

3. RESULTS

Table 1. shows validation results for the development set of the SPEAR dataset. We compare proposed methods with noisy and baseline and present the evaluated results of STOI[13], PESQ-NB[14], segSNR, and SI-SDR[15]. We computed the average results of Dataset1, Dataset2, Dataset3, and Dataset4. DNN_1 is the primary DNN-based speech enhancement followed by the baseline method. DNN_1 + beamforming is DNN_1 followed by proposed superdirectivie beamforimg. DNN_1+beamforming + DNN_2 uses TRU-Net as post filter of DNN_1 + beamforming.

4. CONCLUSION

In this report, we present speech enhancement methods by using DNN-based LSTM-ResUnet as the primary network and TRU-Net as post-filtering. And superdirective beamformer is also applied to extract target speech in a cocktail party scenario.

For future work, we will study on multi-channel speech enhancement model as a primary model and linear spatial filter which can be combined better with non-linear DNN.

5. REFERENCES

[1] Pierre Guiraud, Sina Hafezi, Patrick A. Naylor, Alastair H. Moore, Jacob Donley, Vladimir Tourbabin, and Thomas Lunner, "An introduction to the speech enhancement for augmented reality (spear) challenge," in *2022 International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2022, pp. 1–5.

[2] Jacob Donley, Vladimir Tourbabin, Jung-Suk Lee, Mark Broyles, Hao Jiang, Jie Shen, Maja Pantic, Vamsi Krishna Ithapu, and Ravish Mehra, "Easycom: An augmented reality dataset to support algorithms for easy communication in noisy environments," *CoRR*, vol. abs/2107.04174, 2021.

[3] Giso Grimm, Joanna Luberadzka, and Volker Hohmann, "A toolbox for rendering virtual acoustic environments in the context of audiology," *Acta Acustica united with Acustica*, vol. 105, no. 3, pp. 566–578, 2019.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[5] Yuxuan Wang, Arun Narayanan, and DeLiang Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[6] Hyeong-Seok Choi, Sungjin Park, Jie Hwan Lee, Hoon Heo, Dongsuk Jeon, and Kyogu Lee, "Real-time denoising and dereverberation wtih tiny recurrent u-net," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5789–5793.

[7] Zhong-Qiu Wang, Gordon Wichern, Shinji Watanabe, and Jonathan Le Roux, "Stft-domain neural speech enhancement with very low algorithmic latency," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 397–410, 2023.

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger

Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[10] Yuxuan Wang, Pascal Getreuer, Thad Hughes, Richard F Lyon, and Rif A Saurous, "Trainable frontend for robust and far-field keyword spotting," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5670–5674.

[11] Vincent Lostanlen, Justin Salamon, Mark Cartwright, Brian McFee, Andrew Farnsworth, Steve Kelling, and Juan Pablo Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2018.

[12] Harishchandra Dubey, Vishak Gopal, Ross Cutler, Ashkan Aazami, Sergiy Matusevych, Sebastian Braun, Sefik Emre Eskimez, Manthan Thakker, Takuya Yoshioka, Hannes Gamper, et al., "Icassp 2022 deep noise suppression challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9271–9275.

[13] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[14] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra, "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)*. IEEE, 2001, vol. 2, pp. 749–752.

[15] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey, "Sdr–half-baked or well done?," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.