Speech Enhancement using Deep Convolutional Neural Network

Student: Tran Anh Quan

Instructor: Bui Thi Loan

Bachelor of Computer Science
Hoa Lac Campus - FPT University
4 May 2021

Table of Contents

Abstract

This work presents a supervised speech enhancement method using a deep convolutional neural network (CNN). The proposed CNN is based on a Convolutional Autoencoder architecture with symmetric skip-connections. Additionally, we focus on building a novel and robust dataset for this task. The data contains a clean speech dataset and a noise dataset, and each outweighs its counterpart used in recent works. Finally, we investigate the performance of the system on many levels of noise by performing the evaluation using objective metrics that are commonly used in this area.

Index terms: Speech enhancement, Speech denoising, Convolutional Neural Network, Convolutional Autoencoder, Unet, Deep learning.

Acknowledgements

I would like to thank my instructor, Ms. Bui Thi Loan, as well as teacher Phan Duy Hung, who had instructed me enthusiastically throughout the research process, as well as given me numerous precious advice for improvement.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

There is an increasing need for ways to enhance the quality of speech. The task of speech enhancement is to improve the perceptual quality of speech signals, especially by suppressing background noises. Noises from the environment, animals, machines, etc, exist in the majority of verbal recordings. Hence, speech enhancement plays an important role in audio-related forms of communication like voice calls, helping people with hearing loss, or serving as the preliminary for speech recognition systems. With the rising computing strength of hardware such as GPU, a trend of using deep neural networks (DNNs) rather than traditional approaches to denoise speech has emerged. There has been an increasing attention in DNN-based speech denoising solutions in recent years.

Pioneers in this area have emerged over a decade ago. For example, in [1], the authors use a very simple neural network model (compared to contemporary models) with only one hidden layer to get rid of reverberation. Another thesis also released in 2009 is [2], which included a simple model called ADALINE, with some hidden layers. As this area gains more attention, the models have increased dramatically in both size and complexity. Researchers have tried many ways to raise performance, such as applying Bayesian Wavenet [3], using multiple deep neural networks [4], or resorting to hybrid approaches [5]. Over time, approaches are now converging to some common methods:

- Using Recurrent Neural Network (RNN): This straightforward approach is effective when the data is chain-like. RNN is difficult to train in general because the gradient of RNN vanishes or explodes at an exponential rate when backpropagation to the same layer is repeated. The notorious vanishing/exploding gradient problem is always concerning researchers using RNN [6]. The most commonly used among the methods which have been proposed as a solution [7–9] is long short-term memory (LSTM) [9]. LSTM partially solves the vanishing gradient problem by integrating three gated units (input gate, forget gate, and output gate). LSTM and the bidirectional LSTM (BLSTM) have been applied to speech denoising and worked better than the traditional approaches at the time because they can be trained efficiently in practice [10–13]. Nevertheless, the reduction of the vanishing gradient problem when using LSTM is attached to the high computational cost as it has a large number of parameters. Thus, if the gradient problem can be solved in a different way, a simpler RNN should be more suitable than LSTM, particularly for real-time speech enhancement [14].

- Using Convolutional Neural Network (CNN): This type of model is effective when the data is image-like, or matrix-like. Because of their weight sharing property, Convolutional Neural Networks (CNN) generally have fewer parameters than FNNs and RNNs. CNNs have already demonstrated their effectiveness in extracting features in speech recognition [15, 16] and removing noise from images [17,18]. Recent research suggests that a convolutional neural network (CNN) may be used as a convolutional denoising autoencoder (CDAE). This type of DNN architecture is most commonly used in image classification and feature detection, where it outperformed all other methods [19].

- Using Convolutional Recurrent Neural Network (CRNN): Recently, researchers started to apply combinations of these two models to utilize the advantage of both. The common mixture is the convolutional encoder and decoder part placed at both the beginning and the end of the model, handling the spectrogram data, while the RNN, which tend to be LSTM chains, is at the middle to extract the characteristic of 'chain like' data. This model structure is getting impressive performance, some of them are even designed for real-time speech enhancement with high accuracy [20].

With two out of three main approaches having convolutional layers at the beginning of the model, it is reasonable that researchers recently tend to use spectrogram data, rather than signals in the time domain as the input of the DNNs. Each work has its variation, but in general, the flow is standardized: The researchers convert their audio data to spectrogram form and then feed the model, use the model to denoise this spectrogram to obtain the 'clean' spectrogram, and eventually convert it back to time-domain waveforms [20–25].
Respecting the task of data pre-processing, the common procedures are to collect clean voice data and noise data, blend them to make a noisy voice dataset for training neural networks, and use the original clean voice to evaluate the predicted clean voice [20–26].
There are still multiple headaches for researchers in this area. One of the biggest hindrances is data shortage. Compared to other types of data like image or text, audio data is much harder to collect, refine, and examine in general. Therefore the number of reliable audio datasets is rather limited. Besides, almost all present datasets contain only one language in their corpus, and models trained on these datasets are not promised to perform well on speech in other languages [23].

1.2 Idea and motivation

The rising tide of using the Convolutional Encoder-Decoder and its variations has proved the efficiency of this model [22–24]. This type of model needs the input to be in image form, so we have to convert our original audio data - time-domain waveform - to data in image form. In this thesis, we follow the common pattern, which feeds the STFT spectrogram of the noisy speech to the model, predict a spectrogram version of the clean voice, and eventually convert it back to waveform as the final product. In this approach, we predict the noise spectra instead of directly predicting the clean voice spectra. The reconstruction of noise, which normally makes up a minority of a record, would be easier to execute than rebuilding the clean voice. Therefore we set the noise spectra as the output of our model, then use itself and the original noisy voice spectra to get the clean voice spectra by the simple subtraction.

There is also a motif concerning data collecting and pre-processing among recent works in this area. That is to take a dataset of clean speech and noise, then blend them to get the noisy speech, then use the noisy speech for the input and compare the output with the original clean speech. Combined with some data augmentation techniques, a small dataset can turn into an acceptably large collection of speech signals for training and testing. However, a small initial dataset would still be an obstacle for works in the present and future to reach the stage of production.

About clean speech, there are a bunch of datasets used. In [2], the authors use RSR2015, which contains 71 hours of clean speech spoken by 300 speakers. In [5], [21], and [22] the TIMIT database is used for clean speech, which was released in 1988 with 52 hours and 630 speakers. In [27], [28], the authors use IEEE corpus for clean speech, which contains only about 1 hour of audio material. In [20], [24] and [29], the data for clean speech is the Wall Street Journal dataset (WSJ0), which contains 43 hours of speech recorded by 91 speakers. In a world that is home to nearly 8 billion people, these datasets still seem deficient.

About the noise, reliable sources of data are rather scarce. In [4], [27], [28], the authors use NOISEX, NOIZEUS, and AURORA respectively. Each of these datasets contains about 10 classes of ambient noises, which is not enough for production in the real world with uncountable sources of environmental sound.

To enhance the performance of a speech enhancement system, many researchers choose to apply a more complicated model such as the mixture of CNN and RNN [20, 21, 26]. But the increase of layers, parameters put more computing and memory strain on the system, making it hard to install it on light devices such as smartphones or other gadgets in the 4.0 ecosystem. Therefore, we decide to use a common model rather than invest the effort in improving it, as well as common methods of collecting and preprocessing data. Instead, we get more data in both clean speech and noise, examine, refine, and thoroughly preprocess them so that they help our model achieve state-of-the-art performance.

1.3 Related Works

- A fully convolutional neural network for speech enhancement (Park & Lee, 2016) [22]:

  The model proposed in this thesis is an alternative convolutional network architecture - Redundant Convolutional Encoder-Decoder (R-CED). The difference between it and other common Convolutional Encoder-Decoders lies in getting rid of the pooling and upsampling layers. It still reserves the symmetric characteristics, but in contradiction to CED, R-CED encodes the features into higher dimensions alongside the encoder and achieves compression alongside the decoder.

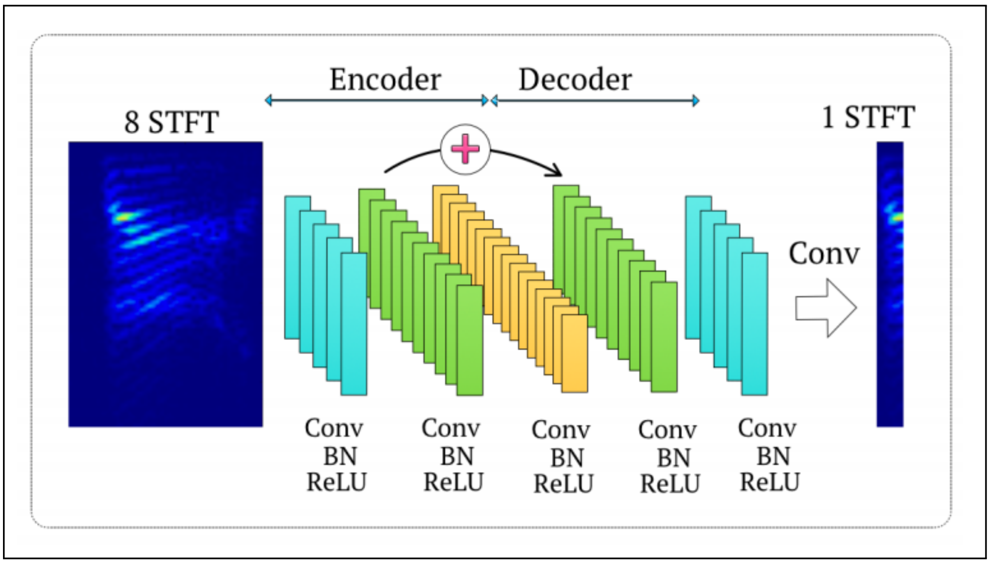


Figure 1.1: R-CED

- A Convolutional Recurrent Neural Network for Real-Time Speech Enhancement (Tan & Wang, 2018) [20]:

  The model is a typical combination of CNNs and RNNs. By integrating the two topologies, the proposed CRN can utilize both the ability to learn features of CNNs and the ability to model transient dependencies of RNNs.

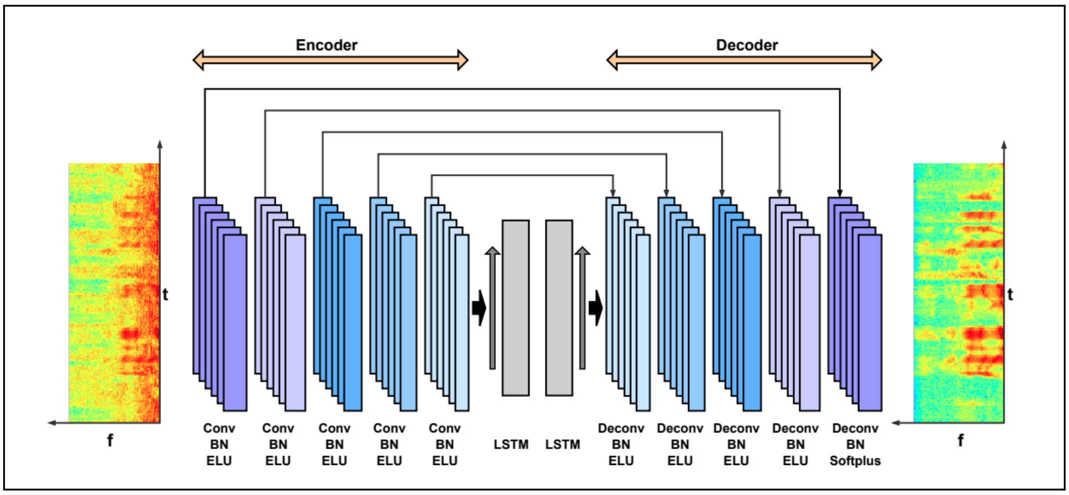


Figure 1.2: CRN

- TCNN: Temporal Convolutional Neural Network for Real-time Speech Enhancement in the Time Domain (Pandey & Wang, 2019) [24]:

The figure below demonstrates the architecture of the proposed TCNN. The encoder, decoder, and TCM are the three main components of the model. While the encoder and the decoder largely resemble their counterparts in other works, the TCM comprises dilated causal convolutional layers which only have one dimension.

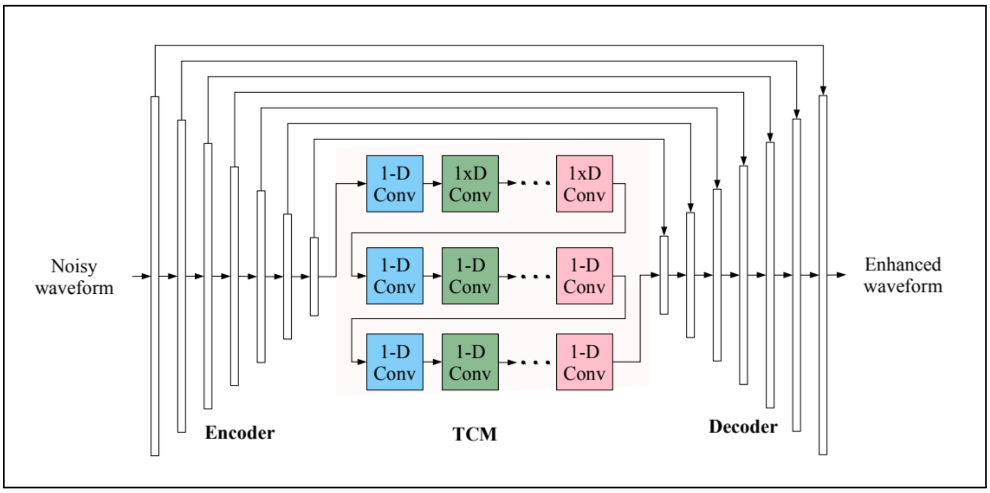


Figure 1.3: TCNN

- Speech enhancement using progressive learning-based convolutional recurrent neural network (Li et al., 2020) [21]:

  Figure 1 represents the pipeline of PL-CRNN. Instead of using just one CRNN, the framework consists of multiple CRNNs with a modest size compared to a typical CRNN, each of them serving as a subnet. The explanation for using smaller CRNNs is that the model will compensate for performance differences when a simpler network is used instead of a network with more parameters.

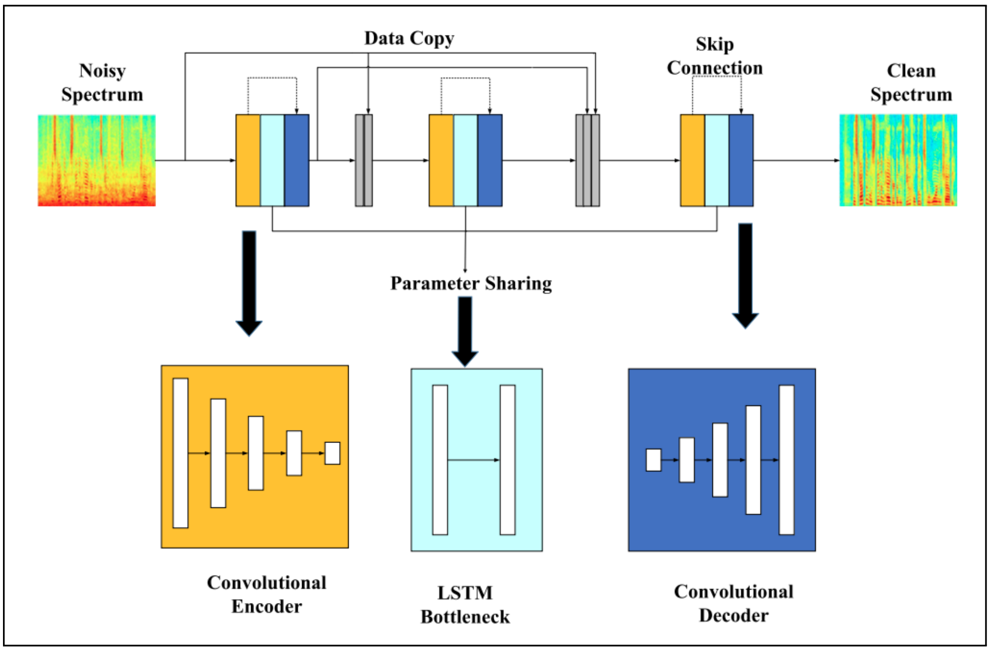


Figure 1.4: PL-CRNN

- Real Time Speech Enhancement in the Waveform Domain (Defossez et al., 2020) [26]:

  The framework of DEMUCS is quite similar to other CRNNs. More specifically, it has the LSTM serving as the bottleneck between convolutional encoder and decoder. The novel element of this work is it uses raw waveform as the input to the model rather than spectrogram.

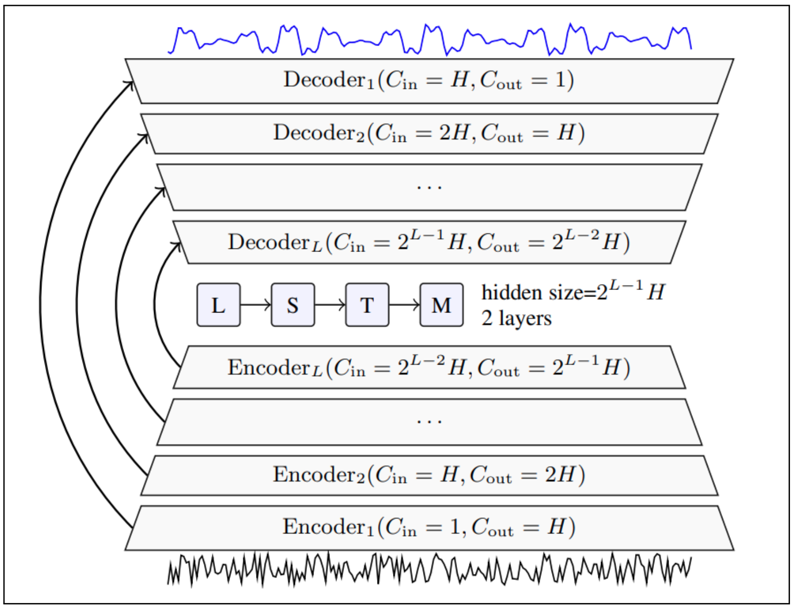

Decoder$_1$($C_{\text{in}} = H, C_{\text{out}} = 1$)
Decoder$_2$($C_{\text{in}} = 2H, C_{\text{out}} = H$)
...
Decoder$_L$($C_{\text{in}} = 2^{L-1}H, C_{\text{out}} = 2^{L-2}H$)
L
S
T
M
hidden size=$2^{L-1}H$
2 layers
Encoder$_L$($C_{\text{in}} = 2^{L-2}H, C_{\text{out}} = 2^{L-1}H$)
...
Encoder$_2$($C_{\text{in}} = H, C_{\text{out}} = 2H$)
Encoder$_1$($C_{\text{in}} = 1, C_{\text{out}} = H$)

Figure 1.5: DEMUCS

From simple models [1,2], many complicated networks have been proposed over a decade and proved their efficiency. However, as aforementioned, the data these works use are quite limited. Hence, the main contribution of this work is a new robust dataset with a wide range of voice and noise which were carefully chosen and processed. The consequence of using large datasets is a remarkably long time of training, but it proved that it is worth the effort.

1.4 Contribution

In this thesis we propose a robust dataset designed for speech enhancement, using a popular, common structure of neural networks. We create the dataset by taking one of the largest and most diverse contemporary dataset for people voice - Librispeech as a clean voice source, merge noise datasets together (namely ESC-50 and UrbanSound) to create a big set of noises. Then we blend the clean voice with noise at different signal-to-noise ratios (SNRs), which returns noisy voice records. We use an end-to-end system with the model taking noisy voice as input and noise as output, then obtain the clean voice by subtracting the noise from the noisy voice.

Chapter 2

Data collecting and preprocessing

2.1 Data collecting

It is reasonable that recording noisy speech and clean speech independently (with the condition that the speeches in the noisy records match the clean records exactly) is uneconomical and extremely time-consuming. Therefore, until now, following the common pattern of collecting and preprocessing data, which means to get clean voice data and noise data separately, then blend them, is still the most realistic and versatile methodology.

Here are the clean speech datasets we use for experiments, all of which are taken from LibriSpeech [30]. After preprocessing (blended with noise), dev-clean would be treated as validation (development) set, test-clean as test set, and train-clean-100 as the training sets:

Table 2.1: Clean voice data

subset	hours	per-spkr minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
train-clean-100	100.6	25	125	126	251

The dev-clean and test-clean datasets are carefully examined and refined by the authors of LibriSpeech so that they have similar data distribution. Thus, using the dev-clean as the evaluation set while training the model assures the analogous performance on the test-clean dataset as the testing set. This helps us investigate the working of the DNNs more conveniently and accurately. Compared to other works which choose the testing set as a part of a big training set and have no genuine evaluation set [21–23], it is a huge improvement in model evaluation.

Table 2.1 also shows the big difference in size between the proposed dataset and other familiar ones. The total duration of records in this dataset is over 110 hours, while the numbers of RSR2015 [31], TIMIT [32], and WSJ0 [33], are 71, 52, and 43 respectively.

The noise dataset is formed by two parts: one for mixing with the training set, and one for blending with the validation and testing set. The former is the entire ESC-50 [34], with approximately 2.5 hours of noise, and contains 50 different classes. Table 2.2 reveals the diversity of the dataset. The noises vary from environmental to domestic sounds, from artificial to natural sounds, and from low, smooth to loud, fierce sounds.

Table 2.2: Classes in noise dataset

Animals	Natural soundscapes and water sounds	Human, non-speech sounds	Interior/ domestic sounds	Exterior/ urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

The latter is taken from fold 1 of the UrbanSound dataset [35] (this dataset is divided into 10 folds), which contains 10 different types of noises: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The duration of this part is nearly equal to the former.

2.2 Data preprocessing

Speech records are first concatenated into a long vector. The short samples are taken by continuing shifting the cutting point forward a particular number of frames (hop-length-frame) to obtain the noise-only signal with a fixed frame length, which are subsequently mixed with a randomly chosen short sample of noise. Then each sample of clean voice is blended with a random sample of noise with the formula:

$$x = s + \alpha \times n \tag{2.1}$$

where x is the noisy voice, s is the speech, n is the noise, and alpha is a number that is calculated from a variable call SNR_dB (signal to noise ratio (SNR), measured in decibels (dB)) with the formula:

$$\alpha = 10^{SNR\_dB/10} \tag{2.2}$$

To measure the efficiency of the system on different levels of noise, we examine it with different levels of SNR, from -10dB (loudest noise) to 15dB (lowest noise), which means the α is chosen from a set of fixed values.

The short samples of noise are equal in length to those of speech and are obtained with similar techniques. After blending the utterances with noise, a complete dataset of noisy-clean pairs of short samples is created. Then we transform them to spectra in the time-frequency domain using Short-time Fourier

Transform (STFT). Eventually, the data are rescaled and shaped to serve as the input to our model.

Chapter 3

Methodology

3.1 Problem formulation

The objective of the system is to reconstruct the target clean speech signal s_t from an input signal x_t, which is contaminated by noise n_t:

$$x_t = s_t + n_t \tag{3.1}$$

where x_t, s_t, n_t are time-domain waveforms with t as the time index. In STFT domain, this formula equals to:

$$X_{\omega,\tau} = S_{\omega,\tau} + N_{\omega,\tau} \tag{3.2}$$

where $X_{\omega,\tau}$, $S_{\omega,\tau}$, $N_{\omega,\tau}$, are the spectrograms of the noisy speech, clean speech, and noise, respectively, generated by STFT. ω denotes the frequency bin, and τ denotes the time frame.
Figure 3.1 clearly demonstrates the entire process.

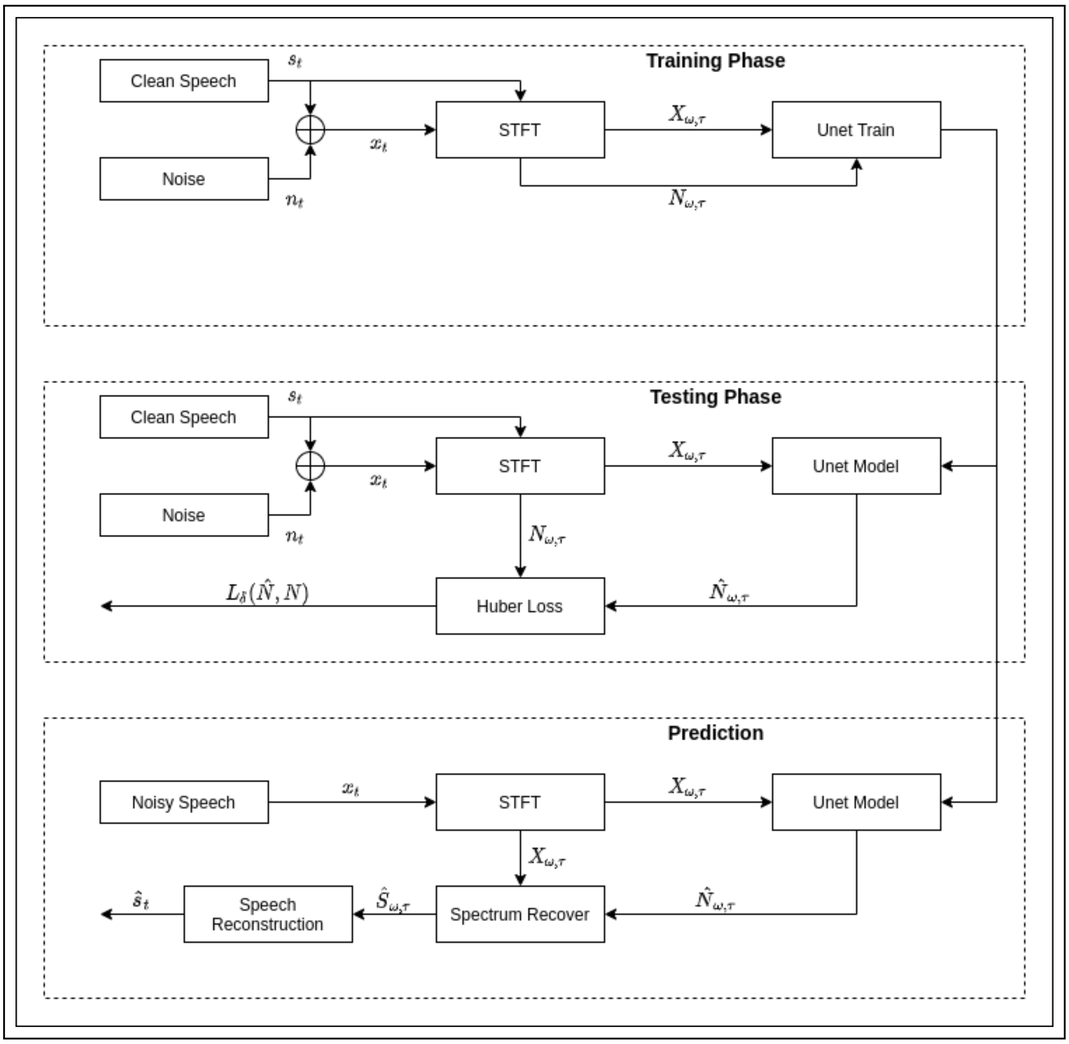


Figure 3.1: System pipeline

About the output of the model, there are two directions that researchers normally follow: predict the clean speech spectra $\hat{S}_{\omega,\tau}$ directly [22], or predict the mask $G_{\omega,\tau}$ in time-frequency domain and recover the clean speech by calculating the element-wise multiplication of G and X [14, 25]. In this work, we apply the former, but choose to predict the noise spectra $\hat{N}_{\omega,\tau}$ rather than $\hat{S}_{\omega,\tau}$, as the noise tends to occupy a lesser amount in an audio record, making the work of predicting itself tend to be easier. The estimated noise spectra is used to obtain the clean speech spectra by the subtraction:

$$\hat{S}_{\omega,\tau} = X_{\omega,\tau} - \hat{N}_{\omega,\tau} \tag{3.3}$$

Eventually the clean speech spectra is scaled and converted back to time-domain waveforms. The ultimate product of the system is the predicted clean speech $\hat{s}_t$.

3.2 Model

The proposed deep CNNs, called Unet, is a Convolutional Encoder-Decoder (CED) architecture, which is characterized by symmetric skip concatenation between the Encoder and Decoder. In the beginning, this architecture was used in Bio-Medical Image Segmentation [36].

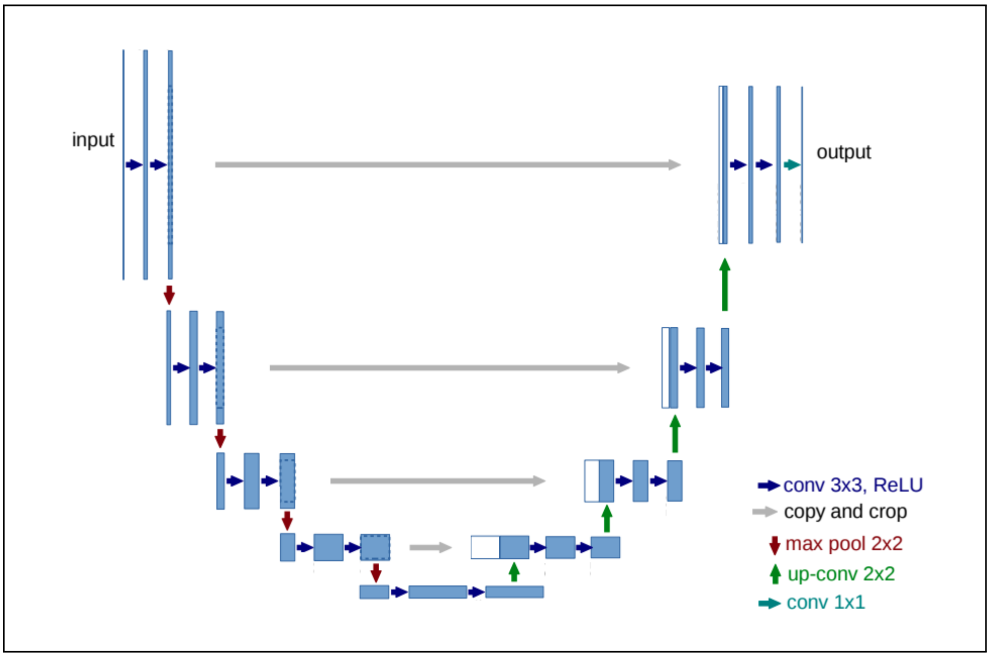


Figure 3.2: Pipeline of the Unet

Figure 3.2 and Table 3.1 show the framework of the Unet. The Encoder comprises repetitious chains of a convolution, batch normalization, max-pooling, and a RELU activation layer. It compresses the features through its length. Compared to the Encoder, the Decoder has an opposite pipeline consisting of the upsampling layers rather than max-pooling ones, and an opposite task of decompressing the features. Together, they form a pair of contracting - expansive paths.

In the Encoder (contracting path), after each pair of 3x3 unpadded convolutions is a rectified linear unit (ReLU) followed by a max-pooling layer with the kernel size of 2x2 and stride 2. After this downsampling operation, the feature channels are doubled in their number. The Decoder (expansive path) consists of repeated application of an upsampling step, a 2x2 ”up-convolution” where the number of feature channels is halved, a skip-connection, and two 3x3 unpadded convolutions that each is followed by a RELU. The skip-connections copy and crop the feature map from the encoder to the corresponding layers in the decoder in order to retain valuable information which may be lost alongside the encoder. A 1x1 convolutional layer with the Hyperbolic tangent activation (tanh) is placed at the end of the model to retrieve the desired noise spectra in the scale of [-1, 1]. Overall, the whole pipeline of Unet has a total of 23 convolutional layers.

Table 3.1: Architecture of the Unet.

(For clarity and simplicity, we omit the ReLU, maxpooling and upsampling layers)

layer name	output shape	hyperparameters	connected to
input_1	(None, 128, 128, 1)	_	_
conv2d	(None, 128, 128, 16)	160	input_1
conv2d_1	(None, 128, 128, 16)	2320	conv_2d
conv2d_2	(None, 64, 64, 32)	4640	conv2d_1
conv2d_3	(None, 64, 64, 32)	9248	conv2d_2
conv2d_4	(None, 32, 32, 64)	18496	conv2d_3
conv2d_5	(None, 32, 32, 64)	36928	conv2d_4
conv2d_6	(None, 16, 16, 128)	73856	conv2d_5
conv2d_7	(None, 16, 16, 128)	147584	conv2d_6
conv2d_8	(None, 8, 8, 256)	295168	conv2d_7
conv2d_9	(None, 8, 8, 256)	590080	conv2d_7
conv2d_10	(None, 16, 16, 128)	131200	conv2d_9
concatenate	(None, 16, 16, 256)	_	conv2d_7, conv2d_10
conv2d_11	(None, 16, 16, 128)	295040	concatenate
conv2d_12	(None, 16, 16, 128)	147584	conv2d_11
conv2d_13	(None, 32, 32, 64)	32832	conv2d_12
concatenate_1	(None, 32, 32, 128)	_	conv2d_5, conv2d_13
conv2d_14	(None, 32, 32, 64)	73792	concatenate_1
conv2d_15	(None, 32, 32, 64)	36928	conv2d_14
conv2d_16	(None, 64, 64, 32)	8224	conv2d_15
concatenate_2	(None, 64, 64, 64)	_	conv2d_3, conv2d_16
conv2d_17	(None, 64, 64, 32)	18464	concatenate_2
conv2d_18	(None, 64, 64, 32)	9248	conv2d_17
conv2d_19	(None, 128, 128, 16)	2064	conv2d_18
concatenate_3	(None, 128, 128, 32)	_	conv2d_1, conv2d_19
conv2d_20	(None, 128, 128, 16)	4624	concatenate_3
conv2d_21	(None, 128, 128, 16)	2320	conv2d_20
conv2d_22	(None, 128, 128, 2)	290	conv2d_21
conv2d_23	(None, 128, 128, 1)	3	conv2d_22
total parameters		1,941,093	

3.3 Objective function

Given a segment of noisy spectra X and clean spectra S, the objective is to directly estimate the noise spectra $\hat{N}$ that approximates the observed noise spectra N. We use the Huber loss [37] as a compromise between L1 and L2 loss to evaluate:

$$L_{\delta}(\hat{N},N) = \begin{cases} \frac{1}{2}(\hat{N}-N)^2 & \text{for } |\hat{N}-N| \leq \delta, \\ \delta|\hat{N}-N| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{3.4}$$

with

$$N = X - S \tag{3.5}$$

where $\hat{N}$ and N denote predicted noise and evaluation noise spectra respectively, and $\delta = 1$.

Chapter 4

Experiments

In order to examine the effectiveness of the proposed method, the performance of speech enhancement was investigated by training and evaluating the model on 6 different datasets, all of them use the same clean voice dataset and noise dataset, but each blend the two datasets at its own SNR level. Finally, we compared the evaluation results between unprocessed noisy speeches and de-noised speeches.

4.1 Datasets

With each level of SNR, the total duration of audio samples is over 100 hours with the training set, and over 5 hours with the validation set and the testing set. All the audio data are sampled at 8kHz, or 8000 frames in 1 second, and are broken into short samples with a duration of slightly above 1 second (8064 frames). In more detail, after preprocessing, there are 344909 short samples for training, 17872 for validation, and 17980 for testing. About the noise datasets, the combination of ESC-50 and UrbanSound is 5 hours in length. The noises from ESC-50 are used for training, while the rest are used for validation and testing, which means the task of the model is to predict unseen noises to the best extent possible.

4.2 Experiment settings

All the experiments are performed on the environment of Google Colab and Tensorflow, using the hardware of GPU NVIDIA Tesla K80. The audio data is converted to spectra by using the Short-time Fourier Transform (STFT) with 256 points Hann window and 64 points window shift. We set the batch size to 64 and set the learning rate to 0.001. We train the model with 10 epochs with the Adam optimizer [38].

4.3 Evaluation

To assess the quality of model predictions, we use two evaluation metrics: STOI (Short-Time Objective Intelligibility) [39] and PESQ (Perceptual Evaluation of Speech Quality) [40]. While Table 4.1 and 4.2 present STOI and PESQ scores, respectively, of unprocessed and processed signals, Figure 4.1 describes the before-after improvements in both the evaluation scores on each level of SNR:

Table 4.1: STOI measure results

Evaluation metrics	STOI(%)						
SNR	-10	-5	0	5	10	15	Avg.
unprocessed	61	72.5	79.3	87.7	91.6	94.6	81.1
train-clean-100	73.8	83.1	87.8	91.8	94.7	96.2	87.9

Table 4.2: PESQ measure results

Evaluation metrics	PESQ						
SNR	-10	-5	0	5	10	15	Avg.
unprocessed	1.68	2.07	2.32	2.72	2.96	3.22	2.50
train-clean-100	2.27	2.68	2.91	3.22	3.35	3.69	3.02

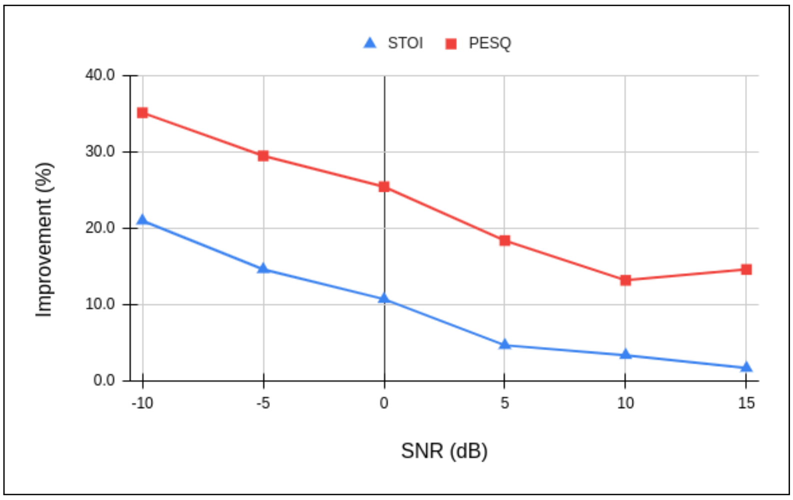
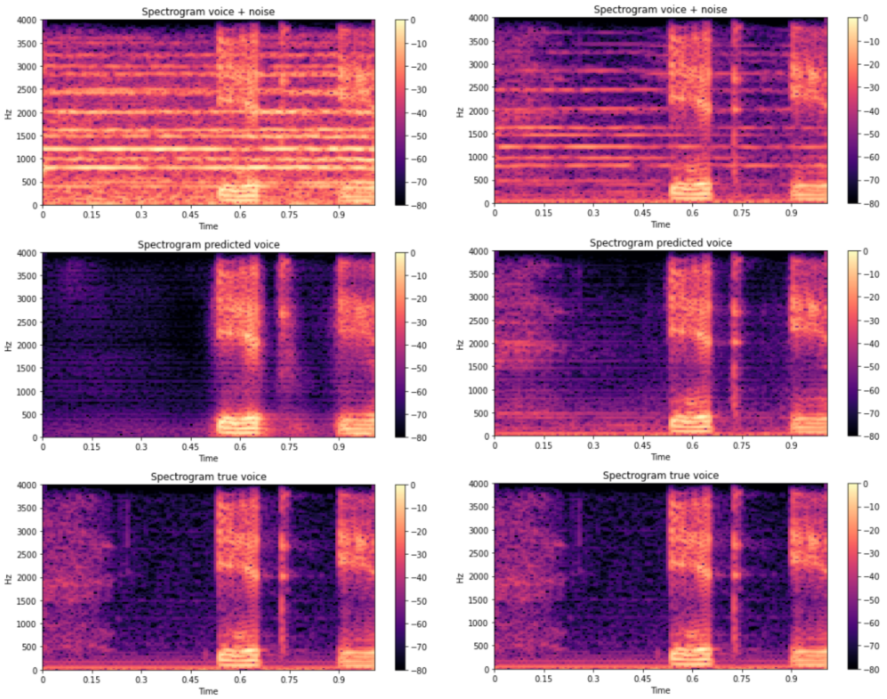


Figure 4.1: Improvements in Evaluation Scores

It is evident that the improvements peak at the case of loudest noises (lowest SNR) and slowly decrease when the noises get fainter. In the most challenging condition, where the utterances are blended with the noise at the SNR level of -10 dB, compared to the unprocessed speech, the processed speech has a gain of 21,0% and 35,1% in STOI score and PESQ score respectively. Meanwhile, the improvements in the case of SNR level of 15 dB are insignificant, with only 1,7% for STOI score and 14,6% for PESQ score.

4.4 Results

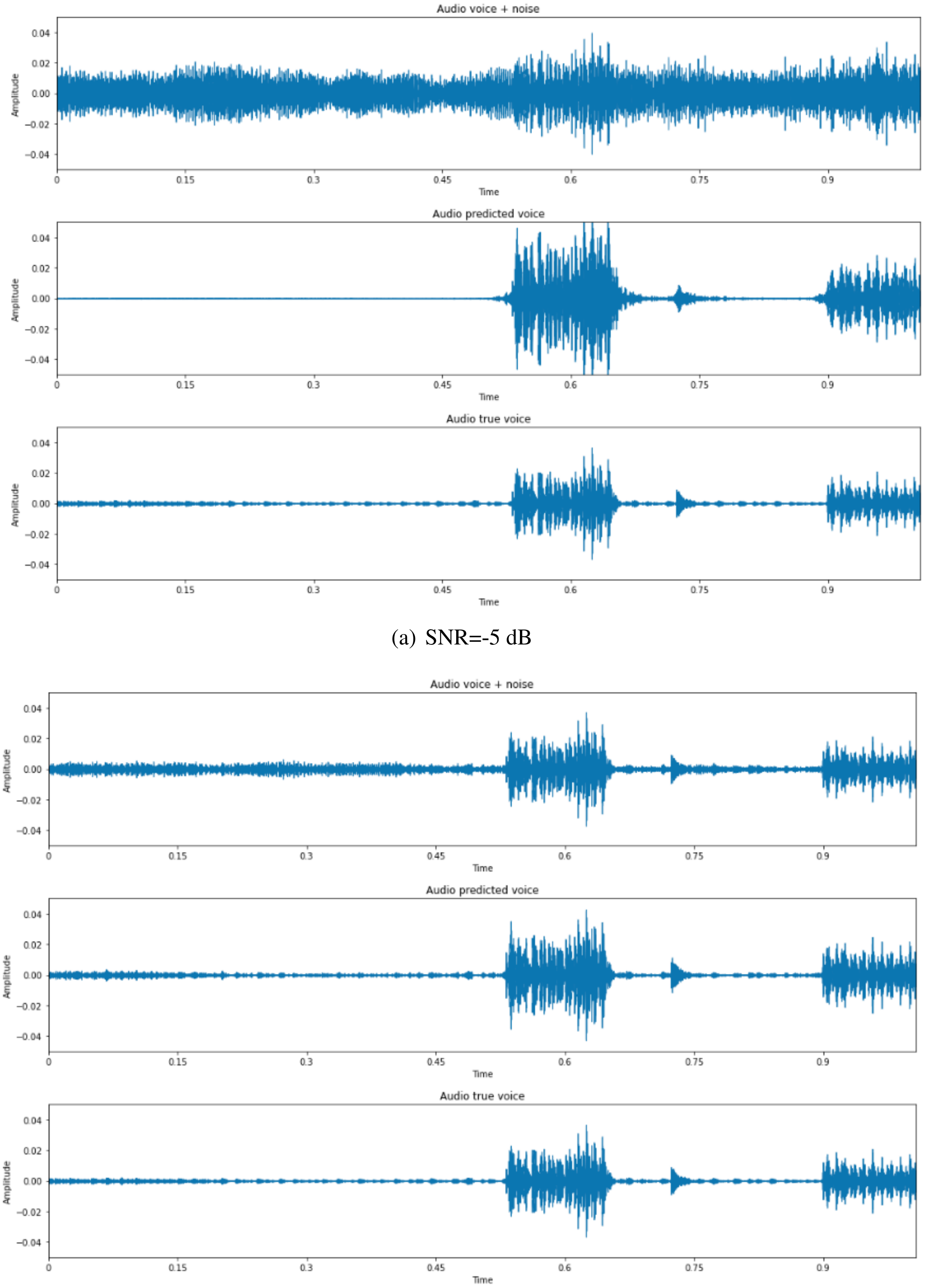


(a) SNR=-5 dB

(b) SNR=10 dB

Figure 4.2: Spectrograms with different levels of SNR

Looking at Figure 4.2 and Figure 4.3, it can be easily observed that the model tends to over-suppress the background noises if the noises are too loud, while the noises that happen concurrently with the speech are not removed radically.



(a) SNR=-5 dB

(b) SNR=10 dB

Figure 4.3: Waveforms with different levels of SNR

Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, we aimed to find a memory-efficient speech denoising method that can be implemented in an embedded system. Inspired by the precedented success of the Convolutional Encoder-Decoder, we hypothesized that this model architecture can more effectively work in real-life applications simply by providing it more data. We set up an experiment to denoise human speech from babble noise which is a major discomfort to people in modern worlds, especially ones using hearing aids, making film or videos, doing phone calls, listening to music, or using speech recognition. In the end, we observed that with the significantly large training set, the model can recover the targeted speech with high accuracy.

5.2 Future Works

There are still various unsatisfying points in this work. The training time is rather long, with 45 minutes to 1 hour to train 1 epoch of the entire train-clean-100. This is partly because the model is still large, with nearly 2 million trainable parameters. Secondly, our conditions do not allow us to train the model with more data, especially with the clean speech corpus of LibriSpeech train-clean-360, which contains up to over 360 hours of audio, with the number of speakers is 921. This is a huge obstacle that prevents the model from additional performance gains. In addition, the objective function of Huber loss seems to be somewhat incompetent in this task.

With the aforementioned shortcomings, there are multiple directions for us to improve the system further. To reduce the size of the model, we can remove some CNN layers. To raise the performance of the model, we can add some LSTMs in the middle between the encoder and decoder. Besides, the present amount of data is already acceptable, however, we can enrich the data by applying some augmentation steps with both the speech and noise dataset. In the future, with these problems being solved, this model would become more reliable to be used in real-time speech enhancement.

References

[1] Bisrat Derebssa Dufera and Tetsuya Shimamura. Reverberated speech enhancement using neural networks. In *2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 441–444. IEEE, 2009.

[2] Khaled Daqrouq, Ibrahim N Abu-Isbeih, and Mikhled Alfauri. Speech signal enhancement using neural network and wavelet transform. In *2009 6th International Multi-Conference on Systems, Signals and Devices*, pages 1–6. IEEE, 2009.

[3] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, Dinei Florêncio, and Mark Hasegawa-Johnson. Speech enhancement using bayesian wavenet. In *Interspeech*, pages 2013–2017, 2017.

[4] Pavan Karjol, M Ajay Kumar, and Prasanta Kumar Ghosh. Speech enhancement using multiple deep neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5049–5052. IEEE, 2018.

[5] Jean-Marc Valin. A hybrid dsp/deep learning approach to real-time fullband speech enhancement. In *2018 IEEE 20th international workshop on multimedia signal processing (MMSP)*, pages 1–5. IEEE, 2018.

[6] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[7] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR, 2016.

[8] Scott Wisdom, Thomas Powers, John R Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. *arXiv preprint arXiv:1611.00035*, 2016.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 708–712. IEEE, 2015.

[11] Han Zhao, Shuayb Zarar, Ivan Tashev, and Chin-Hui Lee. Convolutional-recurrent neural networks for speech enhancement. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2401–2405. IEEE, 2018.

[12] Naijun Zheng and Xiao-Lei Zhang. Phase-aware speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):63–76, 2018.

[13] Yuma Koizumi, Kohei Yaiabe, Marc Delcroix, Yoshiki Maxuxama, and Daiki Takeuchi. Speech enhancement using self-adaptation and multi-head self-attention. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 181–185. IEEE, 2020.

[14] Daiki Takeuchi, Kohei Yatabe, Yuma Koizumi, Yasuhiro Oikawa, and Noboru Harada. Real-time speech enhancement using equilibriated rnn. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 851–855. IEEE, 2020.

[15] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[16] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016.

[17] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. *arXiv preprint arXiv:1603.09056*, 2, 2016.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Kun Han, Yuxuan Wang, and DeLiang Wang. Learning spectral mapping for speech dereverberation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4628–4632. IEEE, 2014.

[20] Ke Tan and DeLiang Wang. A convolutional recurrent neural network for real-time speech enhancement. In *Interspeech*, pages 3229–3233, 2018.

[21] Andong Li, Minmin Yuan, Chengshi Zheng, and Xiaodong Li. Speech enhancement using progressive learning-based convolutional recurrent neural network. *Applied Acoustics*, 166:107347, 2020.

[22] Se Rim Park and Jinwon Lee. A fully convolutional neural network for speech enhancement. *arXiv preprint arXiv:1609.07132*, 2016.

[23] Tomas Kounovsky and Jiri Malek. Single channel speech enhancement using convolutional neural network. In *2017 IEEE International Workshop of*

Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), pages 1–5. IEEE, 2017.

[24] Ashutosh Pandey and DeLiang 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)*, pages 6875–6879. IEEE, 2019.

[25] Yangyang Xia, Sebastian Braun, Chandan KA Reddy, Harishchandra Dubey, Ross Cutler, and Ivan 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)*, pages 871–875. IEEE, 2020.

[26] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. *arXiv preprint arXiv:2006.12847*, 2020.

[27] Yan Zhao, Buye Xu, Ritwik Giri, and Tao Zhang. Perceptually guided speech enhancement using deep neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5074–5078. IEEE, 2018.

[28] Babafemi O Odelowo and David V Anderson. A study of training targets for deep neural network-based speech enhancement using noise prediction. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5409–5413. IEEE, 2018.

[29] Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Monaural speech enhancement using deep neural networks by maximizing a short-time objective intelligibility measure. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5059–5063. IEEE, 2018.

[30] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In

2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[31] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li. Rsr2015: Database for text-dependent speaker verification using multiple pass-phrases. In *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[32] John S Garofolo. Timit acoustic phonetic continuous speech corpus. *Linguistic Data Consortium, 1993*, 1993.

[33] John Garofolo, David Graff, Doug Paul, and David Pallett. Csr-i (wsj0) complete ldc93s6a. *Web Download. Philadelphia: Linguistic Data Consortium*, 83, 1993.

[34] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015.

[35] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, 2014.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[37] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[39] 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)*, volume 2, pages 749–752. IEEE, 2001.

[40] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 4214–4217. IEEE, 2010.

Appendix A

Short-time Fourier Transform

Forward continuous-time STFT

The fourier transform is calculated by the equation:

$$\mathbf{STFT}\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}\,dt \tag{A.1}$$

where τ is the time index and ω is the frequency bin, while the window function and the signal to be transformed are denoted by $w(\tau)$ and $x(t)$, respectively.

Inverse continuous-time STFT

We have

$$\int_{-\infty}^{\infty} w(\tau)\,d\tau = 1. \tag{A.2}$$

It follows that

$$\int_{-\infty}^{\infty} w(t-\tau)\,d\tau = 1 \quad \forall\ t \tag{A.3}$$

and

$$\begin{aligned} x(t) &= x(t)\int_{-\infty}^{\infty} w(t-\tau)\,d\tau \\ &= \int_{-\infty}^{\infty} x(t)w(t-\tau)\,d\tau \end{aligned} \tag{A.4}$$

(A.1) equals to

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} \, dt. \tag{A.5}$$

Substituting $x(t)$ from (A.5):

$$\begin{aligned} X(\omega) &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(t) w(t-\tau) \, d\tau \right] e^{-j\omega t} \, dt \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) w(t-\tau) \, e^{-j\omega t} \, d\tau \, dt. \end{aligned} \tag{A.6}$$

Swapping order of integration:

$$\begin{aligned} X(\omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t) w(t-\tau) \, e^{-j\omega t} \, dt \, d\tau \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x(t) w(t-\tau) \, e^{-j\omega t} \, dt \right] d\tau \\ &= \int_{-\infty}^{\infty} X(\tau, \omega) \, d\tau. \end{aligned} \tag{A.7}$$

From (A.7) we can the inverse Fourier transform:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{+j\omega t} \, d\omega, \tag{A.8}$$

then $X(\tau, \omega)$ can be used to restore $x(t)$:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(\tau, \omega) e^{+j\omega t} \, d\tau \, d\omega. \tag{A.9}$$

or

$$x(t) = \int_{-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} X(\tau, \omega) e^{+j\omega t} \, d\omega \right] d\tau. \tag{A.10}$$