International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Speech Enhancement Using Deep Learning Techniques

Ghanta Pavankalyan[1], Gowtham Bobbili[2]

[1,2]Department of ECE, SRM Institute of Science and Technology, Kattankulathur, India

Abstract

In current era of digital technology, speech enhancement has become one of the important challenges in day-to-day life for smoother conversation and understanding. Most of the recorded or live speech audio signals contain some form of noise such as street noise, barking sounds of dogs, conversation between multiple people, construction noise, car honking etc. The key feature of speech enhancement system is to execute in real time on a recorded waveform with minimal possible lag on any localized hardware with minimum computational resource. Hence, speech enhancement issue is of practical importance in the industry as well as research community to improve the hearing ability of the human being. The proposed algorithm of speech enhancement utilizing the encoder-decoder architecture makes it efficient in extracting the noise from the recorded audio-waveform that runs real-time on a low computational device with minimal time delay.

Keywords: Speech Enhancement, Encoder, Decoder, Deep Learning.

1. INTRODUCTION

One of the most basic types of communication, speech is crucial to many facets of daily life. However, voice signals are frequently distorted by several kinds of noise, making them challenging to comprehend. A number of environmental elements, such as wind and road noise, technological interference, and reverberation, can cause noise in speech signals. The use of voice processing in everything from telecommunications to speech recognition systems has advanced significantly over time. However, the presence of noise in audio signals is one of the major difficulties in speech processing. This noise may come from a variety of sources, including ambient noise, microphone interference, or external influences. The quality and understandability of speech signals can be severely impacted by noise, which lowers the accuracy of speech recognition and other related applications. By eliminating noise from the original signal, speech enhancement techniques strive to increase the quality of voice transmissions. The development of reliable and effective speech augmentation algorithms has attracted a lot of study in recent years. The desire for high-quality speech signals in different applications, including hearing aids, telecommunications, and speech recognition systems, has fueled the development of these algorithms. The effectiveness of our proposed system is evaluated by measuring the signal-to-noise ratio (SNR) and the perceptual evaluation of speech quality (PESQ) for both clean and noisy speech signals. The results demonstrate the proposed system achieves significant improvement in speech quality, particularly in noisy environments.

The proposed speech enhancement system can be applied to a wide range of applications and can provide significant benefits in improving speech signal quality. The results of this study have the potential to contribute to the development of more efficient and robust speech enhancement algorithms in the future.

The presence of noise in speech signals can significantly reduce their quality and intelligibility, thereby limiting their effectiveness in various applications such as telecommunications, hearing aids, and speech recognition systems. The challenge is to develop an effective speech enhancement system that can remove noise from speech signals while preserving their key characteristics such as voice quality, intelligibility, and naturalness. The proposed system should be able to work in real-time, handle various types of noise, and be computationally efficient primarily by eliminating background noise. Since most conversational speech recordings include various forms of noise, like street sounds, dogs barking, or keyboard typing.

Speech enhancement is crucial for audio and video calls [1], hearing aids [2], and even automatic speech recognition (ASR) systems [3]. For many such applications, a crucial feature of a speech enhancement system is its ability to operate in real-time and with minimal delay (online), on communication devices, and ideally on readily available hardware. Previous research in speech enhancement has shown effective techniques that estimate the noise model and use it to recover clear speech [4, 5]. However, these methods still struggle with common noises like non-stationary noise and babble noise that greatly reduce speech intelligibility [6]. In recent years, deep neural network (DNN)-based models have demonstrated significantly better performance in handling these types of noise, resulting in higher quality speech in both objective and subjective evaluations compared to traditional approaches [7, 8]. Furthermore, deep learning methods have also been found to outperform traditional methods for the related task of single-channel source separation [9, 10, 11].

Motivated by recent advancements in speech enhancement using deep neural networks (DNN) and developed a real-time version of the DEMUCS architecture [11] specifically for speech enhancement. Our model is based on convolutions and Long Short-Term Memory (LSTM) networks, has a frame size of 40ms and a stride of 16ms, and can run faster than real-time on a single laptop CPU core. To ensure high-quality audio, a waveform-to-waveform hierarchical generation approach with skip-connections, similar to U-Net [12]. Our model is optimized to directly output a "clean" version of the speech signal by minimizing a regression loss function (L1 loss[13]). Despite the real-time constraint on model runtime, our model's performance is comparable to state-of-the-art models according to both objective and subjective measures.

Multiple metrics have been developed to measure the performance of speech enhancement systems, but studies have shown that these metrics do not necessarily correlate with human evaluations. Therefore, report results using both objective metrics and human evaluation. Additionally analyze the artifacts of the enhancement process by measuring the Word Error Rates (WERs) produced by an Automatic Speech Recognition (ASR) model.

According to the results, proposed method performs similarly to the current state-of-the-art model in all metrics when applied directly on the raw waveform. Additionally, the enhanced speech samples are useful in enhancing the performance of an ASR model in noisy environments.

2. SOFTWARE ARCHITECTURE

Notations and problem setting

Objective is to enhance speech using a single microphone (1) for real-time applications. Work with an audio signal x which consists of clean speech y that has been corrupted by an additive background signal n, such that x = y + n. The length of the input utterances can vary, so T is not a fixed value across samples (2). Aim is to find an enhancement function f that can approximate y, such that f(x) ≈ y. To accomplish this the encoder-decoder architecture (11), which was originally designed for music source separation, and adapt it for causal speech enhancement. Figure 1 provides a visual representation of the model.

Encoder-Decoder Architecture.

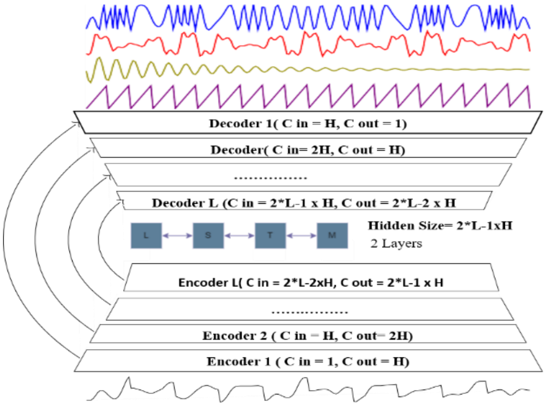
Decoder 1(C in = H, C out = 1)
Decoder(C in= 2H, C out = H)
....
Decoder L (C in = 2*L-1 x H, C out = 2*L-2 x H
L
S
T
M
Hidden Size= 2*L-1xH
2 Layers
Encoder L(C in = 2*L-2xH, C out = 2*L-1 x H
....
Encoder 2 (C in = H, C out= 2H)
Encoder 1 (C in = 1, C out = H)

Fig. 1 Higher-Level Architecture Diagram
Défossez, Alexandre, . "Music source separation in the waveform domain." ***arXiv preprint arXiv:1911.13254*** **(2019).**

The Encoder-Decoder model is composed of a convolutional encoder and decoder with U-net [12] skip connections, as well as a sequence modeling network applied on the encoders' output. It is defined by several parameters: the number of layers L, initial number of hidden channels H, layer kernel size K and stride S, and resampling factor U [1]. The encoder and decoder layers are numbered from 1 to L, with layers at the same scale having the same index [2]. Since we are focusing on monophonic speech enhancement, the input and output of the model contain a single channel. The encoder network E takes the raw waveform as input and produces a latent representation E(x) = z. Each layer of the encoder includes a convolution layer with a kernel size of K and a stride of S. This is followed by a ReLU[15] activation, a "1x1" convolution and a GLU[16] activation that converts the number of channels back. Fig 2 provides a visual description of the process. The final step is to pass the z-hat (1) through the decoder network D, which generates an estimation of the clean signal denoted as D(z-hat) = y-hat. The i-th layer of the decoder takes as input 2

i−1H channels, applies a 1x1 convolution with 2 iH channels, and outputs 2 i−1H channels through a GLU activation function. This is followed by a transposed convolution with a kernel size of 8, stride of 4, and 2i−2H output channels. Finally, a ReLU function is applied, except for the last layer, where the output is a single channel without ReLU. A skip connection is established between the output of the i-th layer of the encoder and the input of the i-th layer of the decoder, as depicted in Figure 2.

DATASET DESCRIPTION

The Valentini[17,18,19] Audio-Visual Dataset for Noise Reduction and Source Separation is a dataset created for the purpose of evaluating algorithms for noise reduction and source separation in real-world scenarios. It was developed by researchers at the University of Trento, Italy. The dataset consists of audio recordings that are corrupted by different types of noise, such as white noise, car noise, and cafe noise. It also includes corresponding clean versions of the recordings and video frames from which the audio was extracted. The dataset was created to provide a benchmark for audio source separation and noise reduction algorithms in real-world scenarios. The dataset contains 20,000 audio files, with a total duration of about 14 hours. The files are divided into two sets: a training set and a testing set. The training set contains 10,000 files and the testing set contains another 10,000 files. Each audio file in the dataset is a 16-bit PCM WAV file with a sampling rate of 48 kHz. The audio files are mono and have a duration of 4 seconds. The audio files are organized into folders according to the type of noise present in the file. The types of noise included in the dataset are white noise, car noise, cafe noise, and street noise.

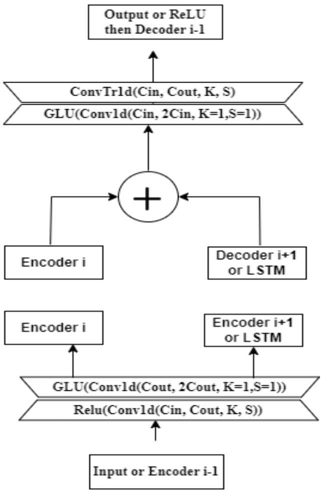
Output or ReLU then Decoder i-1
ConvTr1d(Cin, Cout, K, S)
GLU(Conv1d(Cin, 2Cin, K=1,S=1))
+
Encoder i
Decoder i+1 or LSTM
Encoder i
Encoder i+1 or LSTM
GLU(Conv1d(Cout, 2Cout, K=1,S=1))
Relu(Conv1d(Cin, Cout, K, S))
Input or Encoder i-1

Fig.2 Lower-Level Architecture

Défossez, Alexandre, . "Music source separation in the waveform domain." ***arXiv preprint arXiv:1911.13254*** **(2019).**

Data sets consist of 28 and 56 speakers and are commonly used in speech recognition research. The Valentini data sets will be collected from a publicly available source and loaded into the study using PyAudio, a Python package for working with audio data. The data set will be split into a training and validation set using a 70/30 split. This split ensures that the model is trained on a sufficient amount of data while also having a validation set to evaluate the model's performance. The data set will be preprocessed by converting the audio files into spectrograms, a visual representation of the audio signal. The spectrograms will be used as input to the model.A convolutional neural network (CNN) will be used for this study. The CNN will consist of several convolutional layers followed by fully connected layers. The model will be trained on the training set and evaluated on the validation set.

The data set will be split into batches for training. Each batch will contain a fixed number of spectrograms. The model's performance will be evaluated on the validation set. These metrics will provide an insight into how well the model is performing on unseen data.The results obtained from the model will be analyzed to identify any patterns or insights. The analysis will also include a comparison of the results obtained from the 28 and 56 speaker data sets to identify any differences in performance.

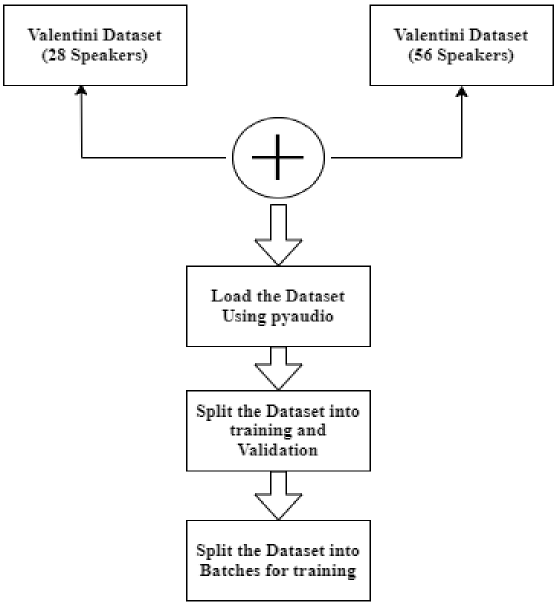
Valentini Dataset (28 Speakers)
Valentini Dataset (56 Speakers)
Load the Dataset Using pyaudio
Split the Dataset into training and Validation
Split the Dataset into Batches for training

Fig 3. Data Pipeline Flowchart

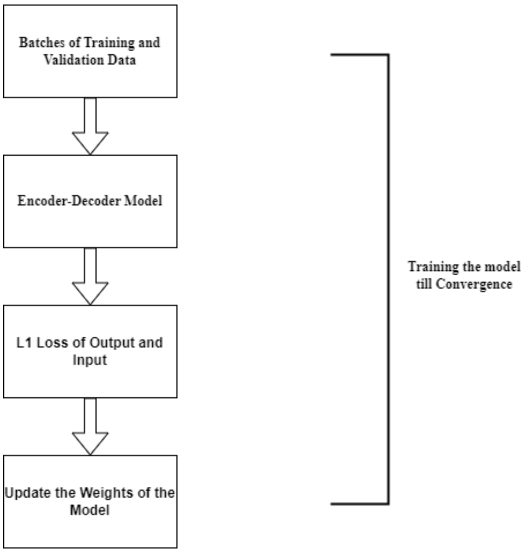
Batches of Training and Validation Data
Encoder-Decoder Model
L1 Loss of Output and Input
Update the Weights of the Model
Training the model till Convergence

Fig 4. Model Training Flowchart

Model training flow chart:

The training and validation data is processed in batches and fed into the encoder-decoder model and the encoder-decoder model is trained using the training data to learn the patterns and relationships between the input and output data, then the model is evaluated using the LI loss function to determine the difference between the predicted output and the actual output. The model weights are updated using backpropagation to minimize the loss function, improving the accuracy of the model and the model's convergence is checked to see if it has reached a satisfactory level of accuracy.

If the model has converged, the training process is stopped, and the final model is used for predictions. If not, the process is repeated until convergence is reached. The model is evaluated on the validation set to determine its performance. If the model's performance is satisfactory, the training process is stopped, and the final model is used for predictions. If not, the process is repeated until the model's performance is satisfactory.

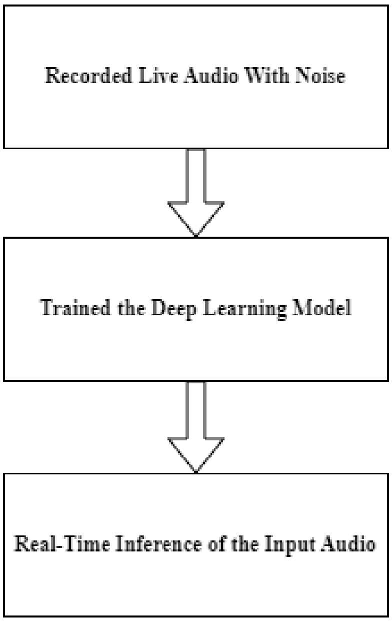
Recorded Live Audio With Noise
Trained the Deep Learning Model
Real-Time Inference of the Input Audio

Fig 5. Model Inference Flowchart

Model Inference Flowchart:

The recorded audio input is preprocessed to reduce noise and prepare it for use with the trained model and then the preprocessed audio input is fed into the trained deep learning model and the trained model uses its learned patterns and relationships to make real-time inferences on the input audio. The output audio is generated by the model and can be used for further analysis, processing, or playback.

RESULTS

The performed experiments with python language along with several libraries such as Numpy, Scipy, torch audio, Sklearn etc. The complete end to end model development code is done using Pytorch framework for model training and inference. Utilizing 16 GPU for faster computation for both during training and inference.

Performed the experiments with multiple input noisy audio signals to check the performance of the model under different conditions. The model has done its significant contribution in eliminating the noise from the input audio waveform. Two test inputs are considered to evaluate the models ability to remove the noise. The amplitude plots of the waveforms to check the content of the noise removal.

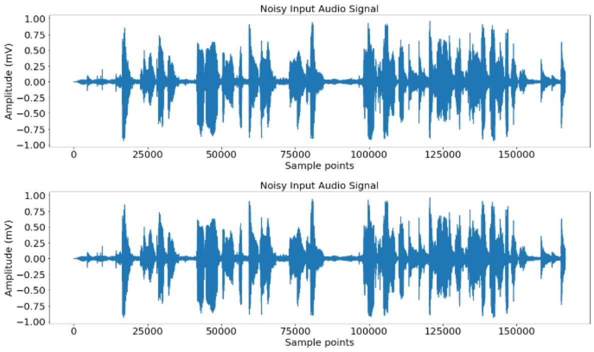

Fig.6 Input noisy audio signal and output clean signal of a test case input 1

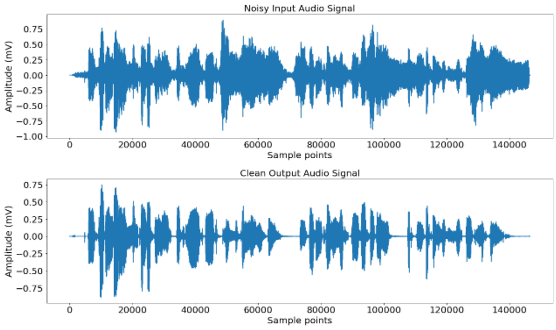

Fig.7 Input noisy audio signal and output clean signal of a test input 2

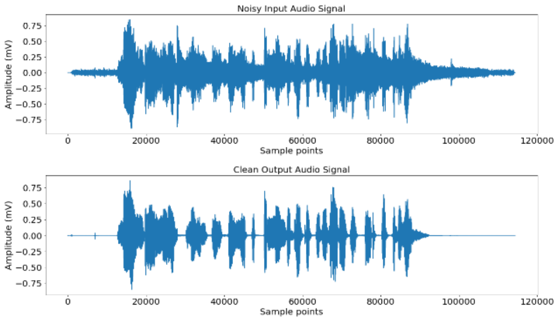

Fig.8 Input noisy audio signal and output clean signal of a test input 3

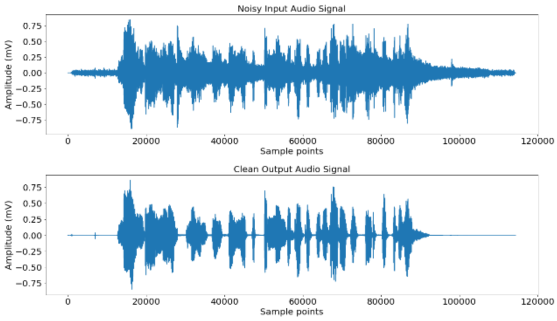

Fig.9 Input noisy audio signal and output clean signal of a test input 4

Figure 6 displays the audio output waveforms before and after the noise suppression. The X-Axis represents the sample points of the audio waveform, whereas the Y-axis displays the Amplitude. The length of the signal is around 10 sec.The given input signal is recorded with the statement “Hi, this is XXXX, Performed Noise Cancellation Experiment” along with recording noise i.e Iphone ringtone. The output clean signal just displays only the text which was given as the input and removes the iphone ringtone. Fig.7 displays the audio output waveforms before and after the noise suppression. The X-Axis represents the sample points of the audio waveform, whereas the Y-axis displays the Amplitude. The length of the signal is around 10 sec. The given input signal is recorded with the statement “Hi, this is XXXX, Performed Noise Cancellation Experiment” along with recording noise i.e baby cry. The output clean signal just plays only the text which was given as the input and removes the background construction noise.

Fig.8 displays the audio output waveforms before and after the noise suppression. The X-Axis represents the sample points of the audio waveform, whereas the Y-axis displays the Amplitude. The length of the signal is around 9 sec. The given input signal is recorded with the statement “Hi, this is XXXX, Performed Noise Cancellation Experiment” along with recording noise i.e calling bell. The output clean signal just plays only the text which was given as the input and removes the background construction noise.

Fig.9 displays the audio output waveforms before and after the noise suppression. The X-Axis represents the sample points of the audio waveform, whereas the Y-axis displays the Amplitude. The length of the signal is around 9 sec. The given input signal is recorded with the statement “Hi, this is XXXX, Performed Noise Cancellation Experiment” along with recording noise i.e mobile ringtone. The output clean signal just plays only the text which was given as the input and removes the background construction noise.

Fig.10 displays the audio output waveforms before and after the noise suppression. The X-Axis represents the sample points of the audio waveform, whereas the Y-axis displays the Amplitude. The length of the signal is around 11 sec.

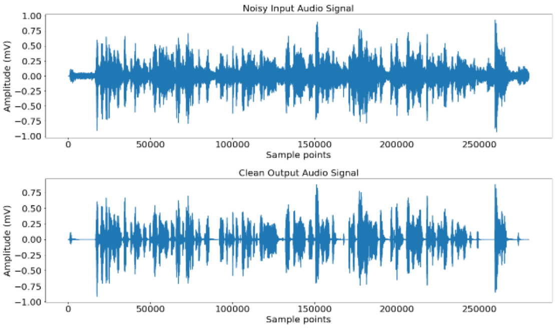

Fig.10 Input noisy audio signal and output clean signal of a test input 5

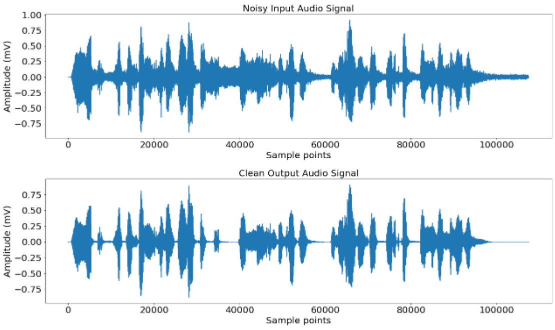

Fig.11 Input noisy audio signal and output clean signal of a test input 6

The given input signal is recorded with the statement “Hi, this is XXXX, Performed Noise Cancellation Experiment” along with recording noise i.e music retro. The output clean signal just plays only the text which was given as the input and removes the background construction noise.

Fig.11 displays the audio output waveforms before and after the noise suppression. The X-Axis represents the sample points of the audio waveform, whereas the Y-axis displays the Amplitude. The length of the signal is around 10 sec The given input signal is recorded with the statement “Hi, this is XXXX, Performed Noise Cancellation Experiment” along with recording noise i.e notifications on mobiles. The output clean signal just plays only the text which was given as the input and removes the background construction noise.

CONCLUSION

The encoder-decoder architecture, which was originally designed for music source separation in the waveform domain, has been successfully repurposed into a causal speech enhancer that can process audio signals in real-time using a consumer-level CPU. Team conducted tests on the standard Valentini [17,18,19] benchmark and achieved state-of-the-art results without needing additional training data. Through empirical testing, we demonstrated that the model is capable of enhancing speech signals that contain noise. Furthermore, found that model can improve the performance of automatic speech recognition (ASR) models in noisy conditions even without requiring retraining of the ASR model.

ACKNOWLEDGEMENT

We would like to extend our thanks to SRM University, for providing us the platform and opportunity to bring our ideas to action.

REFERENCES

1. C. K. Reddy ., “A scalable noisy speech dataset and online subjective test framework,” preprint arXiv:1909.08050, 2019.
2. C. K. A. Reddy ., “An individualized super-gaussian single microphone speech enhancement for hearing aid users with smartphone as an assistive device,” IEEE signal processing letters, vol. 24, no. 11, pp. 1601–1605, 2017.
3. C. Zorila, C. Boeddeker, R. Doddipatla, and R. Haeb-Umbach, “An investigation into the effectiveness of enhancement in asr training and test for chime-5 dinner party transcription,” preprint arXiv:1909.12208, 2019.
4. J. S. Lim and A. V. Oppenheim, “Enhancement and bandwidth compression of noisy speech,” Proceedings of the IEEE, vol. 67, no. 12, pp. 1586–1604, 1979.
5. Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” IEEE Transactions on acoustics, speech, and signal processing, vol. 32, no. 6, pp. 1109–1121, 1984.
6. N. Krishnamurthy and J. H. Hansen, “Babble noise: modeling, analysis, and applications,” IEEE transactions on audio, speech, and language processing, vol. 17, no. 7, pp. 1394–1407, 2009.
7. S. Pascual, A. Bonafonte, and J. Serra, “Segan: Speech enhancement generative adversarial network,” preprint arXiv:1703.09452, 2017.
8. H. Phan ., “Improving gans for speech enhancement,” preprint arXiv:2001.05532, 2020.
9. Y. Luo and N. Mesgarani, “Conv-TASnet: Surpassing ideal time– frequency magnitude masking for speech separation,” IEEE/ACM transactions on audio, speech, and language processing, vol. 27, no. 8, pp. 1256–1266, 2019.
10. E. Nachmani, Y. Adi, and L. Wolf, “Voice separation with an unknown number of multiple speakers,” arXiv:2003.01531, 2020.
11. A. Dfossez., “Music source separation in the waveform domain,” 2019, preprint arXiv:1911.13254.
12. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention, 2015.
13. R. Yamamoto, E. Song, and J.-M. Kim, “Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” preprint arXiv:1910.11480, 2019.

14. Yamamoto, Ryuichi, Eunwoo Song, and Jae-Min Kim. "Probability density distillation with generative adversarial networks for high-quality parallel waveform generation." *arXiv preprint arXiv:1904.04472* (2019).
15. Y. N. Dauphin., "Language modeling with gated convolutional networks," in ICML, 2017.
16. K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in ICCV, 2015.
17. J. Smith and P. Gossett, "A flexible sampling-rate conversion method," in ICASSP, vol. 9. IEEE, 1984, pp. 112–115.
18. C. Valentini-Botinhao, "Noisy speech database for training speech enhancement algorithms and tts models," 2017.
19. C. K. A. Reddy ., "The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework," 2020