

Bangla Speech Isolation from Noisy Auditory Environment Using Convolutional Neural Network

by

K M Tahzeem Zaman
17101212
Zahid Hasan
17101466
Mohd. Ibrahim Hossain
17201021

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

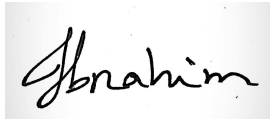
Student's Full Name & Signature:



K M Tahzeem Zaman
17101212



Zahid Hasan
17101466



Mohd. Ibrahim Hossain
17201021

Approval

The thesis titled “Bangla Speech Isolation from Noisy Auditory Environment Using Convolutional Neural Network” submitted by

1. K M Tahzeem Zaman (17101212)
2. Zahid Hasan (17101466)
3. Mohd. Ibrahim Hossain (17201021)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in January, 2023.

Examining Committee:

Supervisor:
(Member)



Md. Ashraful Alam, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

We hereby state that all the work that has been done to prepare this thesis, from research to the implementation of our proposed work and model, has been done on our own. Inspiration and all external resources such as datasets and derivations are acknowledged. We also refrained from using unethical means to obtain any gain here. We hereby certify that this work has not been submitted in whole or in part to any other university or institution for the purpose of completing the degree.

Abstract

In recent years, the primary solution to sound enhancement has gained popularity. There is a rich research contribution from academia and industry to remove noise and enhance sound quality. With the advance in machine learning and deep learning algorithms, well-performing audio enhancement models now exist. But such a sophisticated and well-researched model has not existed utilizing the language of Bangla. Although there have been models trained and tested to comprehend the language, no such model exists that can process real-time Bangla speech. Also, no such dataset exists that contains a substantial amount of speeches conducted in the Bangla language spanning over multiple hours. In this research, we studied the existing models that are working to separate noise in composite auditory environments, and on the basis of that study, we designed and implemented a U Net architecture model that has been trained in the Bangla language and is able to isolate and separate external noise from Bangla language speeches providing a clean feed to the listeners. Implementation of convolution neural networks in digital signal processing is a different approach and we achieved our desired results through it.

Keywords: Short-time Fourier Transform (STFT), U-Net, Singal to Distortion Ratio (SDR), speech separation.

Acknowledgement

To begin, we thank the Almighty for allowing us to continue with this research despite all of the challenges we had during the pandemic of the last 2 years. We would not be here without divine blessing. We would like to thank our supervisor for guiding us throughout this journey, for correcting our mistakes, and for responding to us whenever we need to improve our work ethic. He has given us sound guidance from his own experience and has taken time out of his busy schedule for us. We would like to thank our peers and friends for their help and support wherever possible in our challenging times. And finally, we thank our parents for their unconditional love for us.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	v
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	ix
Nomenclature	x
1 Introduction	1
1.1 Motivation	1
1.2 Problem Formulation	2
1.3 Research Objectives	2
2 Literature Review	3
3 Dataset Description	6
3.1 A Novel Bangla Audio Dataset	6
3.2 Data Pre-processing	6
4 Methodology	8
4.1 Architecture of the Proposed Model	8
4.1.1 STFT	9
4.1.2 FFT	10
4.1.3 Convolutional Neural Network (CNN):	11
4.1.4 U-NET:	11
4.1.5 Residual U-net	12

5	Experimental Results	13
5.1	Our Output Evaluation Method	13
5.2	Evaluation metric	13
5.3	Training Scheme	14
5.3.1	Mean Absolute Error (MAE) and Mean Square Error (MSE) .	14
5.4	Comparison study of Signal to Distortion Ratio (SDR)	15
5.4.1	Outputs based on model trained at 8.2 SDR (Uniform)	17
5.4.2	Outputs based on model trained at -3.4 SDR (Uniform)	17
5.4.3	Outputs Based on Model trained on Randomized SDR Level (-3.4-8.2)	18
5.5	Result Evaluation	18
5.5.1	Evaluation on Real-World Examples	19
6	Conclusion and Future Work	23
	Bibliography	25

List of Figures

3.1	Collected Online Video clips	6
3.2	Data Pre-processing flowchart	7
4.1	Model Architecture	8
4.2	Structure of the Residual Blocks	10
5.1	Train with 8.2 SDR	15
5.2	Train with -3.4 SDR	15
5.3	Train with Random ($-3.4 \sim 8.2$) SDR	16
5.4	Input Audio with added Noise, Ground Truth and Prediction; (Tested on 8.2 SDR)	17
5.5	Input Audio with added Noise, Ground Truth and Prediction; (Tested on 0.43 SDR)	17
5.6	Input Audio with added Noise, Ground Truth and Prediction; (Tested on -3.4 SDR)	18
5.7	Input Audio with added Noise, Ground Truth and Prediction; (Tested on 8.2 SDR)	18
5.8	Input Audio with added Noise, Ground Truth and Prediction; (Tested on 0.43 SDR)	19
5.9	Input Audio with added Noise, Ground Truth and Prediction; (Tested on -3.4 SDR)	19
5.10	Input Audio with added Noise, Ground Truth and Prediction (Tested on 8.2 SDR)	20
5.11	Input Audio with added Noise, Ground Truth and Prediction (Tested on 0.43 SDR)	20
5.12	Input Audio with added Noise, Ground Truth and Prediction (Tested on -3.4 SDR)	20
5.13	Effect of noise in training phase	21
5.14	Real World Audio input performing variably at differently trained models	22

List of Tables

3.1	Train-Test-Validation Split	7
5.1	Compariosn of SDR	16

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN Convolutional Neural Network

DSP Digital Signal Processing

FClayers FC layersFully Connected Layers

ML Machine Learning

SNR Signal to Noise Ratio

STFT Short-time Fourier Transform

Chapter 1

Introduction

1.1 Motivation

On paper, noise isolation may appear too elementary and unnecessary to pursue, and its significance may seem questionable. However, there are innumerable reasons why sound isolation is a crucial practice to pursue and master, particularly in this rapidly advancing digital age in which technology's accuracy has a profound impact on people's lives. Therefore, noise separation is essential and crucial for reducing and mitigating noise pollution in audio. On a more technical level, however, digital audio files require sound isolation to prevent sound leakage of any kind. It could be a piece of music, a film, or even an audiovisual creation that an audio leak has contaminated. Our solution can help by isolating noise from the video feed and minimizing noise mistakes. For example, in conference sessions, there are numerous noise interruptions. An effective noise-isolating system might help minimize errors/risks in these scenarios. The computational auditory isolation system can also help with noiseless audio streaming. These are a few reasons why computational auditory isolation is a crucial technology to adapt and master, particularly in today's technologically advanced environment.

1.2 Problem Formulation

When people speak in a crowded area, their voices are indistinguishable, causing numerous audio issues. In online audio streams, the sound is processed through the microphone, re-compressed, uploaded to the central server, and then provided to the receiver. Here, the audio signal requires powerful compression, and noise occupies significant data bits in compressed signals. Consequently, voice quality in online conversations and video conferencing often needs to improve. A Computational Auditory Isolation system can help mitigate this issue. We intend to offer an improved system that can take Bangla audio as input for audio processing, extract clean, noise-free audio signals from the audio, and provide individual voices separately.

A model such as this can be beneficial in the field of Bangla contents, especially for archival work, restoring old audio content and further improving the field of online audio communications.

1.3 Research Objectives

In this research, we aim to develop a robust sound isolation system that provides a fast and hassle-free way to extract the targeted speaker's speech in Bangla from a single or multichannel audio source. Audio-only approaches have been around for quite a while, but they have some significant limitations. Also, as stated earlier, such models in the Bangla language do not exist.

There are scopes to improve the effectiveness of the existing models. The objectives of this particular research are:

- To train the model in the Bangla language in order to make it more efficient in isolating multi speaker auditory environments.
- Creating a new dataset consisting of Bangla speech spanning over multiple hours
- Incorporating new audio features to improve speech detection

Chapter 2

Literature Review

To date, many types of research have been conducted to effectively isolate vocal/instrumental sound from mixed audio. A lot of them used audio-only approaches toward the Cocktail Party problem, while contemporary approaches leveraged both audio and visual signals to yield enhanced results.

Among the literature we studied, we have been able to single out various multi-faceted researches that would aid us in executing our desired aim of the project. Here is a rundown of the findings we have been able to collect from the resources and papers.

The human voice serves as a feasible mode of connection as a discourse. When a comparable voice is articulated with sustained resonance and cadence, it becomes something melodious: the singing voice. The performer's voice has become an integral aspect of the modern music industry. In addition to its employment as a main performance voice in songs, it is also utilized in rap music, theater, mumbling, and boom boxing for instance. The genre cappella refers to a vocal performance with one or more vocals with no musical accompaniment. The Sound Isolation approach suggested in this study utilizes a dataset titled A cappella [15]. Using the dataset A cappella, which consists of around a total of 50 hours of Cappella Songs from the internet. There are primarily four distinct ethnic language groups - Western, Hispanic, Indian and the rest - from their analyzed data. The A cappella Dataset consists of two Neural Networks, specifically-

1. Y-net: An innovative Singing Voice Separation Model
2. Audio Visual Speech Separation Model (U-Net)

The instances in their collection are defined by the timestamps. These timestamps are included in the CSV file. They manually picked the sectionst to exclude from the recordings which didn't strictly meet these criteria: single front facing views of the faces where there isn't any obstruction, little contextual noise; absence of boom boxing, finger fidgeting; and tracks with verses (for example they abstained from murmuring as well as warbling).

Chung et al. [11] attempted to separate an objective speaker's discourse from a combination of two speakers using a radically different media discourse detachment structure. In contrast to previous efforts that employed lip development on video

clips or pre-selected speaker data as a helper dependent component, a facial image of each subject was employed. In this endeavor, face appearance was included into cross-modular biometric tasks including aural and visual personality depictions in inactive space. This research aims to target and separate speech emanating from a composite environment, such as a Cocktail Party, utilizing an unique Audio Visual Speech Separation, AVSS, focused on FaceFilter [11], targeting a single picture of the Target Speaker. The applicable methods for this surgery are -

1. Cross-Modal Representation of Identities
2. Audio-Visual Speech Distinction

During this phase, voice and facial appearances are extracted and stored into a latent space in preparation for audio-visual fusion. The generation of general media joint components to address speaker data and horrifying data to be separated is a component of the general media combination stage. By integrating discourse embeddings and visual embeddings as well as the channel pivot, it is possible to obtain joint highlights from several media types. They employ two methods for assigning visual identities to voice embeddings [13]. The first one employs identical speaker information for each frame of voice embeddings. The alternative method relies on a self-reflection component to provide uniquely weighted partition needs. Due to the fact that each discourse outline includes distinct states such as silence, target discourse only, meddling discourse alone, and covered discourse, the level of the partition must vary depending on the nature of each outline. The cover assessor forecasts a time-frequency veil, leaving just the ideal voice of the observed sign. Specifically, it identifies the objective speaker using standard media components and builds a fine veil that considers the force ratio between the objective and interfering indicators.

Simpson et al. (2015) [5] [6] trained a Convolutional Deep Neural Network to produce predicted approximations for every optimal bare mask for separating speech signals within bona fide music compositions. It was shown that a convolutional DNN is able to distinguish vocal sounds from common musical combinations. Their almost one billion-parameter network was trained with surprisingly little information. They compared this performance to more conventional linear algorithms that were appropriately scaled. The DNN was proven to provide accurate predictions based on the identification of the voice sounds positively.

In [4][12], Zhang et al. (2020) provide an original audio-visual speech separation prototype that deploys and uses an uniquely disassembled technique to extract speech-related visual characteristics from video footages and utilize them to aid sound separation. They noticed that face movements include all speech-related information, as opposed to lip motions. The outcomes demonstrated that their methodology performs very well in busy conditions compared to other study [8].

In paper [14], Grauman et al. (2021) propose a similar approach as [10] [4] where they used a multi-task learning framework to simultaneously learn audio-visual speech separation and crossmodal face-voice embeddings. They used the complementary cues between lip movements and crossmodal speaker embeddings for vocal sound separation. Those embeddings assist in identifying the voice characteristics and

enhance speech separation; well-separated vocal sounds produce better audio embeddings. Their vocal separator model was fed with synthetically coupled voice signal to analyze the lip movement and the facial movements to track to extract the voice of the corresponding speaker.

Sagnik et al. (2021) [14] introduced the job of active audio-visual source separation and a mobile robot tool for navigating the environment in article [10]. Using signal processing, they developed a reinforcement learning (RL) framework for their mobile robot in which an agent learns a policy on how to move to better hear. In addition, researchers created reinforcement learning structures to address a wide range of visual navigation problems.

Although they have done very well in this case [3], they have faced some difficulties with their agent. Their agents easily get distracted by abnormal motion and produce poor quality for the near target and agents for distant targets sometimes lack a direct path due to the noisy environment.

To distinguish between monaural and binaural channels, Simpson (2015) [6] [7] they employed two distinct convolutional auto-encoder deep neural networks (DNN). With binaural audio, feed-forward DNN units of the dimensions 1000x2500x2000 were set as auto-encoders. This discriminatory design of the output layer required the auto-encoder to synthesize the samples representing the monaural speech of the male voice in the first 1000 output units and the samples representing the monaural speech of the female voice in the second 1000 output units. In addition, Separation quality (for the test data) was determined using the BSS-EVAL toolbox and is quantified by signal-to-distortion ratio (SDR), signal-to-artifact ratio (SAR), and signal-to-interference ratio (SIR).

But the system had drawbacks such as High frequency overlapping particles which cannot be separated for comparative advantage, Not the perfect solution for binaural audio systems. Most importantly this method does not work as efficiently and well as it does in case of male voices.

Chapter 3

Dataset Description

3.1 A Novel Bangla Audio Dataset

There was no Bangla speech dataset available that could be utilized in the training of our model for such a large magnitude. This prompted us to generate a large-scale dataset on our own.

YouTube was chosen to collect data in order to improve feasibility. We viewed and listened to Bangla podcasts, interviews, video lectures, news, reality programs, and a few dramas; then manually annotated the videos with chunks of audible voice. Typically, data are collected automatically in situations like this; however, we were unable to automate the procedure reliably. The data collection contains 15,072 audio clips of Bangla speech, each of which is three seconds in duration, for a total of 12.5 hours of audio. These snippets of audio were taken from a variety of videos found on YouTube, each of which featured a single speaker and contained only a trace amount of ambient noise. We divided the total dataset into 80:10:10 portions for training, validation, and testing, respectively. Additionally, we compiled a sub-dataset from YouTube that includes a wide array of background noises recorded in a variety of settings, such as busy roads, rain and thunder, and coffee shops. To train the model, these noises were added to the audio samples to train the model.



Figure 3.1: Collected Online Video clips

3.2 Data Pre-processing

The audio clips were collected from YouTube videos of varying lengths, but were segmented into 3-second clips for consistency. We extracted the audio from these

Total	Train	Test	Validation
15,072	12,000	1,499	1,500

Table 3.1: Train-Test-Validation Split

video clips, and the audio files were sampled at 16000 Hz. Each audio clip was then converted to a 1D array of 48,298 length.

The speech in the audio files were very audible and mostly noise-less, yet we used Librosa Library to perform Spectral Gating on the audio for further speech enhancement and noise suppression.

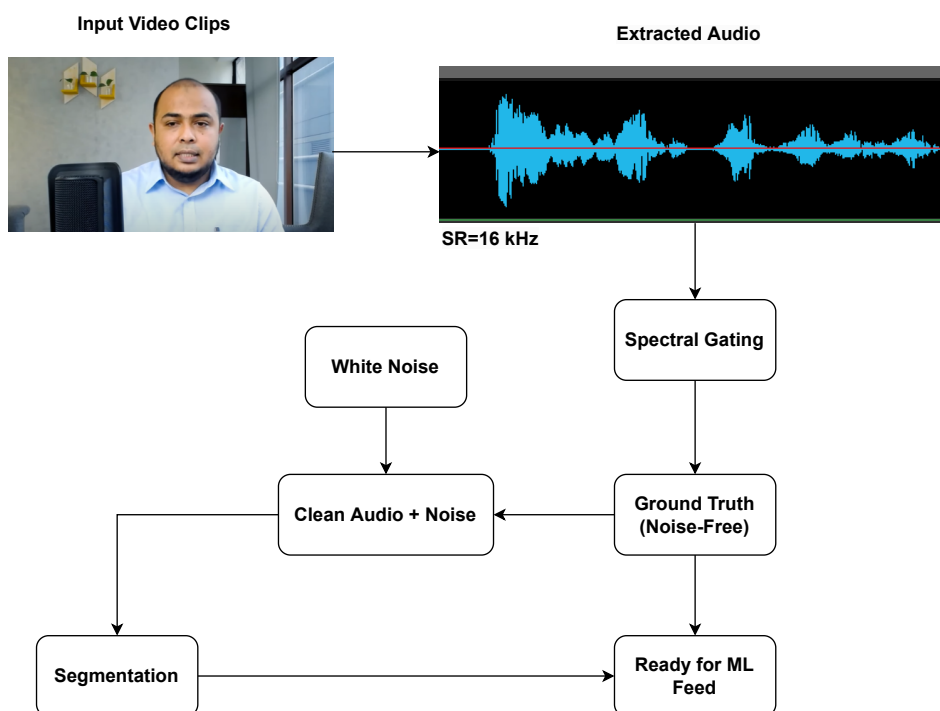


Figure 3.2: Data Pre-processing flowchart

The audio files were loaded from the directory and shuffled to ensure randomness in the dataset. All the audio clips were then concatenated into a single file.

Background noises collected were from various environments and added to the audio clips to create a diverse dataset for training the model.

Chapter 4

Methodology

4.1 Architecture of the Proposed Model

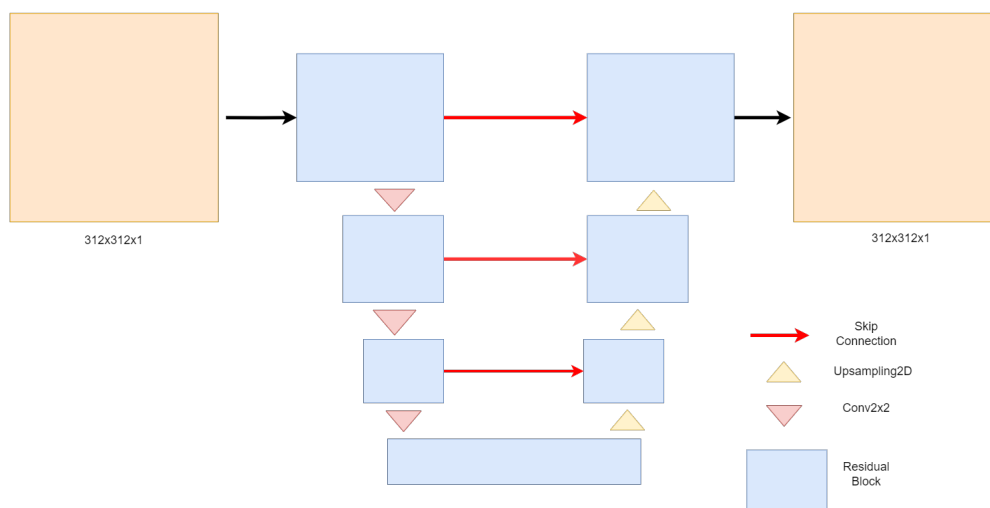


Figure 4.1: Model Architecture

As mentioned in the Dataset above, we have a total of 15072 results in our dataset. In order to efficiently feed the results into the model, we sliced together 100 results to form one single segment, totalling 1500 segments and then these 1500 segments were fed into the model. The model architecture we are implementing is the Residual U Net architecture which converts these 1500 segments into STFT. Although it was not necessary to segment the data but it was done to save computational costs and to process the results efficiently. The segmented videos are randomly selected and trained by multiplying by noisy audio, multiplication was done 3 times, once by 0.2, then by 0.8 and the third by a random value between 0.2 and 0.8. Henceforth the model was trained 3 distinct times to validate the results under varying noisy scenarios. Training under these different values made the model differentiate and process data efficiently yielding us improved results even when the magnitude of noise was much higher than average making the model much more versatile. Once processing

and calculation is done, the STFT 2D matrices are again passed through Inverse STFT/ISTFT to achieve the predicted results that are supposedly noise free clean audio. While the initial length of the array is 48298, the final length of the array after ISTFT becomes 48203 indicating minimal loss of dimensions but the audio itself is noise free clean. The model takes inputs x_1 and x_2 which are both height and width of the STFT respectively that are specifically 312 and 312, hence the dimensions of the input STFT are 312x312x1.

The input STFT of the models are first in the amplitude spectrogram. Since amplitude spectrograms are not distinguishable by the naked eye, we will be converting them to db spectrograms for visualization purposes and also to better explain and analyze our results. The input matrices of dimensions 312x312x1 are passed into the residual U net, the total features will be 16. The STFT matrices are maxpulled by U net 3 times shirking the total dimensions to 156, 78 and 39. We avoid further maxpulling of the matrices as it may lead to cumulative feature loss. Then by upsampling using the function `upsampling2D`, the matrices are again convoluted 3 times using `Conv2x2` making the output matrices of dimensions 312x312x16 again.

Linear activation/ Identity Activation has been used in the last layer of the model to extract the outputs which are spectrograms of shape 312x312x1 by performing convolution operation with the number of filters set to 1 and kernel size 3x3. ISTFT/ Inverse STFT function is performed on the STFT matrices to then convert them into noise free clean audio which are our resultant outputs.

4.1.1 STFT

Short-time Fourier transform (STFT) is a Fourier-related transform implemented and deployed primarily in digital signal processing projects to detect and determine the sinusoidal frequency and phase content of local parts of an evolving signal and also to calculate minute frequencies of signals. [2]. In practical applications, STFTs are computed by first dividing a larger temporal signal into shorter segments of equal length, and then independently computing the Fourier transform on each segment. These segments are then combined back together to form the original signal. The Fourier spectrum is illustrated here for each successively shorter segment. In the scenario of discrete time, the data that needs to be modified might be segmented into chunks or frames. Each segment is Fourier converted, and the complex output is added to a matrix that stores the magnitude and phase for each time and frequency point. This can also be stated as the following formula:

$$STFTx[n](m, \omega) = X(m, \omega) \equiv \sum_{n=-\infty}^{n=\infty} x[n]w[n - m]e^{-i\omega n} \quad (4.1)$$

likewise, with signal $x[n]$ and window $w[n]$. In this instance, m is discrete and continuous, but the STFT is often performed on a computer using the fast Fourier transform, so both variables are discrete and quantized.

As in our method to solve the problem, it is important to know the variation of frequency with respect to time. That's why STFT is used as the audio feature in the model

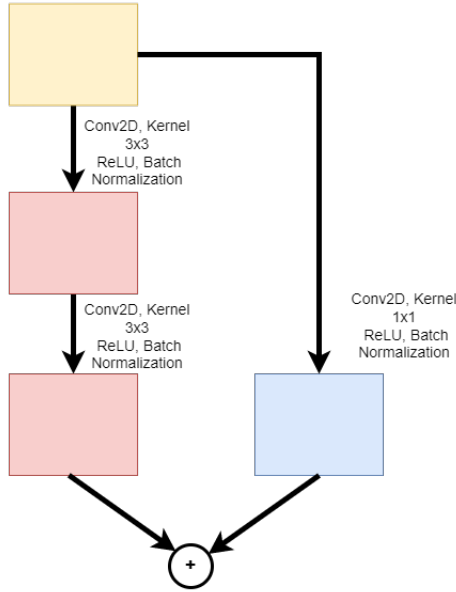


Figure 4.2: Structure of the Residual Blocks

In the main paper, [9], All audio is resampled to 16kHz and converted to mono by eliminating the right channel from stereo audio. STFT is computed with a 25ms Hann window, a 10ms hop length, and an FFT size of 512, providing an input audio feature. of $257 \times 298 \times 2$ scalars.

4.1.2 FFT

A fast Fourier transform (FFT) is an algorithm that calculates the discrete Fourier transform (DFT) of a sequence; the discrete Fourier transform is a method for converting particular sequences of functions into other types of representations. The discrete Fourier transform can also be described as the transformation of the structure of a waveform’s cycle into sine components. A rapid Fourier transform can be utilized in numerous signal processing applications. It may be beneficial for reading sound waves and image processing technology. Various types of equations can be solved with a quick Fourier transform, and various forms of frequency activity can be displayed in helpful ways. FFT, or Fast Fourier Transformation, is very useful in sound engineering, which is directly applicable to our research and justifies its applicability to our methodology.

The DFT equation $X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}$ gets divided into a series of minute conversions and then repurposed and reformulated in the Fast Fourier Transformation calculation. The fundamental formulas are known as radix-2 or radix-4, however there are alternative radix-r variants for $r = 2^k$ and $r \geq 4$. In order to comprehend how the FFT algorithm functions, consider the example when $k, n \in \mathbb{N} = 2n$. Applying

this particular DFT equation, one harmonic X requires N complex multiplications (k). As it is observed, already there are N number of harmonics, there must be N^2 complex multiplications. Assume that the time series $x(n)$ is decomposed into two $N/2$ -point time series consisting of even and odd samples. The number of complex multiplications per tiny transformation is $(N/2)^2 = N^2/4$. The total number of complex multiplications would thus be $(N^2/4) + (N^2/4) = N^2/2$, or one-half of what was originally required. Continuing with this partitioning strategy reduces the budget for complex multiplication to $(N/2) \log_2(N)$, which is much less than the direct mechanization of the DFT. Because of this significantly reduced need for multiplication, the fast Fourier transform can achieve significantly higher speeds.

4.1.3 Convolutional Neural Network (CNN):

Convolutional Neural Networks, often known as CNNs, are very similar to classic artificial neural networks (ANNs) in the sense that both types of networks are made up of neurons that can improve themselves through the process of learning. Each neuron will continue to receive an input and execute an operation (such as a scalar product followed by a nonlinear function) - the fundamental building blocks of innumerable ANNs. From input raw picture vectors to output class score, the complete network will continue to represent a single perceptual score function (the weight). The final layer will include loss functions connected with the classes, and all of the standard tips and tactics that were created for conventional ANNs will still be applicable. The main significant distinction between CNNs and conventional ANNs is that CNNs are predominantly employed in the field of picture pattern recognition. This enables us to encode image-specific characteristics into the network's architecture, making it better suited for image-centric tasks, while also lowering the number of parameters required to set up the model. CNNs concentrate mostly on the assumption that the input will consist of images. This narrows the emphasis of the architecture to be set up in a way that is most suited to meet the requirements for working with the particular kind of data.

4.1.4 U-NET:

U-net is a technology for segmenting images that was designed particularly for the sake of picture segmentation jobs. Because of these characteristics, U-net has a high utility within the medical imaging community. As a result, U-net has seen widespread adoption as the principal tool for performing segmentation tasks in medical imaging, which has led to its widespread adoption. U-net's use in practically all major image modalities, from CT scans and MRI to X-rays and microscopes, demonstrates its success. In addition, while U-net is primarily a segmentation tool, there have been occasions where it has been used for other purposes. U-net is an architecture of neural networks built primarily for image segmentation. The fundamental architecture of a U-net consists of two pathways. The first path is the contraction path, also known as the encoder or the analytic path, which offers classification information similar to a conventional convolution network. The second path is the expansion path, also known as the decoder or the synthesis path, which consists of up-convolutions and concatenation of the contracting path's characteristics. This enhancement enables the network to get localized classification data. In

addition, the expansion path enhances the output's resolution, which can then be passed to a final convolutional layer to get a fully segmented image. The resulting network is nearly symmetrical, taking the form of a U.

4.1.5 Residual U-net

RESUNET or Residual U-Net Architecture relates to the architecture of Deep Residual Networks. RESUNET is a fully convolutional neural network created to provide excellent performance with less parameters. It is superior to the present UNET design. RESUNET benefits from both the UNET design and Deep Residual Learning. When developing a deeper network, the utilization of residual blocks is helpful since it eliminates the need to worry about the problem of vanishing gradients or inflating gradients. It also facilitates straightforward network training. Rich skip connections in RESUNET improve the flow of information between different layers, which in turn improves the gradient flow during training. Similar to a U-Net, the RESUNET comprises an encoding network, a decoding network, and a bridge connecting the two. The U-Net is built with two iterations of the 3x3 convolutional layer, with each iteration being followed by a ReLU (Rectified Linear Unit) activation function. In the case of RESUNET, these layers are replaced by a residual block that has already been activated.

The final output of the results are produced through linear activation layer

Chapter 5

Experimental Results

5.1 Our Output Evaluation Method

As we have worked on a novel dataset, a proper comparison study is not feasible for this project.

We have trained our model four times with different levels of noise added each time with the audio. We also evaluated our models. In the later sections, we have shown some comparisons using SDR and visual comparison was done using STFT by performing dB spectrogram, denoting differences in our outputs.

Additionally, we recorded some short video and audio in the streets where we authors were speaking in noisy environments. We observed trained models with different noise multipliers performing differently in isolating noise. As we do not have Ground Truths for these recorded files, an evaluation could not be performed over them. Those experiments can be found at our [GitHub](#).

5.2 Evaluation metric

Vincent et al.[1] established the signal-to-distortion ratio (SDR) as one of a family of measures for evaluating Blind Audio Source Separation (BASS) algorithms when the original source signals are available as ground truth. SDR is the most common score reported for speech separation algorithms. It is measured in decibels and its definition is:

$$SDR := 10 \log_{10} \left(\frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} \right) \quad (5.1)$$

This metric can correspond well with the amount of residual noise in the separated audio. Higher SDR indicates a higher quality of isolated sound.

It may be noted that in our project, SDR value calculations have been done automatically with a python library named `mir_eval`.

5.3 Training Scheme

For the training scheme, we first prepared the three types of train data set of 8.2 SDR, 0.43 SDR, and randomly selected SDR from 0.43 to 8.2 SDR. For all the training data sets, the training parameters were the same. The batch size was 20, Adam optimizer with learning rate 0.001, and loss function was “noise to signal loss”. The training epoch was 30. We distinctly trained the model under 3 separate noisy scenarios where we multiplied the noise by 0.2, 0.8 and a random number between 0.2 and 0.8 to validate the best results. Multiple training ensured us the best of results which are described and shown below in the following paragraphs.

5.3.1 Mean Absolute Error (MAE) and Mean Square Error (MSE)

In the fields of meteorology, air quality, and climate research, the root mean square error (RMSE) has served as a standard statistical tool that has been utilized to measure the performance of model predictions. The mean absolute error (MAE) is an additional helpful metric that is commonly included in model evaluations. While both have been utilized for many years to evaluate model performance, there is no consensus on the most acceptable metric for model mistakes.

As each statistical measure condenses a huge quantity of variables into a single value, it provides just a single projection of the model errors that emphasizes a particular feature of the error characteristics of the model’s performance. Similarly, it is straightforward to demonstrate that for multiple sets of mistakes with the same RMSE, the MAE would differ between sets. Since statistics are merely a collection of tools, it is the responsibility of the researcher to select the most applicable tool for the subject being addressed. Because the RMSE and MAE are defined differently, we should anticipate different outcomes. Occasionally, additional 25 measures are necessary to provide a comprehensive picture of error distribution. When the error distribution is anticipated to be Gaussian and sufficient samples are available, the RMSE is superior to the MAE for illustrating the error distribution. For the purposes of simplification and calculation, we will assume that we already have n samples of model errors calculated as $(e_i, i = 1, 2, \dots, n)$. This will allow us to calculate more accurately.

The assumption underpinning the presentation of the RMSE is that mistakes are independent and follow a normal distribution. Consequently, utilizing the RMSE or the standard error (SE) provides a more full picture of the error distribution. So, using the estimated RMSEs, one may re-construct the error distribution near to its “truth” or “precise solution,” with its standard deviation within 5 percent of its truth, that is, $SE = 1$. Reconstructing the error distribution using RMSEs will be increasingly reliable as the sample size increases.

When compared to MAEs, RMSEs have the distinct advantage of avoiding the usage of absolute value, which is something that is highly undesired in many mathematical computations. This is one of the numerous ways in which RMSEs excel over MAEs. For instance, it may be challenging to calculate the gradient or sensitivity of the

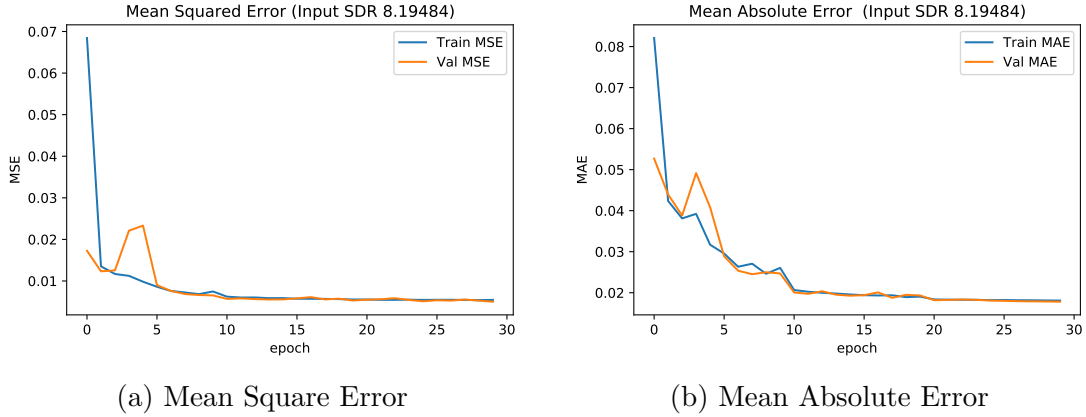


Figure 5.1: Train with 8.2 SDR

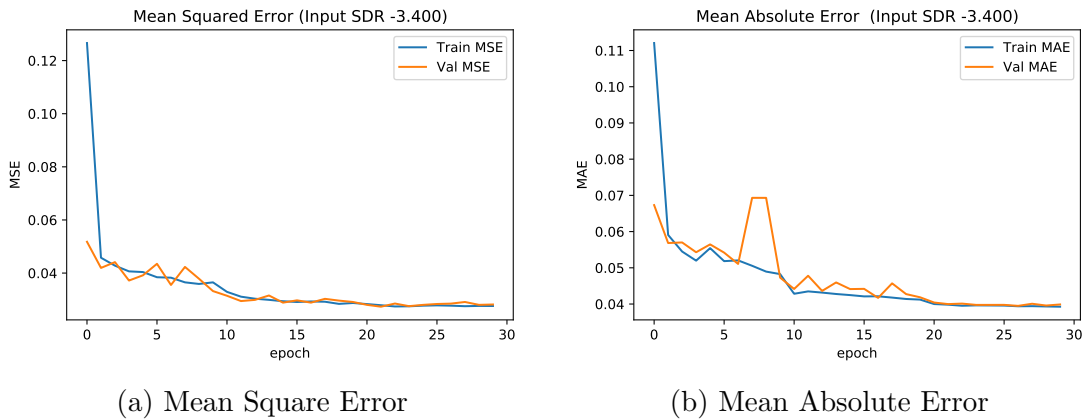


Figure 5.2: Train with -3.4 SDR

MAEs relative to particular model parameters. In addition, the sum of 20 squared errors is frequently stated as the cost function to be minimized by altering model parameters in the field of data assimilation. In such cases, penalizing significant errors by defining least-square terms appears to be a highly effective method for enhancing model performance. When it comes to estimating the sensitivity of model errors or using data assimilation, MAEs are most certainly not favored over RMSEs in any way, shape, or form.

5.4 Comparison study of Signal to Distortion Ratio (SDR)

All the experiments are done on the same train-validation-test split mentioned in chapter 3. We split the training dataset as 80 : 10 : 10 for training, testing and validation. The best model was chosen for the highest average validation SDR score. The SDR score shown in the Tab.5.1 is for the test dataset.

The figures in the following subsections show on the left that the noise added audio files visibly look cloudy and the voices are little hard to identify. The spectrogram in the middle were the ground truths for these particular audio files. On the right, the spectrograms show how much the model was able to identify noises and remove them.

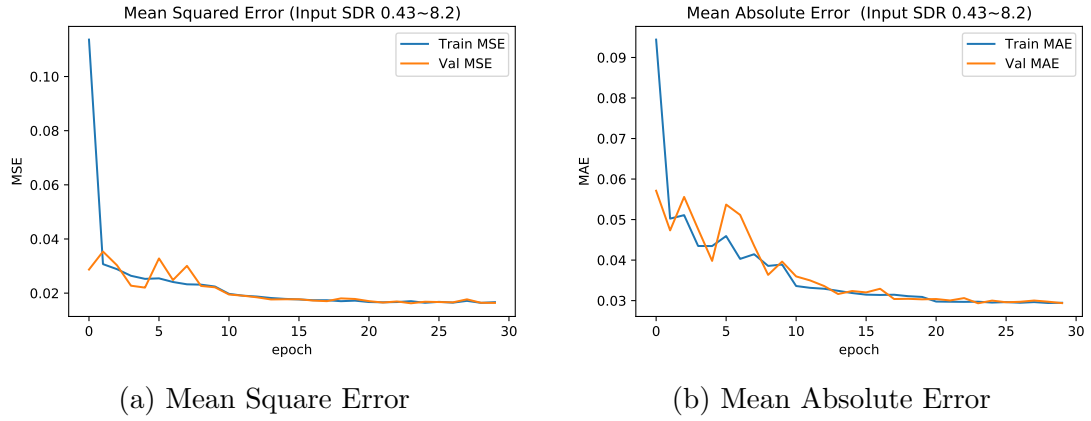


Figure 5.3: Train with Random (-3.4 ~8.2) SDR

Trained on	Avg. Input Test SDR	Avg. Output Test SDR
8.2 SDR	8.2	16.26
	0.43	4.772
	-3.4	-0.811
-3.4 SDR	8.2	14.87
	0.43	10.774
	-3.4	8.004
Random (-3.4 ~8.2)	8.2	15.88
	0.43	10.746
	-3.4	7.7412

Table 5.1: Compariosn of SDR

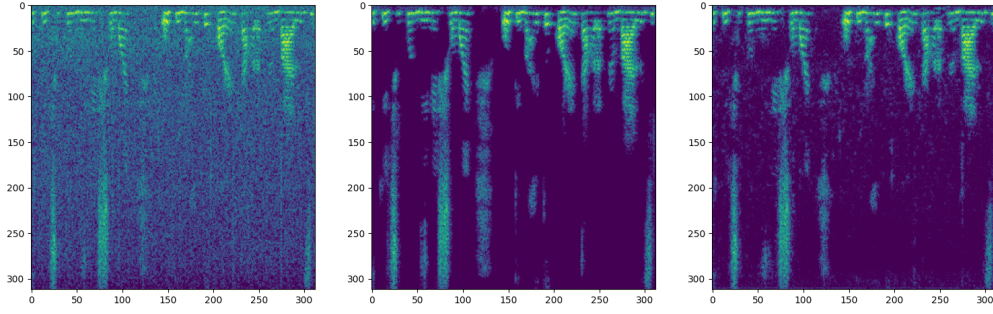


Figure 5.4: Input Audio with added Noise, Ground Truth and Prediction; (Tested on 8.2 SDR)

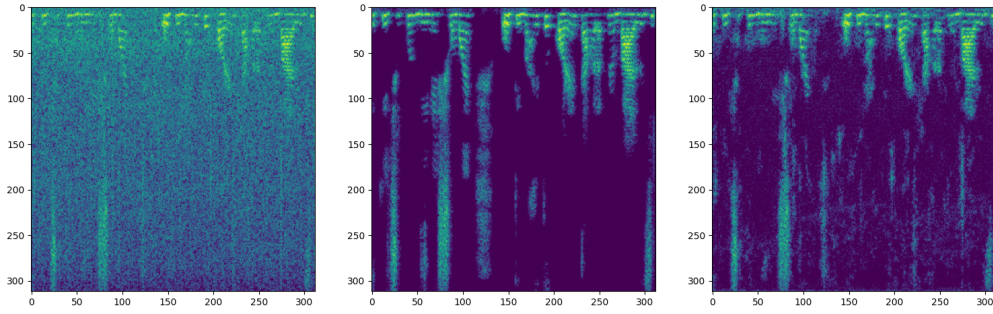


Figure 5.5: Input Audio with added Noise, Ground Truth and Prediction; (Tested on 0.43 SDR)

5.4.1 Outputs based on model trained at 8.2 SDR (Uniform)

This is the least level of noise we added to the clean feed, and we have tested the model with 3 different levels of SDR. Here are some amplitude-DB spectrogram for visual inspections.

The following figures: [5.4](#) , [5.5](#), [5.6](#) show how few of the operations performed over the different files for this noise level.

From the [5.1](#) and the aforementioned Spectrograms, it is significantly evident and visible that audio with minimal noise allows the model to perform at its best.

5.4.2 Outputs based on model trained at -3.4 SDR (Uniform)

This is the maximum level of noise we added to the clean feed, and we have tested the model with 3 different levels of SDR. Here are some amplitude-DB spectrogram for visual inspections.

The following figures: [5.7](#) , [5.8](#), [5.9](#) show how few of the operations performed over the different files for this noise level.

From the [5.1](#) and the aforementioned Spectrograms, it is yet again noticed that audio with maximum noise still allows the model to perform its best; and the efficiency suffers in low noise scenarios.

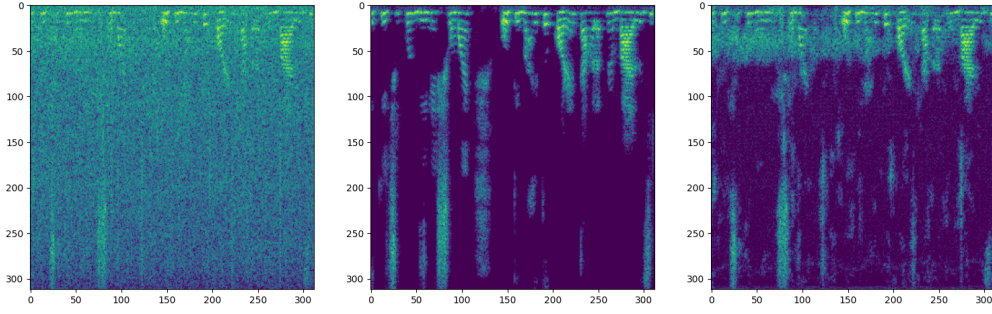


Figure 5.6: Input Audio with added Noise, Ground Truth and Prediction; (Tested on -3.4 SDR)

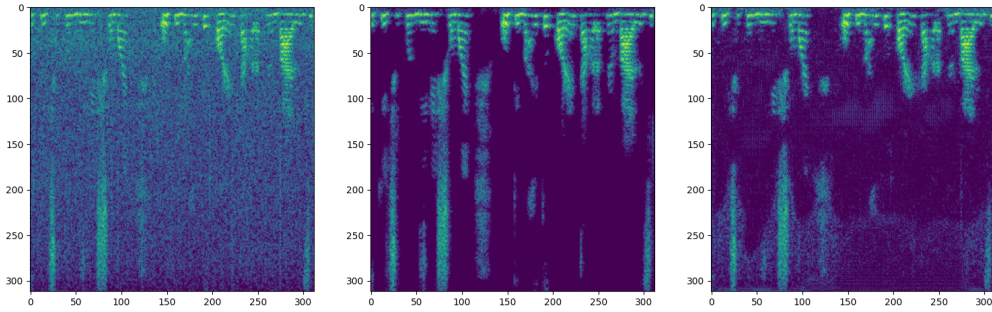


Figure 5.7: Input Audio with added Noise, Ground Truth and Prediction; (Tested on 8.2 SDR)

5.4.3 Outputs Based on Model trained on Randomized SDR Level (-3.4-8.2)

The level of noise we added to the clean feed are randomized from -3.4 to 8.2 SDR, and we have tested the model with 3 different levels of SDR. Here are some amplitude-dB spectrogram for visual inspections.

The following figures: [5.10](#) , [5.11](#) , [5.12](#) show how few of the operations performed over the different files for this noise level.

From the [5.1](#) and the aforementioned Spectrograms, this model not only performs moderately on every scenario, but its efficiency goes nearly as high as possible in every area, as can be seen from the fact that it is possible to see both of these things.

5.5 Result Evaluation

We summarized all our experimental funding in Fig.[5.13](#). As shown in the figure, there exists some dependency on the presence of noise in the training phase. For example, train with uniform noise 0.2 performs worst compared to the other two schemes. In fact, train with Random Noise performs slightly better compared to the uniform noise 0.5. Henceforth, it is observed that the model trained in the highest noise multiplier performs best in terms of low noisy scenarios and the model trained in lower noise values such as 0.2 or 0.5 performs well when multiplied by similar value of noise but performance begins to decline when the model is tested with higher noisy values such as 0.8.

When model is trained on uniform noise 0.2, average Output SDR is 16.26, the best

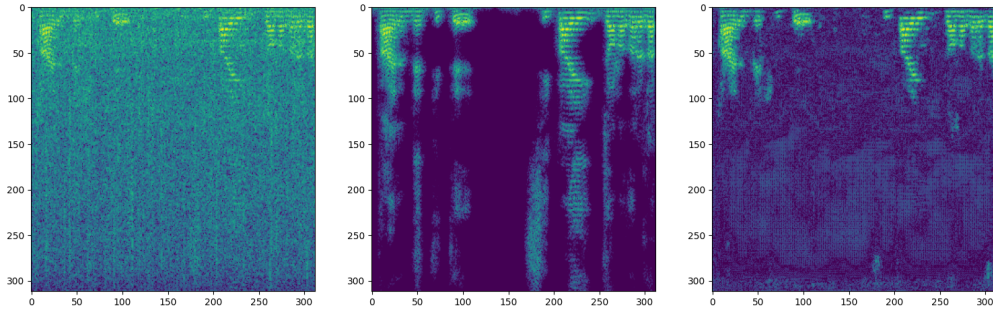


Figure 5.8: Input Audio with added Noise, Ground Truth and Prediction; (Tested on 0.43 SDR)

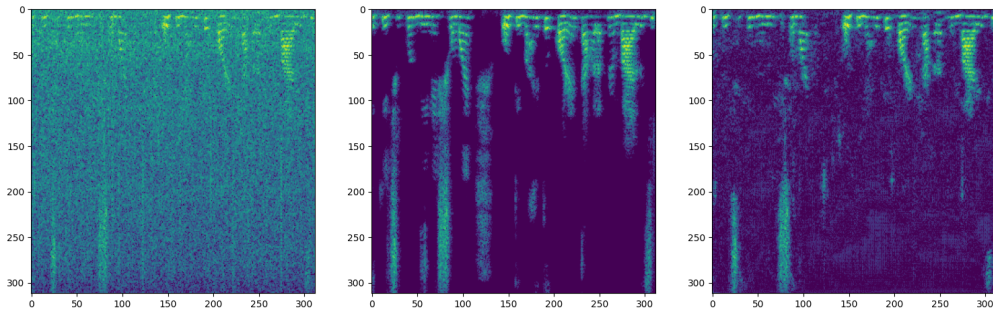


Figure 5.9: Input Audio with added Noise, Ground Truth and Prediction; (Tested on -3.4 SDR)

results are seen on testing with 0.2, good but significantly lower results on testing with 0.5 and very poor results with 0.8 resulting in a negative average Output SDR of -0.811. Similar testing were done on two more occasions with 0.8 and random values between 0.2 and 0.8. We yielded the best results while testing the model under the random values resulting in the best average Output SDR values of 15.88, 10.746 and 7.7412 respectively.

5.5.1 Evaluation on Real-World Examples

In the diagram above, we created an Unknown Example as a real world scenario and how our model might perform in an unknown use case where the input audio has no ground truth. As we can see in 5.14, the input spectrogram looks to be very noisy and below are the three outputs that have been produced by implementing the model three times.

In each of the spectrograms below the input, we can clearly see and distinguish the differences between the noisy spectrogram and the clean noise free spectrogram of the outputs indicating that our model has successfully isolated and separated noise from an unknown example. Henceforth the model can be deemed useful and applicable in real world scenarios as well.

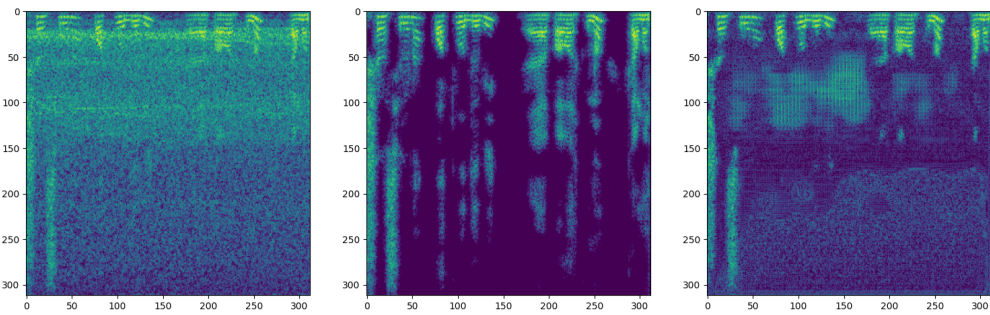


Figure 5.10: Input Audio with added Noise, Ground Truth and Prediction (Tested on 8.2 SDR)

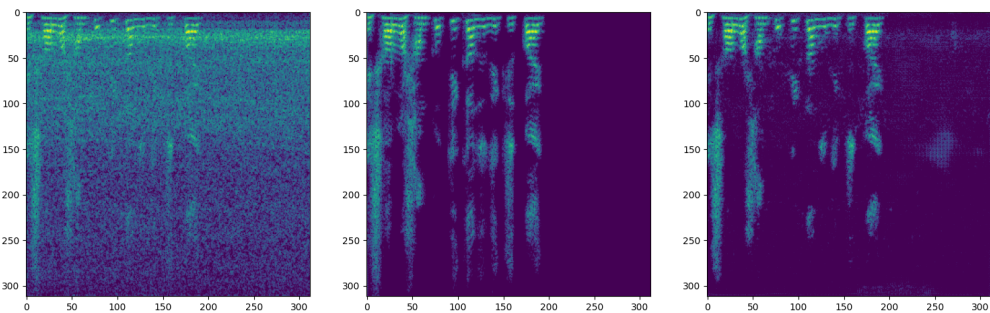


Figure 5.11: Input Audio with added Noise, Ground Truth and Prediction (Tested on 0.43 SDR)

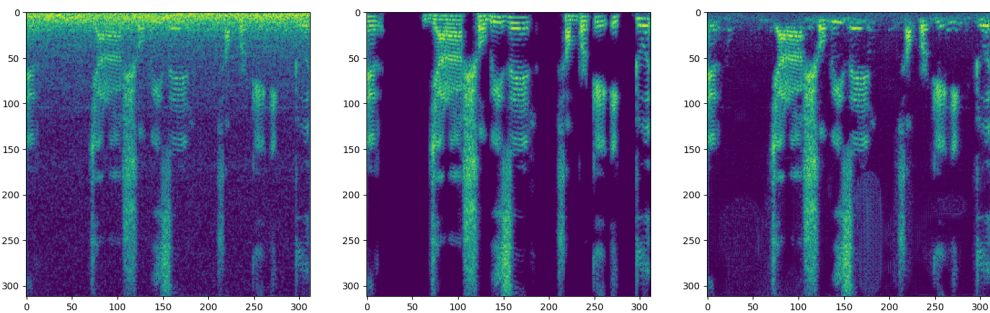


Figure 5.12: Input Audio with added Noise, Ground Truth and Prediction (Tested on -3.4 SDR)

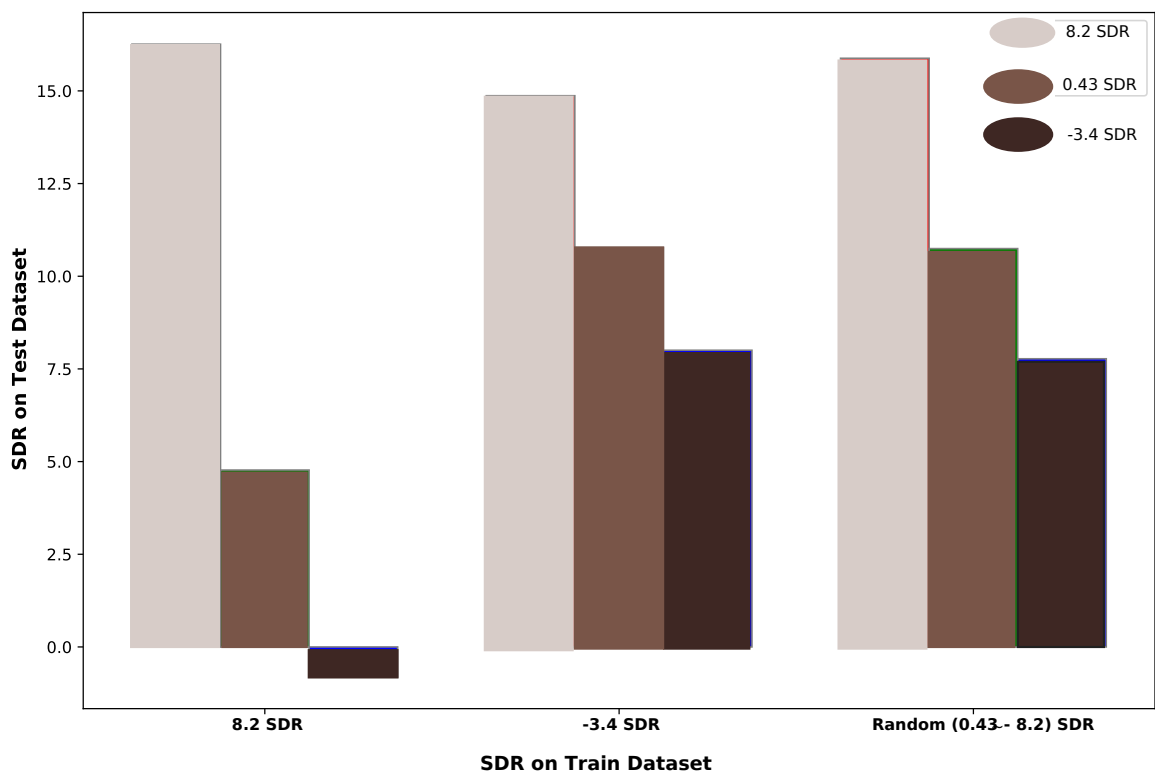
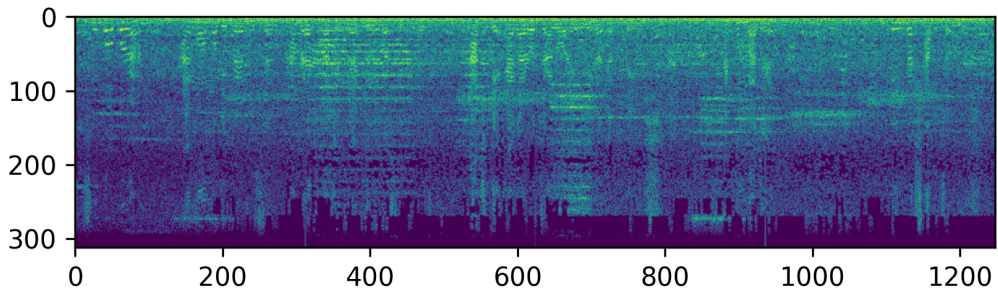
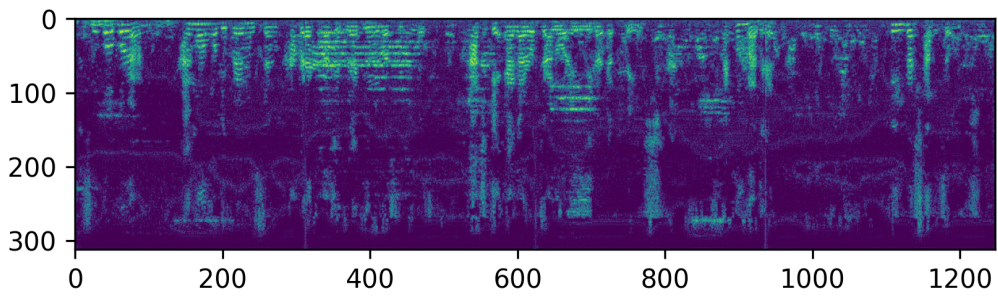


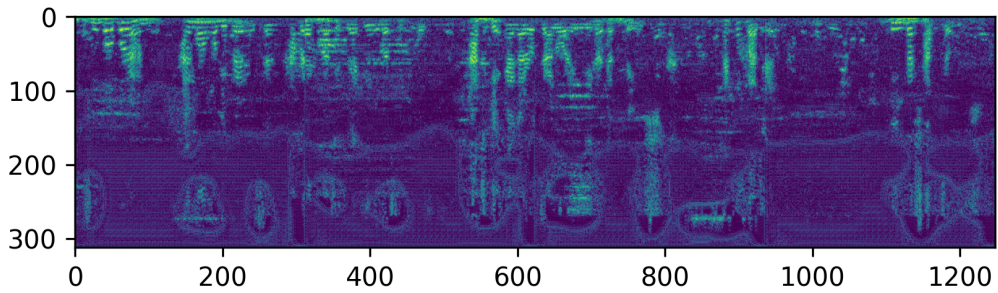
Figure 5.13: Effect of noise in training phase



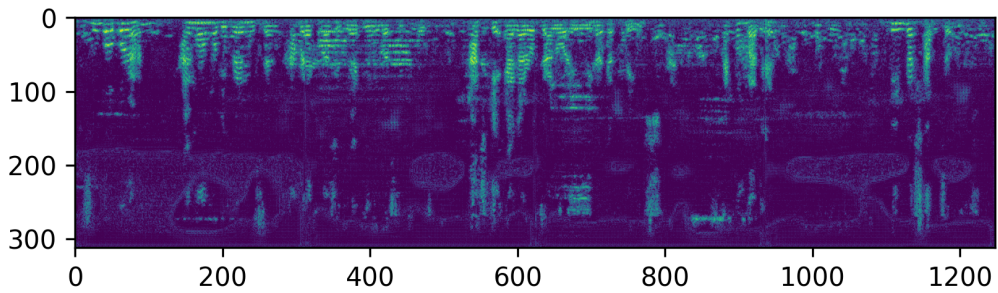
(a) Input Spectrogram of Input (Real World Audio Record)



(b) Output Spectrogram, by Model Trained at 8.2 SDR



(c) Output Spectrogram, by Model Trained at -3.4 SDR



(d) Output Spectrogram, by Model Trained at Randomized SDR Ranging from -3.4 to 8.2

Figure 5.14: Real World Audio input performing variably at differently trained models

Chapter 6

Conclusion and Future Work

Speech isolation can play a crucial role in facilitating audio enhancement in this hyper-connected world. Since the pandemic, the importance has only grown bigger; with more online video lectures or online meetings. Though researchers have started working on this issue earlier, our novel approach of using Bangla language to the speech isolation model may contribute to further development of this technology. Therefore, we hope researchers worldwide will come forward to make this technology better and, more importantly, accessible to all. The project's scope is enormous, and it is not feasible for us to execute a project of such magnitude. So instead, we plan to expand the project incrementally by further adding the dimension of video where the model can also recognize Bangla Speech in videos as well and perform speech separation and noise isolation. Eventually, we intend to deploy the model in a real-world scenario and observe how it performs compared to the other models.

In this paper, all our architectures focus on improving audio features. We implemented Residual U Net architecture on a brand new dataset which although gave us promising results but further iterations and expansion of the dataset will yield us better results. Henceforth we intend to further expand the dataset manifold so as to use it as a standard for Bangla Speech Separation. Since this is the first of this kind of research specifically involving the language of Bangla, we could not compare it to any existing model.

Henceforth, our future goal would be to implement the model again including Videos of Bangla Speech and compare the two models for further research. This will enable us to amplify the reach of our research enabling us to implement much more complex computations such as Multi Speaker Speech separation and identification. The scope of this research is enormous and the possibilities are truly endless.

Bibliography

- [1] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [2] E. Sejdić, I. Djurović, and J. Jiang, “Time–frequency feature representation using energy concentration: An overview of recent advances,” *Digital signal processing*, vol. 19, no. 1, pp. 153–183, 2009.
- [3] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.
- [4] H. Erdogan, J. R. Hershey, S. Watanabe, and J. 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)*, IEEE, 2015, pp. 708–712.
- [5] A. J. Simpson, “Deep transform: Cocktail party source separation via probabilistic re-synthesis,” *arXiv preprint arXiv:1503.06046*, 2015.
- [6] A. J. Simpson, G. Roma, and M. D. Plumbley, “Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network,” in *International Conference on Latent Variable Analysis and Signal Separation*, Springer, 2015, pp. 429–436.
- [7] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, “Synthesizing normalized faces from facial identity features,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3703–3712.
- [8] K. Hoover, S. Chaudhuri, C. Pantofaru, M. Slaney, and I. Sturdy, “Putting a face to the voice: Fusing audio and visual signals across a video to determine speakers,” *arXiv preprint arXiv:1706.00079*, 2017.
- [9] A. Ephrat, I. Mosseri, O. Lang, *et al.*, “Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation,” *arXiv preprint arXiv:1804.03619*, 2018.
- [10] J.-C. Hou, S.-S. Wang, Y.-H. Lai, Y. Tsao, H.-W. Chang, and H.-M. Wang, “Audio-visual speech enhancement using multimodal deep convolutional neural networks,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 117–128, 2018.
- [11] S.-W. Chung, S. Choe, J. S. Chung, and H.-G. Kang, “Facefilter: Audio-visual speech separation using still images,” *arXiv preprint arXiv:2005.07074*, 2020.

- [12] P. Zhang, J. Xu, Y. Hao, B. Xu, *et al.*, “Audio-visual speech separation with adversarially disentangled visual representation,” *arXiv preprint arXiv:2011.14334*, 2020.
- [13] R. Gao and K. Grauman, “Visualvoice: Audio-visual speech separation with cross-modal consistency,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2021, pp. 15 490–15 500.
- [14] S. Majumder, Z. Al-Halah, and K. Grauman, “Move2hear: Active audio-visual source separation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 275–285.
- [15] J. F. Montesinos, V. S. Kadandale, and G. Haro, “A cappella: Audio-visual singing voice separation,” *arXiv preprint arXiv:2104.09946*, 2021.