INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

MSAF: A Multiple Self-Attention Field Method for Speech Enhancement

Minghang Chu[1], *Jing Wang*[1], *Yaoyao Ma*[1], *Zhiwei Fan*[1],*Mengtao Yang*[1],*Chao Xu*[1], *Zhi Tao*[1,*], *Di Wu*[1,*]

[1]School of Optoelectronic Science and Engineering, Soochow University, China
[*]Corresponding authors: Di Wu, Zhi Tao
wudi@suda.edu.cn, taoz@suda.edu.cn

Abstract

Speech enhancement (SE) systems, based on generative adversarial networks (GANs), are limited in improving speech quality and intelligibility. In this study, we propose a novel multiple self-attention field method for speech enhancement (MSAF). The models with different positions of the self-attention layers focus on different features. The output of each model is assigned a different feature weight, which is obtained by training. Then, we fuse the models according to the feature weights to obtain a clean speech signal. For speech quality, the proposed method improves by 8.22%, 8.52%, 9.28%, and 9.40% in CBAK, CSIG, COVL, and PESQ on average compared with the baseline SASEGANs. The results show that the MSAF comprehensively improves the performance of the baseline SASEGAN and performs better than the mainstream GAN-based SE methods. Importantly, the proposed method can be extended to other GAN-based SE methods.

Index Terms: deep learning, multi-field self-attention, generative adversarial network (GAN), speech enhancement (SE)

1. Introduction

In recent years, more and more speech-related applications have become popular, including controlling smart devices through voice commands [1], supporting voice search in various scenarios [2], real-time meeting records, etc. However, various noises exist in real-world scenarios, which affects the performance of speech-related applications. Speech enhancement (SE) is used to attenuate noise in speech signals, thereby reducing the impact of noise on speech-related applications.

Speech enhancement is divided into deep learning-based methods and traditional speech signal processing methods [3]. With the rapid development of deep learning (DL), DL-based SE methods [4, 5] perform better than the classic SE methods [6, 7]. Deep learning-based SE methods mainly include convolutional neural networks (CNNs) [8, 9], recurrent neural networks (RNNs) [10, 11], and generative adversarial networks (GANs) [12, 13, 14, 15]. As reported in [16, 17], a deep neural network (DNN) was adopted as a mapping network. The input of the network is the logarithmic power spectrum (LPS) features of noisy speech and the output of the network is that of clean speech. Park et al. [18] adopted a fully convolutional network to map the short-time Fourier transform (STFT) of noisy speech to that of clean speech.

Recently, SEGAN-based variants have improved SEGAN in various aspects. In terms of input types, raw waveforms [19] and spectrograms [20] have been utilized. However, these SEGAN variants still rely heavily on convolutional layers. The receptive field, as a property of convolution operations, limits the ability of SEGAN and SEGAN variants to capture long-range dependencies of input sequences. To address this issue, Phan et al. [15] coupled a self-attention layer [21] with the convolutional layer of SEGAN. The findings demonstrated that, in all objective evaluation metrics, the SASEGAN proposed by Phan performs better than the baseline SEGAN. But the performance of SASEGAN is influenced by the position of the self-attention layer.

In this paper, to solve the above issues, we propose a novel multiple self-attention field method for speech enhancement (MSAF). The models with different positions of the self-attention layers focus on different features. Considering the effect of the position of the self-attention layers on SE performance, SASEGANs with self-attention layers at different positions are fused. Importantly, our method may apply to other GAN-based speech enhancement systems.

Section 2 introduces the proposed method; Section 3 presents experimental settings; Section 4 is the results and discussion, and conclusions are given in Section 5.

2. Method

2.1. Multiple Self-Attention Field Method

The mainstream of model fusion is divided into three categories: bagging [22], stacking [23], and boosting [24]. These three methods are mostly used in the field of machine learning, such as random forests.

Bagging is divided into using the same algorithm and using different algorithms. For using the same algorithm, we sample a subset from the training set and construct a new model on the subset. Suppose we generate M bootstrap data sets and the

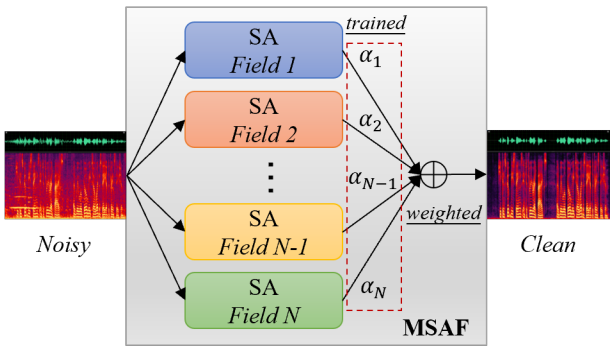


Figure 1: *Schematic diagram of the MSAF.*

10.21437/Interspeech.2023-886

committee prediction is given by:

$$y_{com}(x) = \sum_{m=1}^{M} y_m(x) \tag{1}$$

where $y_{com}(x)$ denotes the committee prediction and $y_m(x)$ denotes the output of the m_{th} model. This procedure is known as bagging. As shown in Figure 1, every model is input the same training set when using different algorithms. Taking N models as an example, α_i (i=1,2,...,N) represents the weight coefficient of the i_{th} model output accounting for the final output. Fusion output is calculated by the following formula:

$$y_{fusion}(x) = \sum_{i=1}^{N} \alpha_i \cdot y_i(x) \tag{2}$$

where $y_{fusion}(x)$ denotes the output of the model fusion. One advantage of bagging is that it allows parallel training of base models. However, it assumes that the errors due to individual models are uncorrelated, which is often not the case, but it can still achieve better performance than any individual model.

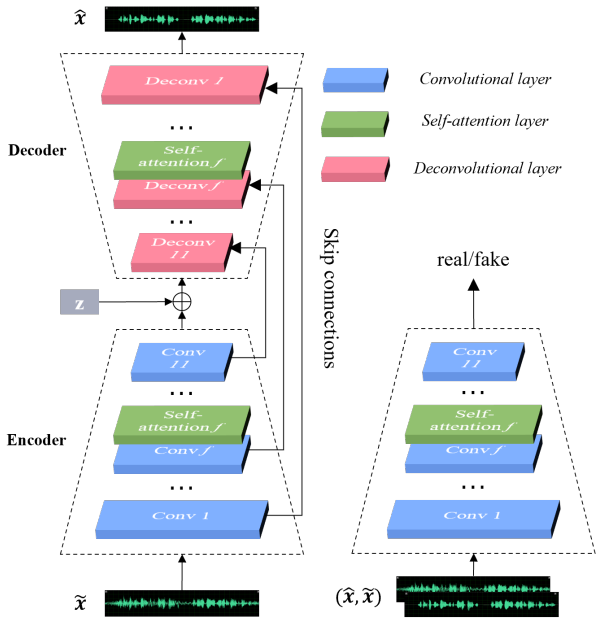
The ideas of stacking and blending are similar. The prediction of the basic model (level 0) is adopted as the input to build the meta-model (level 1). The implementation steps of stacking are all on the training set. First, the training data set is split into N folds and the $(N-1)_{th}$ fold data is used to train the first basic model. Then, making predictions on the test folds, the prediction results are saved. Continue the above steps N times for the base model until it produces a prediction dataset of the same size as the original training dataset. Repeating the above steps for each baseline model will eventually result in predictions for all base models for the entire training dataset. Now we build the level 1 model, also known as the meta-model, which uses the predictions of all the base models as input features and the target values from the original training set as targets.

Schematic diagrams of the multi-field self-attention method are shown in Figure 1. Noisy denotes a noise-corrupted audio signal and clean denotes a clean audio signal. SA Fields indicate that the model has different characteristic receptive fields for noise. α represents the weight coefficient of the model of different fields. Noisy signals are enhanced to clean signals by a model with multiple fields.

2.2. Network Architecture of the proposed method

For speech enhancement, the goal is to find a function $f\left(\widetilde{\mathbf{x}}\right) : \widetilde{\mathbf{x}} \mapsto \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^T$ denotes a clean audio signal and $\widetilde{\mathbf{x}} \in \mathbb{R}^T$ denotes a noise-corrupted audio signal, to recover the clean audio signal $\mathbf{x}$ from the noise-corrupted audio signal $\widetilde{\mathbf{x}}$. T denotes the time length. The noise-corrupted audio signal $\widetilde{\mathbf{x}}$ is defined as $\widetilde{\mathbf{x}} = \mathbf{x} + \mathbf{n} \in \mathbb{R}^T$, where $\mathbf{n} \in \mathbb{R}^T$ denotes the additive background noise signal. By specifying the generator as the enhancement mapping, SEGAN methods [18, 25, 26] accomplish this goal. To improve the attention of the network in the correlation of time dimension, a self-attention speech enhancement generative adversarial network (SASEGAN) [15] was proposed. Inspired by SASEGAN, the network architecture of base models of MSAF is shown in Figure 2.

The generator consists of an encoder and a decoder, as shown in Figure 2(a). Each convolutional layer of the encoder is skip-connected to a convolutional layer of the decoder. z is a matrix randomly sampled from the Gaussian distribution, and the size of z is the same as that of the output of the encoder.



(a) Generator (b) Discriminator

Figure 2: *Network architecture of MSAF.*

Based on this, the robustness of the network is improved. Gray blocks represent self-attention layers, which can be placed after any convolutional layer in the encoder. In the decoder, the location of the self-attention layer mirrors that of the encoder. f denotes the field label of the self-attention layer.

For the discriminator, the network architecture is similar to the generator's encoder. The difference is that the input of the discriminator is a pair of audios $\left(\hat{x}, \widetilde{x}\right)$.

Due to the uncertainty of the self-attention layers' position in SASEGANs on the effect of speech enhancement, we observe that the self-attention layer learns different features at different positions in SASEGAN. As shown in Figure 3, base models with self-attention layers at different positions are fused by bagging. Here, we set N as 11.

Various losses were proposed to improve adversarial training for GAN, such as Wasserstein loss [**?**], least-squares loss [**?**], and relativistic loss [19]. Here, the least-squares loss is adopted. We make some improvements to it, and the objective functions of the discriminator and the generator are written as:

$$\begin{aligned} \min_D \ \mathcal{L}_{MMLS}(D) &= \mathbb{E}_{x,z}[\frac{1}{2}\sum_{l=1}^{11}(D(x,\widetilde{x})-1)^2] \\ &+\mathbb{E}_{x,z}[\frac{1}{2}\sum_{l=1}^{11}D(\alpha_l \cdot G_l(z,\widetilde{x}),\widetilde{x})^2] \end{aligned} \tag{3}$$

$$\begin{aligned} \min_G \ \mathcal{L}_{MMLS}(G) &= \mathbb{E}_{x,z}[\frac{1}{2}\sum_{l=1}^{11}(D(\alpha_l \cdot G_l(z,\widetilde{x}),\widetilde{x})-1)^2] \\ &+\lambda \cdot \mathbb{E}_{x,z}[\left|\left|\sum_{l=1}^{11}(\alpha_l \cdot G_l(z,\widetilde{x})) - x\right|\right|_1] \end{aligned} \tag{4}$$

where $D(\cdot)$ denotes the discriminator output and $G_l(\cdot)$ denotes the output of the generator with the self-attention layer placed

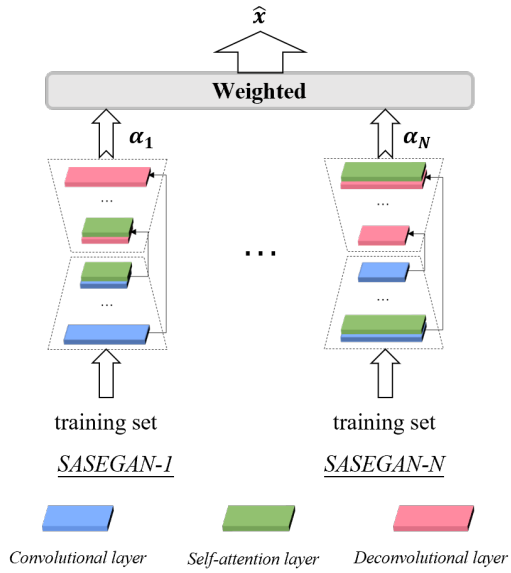


Figure 3: *Illustration of MSAF. SASEGAN-1 denotes the self-attention layer is placed after the first convolutional layer in SASEGAN and SASEGAN-11 denotes the self-attention layer is placed after the last convolutional layer in SASEGAN.*

after the l_{th} convolutional layer. $\alpha_l (l \in [1, 13])$ denotes the weight coefficient of the output of SASEGAN-l to the output of MSAF. λ is a hyperparameter that adjusts the proportion of l1 loss in the full objective loss function.

It can be seen that the objective functions contain the output of multiple models, which are called multi-model least squares (MMLS) loss functions. SASEGANs with self-attention at different positions are forced to learn different latent features by MMLS loss.

3. Experiments

3.1. Dataset

We used a public dataset, which is available[1] . The dataset consists of utterances from thirty speakers taken from the Voice Bank corpus [27]. According to previous work, 28 speakers were adopted for training and 2 speakers were adopted for testing. For the training set, 10 types of noise were combined with the clean speech of 28 speakers at signal-to-noise ratios (SNRs) of 15, 10, 5, and 0 dB. Similarly, 5 types of noise were combined with the clean speech of 2 speakers at SNRs of 2.5, 7.5, 12.5, and 17.5 dB. All speech signals were downsampled to 16kHz.

3.2. Experimental Settings

All our work was performed on the Tensorflow [28] framework. The networks were trained using a mini-batch size of 50, RMSprop [29], and a learning rate (LR) of 0.0002 for 100 epochs. We set the weight of our $L1$ regularization λ to 100 for the whole training. During training, we use a sliding window (50% overlap) of roughly 1 second of speech to extract waveform chunks. Instead, we slide windows without overlapping them throughout the test audio signals and concatenate the results in the test phase. Both in the training and testing phases, a high-frequency pre-emphasis filter with a factor of 0.95 is applied to all input samples.

4. Results and discussion

The proposed method is compared with the baseline method SASEGAN and quantitatively evaluated at PESQ (speech quality), CBAK (the invasiveness of background noise), CSIG (speech signal distortion), COVL (the overall effect), SSNR (segmental signal-to-noise rate), STOI (short-time objective intelligibility).

Results of the MSAF and the mainstream speech enhancement methods are shown in Table 1. The higher the value, the better the performance of the method. We bold the best results for GAN-based methods. The compared methods are divided into two parts: GAN-based methods and non-GAN-based methods. SASEGAN-Ns denote SASEGANs that the self-attention layers are placed in different positions, where N is the index of the self-attention layer position. We find that the proposed method outperforms baseline models for CBAK, CSIG, COVL, and PESQ, as shown in Figure 4. Compared with the baseline SASEGANs on average, the proposed method improves by 9.40%, 8.52%, 8.22%, and 9.28% in PESQ, CSIG, CBAK, and COVL. Similarly, the proposed method outperforms ISEGAN and DSEGAN which are GAN-based SE methods. For Non-GAN-based SE methods, the proposed method outperforms DNN. TSN slightly outperforms the proposed method on PESQ, COVL, and CSIG, but STOI and SSNR of the TSN are close to noisy, as shown in Figure 5.

For a more intuitive comparison, SSNR and STOI are normalized in Figure 5. The higher the bar, the better the performance of the method. It can be seen that DNN and TSN perform far worse than the other methods on SSNR and STOI, even worse than noisy-corrupted speech. This shows that TSN, while capable of improving speech quality, performs poorly in terms of SSNR and speech intelligibility. We find that GAN-based SE methods can improve the STOI and SSNR of noisy-corrupted-speech, under the premise of ensuring the improvement of speech quality. For SSNR, MSAF obtains absolute gains of 1.71 on average compared with the baseline SASEGANs. In terms of speech intelligibility STOI, MSAF obtains absolute gains of 0.77 on average.

The results show that the proposed method MSAF outperforms the mainstream SEGAN-based methods. For non-GAN-based methods, the proposed method MSAF is comparable in improving speech quality and outperforms it in segmental SNR and speech intelligibility. In addition, we also conduct ablation experiments to investigate the impact of self-attention layer positions on the results. For more details, the URL is available[2].

5. Conclusions

In this study, we propose a novel multiple self-attention field method for speech enhancement (MSAF). SASEGANs with self-attention layers at different positions are fused. For speech quality, the proposed method MSAF improves by 8.22%, 8.52%, 9.28%, and 9.40% in CBAK, CSIG, COVL, and PESQ on average compared with the baseline SASEGANs. As to SSNR and STOI, compared with the baseline SASEGANs on average, the proposed method obtains absolute gains of 1.71 and 0.77, respectively. The results show that the MSAF compre-

[1] https://datashare.ed.ac.uk/handle/10283/2791

[2] https://mmf-sasegan.github.io/

Table 1: *Results of MSAF and the mainstream speech enhancement (SE) methods.*

	PESQ	CSIG	CBAK	COVL	SSNR	STOI(%)
Noisy	1.97	3.35	2.44	2.63	1.68	92.10
			Non-GAN-based SE Methods			
DNN [17]	2.45	3.73	2.89	3.09	3.64	89.14
TSN [30]	2.68	3.96	2.94	3.32	2.89	92.52
			GAN-based SE Methods			
SEGAN [13]	2.19	3.39	2.90	2.76	7.36	93.12
DSEGAN [14]	2.35	3.55	3.10	2.93	8.70	93.25
ISEGAN [14]	2.24	3.23	2.95	2.69	8.17	93.29
SASEGAN-3 [15]	2.32	3.51	3.07	2.90	8.53	93.35
SASEGAN-4 [15]	2.36	3.57	3.08	2.95	8.38	93.47
SASEGAN-5 [15]	2.31	3.46	2.94	2.85	7.20	93.22
SASEGAN-6 [15]	2.38	3.46	3.12	2.90	8.86	93.39
SASEGAN-7 [15]	2.30	3.52	2.98	2.89	7.34	93.38
SASEGAN-8 [15]	2.34	3.55	3.03	2.92	8.03	93.24
SASEGAN-9 [15]	2.29	3.45	3.05	2.85	8.48	93.28
SASEGAN-10 [15]	2.41	3.62	3.06	2.99	7.87	93.36
SASEGAN-11 [15]	2.35	3.57	3.03	2.94	7.76	93.19
			Proposed			
MSAF	**2.56**	**3.82**	**3.29**	**3.18**	**9.76**	**94.09**

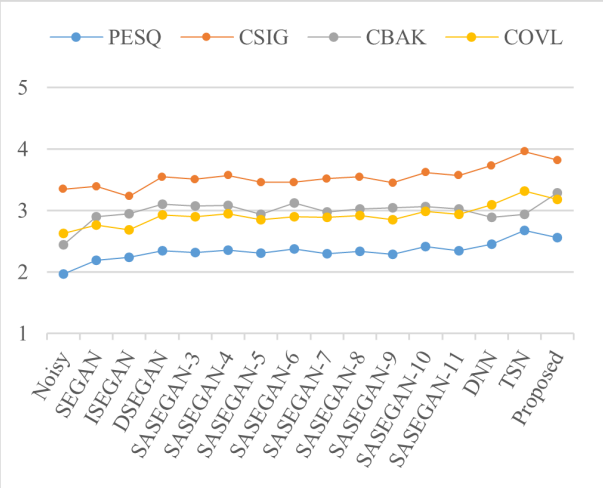


Figure 4: *Results of PESQ, CSIG, CBAK, and COVL for the proposed method and the compared methods.*

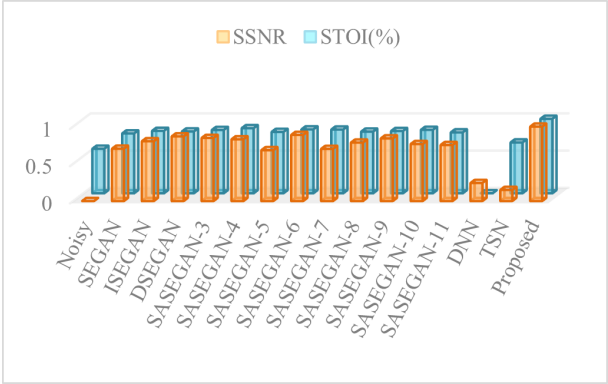


Figure 5: *Results of SSNR and STOI for the proposed method and the compared methods.*

hensively improves the performance of the baseline SASEGAN and performs better than the mainstream GAN-based SE methods. Importantly, the proposed method may generalize to other GAN-based SE methods. In future work, we will try to apply it to other mainstream GAN-based SE methods.

6. References

[1] A. A. Arriany and M. S. Musbah, "Applying voice recognition technology for smart home networks," in *2016 International Conference on Engineering & MIS (ICEMIS)*, Sep. 2016, pp. 1–6.

[2] Y.-Y. Wang, D. Yu, Y.-C. Ju, and A. Acero, "An introduction to voice search," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 28–38, May 2008.

[3] D. Michelsanti, Z.-H. Tan, S.-X. Zhang, Y. Xu, M. Yu, D. Yu, and J. Jensen, "An Overview of Deep-Learning-Based Audio-Visual Speech Enhancement and Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1368–1396, 2021.

[4] J. Richter, G. Carbajal, and T. Gerkmann, "Speech Enhancement with Stochastic Temporal Convolutional Networks," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 4516–4520.

[5] S. K. Roy, A. Nicolson, and K. K. Paliwal, "A Deep Learning-Based Kalman Filter for Speech Enhancement," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 2692–2696.

[6] M. A. Abd El-Fattah, M. I. Dessouky, A. M. Abbas, S. M. Diab,

E.-S. M. El-Rabaie, W. Al-Nuaimy, S. A. Alshebeili, and F. E. Abd El-samie, "Speech enhancement with an adaptive Wiener filter," *International Journal of Speech Technology*, vol. 17, no. 1, pp. 53–64, Mar. 2014.

[7] X. Yan, Z. Yang, T. Wang, and H. Guo, "An Iterative Graph Spectral Subtraction Method for Speech Enhancement," *Speech Communication*, vol. 123, pp. 35–42, Oct. 2020.

[8] A. Pandey and D. Wang, "TCNN: Temporal Convolutional Neural Network for Real-time Speech Enhancement in the Time Domain," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 6875–6879.

[9] Z. Ouyang, H. Yu, W.-P. Zhu, and B. Champagne, "A Fully Convolutional Neural Network for Complex Spectrogram Processing in Speech Enhancement," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5756–5760.

[10] Y. Xia, S. Braun, C. K. A. Reddy, H. Dubey, R. Cutler, and I. Tashev, "Weighted Speech Distortion Losses for Neural-Network-Based Real-Time Speech Enhancement," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 871–875.

[11] D. Takeuchi, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Real-Time Speech Enhancement Using Equilibriated RNN," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 851–855.

[12] Z. Zhang, C. Deng, Y. Shen, D. S. Williamson, Y. Sha, Y. Zhang, H. Song, and X. Li, "On Loss Functions and Recurrency Training for GAN-Based Speech Enhancement Systems," in *Proc. Interspeech 2020*, 2020, pp. 3266–3270.

[13] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network," in *Proc. Interspeech 2017*, 2017, pp. 3642–3646.

[14] H. Phan, I. V. McLoughlin, L. Pham, O. Y. Chén, P. Koch, M. De Vos, and A. Mertins, "Improving GANs for Speech Enhancement," *IEEE Signal Processing Letters*, vol. 27, pp. 1700–1704, 2020.

[15] H. Phan, H. L. Nguyen, O. Y. Chén, P. Koch, N. Q. K. Duong, I. McLoughlin, and A. Mertins, "Self-Attention Generative Adversarial Network for Speech Enhancement," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 7103–7107.

[16] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An Experimental Study on Speech Enhancement Based on Deep Neural Networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[17] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A Regression Approach to Speech Enhancement Based on Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[18] S. R. Park and J. W. Lee, "A Fully Convolutional Neural Network for Speech Enhancement," in *Proc. Interspeech 2017*, 2017, pp. 1993–1997.

[19] D. Baby and S. Verhulst, "Sergan: Speech Enhancement Using Relativistic Generative Adversarial Networks with Gradient Penalty," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 106–110.

[20] Z.-X. Li, L.-R. Dai, Y. Song, and I. McLoughlin, "A Conditional Generative Model for Speech Enhancement," *Circuits, Systems, and Signal Processing*, vol. 37, no. 11, pp. 5005–5022, Nov. 2018.

[21] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-Local Neural Networks," in *CVPR 2018*, 2018, pp. 7794–7803.

[22] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[23] S. Džeroski and B. Ženko, "Is Combining Classifiers with Stacking Better than Selecting the Best One?" *Machine Learning*, vol. 54, no. 3, pp. 255–273, Mar. 2004.

[24] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *ICML*, 1996, pp. 148–156.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.

[26] Y. Xiang and C. Bao, "A Parallel-Data-Free Speech Enhancement Method Using Multi-Objective Learning Cycle-Consistent Generative Adversarial Network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1826–1838, 2020.

[27] C. Veaux, J. Yamagishi, and S. King, "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)*, Nov. 2013, pp. 1–4.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[29] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[30] J. Kim and M. Hahn, "Speech Enhancement Using a Two-Stage Network for an Efficient Boosting Strategy," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 770–774, May 2019.