APPLICATION OF DEEP LEARNING FOR ARRAY MICROPHONE PROCESSING

by

Muhammed Furkan Akyürek

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2020

ABSTRACT

APPLICATION OF DEEP LEARNING FOR ARRAY MICROPHONE PROCESSING

Array microphone processing is a complex application with multiple interlinked components like direction of arrival for the audio sources, beamforming and post-filtering that are dependent on the array geometry. The array microphones gained popularity by the advent of the smart speakers. In this thesis, an end-to-end solution is provided containing all of the array microphone processing components along with the denoising integrated to the core of the system using a deep learning method called autoencoders. The neural network system is trained on the magnitude spectra generated by a dataset created exclusively for this thesis by combining some of the publicly available speech and noise datasets. This thesis proposes a single channel and a multichannel speech enhancement model to solve the beamforming problem. The multichannel autoencoder model is shown to perform better than some of the common conventional beamforming methods by objective evaluation methods. Results from this thesıs indicate the room for improvement in this field by the use of neural networks.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

C	Constraint matrix
$e(k)$	Error signal
$E[.]$	Mathematical expectation
h_C	Capon weight
h_F	Frost weight
$J(.)$	MSE criterion
$r_n(k)$	The residual noise
R_{vv}	Noise covariance matrix
r_{yx}	Cross-correlation vector between y and x
R_{yy}	Covariance matrix of y
$s(k)$	Source signal
t	Time index
T	Sampling Period
$v_{\mathrm{a},n}(k)$	The time shifted version of the noise signal
$v_n(k)$	Noise signal
$w[m]$	Windowed signal
$x_{\mathrm{a},n}(k)$	The time shifted version of the source signal
$x_a(t)$	Analog continuous-time signal
X	Fourier transform of x
$y_{a,n}(k)$	The time shifted version of the signal collected on microphone n
$y_n(k)$	The signal collected on microphone n
$z_{\mathrm{C}}(k)$	The output signal of the MVDR beamformer
$z_{\mathrm{DS}}(k)$	The output signal of the DS beamformer
α_n	Attenuation coefficient
$\mathcal{F}_n(\tau)$	Time difference of arrival between microphone 1 and microphone n

ω	Frequency variable
ω_0	Fundamental frequency
σ_s^2	Covariance of the source signal

LIST OF ACRONYMS/ABBREVIATIONS

1-D	One Dimensional
2-D	Two Dimensional
3-D	Three Dimensional
AC	Air Conditioner
ASR	Automatic Speech Recognition
BN	Batch Normalization
BSS	Blind Source Separation
CED	Convolutional Encoder-Decoder
CLDNN	Convolutional Long Short-term Memory Deep Neural Network
CNN	Convolutional Neural Network
DEMAND	Diverse Environments Multi-channel Acoustic Noise Database
DFT	Discrete Fourier Transform
DOA	Direction of Arrival
DS	Delay-and-sum
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GAN	Generative Adverserial Network
GPU	Graphical Processing Unit
GSC	Generalized Sidelobe Canceller
LCMV	Linearly Constrained Minimum Variance Filter
MAM	Multichannel Autoencoder Model
MER	Match Error Rate
MNIST	Modified National Institute of Standards and Technology
MOS	Mean Opinion Score
MP3	MPEG Audio Layer-3
MSE	Minimum Squared Error
MVDR	Minimum Variance Distortionless Filter

PESQ	Perceptual Evaluation of Speech Quality
PReLU	Parametric Rectified Linear Unit
R-CED	Redundant Convolutional Encoder-Decoder
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RNN	Recursive Neural Network
SNR	Signal-to-noise Ratio
STFT	Short-time Fourier Transform
STOI	Subjective Short-time Objective Intelligibility
TIMIT	Texas Instruments/Massachusetts Institute of Technology
WAV	Waveform Audio File Format
WER	Word Error Rate
WIL	Word Information Lost

1. INTRODUCTION

Smart speakers and the latest smartphones supported with multi-microphones are becoming more popular. The array microphone structure enables the beamforming methods to be applied to the multichannel data. This increases the speech quality and performs better than a single microphone [1]. Specifically, the smart speakers like Amazon Echo, Google Home and Apple HomePod feature various microphone array geometries. These devices record the audio on all their microphones and process the data to improve the performance of their speech recognition engines.

The array microphone structure offers a unique advantage that is not physically possible with a single microphone: spatio-temporal data. As the microphones are placed next to each other with some distance, each microphone collects a similar yet a different version of the signal sources in the room. There are various approaches to exploit the spatial dimension of the multichannel acoustic data. The most basic methods employ a delay detection mechanism to superimpose all the channels and sum them. Other methods try to find the direction of arrival (DOA) of the speech source then use this to assign weights to each channel. These methods are called beamforming algorithms and some are directly linked to the array geometry. The conventional beamforming algorithms like delay-and-sum (DS) beamformer and the minimum variance distortionless filter (MVDR) are some of the most popular ones ([2], [3]). More adaptive beamforming methods which change their response in time also exist but are not explored in this thesis.

In this thesis, we aim to go beyond the conventional beamforming methods that are used in coordination with post-filtering methods. Our goal is to develop an end-to-end solution that can be used on multichannel audio data irrespective of the array geometry and the microphone number using deep learning methods. Not only does the deep learning model implement the beamforming components to combine the signals but also it enhances the speech signal through denoising. Through this structure, we

aim to eliminate the need to build complicated connected array microphone signals processing structures. Our model takes inspiration from the image denoising models making use of the autoencoders due to the similarities between speech denoising and image denoising operations.

This thesis has another important task to make the main goal possible. Due to the need for large datasets to train neural networks for better generalization, we are tasked with creating a dataset made up of hours of audio samples including various speakers, environment configurations and noise types.

We will measure the success of this model by comparing it to some of the most popular conventional beamforming methods. The research papers in this field are difficult to implement or not possible as the models are not available. For the evaluation, speech intelligibility evaluation methods will be used for objective comparisons. The exact same test sets containing audio samples various signal-to-noise ratio (SNR) levels will be used for all methods.

This thesis has been inspired by the latest developments in the deep learning field and the success of the models to tackle some of the most difficult problems like speech recognition and speech generation. We wanted to propose a solution that would eliminate the need to extract noise profiles, build systems that are made for a limited number of array geometries and structures and work with features that will make the generalization difficult. As it can be seen in the developments of the image processing field with the application of deep learning models, the speech can be a field where more progress can be made by training large datasets with various neural network models.

2. BACKGROUND

2.1. Speech Processing Fundamentals

2.1.1. Basics of Speech

Speech is the most natural form of human communication. Denes and Pinson named the speech process from production to perception as the speech chain [4].

The Figure 2.1 displays the entire process of how the speech is produced and perceived by humans. The journey begins with the formulation of the message in the human brain and moved onto the vocal tract system where the speech is created in acoustic waveform. The message could be what we say in a normal conversation which is then converted into the language code. This could be any of the languages humans speak.

The third stage in the Figure 2.1 represents the actions taken by our body to actualize the message we intend to create. The neuro-muscular controls trigger the vocal tract system to physically create an acoustic waveform.

Once an acoustic waveform has been created, we move onto the speech perception sequence where the speech generated into the transmission channel has to be processed and understood. The next step is to convert it into spectral form to be processed by the brain. This conversion is done by the basilar membrane motion stage. The neural transduction step comes right after the spectral form is created. This can be thought of as a biological feature extraction where the sound features are generated in order for the brain to understand the sound. The sound features are used to the create the words and sentences in the language translation process. The chain is complete and the speech is produced and perceived.

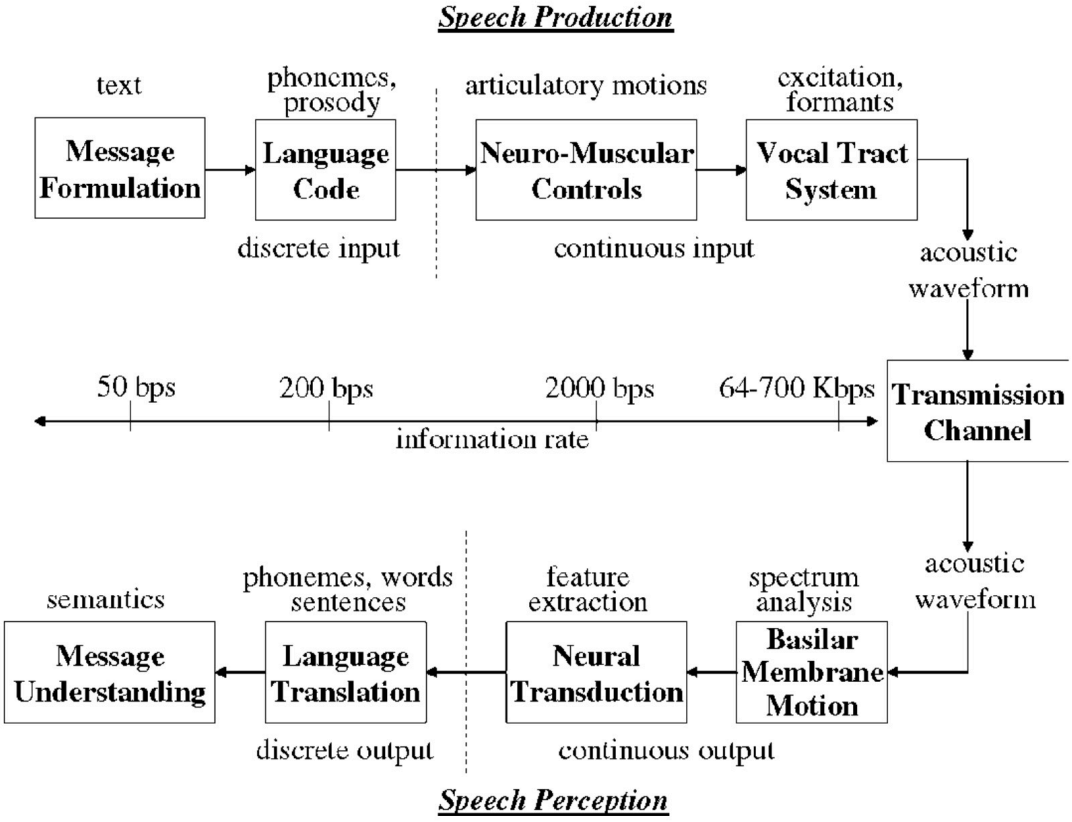


Figure 2.1. The speech chain can be seen above. It's broken down to production and perception. Each section is also separated based on its type [5].

In our project, the acoustic waveform and transmission channel will be the main interest as the entire processing will be done over the digital speech signal obtained in order to improve the quality of the perceived speech.

The human ear can hear in the range of 20 Hz to 20,000 Hz. With the age, the upper limit goes down to 16,000 Hz. This frequency interval will also determine the sampling rate that will be used within the project. Working with anything above 8,000 Hz will be unnecessary as it will not bring any benefit to the end result. The narrowband telephone calls has the frequency range of 300 Hz to 3,400 Hz. Using the narrowband audio, we will be saving data and still end up with highly intelligible speech despite the narrowband audio sounding different than the original audio [5].

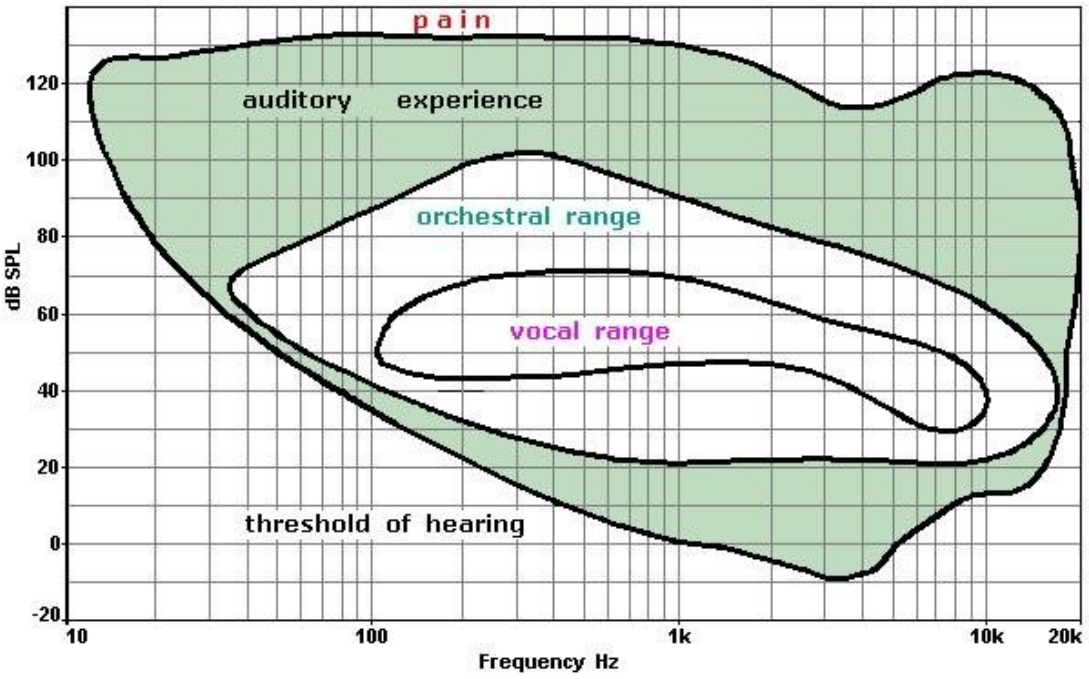


Figure 2.2. The frequency range of human hearing [6].

The noise is an integral part of the speech processing and is one of the core aspects of this thesis. It can be stationary as in the case of an air conditioning noise or non-stationary as in the case of a cafe recording. Despite the changing nature of the audio signal, if it's taken in short-time frames (10-30 ms), its characteristics are mostly stationary. This allows us to perform Fourier analysis on the speech signal with the added noise [7].

2.1.2. Frequency Domain and DFT

Most audio signals can be considered as continuous-time signals which can be simply described as having uncountable number of samples. The acoustic waveform humans produce is a good example of the audio signals in continuous form. However, in order to digitally process the audio signals, it is essential that the work is done on finite number of samples. Therefore, all the speech signals worked on are in discrete-time form.

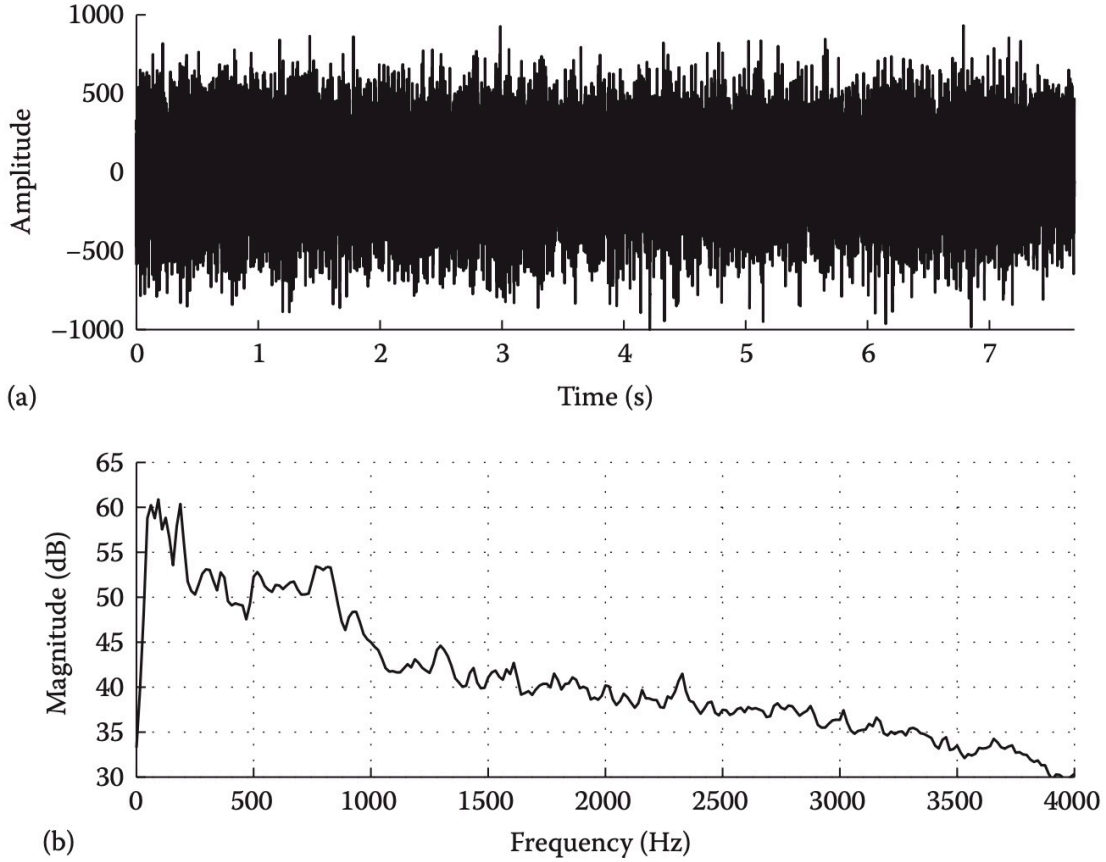


Figure 2.3. The image above displays how in certain cases most of the noise energy is accumulated in lower frequencies while the rest of the spectrum is much less. (a) Noise from a car in time domain. (b) frequency domain representation [7].

The audio signal will have a time axis, t. For this chapter, the notation used for the continuous time signal is $x_a(t)$. On the other hand, the discrete signal will be denoted as $x[n]$ where n is an integer. The value of nth number will correspond to the continuous value, $x_a(t)$ where t is equal to nT.

$$x[n] = x_a(nT) \tag{2.1}$$

T stands for the sampling period. In the Figure 2.4, T can be seen in the discrete-time scenarios. For the 8,000 Hz case, one second will contain 8,000 samples of the signal. In the work concerning the thesis, that sampling frequency is sufficient to process the

signal without a significant loss in information.

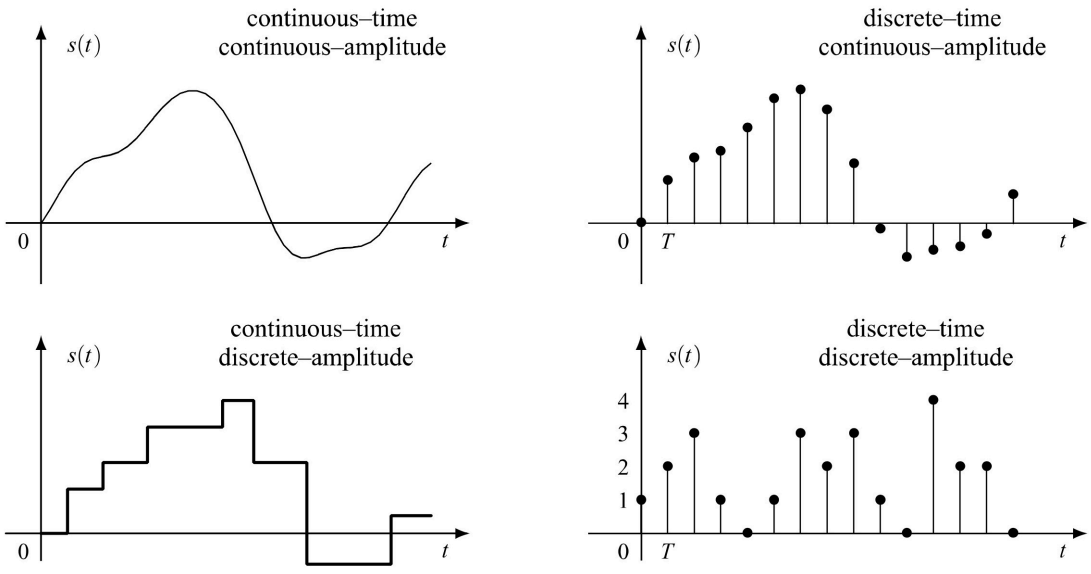


Figure 2.4. The graphs display various cases where time or amplitude values are displayed in the continuous form or discrete form. The discrete-time continuous-amplitude will be used in this thesis [8].

The audio signal is captured in the time-domain and converted to the discrete-time form. The time-domain analysis is used for the audio signals and will be shown in the later sections of this thesis for noise cancellation, and beamforming. The frequency domain is another way we can represent the signal. The time-domain analysis shows how the signal amplitude changes over time. The frequency-domain analysis, on the other hand, displays the energy distribution of the signal across the frequency axis.

A signal can be converted to the frequency domain using the Fourier Transform. The Fourier Transform decomposes a signal into a summation of infinite sine-wave components. Working with the speech signals, the Discrete Fourier Transform (DFT) will be used as the signals that are stored digitally are discrete and have finite duration. For the DFT, only a set of discrete frequencies will be used. Those frequencies are the multiples of the fundamental frequency, ω_0. The k is the coefficient of the fundamental frequency and $k\omega_0$ is in radians/second.

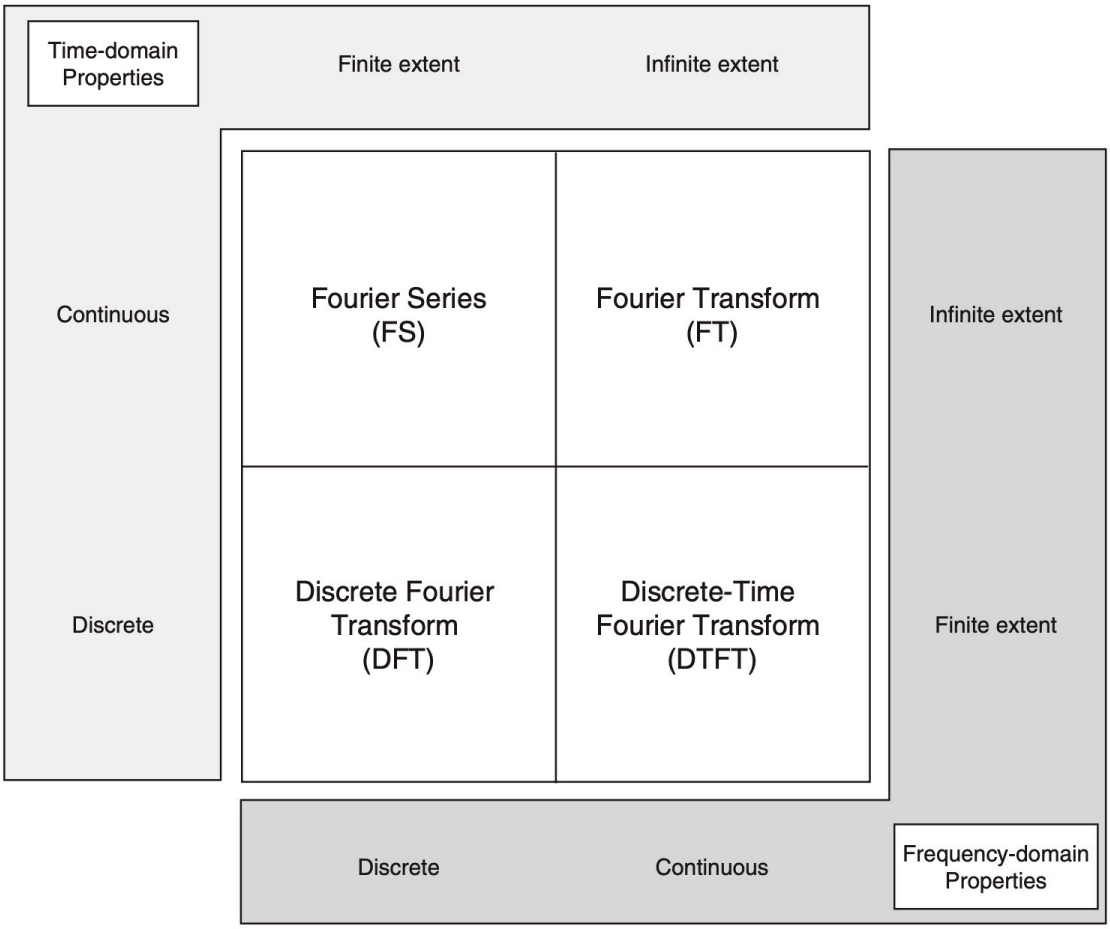


Figure 2.5. The time and frequency domain characteristics for the four different cases [9].

For $x[n]$ of length N, set $\omega_0 = \frac{2\pi}{N}$

$$x[n]: \quad n = 0, 1, \ldots, N-1$$

$$X[k]: \quad k = 0, 1, \ldots, N-1$$

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\omega_0 n} \tag{2.2}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{jk\omega_0 n} \tag{2.3}$$

In Equation 2.2, the time domain components are known and represented with the

variable $x[n]$. The length of the transform is set to be N and the fundamental frequency, ω_0, is equal to $\frac{2\pi}{N}$. The right hand side of the equation has the transform X representing the frequency content of the signal [9].

The next Equation 2.3 is the inverse DFT. It uses the frequency components of the signal that is already known to obtain the original signal. Both of these equations are used frequently in order to work with the frequency characteristics of the signal and generate the original signal using the processed frequency spectrum.

The DFT is always defined thanks to its finite sum property. This allows the calculation to be made on a computer easily. One example method that is frequently used is Fast Fourier Transform (FFT).

2.1.3. Short Time Fourier Transform

The Discrete Fourier Transform is used to obtain the frequency-domain representation of a discrete signal using the sum of sinusoidal waves. The assumption made for the DFT was the frequencies of the sinusoidal waves do not change over time which would mean the signal properties stay the same. In reality, the properties (phase, amplitude and frequency) will have a time-varying nature. Speech signals are non-stationary and exhibit changes in their properties over time. For this reason, the DFT is not enough to analyze the speech signals. To overcome the problem of varying signal properties, a new method is introduced: the short-time Fourier transform [10].

The short-time Fourier transform of a signal is defined as the following

$$X[n,\omega] = \sum_{m=-\infty}^{\infty} x[n+m]w[m]e^{-j\omega m} \tag{2.4}$$

where $x[n]$ represents the signal while $w[n]$ represents a windowed signal. The signal is a one-dimensional sequence of the time variable n. On the other hand, the transform is a two-dimensional sequence with the discrete variable n and the discrete variable

ω. The equation can be viewed as the signal that is shifted by m is multiplied by the window sequence $w[m]$. The window sequence has stationary properties and as the time variable n is shifted, a different part of the signal is processed.

The primary reason behind the sliding window approach is to keep the spectral characteristics of a stationary signal fixed by limiting the sequence to the duration of the window. Based on how frequently the characteristics change, the window should be shorter or longer. However, reducing the window has the adverse effect of frequency resolution decrease. At the same time, a shorter window would mean the increase in the time resolution. This trade-off determines how the window length should be selected.

The spectrogram is a useful representation that will be featured frequently in this thesis. It's the graphical display of the magnitude of the transform. The logarithmic scale is generally used to obtain the graph.

$$S(n, \omega) = \log |X(n, \omega)|^2 \tag{2.5}$$

Continuing the trade-off discussion related to the window size, two different outcomes can be achieved. The wideband spectrogram representation is the first one. It can be achieved by using a short window $w[n]$. This type of window results in poor resolution in frequency domain and good resolution in the time domain. The narrowband spectrogram representation is the opposite of the first approach. It can be created by using a long window $w[n]$. This representation exhibits vertical striations while the wideband spectrogram representation exhibits horizontal striations. This can be better viewed in Figure 2.7.

The narrowband and the wideband spectrogram representations essential show the length of the window that slides through the signal to complete the short-time Fourier transform. In the narrowband case, it could be as long as 30 ms. For the wideband case, it can be 5 ms. Another important factor in the calculation of STFT of

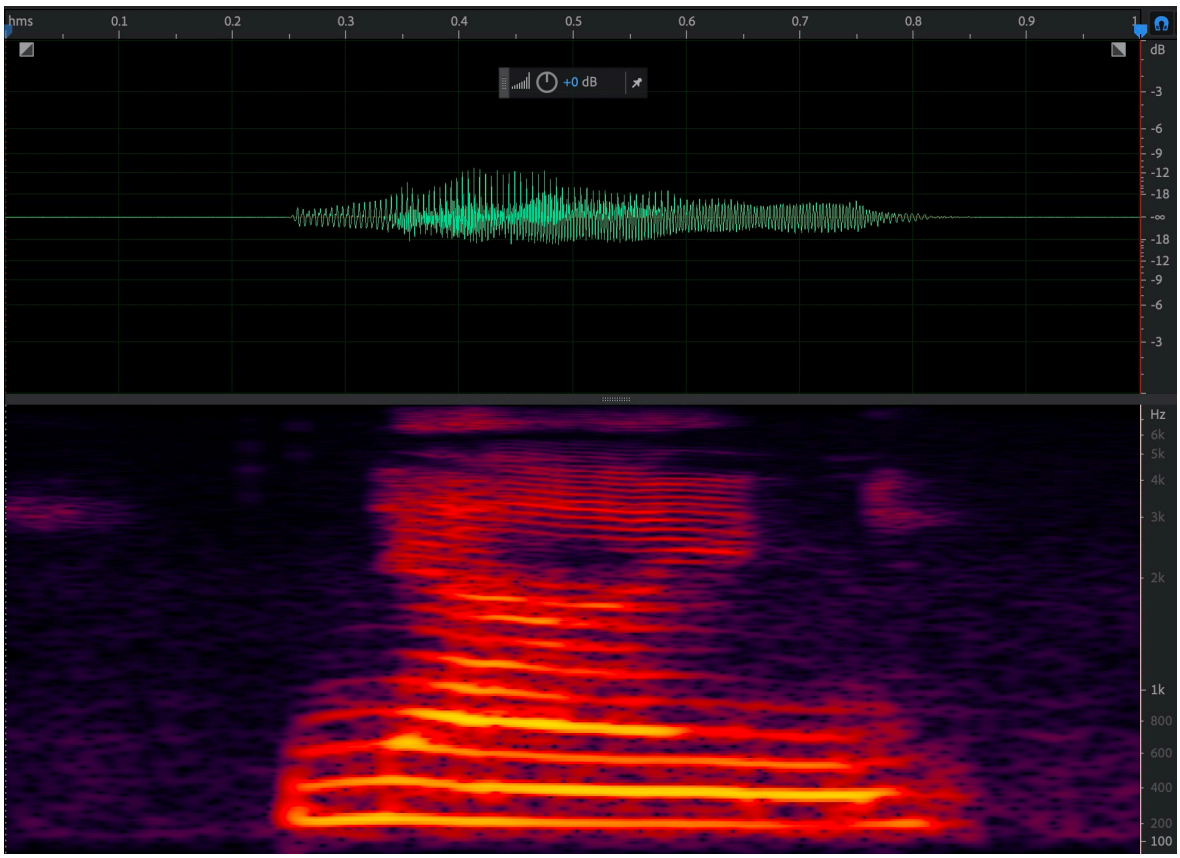


Figure 2.6. The spectrogram of a woman saying "nine". The audio has the sampling frequency of 16 kHz.

a signal is the amount of overlap. Using the overlap percentage, a window function is applied to generate the window segments by multiplying the signal with the function. Each segment passes through Fast Fourier Transform (FFT).

The main rationale behind the windowing function is to localize the signal to give it a stationary nature. The window function $w[n, \tau]$ is usually tapered at its ends. This prevents unnatural discontinuities in the segments. Because of the tapering, the overlap becomes necessary. The reasoning becomes clear in the Figure 2.8 where significant amount of data is lost.

The window function selection impacts the spectral estimate received as the result of the transform. There are many different window functions. The most commonly

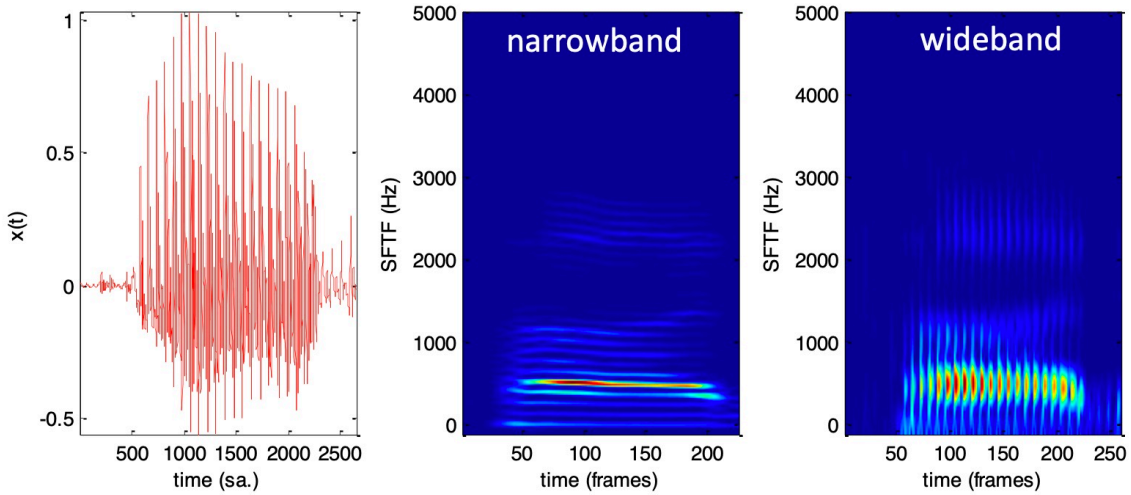


Figure 2.7. The figure displays the time domain representation of a signal (left), the narrowband spectrogram representation (center) and the wideband spectrogram representation (right) [11].

used ones are the Hamming and Hann windows.

Both Hann and Hamming window functions are in the family of generalized cosine windows.

$$w[n] = \sum_{k=0}^{K} (-1)^k a_k \cos\left(\frac{2\pi k n}{N}\right), \quad 0 \leq n \leq N \tag{2.6}$$

In Equation 2.6, when $a_0 = 0.5$ and $K = 1$ are set, the outcome is the Hann window. This function is named after Julius von Hann and also referred as the Hanning function.

$$w[n] = 0.5\left[1 - \cos\left(\frac{2\pi n}{N}\right)\right] = \sin^2\left(\frac{\pi n}{N}\right) \tag{2.7}$$

The Hamming window function is quite similar to the Hann window function with one key difference that can be seen in Equation 2.7. The a_0 variable is set to 25/46 in order to cancel the first sidelobe of the Hann window.

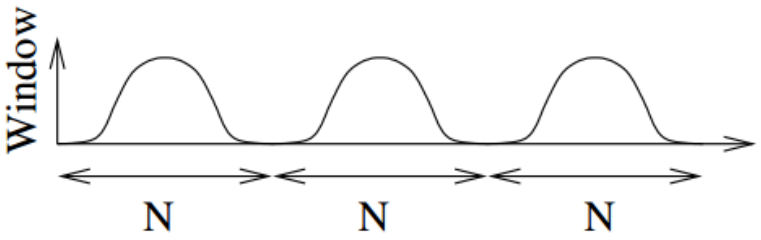


Figure 2.8. The overlap percentage is set to 0%. The windowing function tapers at its ends and causes data loss [12].

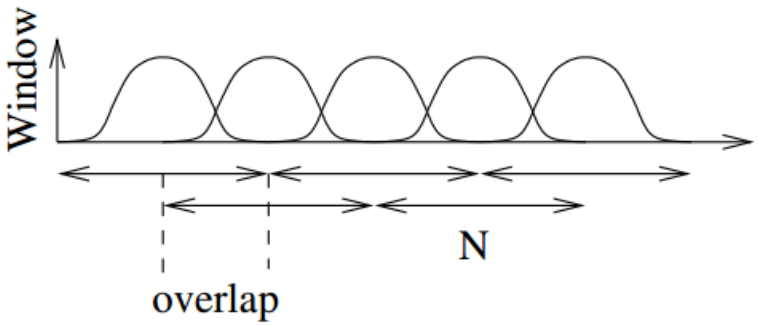


Figure 2.9. The overlap percentage is increased compared to the Figure 2.8 to prevent the data loss [12].

2.2. Microphone Arrays and Beamforming

Speech signals in most cases are collected in noisy environments. The receiving microphone can pick up noisy interference with varying patterns and amplitude. In certain scenarios, the objective would be to amplify the signal from the closest speech source to the microphone while eliminating other speech sources active in the vicinity. The reverberating environment is another factor which negatively impacts the signal quality received on the microphone [2].

To reduce the effect of aforementioned undesired signals, one popular method is using multiple microphones and process the signal collection. A microphone array that

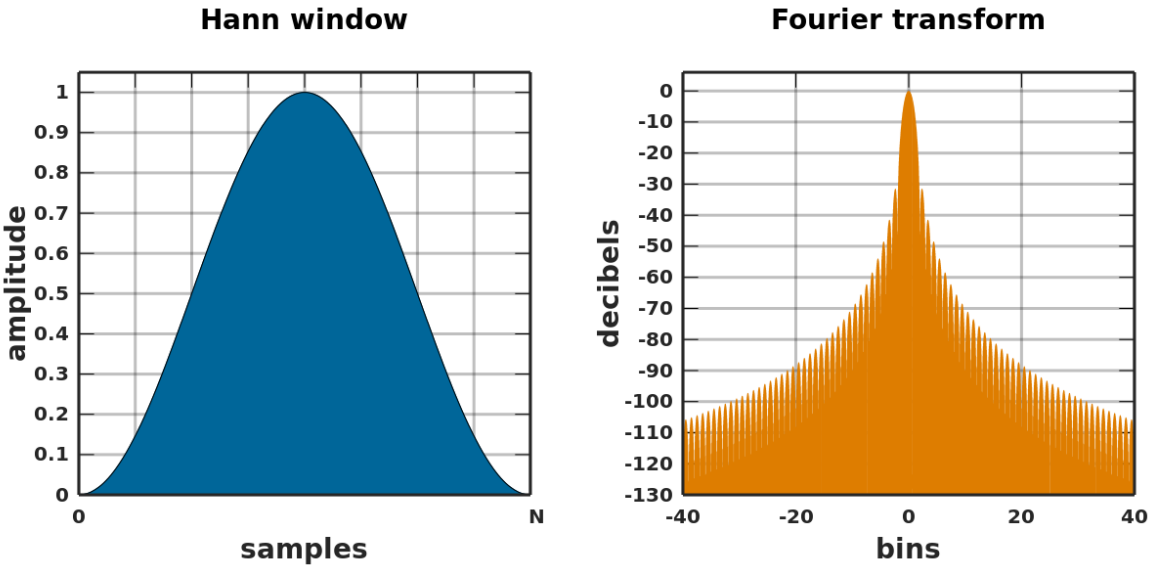


Figure 2.10. The window function based on the number of samples (left) and the Fourier transform (right) of the Hann window function can be seen above [13].

consists of multiple receivers enriches the time-frequency data collected by a single microphone with spatial information. The microphone array can be configured in various geometries such as a line, a circle or a sphere. The microphones placed in various setups captures not only the signals but also the spatial data useful to exploit to improve the quality and intelligibility of the speech source. The goal of using the microphone arrays could be singling out a source of the audio signal or increasing the quality of a speech source by using spatio-temporal data [1].

The geometries of the array vary based on the application. For certain applications like source localization, knowing the exact array geometry is crucial for the success of method. This reason causes line and circle to be popular due to their simplicity. In this thesis, the geometry of the array does not impact the speech enhancing neural network for the multiple array setup.

The array microphone structure is useful to solve a number of problems like noise reduction, dereverberation, source localization and cocktail party. The focus of this thesis will be the noise reduction problem. The main goal with the noise reduction is

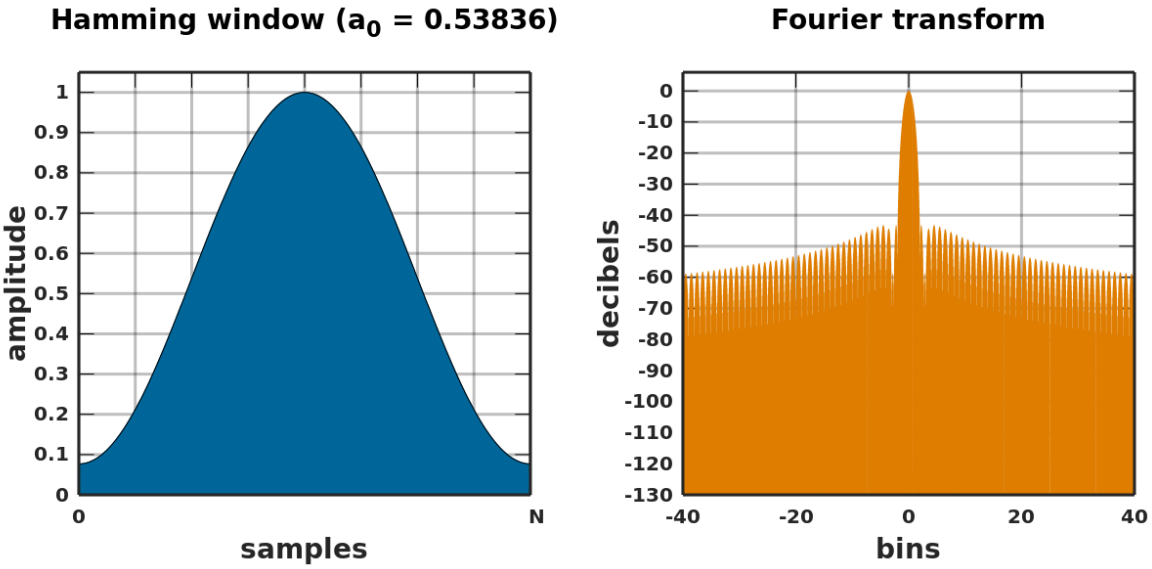


Figure 2.11. The window function based on the number of samples (left) and the Fourier transform (right) of the Hamming window function can be seen above [13].

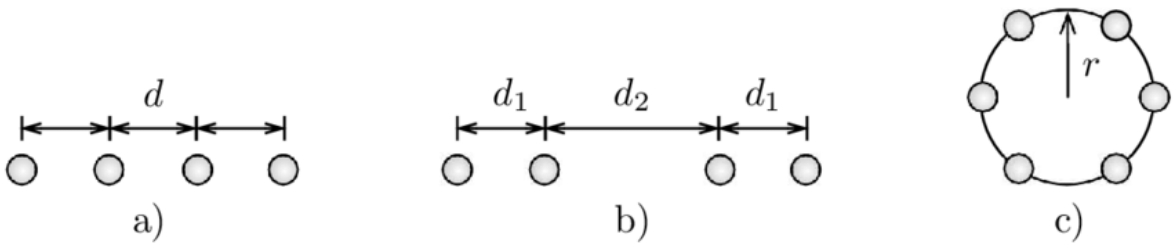


Figure 2.12. Various array geometries can be seen above. The simple geometries reduce the complexity of the problem [14].

to improve the quality and intelligibility of the speech source. The problem with using a single microphone is having the intelligibility of the speech degraded while improving the quality of the speech [15]. The added spatio-temporal data coming from the array microphone should be able to help with the trade-off of quality and intelligibility in theory. In certain applications, directional microphones are used in order to locate where the noise source is and only activate the channels coming from the microphones nearest to the speech source.

A problem the microphone array will have to solve is the signals that are not coming from the direct-path but those that are reflected and reverberated across the environment. This can be seen in the signal as the attenuated and delayed replicas of the original signal. The amount of delay and attenuation will be different in each microphone which indicates the potential location of that specific source.

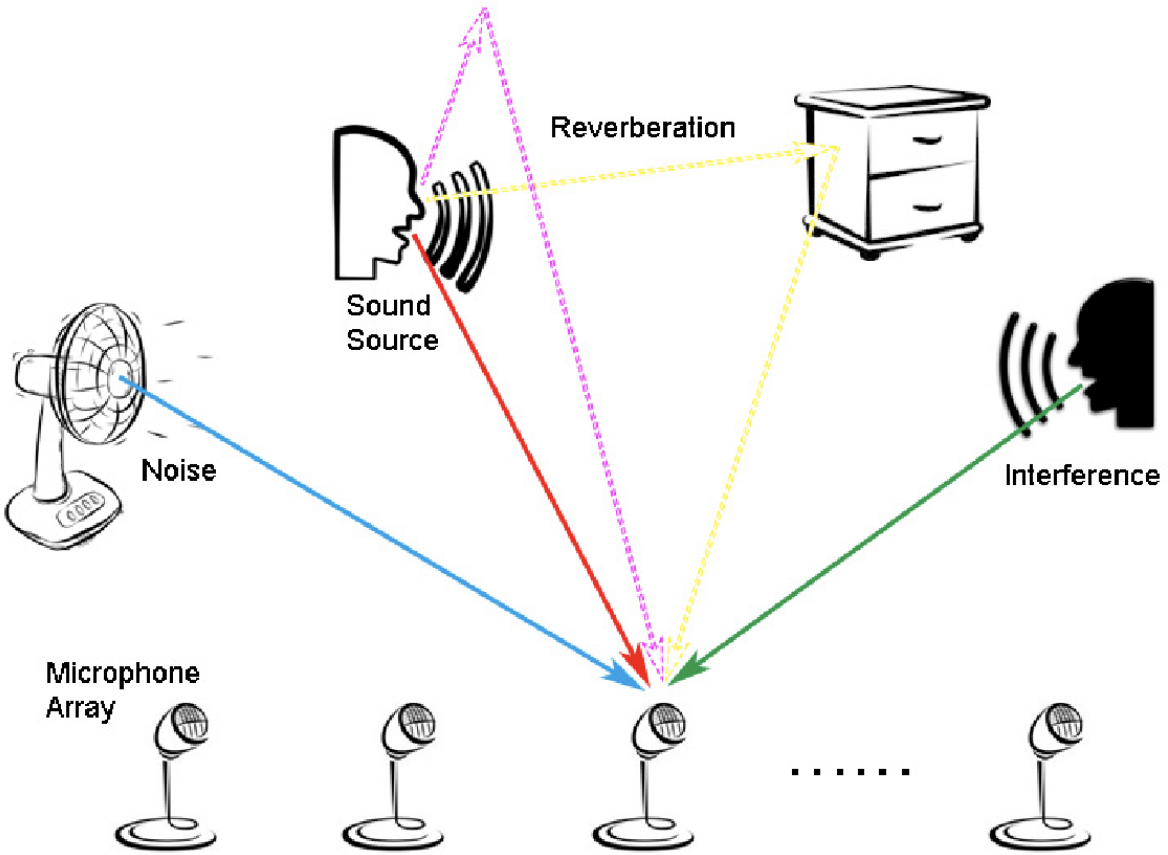


Figure 2.13. The microphone arrays introduce the spatial information missing in the single microphone case and improves the understanding of various sound sources as it can be seen in the diagram above [16].

In this thesis, the blind source separation (BSS) is not going to be investigated as beamforming is the main subject of the neural network that we will try to implement.

Beamformer is a filter that exploits the spatial data that is collected on the outputs of each microphone in order to create a beam pattern. There are traditional methods that employ the cross correlation algorithm to determine the time difference

in each microphone, delay the outputs, multiply with certain weights and sum the outcomes. This is known as the delay and sum beamformer. There are different ways to calculate the weighting coefficients of each microphone which will be explained in the following sections.

The microphone array output can be expressed with the Equation 2.8. This scenario assumes there are N microphones in this array. α_n represents the attenuation coefficient of each microphone, $s(k)$ is the source signal, t is the propagation time from the source signal to the first microphone. $v_n(k)$ corresponds to the noise signal. $\mathcal{F}_n(\tau)$ represents the time difference of arrival between microphone 1 and microphone n. We can say that $\mathcal{F}_n(\tau)$ is equal to 0 [1].

$$\begin{aligned} y_n(k) &= \alpha_n s\left[k - t - \mathcal{F}_n(\tau)\right] + v_n(k) \\ &= x_n(k) + v_n(k), n = 1, 2, \ldots, N \end{aligned} \tag{2.8}$$

The assumption is the $\mathcal{F}_n(\tau)$ and the relative delay can be found or estimated. The source signal and the noise signal are not correlated. The goal is to find $s(k)$ and reduce the effect of $v_n(k)$. This will improve the final output quality of the beamformer.

2.2.1. Delay and Sum Method

The delay-and-sum (DS) beamformer exploits the delay captured by the multiple microphones located close to each other [2]. This beamformer has two core steps. The first one is the time-shift applied to each signal captured at the microphones. A microphone is selected as the reference microphone and the time difference is calculated by aligning the signals onto each other.

$$\begin{aligned} y_{\mathrm{a},n}(k) &= y_n\left[k + \mathcal{F}_n(\tau)\right] \\ &= \alpha_n s(k - t) + v_{\mathrm{a},n}(k) \\ &= x_{\mathrm{a},n}(k) + v_{\mathrm{a},n}(k), n = 1, 2, \ldots, N \end{aligned} \tag{2.9}$$

The subscript a stands for the shifted version of the original signal.

The second step of the delay-and-sum beamformer is the summation. The time-shifted signals are summed up.

$$\begin{aligned} z_{\mathrm{DS}}(k) &= \frac{1}{N}\sum_{n=1}^{N} y_{\mathrm{a},n}(k) \\ &= \alpha_{\mathrm{s}} s(k-t) + \frac{1}{N} v_{\mathrm{s}}(k) \end{aligned} \tag{2.10}$$

where

$$\alpha_{\mathrm{s}} = \frac{1}{N}\sum_{n=1}^{N} \alpha_n \tag{2.11a}$$

$$\begin{aligned} v_{\mathrm{s}}(k) &= \sum_{n=1}^{N} v_{\mathrm{a},n}(k) \\ &= \sum_{n=1}^{N} v_n\left[k + \mathcal{F}_n(\tau)\right] \end{aligned} \tag{2.11b}$$

A good example for the DS beamformer can be seen in Figure 2.14. One improvement to the previous method explained above is using different weights for each microphone. This allows certain signals to be strengthened even further and would be employed if it's known that the specific microphone has higher chance of capturing speech content while the other microphones are receiving more noise content.

2.2.2. Minimum Variance Distortionless Response Filter (MVDR)

This is one of the most widely used adaptive beamformers [3]. As this is an adaptive beamformer, it is expected for an MVDR beamformer to adapt itself to the noise surrounding the microphones. The logic behind the MVDR filter is finding the weights that reduce the output power while being subject to the following constraint

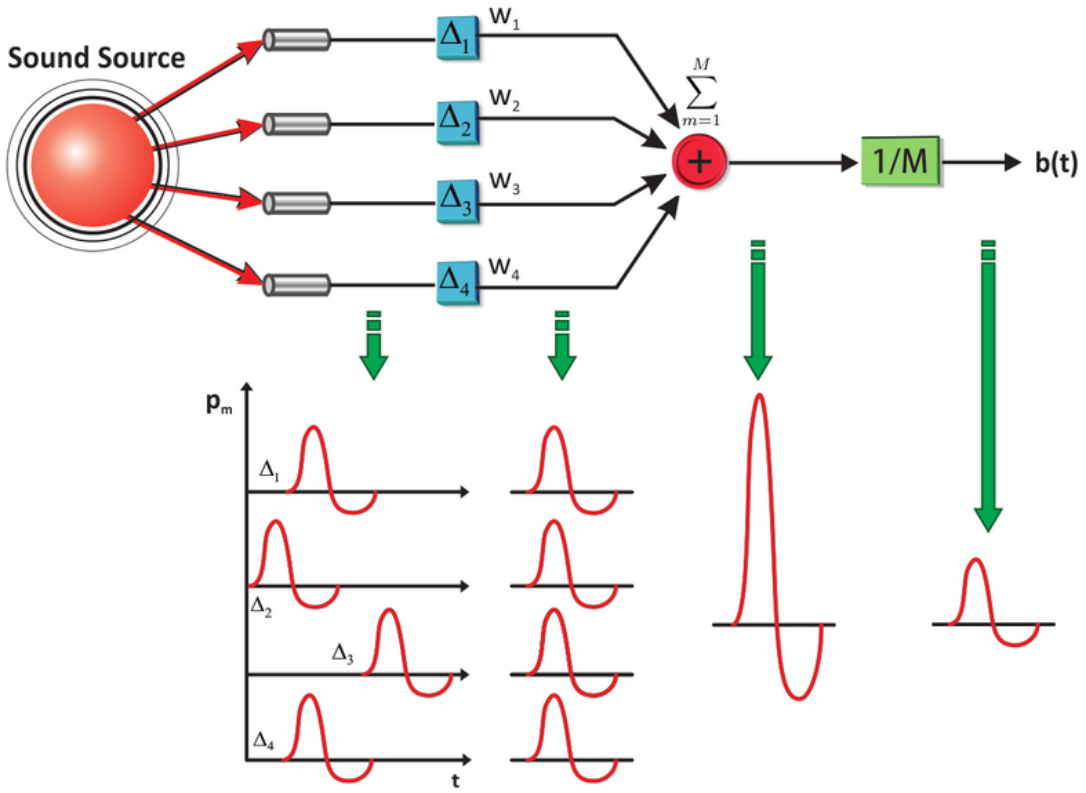


Figure 2.14. A visual explanation of the weighted delay-and-sum beamformer can be seen. Each microphone is shifted in order to align all the signals and multiplied with a coefficient. The aligned and weighted signals are then summed and normalized [17].

in Equation 2.12:

$$\min_{\mathbf{h}} \mathbf{h}^T \mathbf{R}_{y_a y_a} \mathbf{h} \quad \text{subject to} \quad \mathbf{h}^T \boldsymbol{\alpha} = \alpha_1 \tag{2.12}$$

The weights are represented by h while the output power is represented by $E\left[z^2(k)\right] = \mathbf{h}^T \mathbf{R}_{y_a y_a} \mathbf{h}$.

An important variable is the covariance matrix $\mathbf{R}$. It can be expressed as

$$\mathbf{R}_{y_a y_a} = \sigma_s^2 \boldsymbol{\alpha}\boldsymbol{\alpha}^T + \mathbf{R}_{v_a v_a} \tag{2.13}$$

where

$$\mathbf{y_a}(k) = [y_{\mathbf{a},1}(k) y_{\mathbf{a},2}(k) \cdots y_{\mathbf{a},N}(k)]^T \tag{2.14a}$$

$$\mathbf{v_a}(k) = [v_{\mathbf{a},1}(k) v_{\mathbf{a},2}(k) \cdots v_{\mathbf{a},N}(k)]^T \tag{2.14b}$$

$$\boldsymbol{\alpha} = \left[\ \alpha_1 \alpha_2 \cdots \alpha_N \ \right]^T \tag{2.14c}$$

represent the variables.

In this case, $\mathbf{R}_{v_\mathrm{a} v_\mathrm{a}} = E\left[\mathbf{v}_\mathrm{a}(k)\mathbf{v}_\mathrm{a}^T(k)\right]$ stands for the noise covariance matrix. In order to solve this optimization problem, the Lagrange multipliers will be employed.

$$\mathbf{h}_\mathrm{C} = \alpha_1 \frac{\mathbf{R}_{y_\mathrm{a} y_\mathrm{a}}^{-1} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{R}_{y_\mathrm{a} y_\mathrm{a}}^{-1} \boldsymbol{\alpha}} \tag{2.15}$$

The subscript C stands for Capon who is the author of the MVDR filter and another name for this filter in Equation 2.15. The output of the MVDR filter is

$$z_\mathrm{C}(k) = \mathbf{h}_\mathrm{C}^T \mathbf{y}_\mathrm{a}(k) = \alpha_1 \frac{\boldsymbol{\alpha}^T \mathbf{R}_{y_\mathrm{a} y_\mathrm{a}}^{-1} \mathbf{y}_\mathrm{a}(k)}{\boldsymbol{\alpha}^T \mathbf{R}_{y_\mathrm{a} y_\mathrm{a}}^{-1} \boldsymbol{\alpha}} = x_1(k) + r_\mathrm{n}(k) \tag{2.16}$$

where

$$r_\mathrm{n}(k) = \alpha_1 \frac{\boldsymbol{\alpha}^T \mathbf{R}_{y_\mathrm{a}}^{-1} \boldsymbol{y}_\mathrm{a} \mathbf{v}_\mathrm{a}(k)}{\boldsymbol{\alpha}^T \mathbf{R}_{y_\mathrm{a} y_\mathrm{a}}^{-1} \boldsymbol{\alpha}} \tag{2.17}$$

represents the residual noise.

In the case of reverberation, a more general algorithm is called for. The extended version of the MVDR filter is the linearly constrained minimum variance filter (LCMV). It can be said that the MVDR filter is a specific case of the LCMV filter [3]. Certain

beamforming algorithms that employ deep learning focus on finding the MVDR coefficients to improve the beamforming performance. The coefficients found are then employed to the problem at hand.

2.2.3. Linearly Constrained Minimum Variance Filter (LCMV)

Linearly constrained minimum variance (LCMV) beamformer or also known as the Frost beamformer due to its inventor is an adaptive least mean squares method to the same optimization problem.

The LCMV beamformer can be seen as a particular case of the Wiener filter [1]. In real life scenarios, the reference signal is not known.

$$y(k) = x(k) + v(k) \tag{2.18}$$

where $x(k)$ is the zero mean clean speech signal and $v(k)$ is the noise process that is uncorrelated with the speech signal which their summation is the resulting the signal collected. The error signal can be defined as

$$e(k) = x(k) - z(k) = x(k) - \mathbf{h}^T\mathbf{y}(k) \tag{2.19}$$

where

$$\mathbf{h} = \begin{bmatrix} h_0 & h_1 \cdots h_{L-1} \end{bmatrix}^T \tag{2.20}$$

is a finite impulse response (FIR) filter with the length L.

$$\mathbf{y}(k) = [y(k)y(k-1)\cdots y(k-L+1)]^T \tag{2.21}$$

The samples of the observation signal represented with $y(k)$, and

$$z(k) = \mathbf{h}^T \mathbf{y}(k) \tag{2.22}$$

is the output of the filter $\mathbf{h}$. The MSE criterion can be written as follows [18]:

$$J(\mathbf{h}) = E\left[e^2(k)\right] = \mathbf{h}^T \mathbf{R}_{yy} \mathbf{h} - 2\mathbf{r}_{yx}^T \mathbf{h} + \sigma_x^2 \tag{2.23}$$

where $E[\cdot]$ denotes mathematical expectation,

$$\mathbf{R}_{yy} = E\left[\mathbf{y}(k)\mathbf{y}^T(k)\right] \tag{2.24}$$

stands for the correlation matrix, and

$$\mathbf{r}_{yx} = E[\mathbf{y}(k)x(k)] \tag{2.25}$$

is the cross-correlation vector between clean speech signals and noisy speech signals.

The issue with the following formula is with the access to the reference signal $x(k)$. If this is set to zero, the equation above becomes the following:

$$J(\mathbf{h}) = \mathbf{h}^T \mathbf{R}_{yy} \mathbf{h} \tag{2.26}$$

In various applications, the following constraint is also given for the filter $\mathbf{h}$:

$$\mathbf{C}^T \mathbf{h} = \mathbf{u} \tag{2.27}$$

where $\mathbf{C}$ stands for the constraint matrix and $\mathbf{u}$ is defined as

$$\mathbf{u} = \begin{bmatrix} u_0 & u_1 \cdots u_{L_c-1} \end{bmatrix}^T \tag{2.28}$$

The following optimization problem now presents itself:

$$\min_{\mathbf{h}} J(\mathbf{h}) \quad \text{subject to} \quad \mathbf{C}^T\mathbf{h} = \mathbf{u} \tag{2.29}$$

Similar to the MVDR beamformer, in Equation 2.30, the Lagrange multipliers can be employed to solve this optimization problem to find the optimal filter [19]:

$$\mathbf{h}_{\mathrm{F}} = \mathbf{R}_{yy}^{-1}\mathbf{C}\left(\mathbf{C}^T\mathbf{R}_{yy}^{-1}\mathbf{C}\right)^{-1}\mathbf{u} \tag{2.30}$$

However, it can be seen from the equation above that the correlation matrix $\mathbf{R}_{yy}^{-1}$ must be invertible for this to work. Also $\mathbf{C}$ has to have full column rank.

In the case of $L_{\mathrm{c}} = 1$, the constraint matrix $\mathbf{C}$ turns into a constraint vector and the solution presents itself in the form of the MVDR or Capon filter proving that the MVDR filter is a special case of the LCMV or Frost beamformer.

2.2.4. Generalized Sidelobe Canceller

The generalized sidelobe canceller (GSC) tackles the same problem as the LCMV filter. It converts a constrained optimization problem into an unconstrained optimization problem [20].

The details of the GSC will not be presented. LCMV and GSC have been proved to be equivalent [21].

2.3. Performance Evaluation Methods

In order to measure the quality of the speech enhancement systems, certain performance evaluation methods are used. Four common evaluation methods will be referred in this thesis. Those are: subjective short-time objective intelligibility (STOI), perceptual evaluation of speech quality (PESQ), mean opinion score (MOS) and word

error rate (WER).

Mean opinion score (MOS) is the simplest of the aforementioned evaluation methods to implement. It is commonly used to rate the speech quality and intelligibility. The MOS is conducted by presenting audio samples to a test group of certain people and asking those people to rate the audio signal from 1 to 5, where 1 is the lowest perceived quality and 5 is the highest perceived quality. After the test conducted, the arithmetic mean of the rating scores for each sample is calculated. The mean found will be the MOS score of the audio sample [22].

Perceptual evaluation of speech quality (PESQ) is a method introduced as a new way to evaluate the speech quality in communication networks [23]. As opposed to the mean opinion score, the PESQ method is comparatively more objective and reliable. This made PESQ one of the most popular evaluation metrics.

The PESQ score starts from -0.5 and the maximum score is 4.5. A higher number would indicate a better intelligibility score. The method takes the target speech and the audio signal to be evaluated. The PESQ method is a full reference algorithm as it requires a reference signal to function.

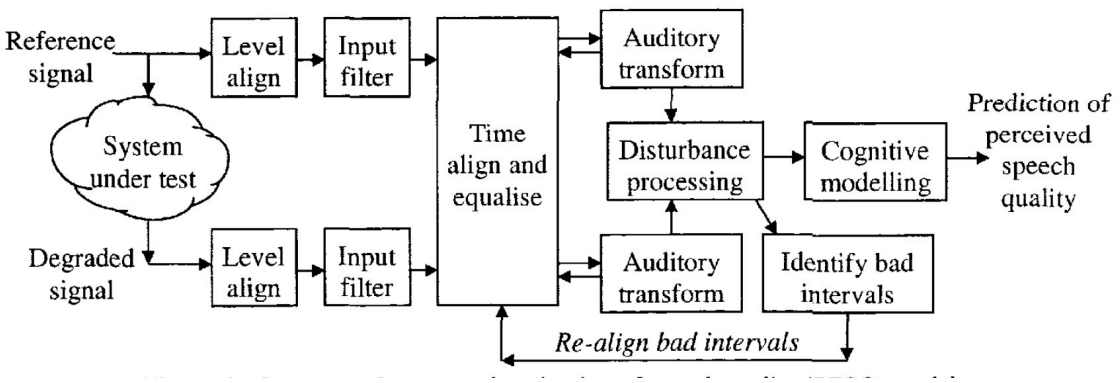


Figure 2.15. The structure of the PESQ method [23].

Subjective short-time objective intelligibility (STOI) is an evaluation method developed as an alternative objective intelligibility measure [24]. The main goal behind

the development of this method is to deliver better evaluation performance for signals processed by time-frequency weighting such as the noise reduction algorithms implemented in this thesis.

In a similar structure to the PESQ method, the STOI evaluation method also requires both the target speech and the processed signal. It produces an average intelligibility score where the higher score would mean better intelligibility is obtained.

The final metric mentioned in this section is the word error rate (WER) [25]. This method is most frequently used in speech recognition algorithms but serves as a useful metric to calculate to find the performance of the speech enhancement systems.

The WER requires both the target speech and the filtered speech signal to go through a speech recognition software. The predicted word sequences would be the inputs of the word error rate method. This method utilizes two measures: match error rate (MER) and word information lost (WIL). The former stands for the erroneous input/output word matches wile the latter is the approximation of the word information lost in the processing. The system uses the Viterbi alignment method for the calculations.

2.4. Deep Learning

The application developed in this master's thesis is built by the deep learning algorithms. This section will review the deep learning concepts as it is an integral part of this thesis.

2.4.1. Motivation

Deep learning is a type of machine learning which utilizes the deep artificial neural networks. Deep learning models generalize well on problems such as image classification, speech recognition and natural language processing [26]. Deep learning

essentially extracts higher level features progressively during training where the raw data provided goes through multiple layers.

The improvements in the computing power through the graphical processing units (GPU) and creation of massive datasets on various fields made it possible for the deep learning algorithms to solve complex problems in nature.

A technique called greedy layer-wise pretraining enabled the models to train much faster by allowing the weights of hidden layers to update minimally [27]. It is shown in image recognition field that deep learning models perform significantly better than the models designed with hand-designed features and methods [28].

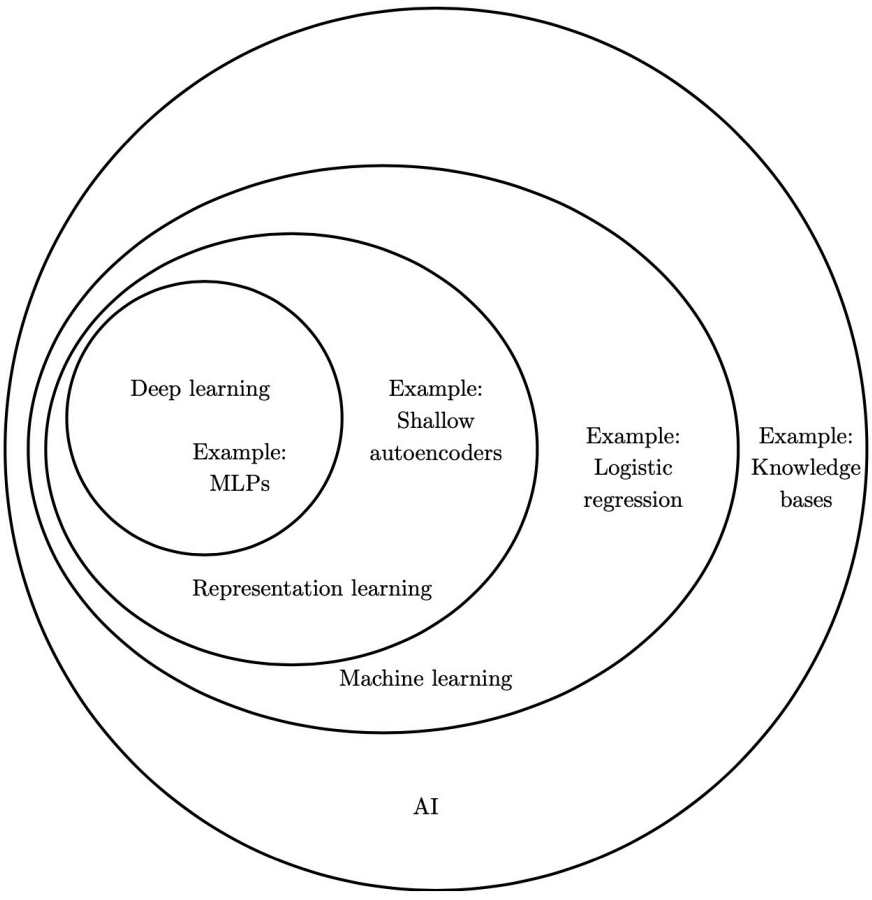


Figure 2.16. The diagram of artificial intelligence (AI) where it can be seen deep learning is a subset of the larger machine learning field. [26].

Speech enhancement is a particular area of interest which the deep learning can be applied to. Non-stationary nature of the added noise and the diversity of the noise types along with the reverberation in the room make the task increasingly difficult. In addition to the variations in noise, the speech signal can vary based on the person and the environment. These challenges make the task of building a generalized model to enhance the speech signals suitable for the deep learning algorithms. In the literature, deep learning based speech enhancement has become more popular in the recent years ([29], [30], [31], [32], [33], [34]). This thesis will be employing deep learning methods used for speech enhancement and configure them for the multichannel scenario.

In the next sections, related work for the beamforming using deep learning will be explained in detail.

2.4.2. Unsupervised and Supervised Learning

Machine learning algorithms can be classified into two main categories: supervised and unsupervised learning algorithms [26]. The supervised learning algorithms are trained on datasets with samples and their associated target or label. As an example, a model made to recognize car models and year will have a dataset with all the car photos and associated model and year name. The model will be trained on this dataset and base its learning on the associated labels for each training sample.

In unsupervised learning, there is no teacher guiding the model towards the goal. The algorithm will need to analyze, understand the dataset and its probability distribution. The most popular unsupervised learning algorithm is the clustering. It is primarily used to extract the patterns and "clusters" present in the data. k-means clustering is a good example to unsupervised learning where k distinct clusters are created based on samples distance to each cluster's center [35].

Another example to the unsupervised learning methods is the autoencoder algorithm. The autoencoder neural network aims to set the input values equal to the

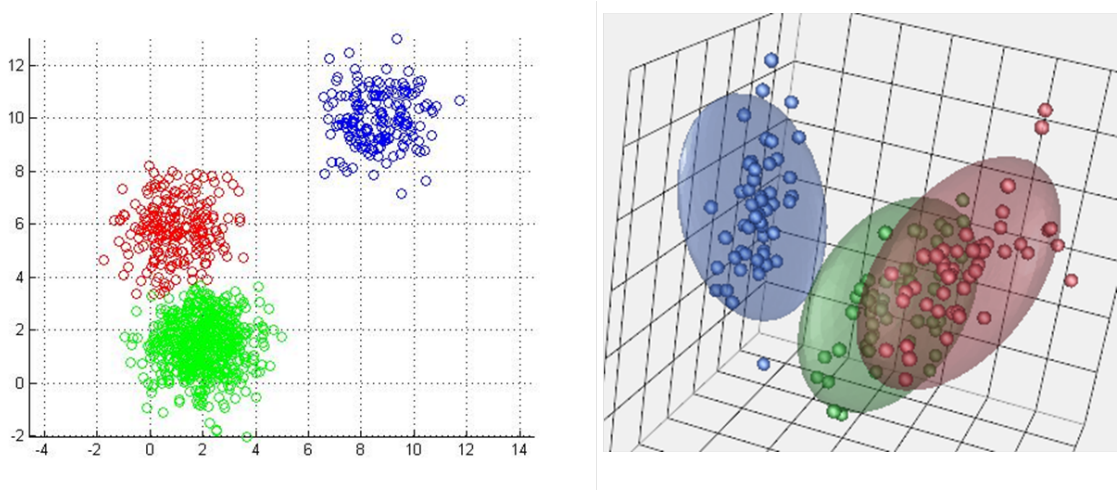


Figure 2.17. A dataset where k-means is applied to a 2D dataset (left) and to a 3D dataset (right). The center of each cluster can be seen and the samples belonging to the clusters are colored differently [36].

target values. What an autoencoder does is to learn a function that will transform the input into the target output. The model can be configured with various constraints and the complexity can be worked on by increasing the level of depth. This thesis will be employing an autoencoder neural network in order to denoise the audio signals.

2.4.3. Convolutional Neural Networks

Convolutional neural network is a specialized class of deep neural network mainly used for analyzing imagery data but increasingly used in other areas [37]. This type of network utilizes a mathematical operation called "convolution". It is a type of linear operation. In Figure 2.18, the convolution operation which is a running matrix operation is displayed. The convolution operation takes place in the layers of the convolutional neural networks (CNN) which is where the name comes from.

The convolution can be in 1-D for a regular time series data, in 2-D for the image data or 3-D for event detection in videos or 3-D medical images. Each type of convolution extracts features from the dimension space it operates. In Equation 2.31, $\mathbf{x}$ is the input function, $\mathbf{w}$ is the kernel or the feature space and $*$ stands for the

convolution operator. The time index **t** only takes integer values and the formula below is the convolution operation for the discrete case.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.31}$$

The data that is worked on in most cases is multidimensional. Therefore, the convolution operation should be conducted in all dimensions. Equation 2.32 stands for a 2-D kernel **K** convolving with a 2-D input function **I**.

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n) \tag{2.32}$$

The operation can be thought of as the kernel striding the input and the dot product is computed at each step.

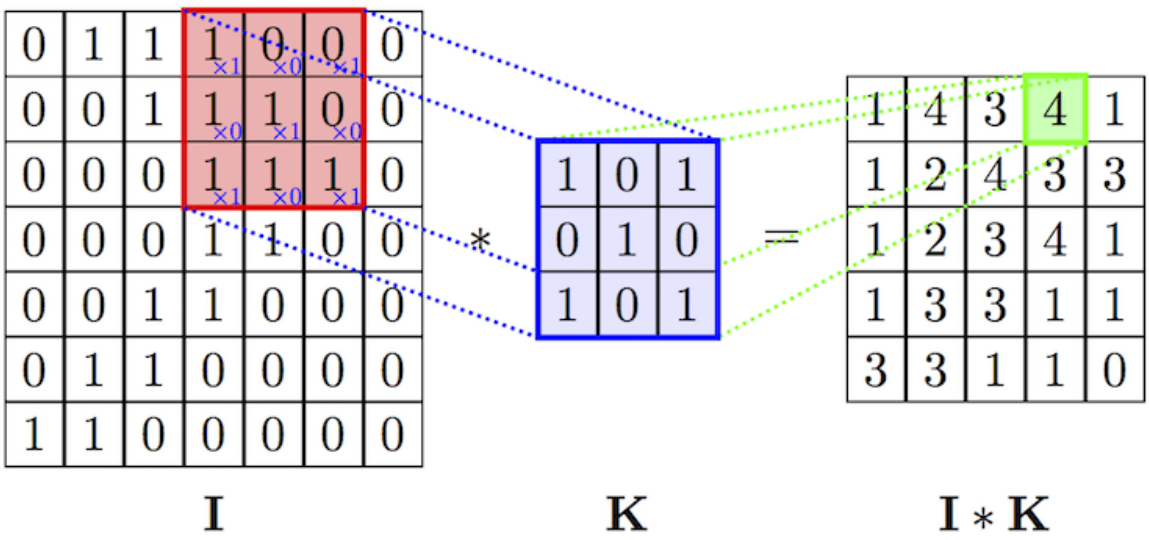


Figure 2.18. A basic visualization of the convolution operation where **I** is going through a convolution operation with the filter **K** resulting in the matrix on the right [38].

A convolutional neural network can be broken down to three segments: input layer, output layer and hidden layers. The hidden layers contain convolutional layers with various kernel sizes and filter numbers followed by the activation function and

additional optional layers such as the normalization, and the pooling layer. In this thesis, convolutional layers are frequently used in the neural network models and the architecture is similar to a CNN.

2.4.4. Autoencoders

Autoencoders are a kind of unsupervised learning algorithms that attempt to copy the input to the output [26]. An autoencoder has two simple parts: encoder and decoder. The encoder can be represented with $\boldsymbol{h} = f(\boldsymbol{x})$ and the decoder can be represented with $\boldsymbol{r} = g(\boldsymbol{h})$. However, the autoencoder's main goal is not to achieve the following result: $g(f(\boldsymbol{x})) = \boldsymbol{x}$.

The autoencoders are trained so that the output data can approximate the input data but not become its exact copy. The model in essence is forced to learn the most critical aspects of the data and drop the least important ones. In a way, autoencoders create a compressed knowledge space and reconstruct from that knowledge space to obtain the output.

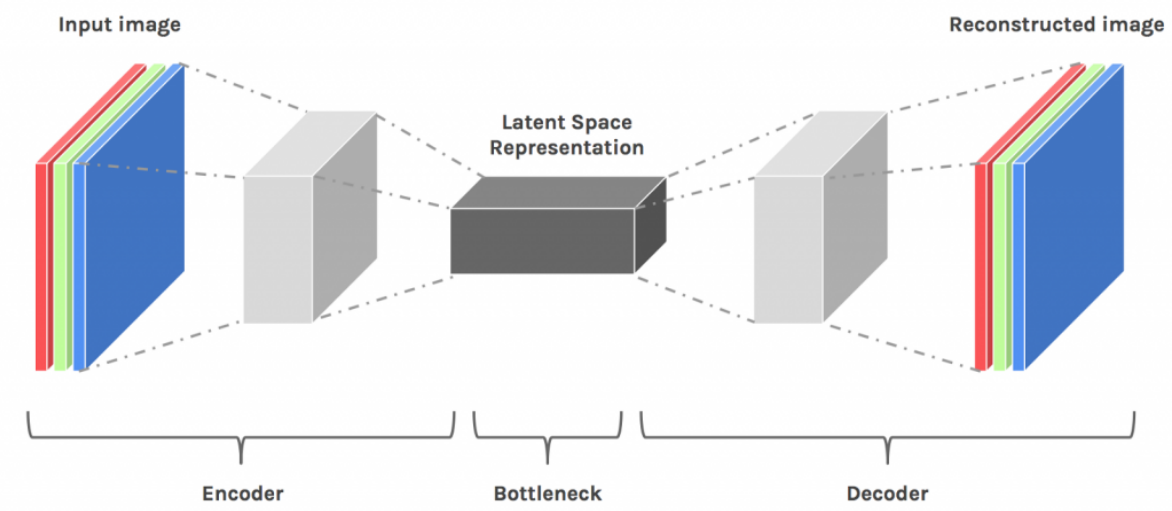


Figure 2.19. A visualization of the stages of an autoencoder. The input goes through the encoding then the decoding phases for reconstruction [39].

In particular, this thesis will be focusing on the denoising autoencoders. These autoencoders tries to minimize the following loss function

$$L(\boldsymbol{x}, g(f(\tilde{\boldsymbol{x}}))), \tag{2.33}$$

where $\tilde{\boldsymbol{x}}$ is the duplicate of $\boldsymbol{x}$ which has been corrupted with some added noise. The main goal of the denoising autoencoders is to eliminate the noise present in the input function.

There are two main expectations from an autoencoder model: sensitive to the input enough to reconstruct the signal and insensitive enough to generalize on the approach thus does not overfit.

3. RELATED WORK

3.1. Single Channel Speech Enhancement Using Deep Neural Networks

The speech enhancement challenge is broken down to two categories in this thesis: single channel and multichannel. In particular, the use of neural networks have become more popular in recent years. Earlier speech enhancements methods focused on the noise profile estimation and its use in order to denoise the speech signals ([40], [41], [42]). The noise-deducted speech using the noise profiles previously does not perform as well when the noise is another human speech or noise that is unbeknownst to the model.

The performance of feature extraction for the speech signals by the use of CNNs have been shown [43] [44]. The CNN use will be a frequent theme in the speech enhancement research that will be investigated in this thesis.

Park and Lee [31] proposes a network of redundant convolutional encoder-decoder (R-CED) in order to achieve high performance when babble noise is present. The aim of this research is to improve the efficacy of the neural networks for the denoising that takes place in hearing aids using a fully convolutional neural network. This approach is an improvement to an earlier model called Convolutional Encoder-Decoder Network (CED) [45].

Proposed R-CED network uses the noisy spectra and the clean spectra to train the autoencoder. The autoencoder is tasked with finding a function that maps the noisy input data to the clean input data. An important factor to note in this research paper is the use of signal STFT as the main input to the system. As opposed to the time-domain, the frequency domain representation allows a compact form of input data to be used. The model uses the TIMIT dataset where the clean speech dataset is corrupted with a variety of noise [46]. The SNR of the noise is also subject to change in the dataset for the better representation of the real world. The input data takes 7

previous STFT frames along with the exact frame match of the clean signal's STFT frame.

The noisy speech input first goes through an encoder with each layer made up of a CNN filter, a batch normalization filter and a ReLU regularization filter with different sizes. After the encoding stage, the decoder stage begins while the structure of each layer remaining similar. A skip connection is also added between the encoder and the decoder stages which are visible in Figure 3.1.

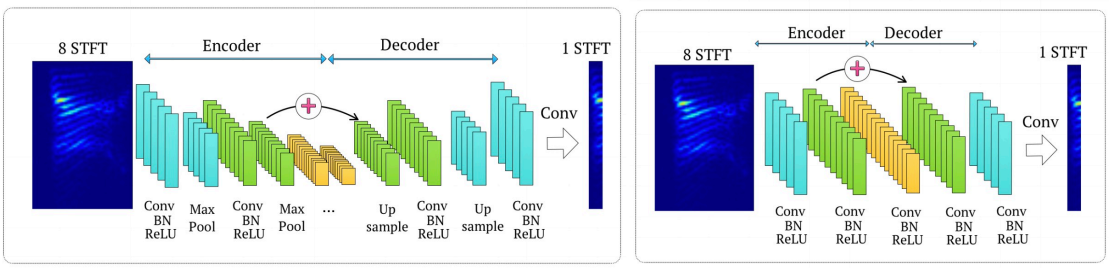


Figure 3.1. The CED network (left) and the R-CED network are displayed [31].

In Figure 3.1, the figure on the left refers to the CED network while the figure on the right refers to the R-CED network. The former model is more complex in terms of the network structure compared to the R-CED and the ladder model is the CED's compressed version with less layers.

In order to train the neural networks to perform speech enhancement, the input type selection is important. Magnitude spectrum as it can be seen in the work of Park and Lee [31] is one of the most popular input features. It brings the audio signals into a compact and normalized form and bring out the frequencies. However, this approach is unable to take the phase into account and the training is only performed on the magnitude spectrum. The assumption is the phase does not change as drastically when the noise is added. In order to alleviate this problem, the use of raw speech signals in time domain is becoming more popular ([32], [33], [47], [48]).

Pascual et al (2017) proposes a model that trains on the raw waveform directly [30]. The model utilizes the generative adversarial networks (GAN) to tackle the speech enhancement problem. It works end-to-end using the Voice Bank corpus [49]. This is the first use of GANs in the speech enhancement area. The advantages of this approach are the use of raw waveform, no effort on feature extraction and the ability to train without a recursive operation as in recursive neural networks (RNN). The model in this network is built in a way that is similar to an autoencoder model. The raw waveform compressed through a number of convolutional layers with a parametric rectified linear units (PReLU) [50]. After the condensed representation is achieved, the decoding phase begins in a structure similar to the encoding stage but in a reversed fashion. An important distinction of this research as opposed to the other ones is the use of 16 kHz frequency rate.

Rethage et al (2018) proposes a speech enhancement model taking advantage of the Wavenet in order to retain the phase information ([33], [51]).

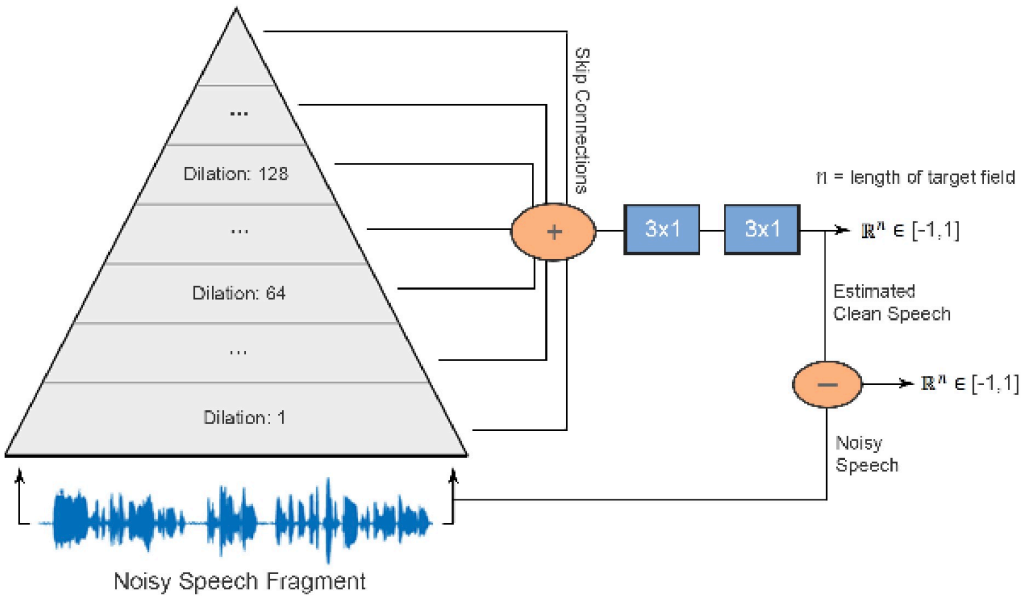


Figure 3.2. The overview of the wavenet for noise reduction model [33].

As previously explained, using the magnitude spectrum has the trade-off of losing the phase information. This research paper constructs an end-to-end system with the raw waveform. Wavenet is essentially developed to synthesize speech. The model uses some of the overlapping needs of noise reduction and speech synthesis. It has access to both the past and future samples of the audio signal. Compared to the baseline Wiener filter, this model performs better in terms of the mean opinion score [42].

3.2. Multichannel Speech Enhancement Using Deep Neural Networks

Multichannel or multi-microphone speech enhancement is a more complex problem compared to the signal channel scenario. Multichannel scenario includes signals that are captured by microphones with some physical distance to each other. This creates certain amount of delay in the signal captured and each channel includes a different projection of the speech signal and the noise variants present in the environment. In addition to these factors, the reverberation is more noticeable in the sample data.

There are various conventional beamforming methods explained in the beamforming section of this thesis. These methods exploit the spatial feature of the multichannel data in order to improve the end result and intelligibility of the speech signal.

A particular difficulty in multichannel scenario is the size of data required to train the models. The data is multiplied by the number of channels added in the training set. Given the importance of using large datasets to be able to capture the hidden patterns, it is important for the model to be trained on as much data as possible. Certain models that will be investigated trained on datasets that were much larger compared to the datasets in the single channel scenario.

Sainath et al (2017) proposes a multichannel signal processing deep learning model to improve the automatic speech recognition (ASR) [52]. This model is another example of an end-to-end model using the raw waveform. It includes localization, beamforming, post-filtering alongside with the acoustic modeling in a single model with

the aim of lower word error rate. This paper, in contrast to models with stages independently tuning the signal, proposes a network architecture where the entire system focuses on the goal of better ASR.

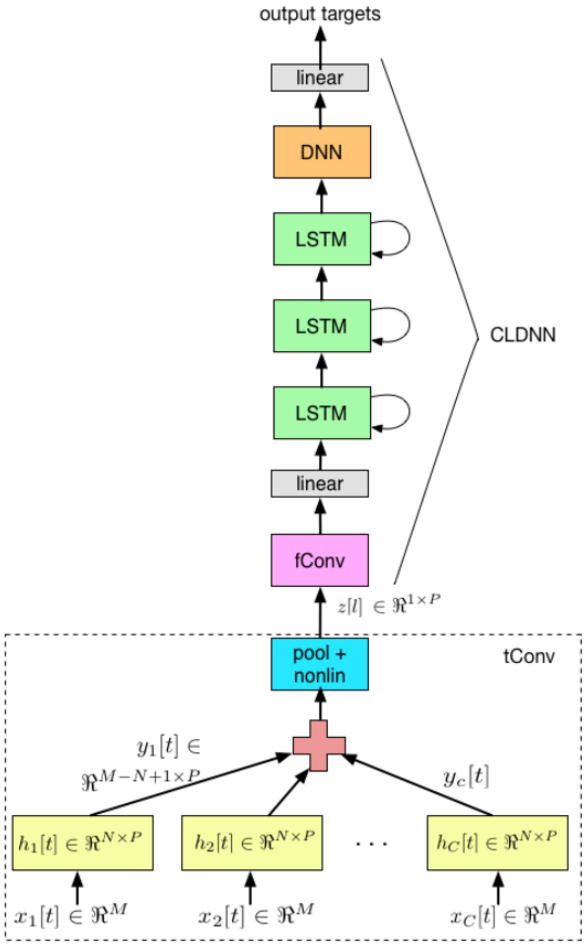


Figure 3.3. The architecture of the CLDNN acoustic model [52].

The first stage of the model takes the inputs from all microphones. Each input signal goes through separate time convolution filters and mapped down to a single time-frequency representation. In this layer, the bandpass filters go through a learning phase. The output of this stage is taken through the convolutional long short-term memory deep neural network (CLDNN) model [53].

The model is trained on a dataset of 2000 hours with varying SNRs of diverse noise catalogue. The audio is captured by 8 microphones. In order to make the data even more authentic, the signals are rerecorded with the help of a mouth simulator, a

7 channel circular microphone array and a living room setting.

The comparisons are made to a CLDNN acoustic model using log-mel features. The improvement in the word error rate (WER) can be seen in the paper. A number of factors like the filter number, the channel count, the type of CLDNN have an impact in the change in WER.

4. DATA and METHODOLOGY

4.1. Motivation

Deep learning has become quite successful in solving problems in a much more efficient way. Image processing, speech recognition and speech enhancement are some of those areas. Denoising or speech enhancement in the multichannel domain is the subject of that

4.2. Dataset Creation

The dataset is one of the most crucial parts of this thesis. Due to the operating structure of the neural networks, the model needs to be trained on a dataset large and diverse enough to cover many diverse scenarios to be able to extract general patterns. The speech enhancement is one of those areas where the type of the speech data can vary based on the source. A male or female voice, different emotions, tones, the environment are all factors that can differentiate the audio signal. The noise cases are as diverse as the speech signal. The noise could be a static signal or a signal which changes with time. The type of the noise signal could also vary by its type. An air conditioner in an office or a car exhaust would have a significantly different sound profiles and spectra.

An important detail to note is the type of the dataset profile created. A model that is trained on a dataset consisting of only a specific kind of noise will perform well if the noise profile in the test set matches the training set's noise profile. However, a different type of noise will cause the model to perform poorly.

One of the challenges in this thesis is to generalize the diverse cases while keeping the dataset limited in order to train the model quickly. In this thesis, the focus is placed on scenarios where the human speech is improved in urban environments such as a cafe, a conference room, an office or a home. The examples can be multiplied

but the general logic is the types of environment and therefore, the types of noise are limited. This fact enabled the dataset to be in a more compact form than anticipated.

Collecting the speech samples is another case where the variation is important for enhancement and overall generalization. In order to achieve this, the dataset had to be a combination of various publicly available speech datasets that includes multiple speakers.

The last part of the dataset creation of this thesis is the multi-microphone setup. A speech enhancement problem with a single channel is a comparatively simple scenario. The noise is added on top of the speech signal. The SNR can be adjusted programmatically. The reference signal and the noise signal would match frame by frame. The case of the multi-microphone requires a different set of adjustments. For the dataset to have variation, it is important for the source signal and the noise signal to come from a 3-D environment. There are a number of ways to achieve this. The details of how it is configured will be explained in the following sections.

4.2.1. Speech Signal Datasets

Speech signal is one of the three components of the dataset creation process. There are a number of options publicly available for the speech enhancement use case. For the neural network model to adapt to the variations of human speech, it is essential to use a diverse set of speech samples.

Another important factor to consider is the quality of the speech samples. Speech signals recorded in already a noisy environment or with a low quality microphone would distort the target signal. The model would have the goal of turning a noisy signal into another noisy signal and the model would perform poorly.

The last consideration made in selecting the speech sample datasets is certain fixed patterns present in the entire dataset. An example is the presence of an air

conditioner that transmits a noise that is present in the exact same form across the entire dataset. The presence of a constant pattern in the target set has the effect of gearing the model towards creating a noisy interference to match the goal.

The first dataset used to generate the dataset is the Texas Instruments Massachusetts Institute of Technology (TIMIT) corpus [46]. TIMIT corpus contains the recordings of 630 speakers each uttering 10 sentences in English language. The data is recorded in a single channel with the sample rate of 16 000. This corpus became the cornerstone of the dataset created for this thesis due to its variety of speakers and the quality of the recordings.

A more recently developed speech dataset came from Google. Speech Commands is developed as an alternative to the other datasets with much longer speech samples [54]. This dataset only includes short phrases such as “yes”, “no”, “bird” and “dog”. It includes 2618 speakers recording 35 different words. The diversity of speakers is a big advantage of this model. However, there is a major disadvantage of this dataset. It only includes a very limited portion of human speech and the dataset only includes short words as opposed to the sentences. This inhibits the ability of the neural network model to learn the general structure of common forms of human speech. This dataset also included certain noise samples such as running water or office noises. The noise dataset is not included in this thesis dataset.

Mozilla Common Voice corpus is the largest of all the datasets that has been considered for this thesis [55]. Its speech catalogue continues to grow with more than 30 languages and 2500 hours of speech content. Two issues regarding the Common Voice corpus makes it difficult to use in speech enhancement case. The first problem is the file format MP3. MP3 is a lossy format for audio signals to compress the audio files [56]. As opposed to the datasets with audio files stored in lossless WAV format, a lossy format like MP3 is not used in this thesis. Another issue with this dataset is the varying quality of recordings. Since the main goal of this dataset is to collect speech in multiple languages to use in speech recognition, there are many recordings with noise

present in the background. The speech quality is not constant and making it a difficult choice to use in the multi-channel dataset task for this thesis.

The LibriSpeech corpus contains 1000 hours of audiobook recordings sampled at 16 kHz [57]. Compared to the rest of the datasets investigated for this thesis, the samples are one of the cleanest in terms of the noise and the recording quality. The corpus includes 2484 speakers with male and female speakers equally distributed.

The last speech dataset considered for this thesis is the CSTR VCTK Corpus. It is a comparatively limited dataset with 110 speakers reading 400 sentences from newspapers [58]. The set contains recordings that have 48 kHz sampling frequency. This dataset has also been used in the WaveNet project [51]. All the recordings have been made in the exact same setup and the microphone allowing the quality to be stable across the entire dataset.

4.2.2. Noise Datasets

The noise corpus is another critical element required to create a multi-channel dataset for neural network training. Unlike the speech datasets, noise datasets are not as common. Certain research papers took noise samples from freely available online resources like YouTube or FreeSound [59].

When selecting noise samples, it is important to find samples that reflect the scenarios the model will be encountering. This thesis will be focusing on the speech enhancement problem in everyday life cases. The environment will mostly include white noise, babble noise, or noise from household goods.

The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND) is a freely available dataset containing various types of noise recorded in a 16-channel microphone setup [60]. It includes 18 different scenarios such as cafeteria, hallway or meeting. Each scenario contains unique set of noise signals. As an example, while the

meeting noise scenario includes background human speech and the restaurant noise, the living room scenario contains AC noise and TV sound in the background.

DEMAND has been used in some of the latest research papers ([30], [51], [61]). In this thesis, DEMAND is the main noise corpus used to generate the samples to train the neural network model.

4.2.3. Multichannel Noisy Speech Dataset

The main objective of this thesis is to build an end-to-end neural network that enhances the multi-channel signal data by utilizing the spatial information hidden in the data. In order to achieve this objective, there needs to be a diverse and realistic dataset containing audio and noise signals collected by different microphones in 3-D environments allowing reverberation.

There are three approaches to generate a multi-microphone noisy speech dataset. The first approach is to use a physical device with multiple microphones attached. This approach requires an extensive effort to ensure the audio recordings are captured in various settings with speech and noise present. The main difficulty is to ensure the consistency of the speeches while generating diverse noise samples and adjusting the signal-to-noise ratio (SNR). Given the requirements of the neural network models, the dataset needs to guarantee the quantity, and quality standards. The second difficulty is the access to diverse speakers. Compared to the speech datasets investigated in the earlier sections containing more than 500 speakers, it will be an arduous process to reach these high unique speaker numbers for this thesis.

In order to solve the second issue raised, Sainath et al (2017) proposed a setup with a mouth simulator [52]. Compared to a speaker playing human speech, a mouth simulator would generate an audio signal with the same characteristics as a speech signal. Using this device, a speech corpus can be utilized to generate various scenarios. This does not negate the fact that all the scenarios will have to be manually created.

The third approach is the one that has been used in this thesis. Instead of generating the signal physically, the multi-microphone noisy speech can be sampled digitally. Pyroomacoustics is a tool that allows users to simulate a room with multiple signal sources collected by multiple speakers in a room defined by the users [62]. The freedom of configuration given by this tool presents a diverse set of scenarios to be generated much quicker than the physical options explored. Pyroomacoustics simulates the signals traveling along the room, the reverberations caused by the reflections in the room, the absorption ratio, the distance between the microphones causing delays in the collected signals. All these parameters make this method sufficient to create the dataset to train the neural network model.

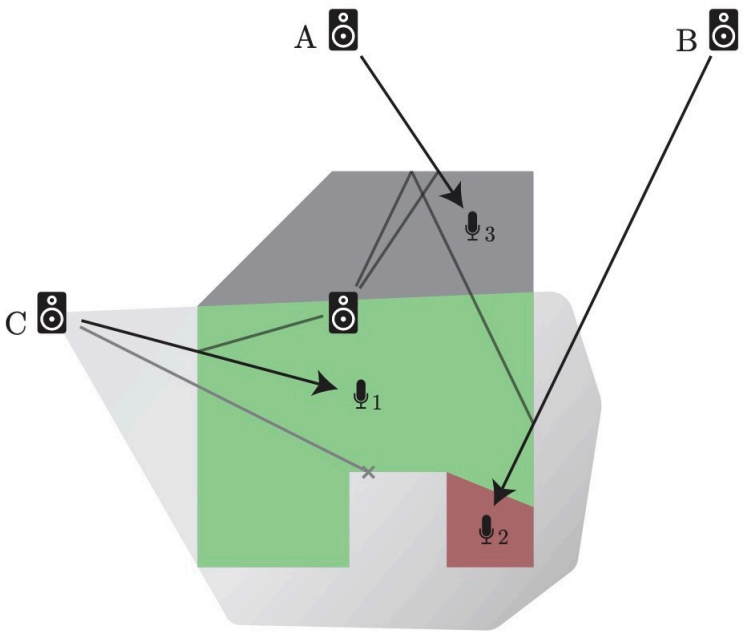


Figure 4.1. A room configuration example from pyroomacoustics where A, B and C represent the signal sources while 1, 2, and 3 are the microphones. The room geometry can be configured freely to enable unique scenarios [52].

The initial selection for the dataset creation in this thesis came from the speech corpus selection. Among all the options explored, TIMIT corpus offered the most

variety in terms of the speaker and sentence combinations. Across the entire TIMIT set, 2500 speech samples are randomly selected. From the DEMAND, cafe, cafeteria, hallway, kitchen, living room, meeting, office and restaurant noise scenarios are selected. From this selection, only a single channel is extracted among the 16 channels arbitrarily and the long recordings are split into 5 second long segments. This clipping operation resulted in 483 noise sample files.

Another speech dataset that is used in this thesis is the Speech Commands corpus. Because of the length of the samples being one second on average, 15 000 samples are selected randomly across 30 different word groups.

Upon selecting the speech and noise datasets, the single channel dataset can be generated for the first type of the single channel neural network model. The dataset is generated by first selecting a random speech sample from TIMIT dataset and a random noise sample from DEMAND. Both signals are initially normalized to have the peak of the signal being equal to -1.0 dB. Among -1.0, -4.0, -7.0 and -10.0, one of the options are arbitrarily selected and applied to the noise sample to create a variation in noise sample and consecutively in the SNR values. The processed noise sample is added on top of the speech signal. If the signal lengths do not match, the noise sample is either curtailed or duplicated to match the speech sample. The algorithm randomly selects 5000 audio samples from the speech and noise dataset and goes through the process explained before.

In the multi-microphone noisy speech dataset creation case, a new variable of room configuration becomes a problem to solve. Uniform rectangular array geometry is selected with each neighboring microphone have 6 cm distance from the closest two microphones. In total, there are 4 microphones in the setup. The microphone array geometry does not change in any of the dataset creation process. This setup is made by considering the commercial microphone array geometries in the market being similar to the one simulated for this thesis.

For the room configuration, certain parameters do not vary with each sample. Absorption value represents how much of the signal is absorbed by the walls and how much is reflected creating reverberations. Pyroomacoustics has a detailed table of what absorption value should be selected for which real life scenarios. In this scenarios, a normal living room setting is preferred, therefore, the value is selected to be 0.2. The sampling frequency for the dataset is selected to be 8 kHz. This frequency allows the dataset to be much smaller while retaining the critical characteristics of the speech allowing the models to train on them.

The room geometry is a varying parameter in a 3-D geometry. The width and the length of the room arbitrarily change between 3 meters and 7 meters while the height is a constant value of 3 m. The noise and speech sources are also arbitrarily placed with a margin of 0.1 m for width and length. For the height, they could be placed between 0.05 m and 2.5 m above the ground. Lastly, the microphone array can be placed in the rectangular room created with the same configuration parameters as the signal sources.

This setup allows a fairly large set of unique scenarios to be generated. One issue that came up with this configuration logic is despite the normalization levels for the loudness of the noise and speech, the location is arbitrarily selected. Therefore, a noise source could be too close to the microphones while having a low peak amplitude value. This deleteriously impacts the ability to accurately detect which samples have high or low SNR value in the naming convention.

The noise files before being placed in the simulation are normalized between -4.0 dB and -11.0 dB. The reason behind starting at -4.0 dB and not -1.0 dB is to prevent the unintelligible samples that has been created due to noise samples with a high peak amplitude being placed next to the microphone array. Using this setup, 2500 samples are created. Each noisy signal has 8 kHz sampling frequency and 4 channels along with the clean target signal to be used in the model.

4.2.4. Speech Dataset in Frequency Domain

The single channel and multi-microphone datasets are created with 2500 noisy signal samples and 2500 clean speech signal samples. In order to run the training on these samples, they need to be converted into the form required by the model designed. The first operation required is the short-time Fourier transform (STFT). This operation transforms the raw time-domain into the frequency domain. The length of the windowed signal is selected to be 256 and to have 8 kHz sampling frequency. The overlap of the window is 75%. Lastly, the window selected for the STFT operation is the hamming window. These parameter do not change in this thesis for consistency.

The output of the STFT operation results in a matrix (129×N) where each column represent a frame which could increased based on the length of the signal.

By following the convention used by Park and Lee [31], for every clean speech frame, the noisy frame and 7 earlier frames are taken. In order for the model to perform well, the clean frame and the noisy frame need to be an exact match. Failure to achieve that will result in issues that will be explored in the next section.

The clean spectrum has the shape of (129×N) and the noisy spectrum has the shape of (129×N×8). This allows each frame to have 8 corresponding noisy frames to operate on. For the case of multi-channel dataset, an additional dimension is added converting the shapes of clean and noisy matrices to (129×N×4) and (129×N×8×4) respectively. Addition of 3 more channels, quadruples the data size of the single channel dataset.

The issue of exact matching between clean and noisy spectra raises a problem with the multi-channel dataset. While the single channel dataset is created by adding noise onto the speech signal, the multi-microphone dataset is a product of simulations. The simulations resulted in delays and reverberation that may cause a difference of almost 300 ms. To prevent any issues with the training model, the dataset needed to be

augmented. By taking the cross-correlation of the clean speech and each corresponding noisy channel in the multi-channel dataset, the delay can be founded. The delay is then applied to have a perfect fit for all the noisy channels and the clean speech. The reverberation is then removed for the clean speech and the noisy speech signal to have the exact same length as the model progresses frame by frame.

After the augmentation and frequency domain conversion, the data is stored in a single h5 file which has separate buckets for clean and noisy speech matrices [63].

4.3. Convolutional Neural Network Model for Beamforming

This section is broken down into two categories. The first category explores the autoencoder implementation for the single channel speech enhancement problem. This stage acts as a launchpad for the multichannel scenario as the model used is going to be similar to the one used in the single channel scenario. The second category is the beamforming with the deep learning methods. The model selected for the task is the autoencoder with the convolutional layers.

4.3.1. Single Channel Speech Enhancement Neural Network

The single channel speech enhancement is the baseline of the multichannel model that will be designed and created. The task of denoising a speech signal and building an end-to-end solution without prior feature extraction required the model to work on the entire frequency spectrum and extract the patterns of speech and noise.

A good example of the use of autoencoders for denoising comes from another domain: images. In [64], the use cases of autoencoders are described. The most basic functionality of an autoencoder is to reconstruct an image after the encoding and the decoding phases. The entire operation compresses the input data and by definition is a lossy process. An autoencoder is only good at reconstructing or decompressing the data it has been trained on. An autoencoder model trained on office noises may

perform poorly on construction site noises.

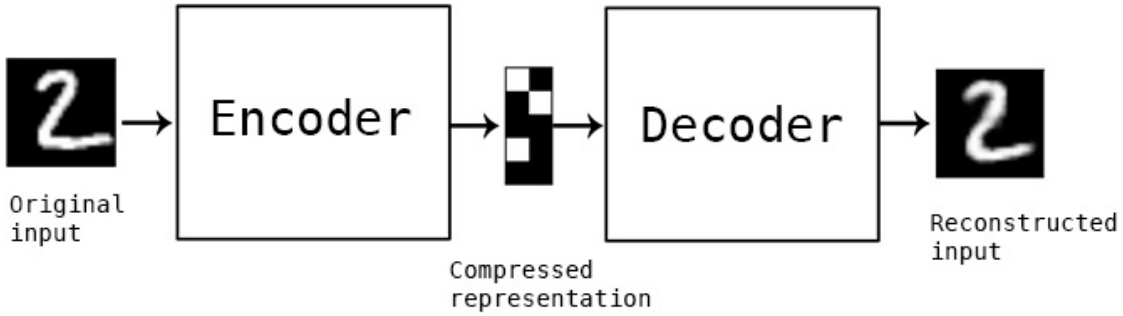


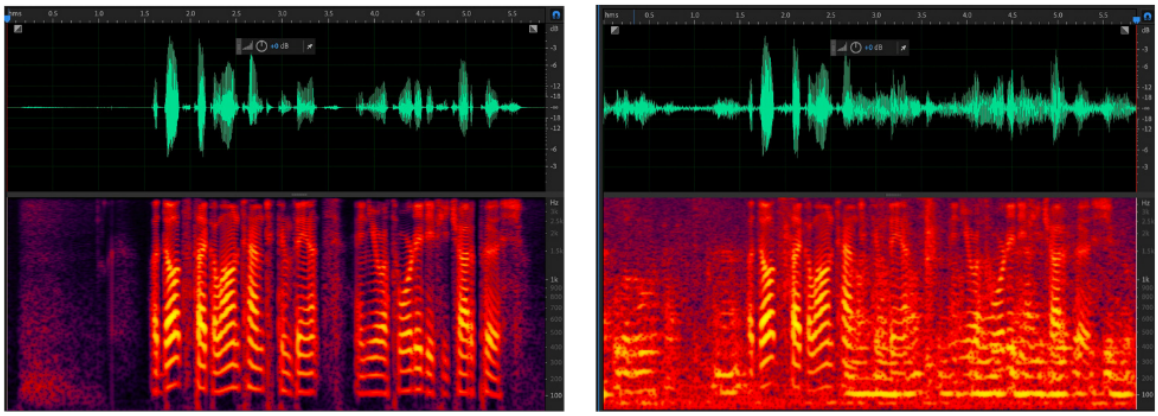
Figure 4.2. The graphical description of the autoencoder for a sample image from the MNIST dataset [64].

Referring back to the MNIST denoising case, the autoencoder is trained on a set of MNIST samples with Gaussian noise added along with the corresponding target samples. The model consists of 2 convolutional layers and 2 max pooling layers in the decoding stage and 3 convolutional layers and 2 upsampling layers for 2-D inputs in the encoding stage. All convolutional layers include ReLU activation function with the exception of the last layer having a sigmoid activation function. The results of this basic model after 50 epochs are quite satisfactory which can be seen in Figure 4.3.

Figure 4.3. The row above contains 9 randomly selected noisy test samples and the row below contains the autoencoder model's denoised output [64].

The denoising for the noisy speech case has similarities with the image denoising operation. In Figure 4.4, the main similarities can be seen. The model will be trained on the magnitude spectrum of the signal which can be considered as a noise added

image sample. The task of the autoencoder model will be to learn the patterns of speech and clear the magnitude spectrum of the noisy speech and produce a clean version of the input. The target will be present for the training and there will be a frame by frame matching of the noisy and target speech spectra.



a) b)

Figure 4.4. Time and frequency domain representation of a clean speech signal from TIMIT corpus (left) and a noise added version of the same speech sample (right) can be seen in the figure.

In order to develop a state of the art autoencoder model for speech enhancement, the model proposed by Park and Lee [31] became a starting point for this thesis. The model contains groups of a convolutional layer, a batch normalization layer and a ReLU activation layer stacked one after another. This is a different approach compared to the model used by the convolutional encoder-decoder (CED) network [45]. That network similar to the MNIST denoising case employs upsampling and downsampling operations. However, in [31], the performance of the redundant convolutional encoder-decoder (R-CED) network is shown to be higher than the CED network.

In light of these developments, in this thesis R-CED's initial structure is used with different number of layers and filter numbers. In the multichannel scenario, more changes are made to the model to improve performance and better train on the existing dataset.

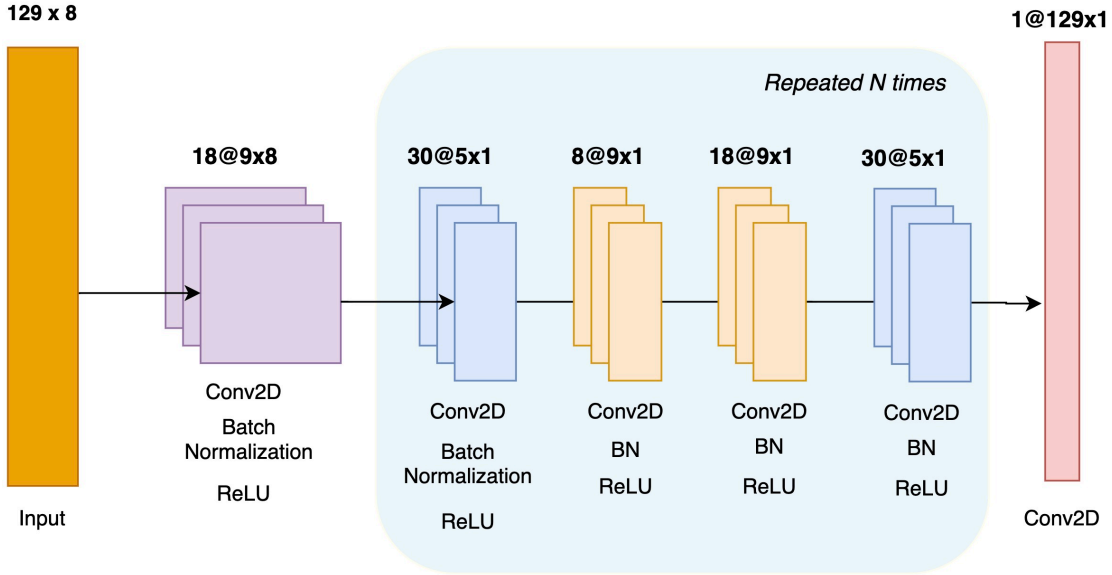


Figure 4.5. The base model used for the single channel speech enhancement can be seen above. The area with the light blue background is repeated 5 times in the first model.

The dataset as explored in the previous section consists of 2000 noisy samples and 2000 clean target samples for training and 500 noisy samples along with the target samples for testing. The dataset employed TIMIT exclusively for the speech signals and certain household and office sound effects from freely available public domain sources as opposed to the DEMAND applied on the multichannel dataset.

The first base model which can be investigated in Figure 4.5 gives an overall picture of the network. The input has the size of (129×8×1) and the target has the size of (129×1×1). The first step is the convolution group made up of 18 2-D convolutional filters (9×8) followed by the batch normalization and ReLU activation step. The batch normalization ensures the stability of the network after each convolution (BN) operation. Batch normalization is followed by the rectified linear unit (ReLU) activation function. The output of this group then moves onto another convolutional layer with 30 2-D convolutional filters of size (5×1). The same batch normalization steps are followed. The rest of the network with the exception of the last layer can be

expanded and repeated. The version that is shown in the Figure 4.5 with 5 repetitions of the center group contains 69973 parameters. The last layer is different than the rest of the layers with a single 2-D convolutional filter of size (129×1).

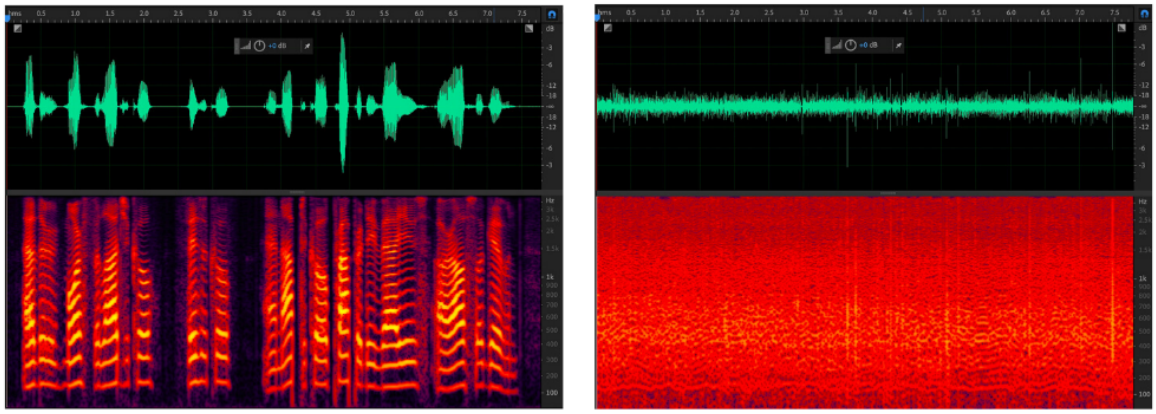
Adam optimizer has been used for the training with default paramaters. For the learning rate selection, when it was below 0.001 and above 0.003, the loss was higher compared to the learning rate interval of 0.001 and 0.003. Within this interval, the loss was not as impacted by the change of the learning rate.

One of the most critical aspects of this model is the selection of the loss function. For the single channel case, mean squared error (MSE) was used. The logic is the target frame and the output frame would be compared and the loss function is calculated for the training and validation sets. Based on the end goal of the project and the model, the loss function can be changed. For certain models, it can be connected to the word error rate to improve the outcome of the acoustic model. Another scenario could be to use one of the intelligibility metrics such as PESQ or STOI. Some research papers create novel loss functions to better train the models for the speech enhancement task ([65], [66]).

The first set of trainings with this model failed. The number of epochs did not decrease the loss value indicating the initial dataset was mismatched. Upon further inspection, it became clear that the dataset needed to be normalized before the training. Both the mean and standard deviation values of the clean and noisy samples have been calculated.

An early mistake made in the dataset creation was not ensuring the clean dataset and noisy dataset matched. Since the dataset is created by the concatenated speech samples, a small mismatch could cause a domino effect in the entire dataset causing the issue in Figure 4.6.

After solving the mismatch issue and introducing the normalization for the dataset, the loss values started to decrease both for the training and validation sets.

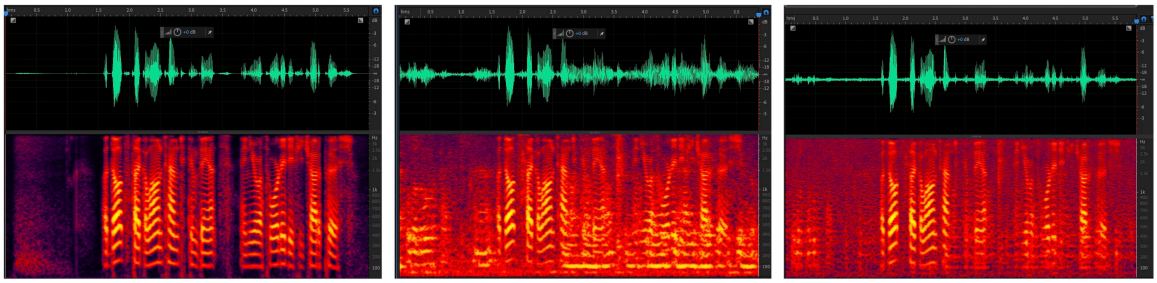


a) b)

Figure 4.6. The output of the first model can be seen above (right). The loss never decreased throughout the training phase and the model did not learn any patterns.

The second model performed well compared to the initial failed model. Upon inspection on Figure 4.7, the cleaned spectra on the right have removed some of the high energy sections introduced in the noisy spectra (center). This model was fairly successful at eliminating noise components like sudden kitchen utensils, TV playing in the background. However, the main issue with this model was the introduction of the white noise in the entire spectrum of the output. The white noise generation issue became a pernicious part of the upcoming models.

In order to solve the white noise issue, a configuration to work on was the size of the network. Increasing the network size to 100 000 parameters had a slight yet beneficial impact on the performance but did not solve the noise introduction problem. In order to better understand the issue, the output of the neural network model is investigated. The model output is normalized and needed to be multiplied by the standard deviation and the mean should be added before converting the magnitude spectrum into the raw waveform. The initial setup used the mean and standard deviation of the noisy dataset. As a solution, different mean and standard deviation values have been



a) b) c)

Figure 4.7. The target signal spectra (left), the noisy spectra (middle) and the cleaned spectra (right) can be observed as the output of the second model.

tested. Changing the standard deviation had a minor impact on the quality of the output signal. On the other hand, changing the mean had a dramatic impact on the presence of the newly introduced noise of the cleaned mixture signal.

In Figure 4.8, the spectra of outputs with various mean values used to normalize can be seen. In the top right quadrant, the clean mixture spectrum with a newly introduced white noise can be clearly seen. This mixture was generated by using the noisy mean obtained by the mean of the noise dataset. Replacing the mean value with the mean of the clean dataset clearly improved the signal quality as it can be see on the bottom left quadrant. Lastly, the after decreasing the mean value by 10%, the best outcome out of these options have been obtained. Reducing the mean value more than the last option started to introduce the noise that was eliminated from the output.

To visualize the issue further, the mean and standard deviation values of all the samples in both the noisy and clean dataset have been graphed. The standard deviation of the noisy and the corresponding clean samples follow a clear line. As it was observed by the trials, the standard deviation change did not have a major influence on the quality of the audio. The distribution of the mean values of mixture samples and its corresponding clean samples display a different case. The mean values are more widespread.

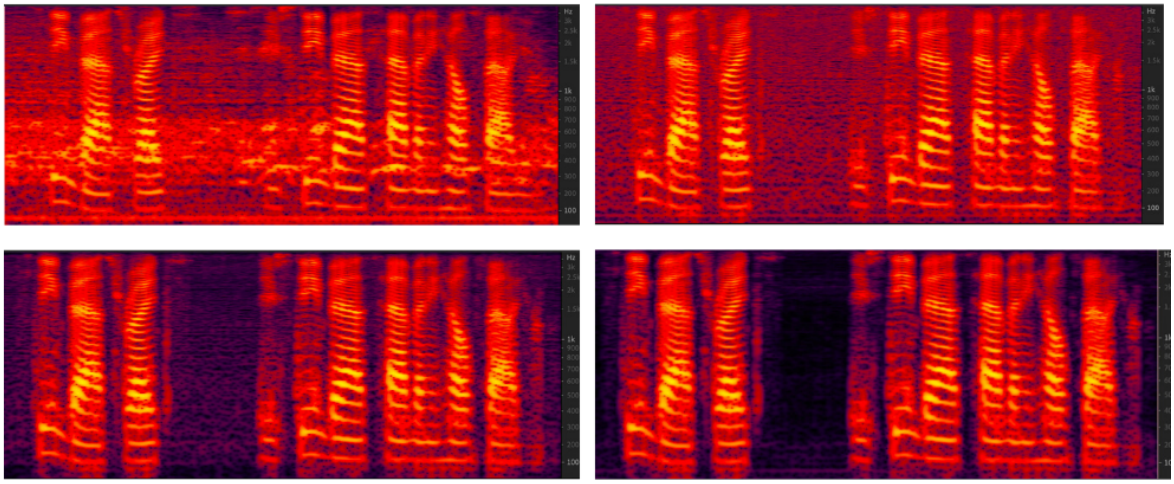


Figure 4.8. The mixture spectrum (top left), the cleaned mixture spectrum with noisy mean (top right), the cleaned mixture spectrum with clean mean (bottom left) and the cleaned mixture spectrum with a custom mean (bottom right) can be seen above.

The last feature of the signal that has been intentionally discarded is the phase. The neural network model is trained on the magnitude spectrum therefore the phase is only used to convert the cleaned mixture spectrum to the audio domain. The difference between the noisy signal phase and the clean signal phase can be felt but it has a minor impact compared to the impact of the other factors of the model and the overall network. The impact of the phase will be investigated in detail in the next section for the multichannel autoencoder model.

In the second model, changing the hyperparameters such as the learning rate, optimization function or the loss function did not improve the performance. However, one change made to the second model reduced the loss. The introduction of two skip connections linking the first convolutional layers with the last one reduced the loss by up to 58% in certain test cases. The benefit of the skip connections is it enables the signal to back-propagate to the bottom layers quickly and eliminate the vanishing gradients problem [67].

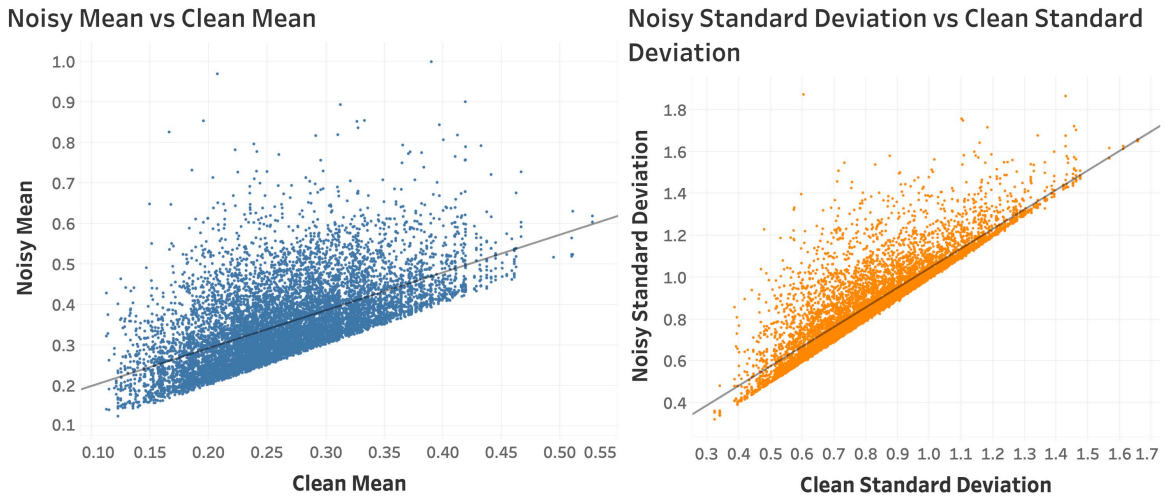


Figure 4.9. The distribution of mean (left) and standard variation (right) values of each noisy and its target clean can be seen.

The dataset used for the single channel autoencoder for speech enhancement had a size of 2 GB. The batch size changed between 64 and 256, but did not have a noticeable impact on the loss. For the models in this section, the epoch numbers were kept between 30 and 50 as the validation loss stabilized around these values.

4.3.2. Multichannel Beamforming with an Autoencoder Neural Network

The learnings of the single channel autoencoder model became a launchpad for the multichannel beamforming model development. The goal of this thesis is to propose a method that takes the audio with multiple channels with minimal data augmentation and offer an end-to-end solution where the output is a single channel cleaned mixture signal without any post-filtering and additional operations required.

To run the models, Google Colaboratory has been used [68]. The models are run on Nvidia's graphical processing unit (GPU) Tesla P100 with 26 GB random access memory (RAM). The size of the RAM enabled larger batches to be trained at once. This detail is important as the multichannel dataset contains data 4 times larger than the previous set while not introducing more diversity but only spatial information.

Two different datasets are used in the multichannel speech enhancement model training. The first dataset contains samples from the Speech Commands corpus [54]. This corpus had the advantage of delivering variety of speakers much quickly as each sample only lasted a second. To make the training quicker, 4 channel data has been reduced to two channels.

Speech Commands multichannel mixture dataset is split into two different files to be able to load each of them separately to the model for training. The initial model was quite similar to the single channel example with the exception of layers being repeated 7 times.

The training loss decreased steadily but the validation loss stabilized after the 30th epoch. Continuing the training caused the model to overfit therefore the training was stopped to prevent the validation loss to increase. One difference in the multi-channel case was the model quickly became overfit while in the single channel model, the training loss also stopped decreasing along with the validation loss which can be seen in Figure 4.10.

One important detail that is not visible initially is the audio quality and the performance of the model. Evaluating a number of random test samples that the model has not been trained on displayed the issue with the approach taken. The model cleaned the outputs from certain noise elements but introduced new unintelligible audio components and worsened the quality of the overall speech to a degree of which the speech became unintelligible. The dataset used for this challenge had the unique feature of very short responses and empty frames with zero energy. However, the empty frames had been distorted in the dataset with noise and reverberation generated by the dataset. This issue of empty frames did not occur in the TIMIT dataset and the single channel dataset was a product of simple addition in comparison to the simulations ran to produce the multichannel dataset.

One approach to solve the autoencoder model's speech distortion problem is changing the dataset. Given the performance of the TIMIT dataset in the single channel scenario, new simulations have been ran to create a new dataset with 2500 speech samples from TIMIT and randomly selected noise samples from DEMAND. As before, the dataset is split into two parts to be able to load the data in different training sessions consecutively.

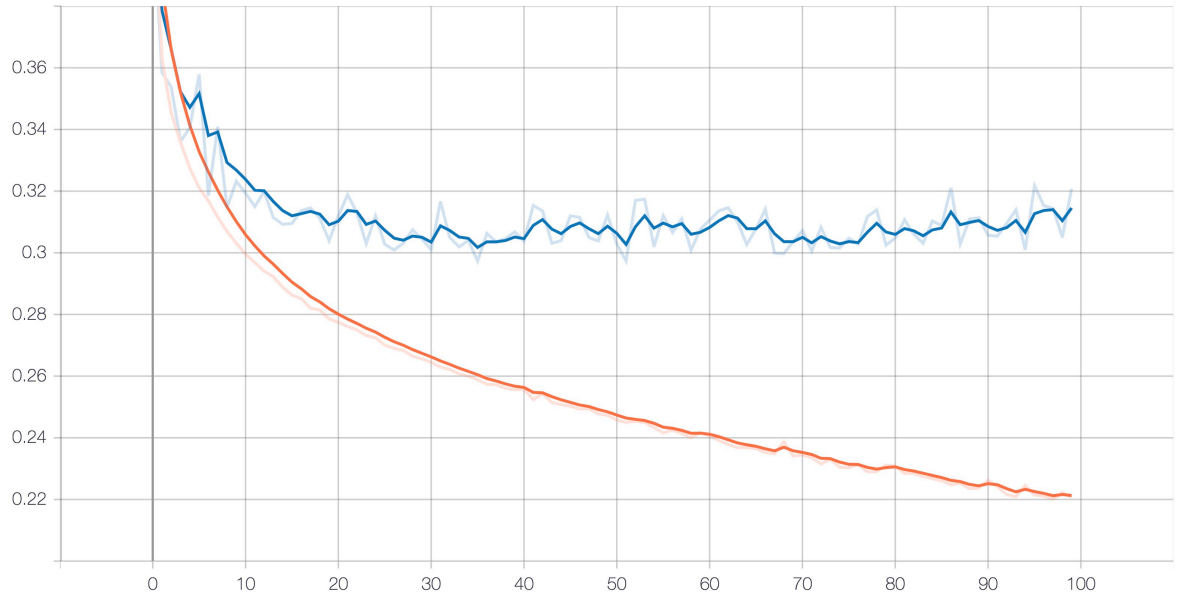


Figure 4.10. X-axis represents the epoch number and the y-axis represents the loss value for the two channel model training. The blue line stands for the validation loss and the orange line stands for the training loss.

The training hyperparameters were kept the same as before with certain changes in the layers. In order to curb the overfitting problem, dropout is introduced before the final layer. Adding the dropout layer after every activation layer caused the validation loss to increase, therefore all but the final dropouts are removed. Another change was to replace the ReLUs with the LeakyReLUs. The last change in the model was the L2 regularizer with 0.01 value to the first and second convolutional layers. The reasoning is reducing the overfitting and increasing the performance of the model.

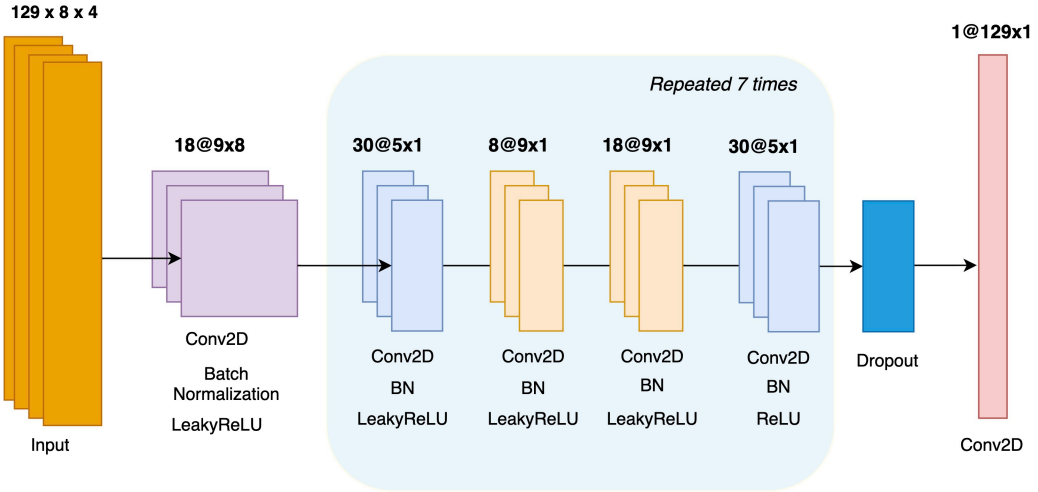


Figure 4.11. The multichannel speech enhancement model share the same structure of the single channel speech enhancement model with certain differences.

Training the new model which can be explored in Figure 4.11 took 100 epochs with two different datasets loaded consecutively in different training sessions. The validation loss decreased 6% compared to the two channel example. The essential part of this exercise is the speech quality.

One unique unmentioned feature of the new dataset created for multichannel training is the newly introduced babble noise and its impact. In the previous signal channel dataset, most of the noise came from objects and rarely from the speech from TV background noise which has a different profile than a human speech recorded directly. Especially on certain samples, the human speech noise effect made it difficult to differentiate the noise even for a human listener.

The results were not as successful as the single channel dataset. In cases where there is the babble noise, the cleaned output performed worse than the mixture the model is aiming to clean. Two examples are worth mentioning for this thesis. The first example can be seen in Figure 4.13. In this example, the mixture includes meeting room speech along with the AC noise in the background. One clear difference between the target signal and the mixture signal is the low frequency areas. The model has

Table 4.1. STOI and PESQ Scores of Different Models

	Noisy Samples	**4 Channel Model**	**1 Channel Model**	**MVDR**	**GSC**
STOI	0.68	0.72	0.29	0.64	0.58
PESQ	1.84	1.79	1.20	1.74	1.46

been able to clean this high energy area successfully and liken the spectrum more to the clean signal. The problem arises on the intelligibility part. The model tries to reduce the babble noise and it is one of the most difficult noise types to eliminate. However, the reduction efforts distort the core speech which the model should preserve, therefore, reducing the intelligibility score.

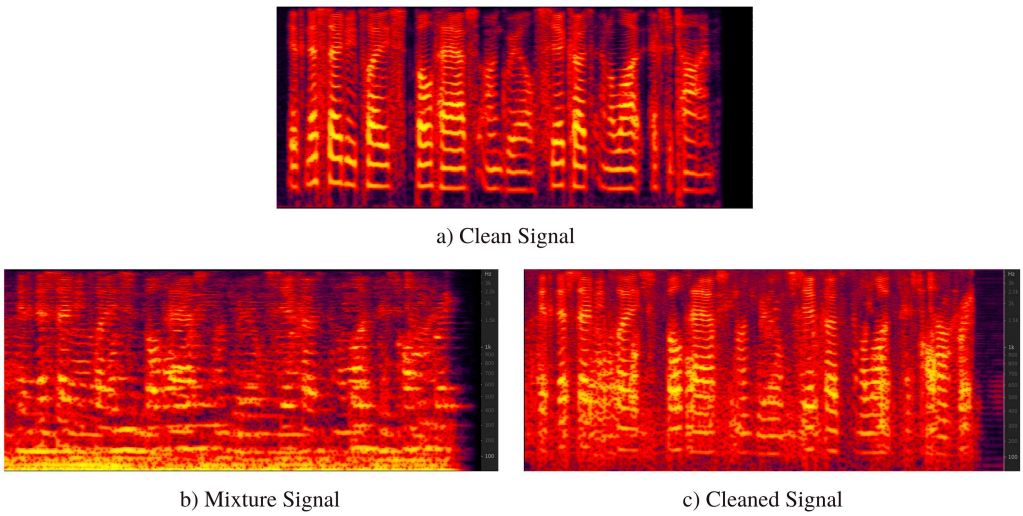


Figure 4.12. The multichannel speech enhancement model share the same structure of the single channel speech enhancement model with certain differences.

Despite the low performance of the model in the babble noise case, the STOI score of the test set evaluated on the multichannel autoencoder model were higher than the noisy samples, the MVDR beamformer, single channel autoencoder model and the GSC model. In the PESQ case, the noisy samples had the highest intelligibility score followed by the multichannel autoencoder model. The scores have been obtained by evaluating 477 test samples by all the methods and taking the mean of all the scores. The results can be explored in Table 4.1.

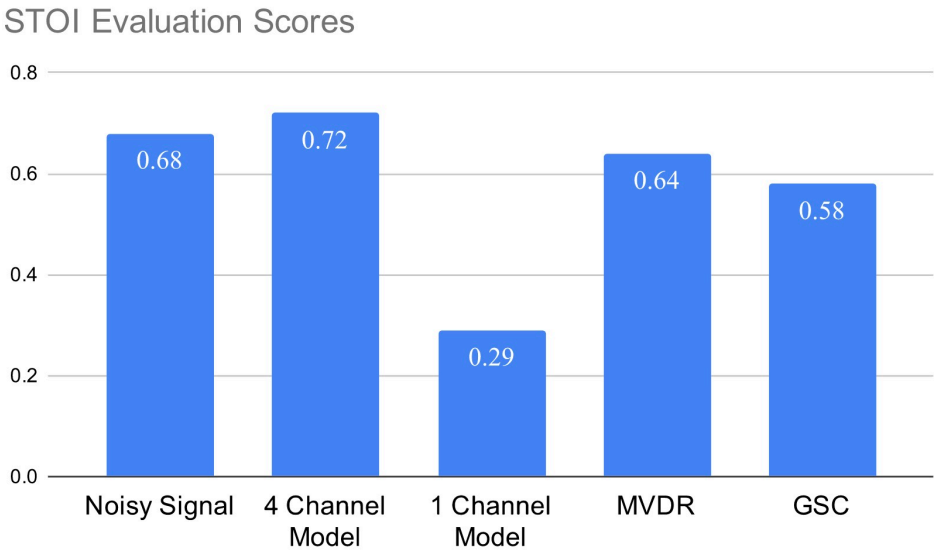


Figure 4.13. The multichannel autoencoder model (MAM) performed the best among all the conventional beamforming methods.

In this thesis, all the neural network models have been trained on the magnitude spectrum. The assumption made in this decision was the magnitude spectrum contains more critical information that the phase. The phase does not change drastically and the noisy phase can be used in order to convert the cleaned mixture signal's magnitude spectrum to the raw waveform. The output of the multichannel autoencoder model has been converted by both the noisy phase information and the clean phase information. The mean of STOI score of the samples converted with clean phase was 4% higher than those converted with noisy phase.

Some recent research papers have used raw waveform in order to retain the phase ([30], [52]). In order to keep the model trained on magnitude spectra, we also created a dataset containing noisy and corresponding phase information of speech samples. The training was conducted on the exact same model used for the magnitude spectra. After 100 epochs, neither the training loss nor the validation loss did not change. The model was not able to learn from the target and noisy phase samples presented to it.

5. CONCLUSION AND FUTURE WORK

In this thesis, the end-to-end neural network approach for the multichannel acoustic beamforming have been explored. Related work in this field showed the variety of approaches are taken to tackle the speech enhancement problem for the multi-microphone scenario. Some research papers focused on the mask estimation and trained the neural networks for spectral mask estimation that are used for beamformer weights ([69], [70], [71]). Overall, the general trend is to improve existing beamforming methods through neural network algorithms. This thesis was focused on the utilizing the spatio-temporal data collected in the multichannel dataset.

The single channel autoencoder showed promising results by successfully denoising some low SNR mixture samples in the test set. Conversion from the normalized magnitude spectra to the raw waveform became a critical issue in this thesis. The use of the correct mean value was partially resolved by using the mean value of the clean dataset. However, this is an area where more progress can be made. The mean value that should be used to convert the signal to the time domain is not a static variable and the mean value should be chosen based on the best outcome. This can be made a part of the neural network model as the best mean value is directly linked to the amount of noise in the mixture sample.

The multichannel autoencoder model is based on the the single channel model architecture adapted for the needs of the multi-microphone setup. One of the most formidable challenges became finding the dataset needed to train the neural network. Creating a dataset made up for noise added speech samples and clean speech samples is a fairly easy task thanks to the plethora of publicly available datasets. On the other hand, the multichannel speech dataset does not exist as commonly in the format required for this challenge. Physically generating the dataset is a costly endeavour that cannot be taken in this thesis. Therefore, a computer simulation generating a wide variety of real-life scenarios accompanied with some popular speech and noise

corpuses have been utilized to create the dataset needed for the task.

After many iterations, the multichannel autoencoder model performed better than conventional beamforming methods in both STOI and PESQ evaluation methods despite the difficult conditions with the babble noise and low SNR. Despite a number of trials and network structures, the phase information by itself could not be trained and the noisy phase information was used to convert the cleaned magnitude spectra.

An important finding of this thesis is the autoencoder network can be used to work on the multichannel data with some data augmentation. However, the speech is distorted in certain cases becoming a problem for the speech recognition models. As the selection of the loss function becomes a crucial part of the neural network training, the loss function could be arranged in a way to improve the word error rate of the final output. As an alternative to the current loss function which is the mean squared error (MSE), another method could be to use one of the speech intelligibility metrics.

The network architecture used for the multichannel task emphasized the frame-by-frame cleaning. This frame approach where each target frame matches exactly with the model worked well as the datasets were artificially created with no spatial data in mind. In the multichannel case, the signal sources reaches the microphones in various ways. The time delay caused by the distance between the microphones and the reflections from the objects in the room can be seen in the audio recordings. Since the noise is treated as a separate source, it will behave differently than a simple addition on top of a speech sample. Therefore, the denoising problem becomes more difficult and different network models might be tested that utilize the spatio-temporal patterns even more. One reason that the some cleaned mixture samples had distorted voice could be due to this reason as it is difficult to match the frames despite the applied cross-correlation.

The dataset for the multichannel dataset contained 2500 samples with 4 channels. Along with the target set, 18 hours of audio content were used in training. This is small

compared to the 2000 hour dataset used by Sainath et al (2017) [52]. More hours of audio content would allow the model to generalize much more when it encounters different noise or speech types. In order to utilize more data, the neural network training setup needs to be configured to allow the data to be processed. For this thesis, Google Colab has been used which only allows a single GPU per session and the sessions are disconnected within 12 hours. A more robust setup is required to run the training models with massive datasets as Google Colab provided GPU models are much slower compared to the latest model GPUs available in the market.

Beamforming has become more popular with the advent of the voice assistant devices like Google Home and Amazon Alexa. Some of the latest smartphones are also using multiple microphones to improve the quality of the recordings. As the neural networks become more powerful and easier to implement, the beamforming can be improved by using the neural network models. This thesis implemented the autoencoder variation of the neural networks to improve the beamforming quality compared to the conventional methods. There is a lot of room for improvement and new models can be implemented with larger datasets to improve the performance of the models for better speech quality.

REFERENCES

1. Benesty, J., J. Chen and Y. Huang, *Microphone array signal processing*, Vol. 1, Springer Science & Business Media, 2008.

2. Brandstein, M. and D. Ward, *Microphone arrays: signal processing techniques and applications*, Springer Science & Business Media, 2013.

3. Capon, J., "High-resolution frequency-wavenumber spectrum analysis", *Proceedings of the IEEE*, Vol. 57, No. 8, pp. 1408–1418, 1969.

4. Whiteside, S. P., "Peter B. Denes and Elliot N. Pinson The Speech Chain: The Physics and Biology of Spoken Language, 2nd edition.", *Journal of the International Phonetic Association*, Vol. 23, No. 2, 1993.

5. Rabiner, L. R. and R. W. Schafer, "Introduction to Digital Speech Processing", *Found. Trends Signal Process.*, Vol. 1, No. 1, pp. 1–194, Jan. 2007, `https://doi.org/10.1561/2000000001`.

6. Dalva, D., U. Guz and H. Gurkan, *Automatic Speech Recognition System for Turkish Spoken Language (Türkçe Dili icin Otomatik Konuşma Tanıma Sistemi)*, Ph.D. Thesis, 2012.

7. Loizou, P. C., *Speech Enhancement: Theory and Practice*, CRC Press, Inc., USA, 2nd edn., 2013.

8. Cariolaro, G., *Unified Signal Theory*, Springer London, London, 2011.

9. Kulkarni, S. R., *Frequency Domain and Fourier Transforms*, 2002.

10. Oppenheim, A. V. and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall Press, USA, 3rd edn., 2009.

11. Gutierrez-Osuna, R., *L6: Short-time Fourier analysis and synthesis*, 2014, http://courses.cs.tamu.edu/rgutier/csce630_14/l6.pdf, (accessed in June 2020).

12. *STFT: why overlapping the window?*, 2014, https://dsp.stackexchange.com/questions/19311/stft-why-overlapping-the-window, (accessed in June 2020).

13. Niemitalo, O., *Hann Window*, 2020, https://commons.wikimedia.org/w/index.php?curid=77818388.

14. Braun, S., *Speech dereverberation in noisy environments using time-frequency domain signal models*, Ph.D. Thesis, 01 2018.

15. Chen, J., J. Benesty and Y. A. Huang, "On the Optimal Linear Filtering Techniques for Noise Reduction", *Speech Commun.*, Vol. 49, No. 4, pp. 305—-316, 2007, https://doi.org/10.1016/j.specom.2007.02.002.

16. Su, F. and C. Joslin, "Acoustic imaging using a 64-node microphone array and beamformer system", *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 168–173, 2015.

17. Fonseca, W., J. P. Ristow, D. G. Sanches and S. N. Gerges, "Bandwidth Comparison on PSFs Simulationsusing Classical Beamforming", *Forum Acusticum*, 2011.

18. Haykin, S., *Adaptive Filter Theory*, Prentice-Hall, Inc., USA, 3rd edn., 1996.

19. Frost, O. L., "An algorithm for linearly constrained adaptive array processing", *Proceedings of the IEEE*, Vol. 60, No. 8, pp. 926–935, 1972.

20. Griffiths, L. and C. Jim, "An alternative approach to linearly constrained adaptive beamforming", *IEEE Transactions on antennas and propagation*, Vol. 30, No. 1, pp. 27–34, 1982.

21. Breed, B. R. and J. Strauss, "A short proof of the equivalence of LCMV and GSC beamforming", *IEEE Signal Processing Letters*, Vol. 9, No. 6, pp. 168–169, 2002.

22. ITU-T Recommendation P.800, 1996, `https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=3638`.

23. Rix, A. W., J. G. Beerends, M. P. Hollier and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs", *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, Vol. 2, pp. 749–752, 2001.

24. Taal, C. H., R. C. Hendriks, R. Heusdens and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech", *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4214–4217, 2010.

25. Morris, A. C., V. Maier and P. Green, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition", *Eighth International Conference on Spoken Language Processing*, 2004.

26. Goodfellow, I., Y. Bengio and A. Courville, *Deep learning*, MIT Press, 2016.

27. Hinton, G. E., S. Osindero and Y.-W. Teh, "A fast learning algorithm for deep belief nets", *Neural computation*, Vol. 18, No. 7, pp. 1527–1554, 2006.

28. O'Mahony, N., S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan and J. Walsh, "Deep learning vs. traditional computer vision", *Science and Information Conference*, pp. 128–144, Springer, 2019.

29. Lu, X., Y. Tsao, S. Matsuda and C. Hori, "Speech enhancement based on deep denoising autoencoder", *Interspeech*, pp. 436–440, 2013.

30. Pascual, S., A. Bonafonte and J. Serra, "SEGAN: Speech enhancement generative adversarial network", *arXiv preprint arXiv:1703.09452*, 2017.

31. Park, S. R. and J. Lee, "A fully convolutional neural network for speech enhancement", *arXiv preprint arXiv:1609.07132*, 2016.

32. Germain, F. G., Q. Chen and V. Koltun, "Speech denoising with deep feature losses", *arXiv preprint arXiv:1806.10522*, 2018.

33. Rethage, D., J. Pons and X. Serra, "A wavenet for speech denoising", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5069–5073, 2018.

34. Michelashvili, M. and L. Wolf, "Audio denoising with deep network priors", *arXiv preprint arXiv:1904.07612*, 2019.

35. Forgy, E. W., "Cluster analysis of multivariate data: efficiency versus interpretability of classifications", *Biometrics*, Vol. 21, pp. 768–769, 1965.

36. Piech, C., *K Means*, 2013, `https://stanford.edu/cpiech/cs221/handouts/kmeans.html`, (accessed in June 2020).

37. LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural computation*, Vol. 1, No. 4, pp. 541–551, 1989.

38. S. Mohamed, I., *Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques*, Ph.D. Thesis, 2017.

39. Nagdev, F. Detschsays, V. Minkovsays and A. E. for Anomaly Detection, *Auto Encoders for Anomaly Detection in Predictive Maintenance*, 2020, `https://iamnagdev.com/2020/03/05/auto-encoders-for-anomaly-detection-in-predictive-maintenance/`, (accessed in May 2020).

40. Boll, S., "Suppression of acoustic noise in speech using spectral subtraction", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 27, No. 2, pp. 113–

120, 1979.

41. Lim, J. S. and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech", *Proceedings of the IEEE*, Vol. 67, No. 12, pp. 1586–1604, 1979.

42. Scalart, P. *et al.*, "Speech enhancement based on a priori signal to noise estimation", *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 2, pp. 629–632, 1996.

43. Abdel-Hamid, O., A.-r. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional neural networks for speech recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 10, pp. 1533–1545, 2014.

44. Zhang, Y., M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks", *arXiv preprint arXiv:1701.02720*, 2017.

45. Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion", *Journal of Machine Learning Research*, Vol. 11, pp. 3371–3408, 2010.

46. Garofolo, J. S., L. F. Lamel, W. M. Fisher, J. G. Fiscus and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1", *NASA STI/Recon technical report n*, Vol. 93, 1993.

47. Fu, S.-W., Y. Tsao, X. Lu and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks", *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 006–012, IEEE, 2017.

48. Pandey, A. and D. Wang, "A new framework for CNN-based speech enhancement in the time domain", *IEEE/ACM Transactions on Audio, Speech, and Language*

Processing, Vol. 27, No. 7, pp. 1179–1188, 2019.

49. Veaux, C., J. Yamagishi and S. King, "The voice bank corpus: Design, collection and data analysis of a large regional accent speech database", *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pp. 1–4, IEEE, 2013.

50. He, K., X. Zhang, S. Ren and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

51. Oord, A. v. d., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "Wavenet: A generative model for raw audio", *arXiv preprint arXiv:1609.03499*, 2016.

52. Sainath, T. N., R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin *et al.*, "Multichannel signal processing with deep neural networks for automatic speech recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25, No. 5, pp. 965–979, 2017.

53. Sainath, T. N., O. Vinyals, A. Senior and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, 2015.

54. Warden, P., "Speech commands: A dataset for limited-vocabulary speech recognition", *arXiv preprint arXiv:1804.03209*, 2018.

55. Ardila, R., M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers and G. Weber, "Common Voice: A Massively-Multilingual Speech Corpus", *arXiv preprint arXiv:1912.06670*, 2019.

56. Brandenburg, K., "MP3 and AAC explained", *Audio Engineering Society Confer-*

ence: 17th International Conference: High-Quality Audio Coding, Audio Engineering Society, 1999.

57. Panayotov, V., G. Chen, D. Povey and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books", *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, IEEE, 2015.

58. Yamagishi, J., C. Veaux, K. MacDonald *et al.*, *CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)*, 2019.

59. Akkermans, V., F. Font Corbera, J. Funollet, B. De Jong, G. Roma Trepat, S. Togias and X. Serra, "Freesound 2: An improved platform for sharing audio clips", *Klapuri A, Leider C, editors. ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011.

60. Thiemann, J., N. Ito and E. Vincent, "The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings", *Proceedings of Meetings on Acoustics ICA2013*, Vol. 19, p. 035081, Acoustical Society of America, 2013.

61. Masuyama, Y., M. Togami and T. Komatsu, "Consistency-aware multi-channel speech enhancement using deep neural networks", *ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 821–825, 2020.

62. Scheibler, R., E. Bezzam and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355, 2018.

63. Folk, M., G. Heber, Q. Koziol, E. Pourmal and D. Robinson, "An overview of the HDF5 technology suite and its applications", *Proceedings of the EDBT/ICDT*

2011 Workshop on Array Databases, pp. 36–47, 2011.

64. Chollet, F., *Building Autoencoders in Keras*, 2016, `https://blog.keras.io/building-autoencoders-in-keras.html`.

65. Xia, B. and C. Bao, "Speech enhancement with weighted denoising auto-encoder", *INTERSPEECH*, pp. 3444–3448, 2013.

66. Zhao, Y., B. Xu, R. Giri and T. Zhang, "Perceptually guided speech enhancement using deep neural networks", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5074–5078, 2018.

67. Mao, X., C. Shen and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections", *Advances in Neural Information Processing Systems*, pp. 2802–2810, MIT Press, 2016.

68. Bisong, E., "Google Colaboratory", *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 59–64, Springer, 2019.

69. Heymann, J., L. Drude and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 196–200, 2016.

70. Jiang, W., F. Wen and P. Liu, "Robust Beamforming for Speech Recognition Using DNN-Based Time-Frequency Masks Estimation", *IEEE Access*, Vol. 6, pp. 52385–52392, 2018.

71. Matsui, Y., T. Nakatani, M. Delcroix, K. Kinoshita, N. Ito, S. Araki and S. Makino, "Online Integration of Dnn-Based and Spatial Clustering-Based Mask Estimation for Robust Mvdr Beamforming", *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 1–5, 2018.