Deep Learning-based Speech Enhancement for Real-life Applications

A Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Soha Abdallah Abdelhafiz Nossier

Assistant Lecturer of Biomedical Engineering, Alexandria University, Egypt

M.Sc. in Biomedical Devices, Alexandria University, Egypt, 2019

B.Sc. in Electrical Engineering, Alexandria University, Egypt, 2014

University of East London

May, 2023

DECLARATION

I here by declare that the thesis entitled "Deep Learning based Speech Enhancement and Recognition in Noisy and Adverse Environments" submitted by me, for the award of the degree of *Doctor of Philosophy* to the University of East London is a record of bonafide work carried out by me under the supervision of Dr. Julie Wall and Prof. Mansour Moniri, Department of Computing and Digital Technologies, School of Architecture, Computing and Engineering, University of East London, London, UK; and Dr. Cornelius Glackin, Intelligent Voice Ltd., London, UK.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this university or any other university or institute.

Name: Soha Abdallah Abdelhafiz Nossier

Date: 16/11/2022 **Signature of the Candidate**

Soha Nossier

ABSTRACT

Speech enhancement is the process of improving speech quality and intelligibility by suppressing noise. Inspired by the outstanding performance of the deep learning approach for speech enhancement, this thesis aims to add to this research area through the following contributions. The thesis presents an experimental analysis of different deep neural networks for speech enhancement, to compare their performance and investigate factors and approaches that improve the performance. The outcomes of this analysis facilitate the development of better speech enhancement networks in this work.

Moreover, this thesis proposes a new deep convolutional denoising autoencoder-based speech enhancement architecture, in which strided and dilated convolutions were applied to improve the performance while keeping network complexity to a minimum. Furthermore, a two-stage speech enhancement approach is proposed that reduces distortion, by performing a speech denoising first stage in the frequency domain, followed by a second speech reconstruction stage in the time domain. This approach was proven to reduce speech distortion, leading to better overall quality of the processed speech in comparison to state-of-the-art speech enhancement models.

Finally, the work presents two deep neural network speech enhancement architectures for hearing aids and automatic speech recognition, as two real-world speech enhancement applications. A smart speech enhancement architecture was proposed for hearing aids, which is an integrated hearing aid and alert system. This architecture enhances both speech and important emergency noise, and only eliminates undesired noise. The results show that this idea is applicable to improve the performance of hearing aids. On the other hand, the architecture proposed for automatic speech recognition solves the mismatch issue between speech enhancement automatic speech recognition systems, leading to significant reduction in the word error rate of a baseline automatic speech recognition system, provided by Intelligent Voice for research purposes. In conclusion, the results presented in this thesis show promising performance for the proposed architectures for real time speech enhancement applications.

Keywords: *automatic speech recognition*, *deep learning*, *hearing aids*, *speech distortion*, *speech enhancement*.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude and appreciation to my supervisors **Dr. Julie Wall**, **Prof. Mansour Moniri**, and **Dr. Cornelius Glackin** for their continuous support and encouragement, which help me to successfully complete my PhD.

I am very grateful to Intelligent Voice Ltd. for funding and sponsoring my PhD, and for providing technical support, resources, guidance, and advice during the experimental work and throughout my PhD.

I am very grateful to the University for granting me the Excellence Studentship, which has enabled me to study at the University of East London.

I owe special thanks and gratitude to my family for their constant encouragement and moral support, without which this PhD would not have been possible.

I would also like to express my sincere thanks to all my friends and colleagues.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TERMS AND ABBREVIATIONS

1D one dimensional

2D two dimensional

ANN Artificial Neural Network

ASR Automatic Speech Recognition

AWGN Additive White Gaussian Noise

BCE Binary Cross Entropy

BTE Behind The Ear

CASA Computational Auditory Scene Analysis

CDAE Convolutional Denoising Autoencoder

cIRM Complex Ideal Ratio Mask

CNN Convolutional Neural Network

CRN Convolutional Recurrent Network

DAE Denoising Autoencoder

DCRN Deep Convolutional Recurrent Network

DCT Discrete Cosine Transform

DDAE Deep Denoising Autoencoder

DE-CADE Deep Encoder - Convolutional Autoencoder DEnoiser

DEMAND Diverse Environments Multichannel Acoustic Noise Database

DNN Deep Neural Network

DNS Deep Noise Suppression

ELU Exponential Linear Unit

ESC Environmental Sound Classification

FCNN Fully Convolutional Neural Network

FF feedforward

FFT Fast Fourier Transform

FFT-Mask Fast Fourier Transform Mask

GAN Generative Adversarial Network

GF Gammatone Frequency

GF Gammatone Filterbank

GFB Gabor Filterbank Feature

GFCC Gammatone Frequency Cepstral Coefficients

GFMC Gammatone Frequency Modulation Coefficients

GPU Graphical Processing Unit

GRU Gated Recurrent Unit

HAAQI Hearing-Aid Audio Quality Index

HASPI Hearing-Aid Speech Perception Index

HASQI Hearing-Aid Speech Quality Index

IBM Ideal Binary Mask

IRM Ideal Ratio Mask

ISTFT Inverse Short Time Fourier Transform

ITE In The Ear

LPC Linear Prediction Coefficients

LReLU Leaky Rectified Linear Unit

LSD Log Spectral Distortion

LSGAN Least Square Generative Adversarial Network

LSTM Long Short-Term Memory

MFCC Mel Frequency Cepstral Coefficients

MLP Multilayer Perceptron

MMSE Minimum Mean Square Error

MOS Mean Opinion Score

MRCG Multiresolution Cochleagram

MSE Mean Square Error

PESQ Perceptual Evaluation of Speech Quality

PITCH Pitch-Based Feature

PLP Perceptual Linear Prediction

PNCC Power-Normalized Cepstral Coefficients

PReLU Parametric Rectified Linear Unit

PSM Phase-Sensitive Mask

ReLU Rectified Linear Unit

RMS Root Mean Square

RNN Recurrent Neural Network

SDR Signal to Distortion Ratio

SEGAN Speech Enhancement Generative Adversarial Network

SI-SDR Scale-Invariant Signal to Distortion Ratio

SMM Spectral Magnitude Mask

SNR Signal to Noise Ratio

SNR_{seg} Segmental Signal to Noise Ratio

SOTA State-of-the-Art

SSN Speech-Shaped Noise

SSNR Segmental Signal to Noise Ratio

STFT Short Time Fourier Transform

STOI Short Time Objective Intelligibility

SWOC Strengths, Weaknesses, Opportunities, Challenges

T-F Time-Frequency

TBM Target Binary Mask

WER Word Error Rate

ZCR Zero Crossing Rate

CHAPTER 1

Introduction

Speech is a sound wave generated by the vibration of the vocal cords, and it is the most common way of communication among human beings, either face to face or remotely, such as on the phone. These sound waves are then sent through the air to our ear, to be first processed by the middle and inner ear and then converted into electrical signals to be sent to the brain for sound interpretation (Hudspeth, 1989). Our brain can easily interpret speech signals if they are received in isolation. However, the speech signal is typically accompanied by other sounds due to the fact that we are surrounded by many environmental sounds, such as nature sounds, animal sounds, urban noise, etc. Whether these other sounds are of interest or not, they negatively affect the interpretation of the speech signal, especially when they are extremely high in intensity (Houtgast, 1981; Sarsenbayeva et al., 2018).

Speech enhancement is a signal processing technique that aims to improve speech quality and intelligibility by removing any other signals propagating with it. There are many applications for speech enhancement, for example, it is an essential process in hearing aids, mobile communication systems, Automatic Speech Recognition (ASR), headphones, and VoIP communication (Loizou, 2013).

The process of speech enhancement may sound simple; however, it is a longstanding issue that has attracted the attention of signal processing researchers for decades, and still has not yet been solved (Loizou, 2013; Wang and Chen, 2018). Many techniques have been proposed to tackle this challenging task, starting from the classical techniques proposed in the 70s (Boll, 1979; Loizou, 2013), which are mainly based on the statistical analysis of the relationship between speech and noise. These techniques were not very effective in removing background noise, especially intrusive noise environments, resulting in unintelligible processed speech (Loizou and Kim, 2010). A published statistic shows that noise levels are very high in places like transportation and on the street (Musa et al., 2022; Neitzel et al., 2009; McAlexander et al., 2015). With the advances in technology and the pervasiveness of speech processing in many devices, especially smart phones, speech enhancement becomes a more challenging process nowadays. As a result, more advanced and efficient techniques are needed to operate in noisy, adverse environments.

In the last decade, researchers have reached more advanced techniques to perform speech enhancement that are based on deep learning, which is a subset of machine learning and artificial intelligence (LeCun et al., 2015; Deng et al., 2014). Deep learning is a data driven approach in which an algorithm is fed with a huge amount of data to gain knowledge about how to perform a specific task, similar to the way our brain works (Schmidhuber, 2015). For deep learning-based speech enhancement, pairs of noisy and clean speech data are needed, in order to learn the mapping function that maps noisy speech to clean speech. With the recent massive availability of speech and noise data, deep learning-based speech enhancement has made a breakthrough in the research area, showing promising performance in eliminating many background noise types and dealing with challenging and intrusive noise environments. This results in generating speech with much better quality and intelligibility in comparison to the classical techniques (Yuliani et al., 2021; Wang and Chen, 2018; Saleem and Khattak, 2019).

A general drawback of any speech enhancement approach is the distortion that occurs during the noise elimination process, especially when processing a speech signal corrupted with intrusive noise environments. This distortion negatively affects the overall quality of the processed speech, more specifically speech intelligibility (Iwamoto et al., 2022). Considering the reported powerful noise suppression ability of most deep learning-based speech enhancement, the speech distortion issue becomes more significant, a fact that makes the unprocessed noisy speech sometimes preferable to human listeners than the distorted processed speech by a speech enhancement approach (Xia et al., 2020*b*). Moreover, speech distortion also affects the performance of the systems where speech enhancement is applied as a preprocessing stage, such as in ASR, as the distorted speech signal may not be understandable by the system (Wang et al., 2019). Consequently, dealing with speech distortion is a current research question in the speech enhancement field.

This thesis aims to contribute to the presented research work in the literature by proposing a new speech enhancement architecture and approach that reduces speech distortion, leading to better overall performance.

1.1 Aims and Objectives of the Thesis

The work in this thesis aims to investigate the deep learning approach for speech enhancement in noisy and adverse environments, and contribute to this research area. This is achieved through the following objectives.

- The thesis first reviews different Deep Neural Network (DNN) architectures in the literature, to understand the advantages and disadvantages of each network type. Moreover, it reviews different deep learning-based speech enhancement

approaches, and how these approaches affect the performance of each architecture type.

- The thesis then covers the full procedure that should be followed to develop a DNN for speech enhancement, including the evaluation metrics used to test the overall quality and intelligibility of the processed speech by the DNN.

- Afterwards, the work focuses on experimenting with different DNN architectures and deep learning-based supervised speech enhancement approaches. These experiments aim to validate the results and conclusions reported in the literature for some speech enhancement architectures using numerical analysis. Furthermore, they fill a gap in the literature by adding more comparisons, critical analysis, and discussions to the newly obtained results from experiments conducted to answer questions that were not investigated in the literature using numerical analysis and visual spectrogram analysis of the processed speech.

- The conclusions of the literature review and the performed comprehensive analysis, discussed above, were then used to develop a new deep learning-based speech enhancement architecture that outperforms State-of-the-Art (SOTA) speech enhancement models in the literature. The plan was to take advantage of the best DNN architectures and speech enhancement approaches proposed in the literature and validated in this thesis, in order to develop a new better speech enhancement model that minimizes speech distortion as the current research question. The proposed architecture should also compromise between performance and complexity, so as to facilitate its applicability for real time applications.

- The work in this thesis also seeks to test and improve the performance of the proposed architecture, to develop another two speech enhancement architectures. These two architectures were designed specifically for two main speech enhancement applications, hearing aids and ASR, which is the application of interest to Intelligent Voice, the sponsoring company of this PhD. The development of these optimized architectures considers improving a specific speech quality evaluation metric that highly affects the performance of the architecture for each application. Additionally, it takes into account limitations regarding network complexity and processing time that differ based on the speech enhancement application.

- The final objective of this thesis is to present a detailed critical analysis of deep learning-based supervised speech enhancement. This analysis highlights the advantages and disadvantages of this approach; moreover, it covers current challenges and issues that warrant further investigation by future research work.

1.2 Thesis Contributions

The contribution of this thesis can be divided into two parts: theoretical and practical. Theoretically, when investigating different proposed DNN based speech enhancement research in the literature, it is clear that most of the research focuses more on analyzing the quality of the output speech, and not enough work was found to compare between the processing done internally in different speech enhancement architectures. Consequently, this thesis fills the above-mentioned theoretical gap by looking deeper into the operations done inside different architectures, so as to come up with an interpretation of how each architecture deals with the speech enhancement task. This interpretation will help in understanding why certain architectures perform better than others; moreover, it will lead to defining the factors that affect the quality of the output, and then finally make some conclusions on how to improve the performance. The research area also lacks the work that show the significance of the signal preprocessing techniques used to prepare the data before the training process. For this reason, the work in this thesis also reveals the importance of these signal processing techniques and how they impact the performance of DNNs for speech enhancement.

From the practical aspect, this work presents a new deep learning-based speech enhancement architecture that outperforms SOTA speech enhancement models in the literature. Additionally, it proposes a two-stage speech enhancement approach that focuses on minimizing speech distortion by applying speech enhancement in the frequency and time domains, to take advantage of different speech features. Furthermore, the architecture was tested, modified, and optimized to be applied to two speech enhancement applications, hearing aids and ASR, where speech enhancement is applied in two different ways. In hearing aids, speech enhancement is the main process; while in ASR, speech enhancement is a preprocessing technique to the main ASR process. For hearing aids, the target is to output speech with high quality and intelligibility by evaluating the output using the well-known speech quality metrics for normal and hearing-impaired listeners. The developed architecture introduces a new speech enhancement technique, named smart speech enhancement. This technique enhances both speech and emergency noise, such as fire alarm, while mitigating undesired noise, to act as an integrated speech enhancement and alert system.

On the other hand, when applying speech enhancement as a preprocessing technique to ASR, as the main process, the performance of the speech enhancement model should also be evaluated when combined with the main process using the evaluation metric of the main process, which is Word Error Rate (WER) for ASR systems. This is due to the fact that DNNs for speech enhancement were proven to perform well on their own but unexpectedly, performance degradation for ASR systems was detected after adding the speech enhancement network. The devolved architecture solves this mismatch problem

and improves the performance of an ASR system, provided by Intelligent Voice for research purposes.

Consequently, the work in this thesis performs a fair evaluation of the proposed architecture by testing it using two different real time applications of speech enhancement.

In conclusion, the contributions of this thesis can be summarized as below:

- investigating and comparing different DNN architectures for speech enhancement using numerical analysis,
- interpreting how DNNs perform the speech enhancement task by spectrogram visualization,
- showing the effect of network hyperparameters on different DNNs for speech enhancement,
- showing the effect of the signal processing techniques used to manipulate the training data,
- comparing different training targets by showing their effect on the performance of DNNs for speech enhancement,
- comparing time and frequency domain approaches for speech enhancement using different DNN architectures,
- proposing a new deep learning-based speech enhancement architecture that outperforms SOTA speech enhancement models in the literature,
- proposing a two-stage deep learning approach for speech enhancement to minimize speech distortion,
- presenting an optimized speech enhancement architecture for ASR that solves the mismatch issue between speech enhancement models and ASR model, and
- presenting an optimized smart speech enhancement architecture for hearing aids that acts as an integrated speech enhancement and alert system.

1.3 Outline of the Thesis

The thesis is organized as follows:

Chapter 1 provides introduction to the thesis, and defines thesis aims, objectives, and contributions.

Chapter 2 reviews the classical and modern speech enhancement approaches, focusing on supervised deep learning-based speech enhancement. It provides a description and illustration of the different DNN architectures that were employed to perform speech enhancement in the literature. Moreover, it highlights the advantages and disadvantages of each architecture type. The chapter also summarizes the evolution of deep learning-based speech enhancement by showing the proposed techniques and approaches in the literature to improve the performance, including the most recently developed DNNs for speech enhancement and the current research gaps.

Chapter 3 presents the full procedure that should be followed to develop a deep learning-based architecture for speech enhancement. The procedure can be described using five sub-processes: data collection, data preprocessing, feature extraction, artificial neural network implementation, training target choice, and evaluation of the processed speech. The chapter explains each sub-process by covering different techniques that can be applied to perform each sub-process.

Chapter 4 compares the performance of seven different DNNs for speech enhancement, belonging to three well-established DNN categories. The comparison is based on processed speech quality, the performance of each network in challenging noise environments, network generalization, complexity, and processing time. The chapter covers a research gap by providing answers to some research questions, such as the factors affecting the choice of learning domain, the effect of signal processing techniques and network hyperparameters on the performance.

Chapter 5 presents a newly developed speech enhancement architecture and proposes a two-stage speech enhancement approach, to deal with speech distortion. The architecture is designed to compromise between performance and complexity by adjusting the hyperparameters of the developed DNN. The two-stage approach minimizes speech distortion, leading to further improvement in speech enhancement performance.

Chapter 6 investigates two speech enhancement main applications, hearing aids and ASR. In this chapter, the developed architecture in Chapter 5 was modified and improved to be applied to each application, to finally show the applicability of the presented work from a real world perspective.

Chapter 7 concludes the thesis using critical analysis of the deep learning-based speech enhancement approach. This analysis highlights the strengths, weaknesses, opportunities, and challenges of the approach. Finally, the chapter gives recommendations for future research.

1.4 Conclusion

This chapter introduced the work presented in this thesis. It included thesis objectives, contributions, and outline. The next chapter will review classical and modern speech enhancement approach, and demonstrates different DNN architectures for speech enhancement.

CHAPTER 2

A Literature Review of Deep Learning-based Speech Enhancement and its Applications

2.1 Introduction

Speech enhancement has two main approaches: the classical and the modern. The classical approach is based on statistical assumptions of the noise presented in the speech signal, and the analysis of the relationship between speech and noise. While the modern approach is based on more advanced techniques using artificial intelligence, or more specifically deep learning algorithms. This chapter mainly reviews the deep learning approach for speech enhancement. However, a brief discussion will be first presented on classical speech enhancement techniques. Afterwards, the deep learning-based speech enhancement approach will be illustrated, followed by a review of the different DNN architectures in the literature that have been employed for speech enhancement. Finally, a discussion will be presented about the evolution of the deep learning-based speech enhancement field, including current challenges and research points under investigation by recent research.

2.2 Classical Speech Enhancement Approach

Classical techniques have been widely used in the field of speech enhancement. These techniques are based on statistical assumptions and models for the speech and noise signals (Loizou, 2013). The challenging part of this approach is how successful these models and assumptions are at describing the relationship between speech and noise. Although some of these techniques were reported to mitigate the noise that exists in the speech signal (Uemura et al., 2009), the way they work is not a generalized way of removing different noise types, because these statistical assumptions are not always fulfilled in some intrusive noise environments. Consequently, these methods are more effective when being applied to environments with a relatively high Signal to Noise Ratio (SNR), or in the case of stationary noise conditions (Hu and Loizou, 2007*b*). Although classical multichannel approaches showed improvement in speech intelligibility (Ortega-García and González-Rodríguez, 1996), it was reported that the classical tech-

niques performance is not satisfactory (Drullman, 1995; Loizou and Kim, 2010). In the following subsections, a brief overview is presented for three different classical speech enhancement techniques.

2.2.1 Spectral Subtraction

Spectral subtraction is one of the first techniques used to remove noise from speech. Here, noise removal is achieved by estimating the magnitude spectrum of the noise, and then subtracting it from the noisy speech magnitude spectrum (Boll, 1979). This technique is based on two assumptions: the first is that the noise is additive so as to be able to do the subtraction procedure; while the second is that the first few frames of the noisy signal have only noise, which is essential to estimate the noise spectrum that is not supposed to vary between frames. Based on these assumptions, the estimation of the clean speech spectrum is obtained by subtracting noise regions (speech pauses) from each frame of the noisy signal, as speech pauses are considered as noise in this technique. However, there are two major drawbacks for spectral subtraction (Malca and Wulich, 1996). The first is that in the case of non-perfect estimation of the noise spectrum, the speech signal will be significantly distorted by the subtraction process (Verteletskaya and Simak, 2011). The second drawback is the presence of unpleasant, unnatural, musical, remnant noise, which accompanies the enhanced output speech signal (Goh et al., 1998). As a result, researchers thought about developing many modified versions of the spectral subtraction technique in order to present solutions to these issues (Kamath and Loizou, 2002; Upadhyay and Karmakar, 2015).

2.2.2 Wiener Filter

Wiener filter aims to obtain an estimate of the clean speech signal by minimizing the mean square error between the estimated and real clean speech. It is assumed in this approach that the noise is additive and its power spectrum is uncorrelated to that of the speech signal (Upadhyay and Jaiswal, 2016). The Adaptive Wiener filter is an example of this technique that reduces the noise estimation error by attenuating each frequency component of the noisy speech by a certain amount that depends on the power of the noise at this particular frequency. It consists of a digital filter stage and an adaptive algorithm stage. The noisy signal first passes through the digital filter to output an estimate of the desired clean speech signal. Afterwards, the error is calculated between the real and estimated clean speech signal generated by the digital filter, to be fed to the adaptive algorithm. The adaptive algorithm is responsible for decreasing this error value by adjusting the coefficients of the digital filter (Abd El-Fattah et al., 2008). Although Wiener filter-based techniques lead to better clean speech estimation than spectral subtraction (Vihari et al., 2016), musical noise still exists in this approach (Amehraye et al.,

2008; Alam and O'Shaughnessy, 2011).

2.2.3 Signal Subspace

The Signal Subspace speech enhancement method is based on the transformation of the noisy speech signal into two uncorrelated and orthogonal subspaces, known as signal and noise subspaces (Ephraim and Van Trees, 1995). This technique is based on two assumptions: the first is that the speech signal follows a specific model and has certain characteristics, and the second is that the interfering noise is uncorrelated additive white noise, which is a very specific noise type. However, other studies conducted some analysis to modify this approach to deal with colored noise as well (Hu and Loizou, 2003; Lev-Ari and Ephraim, 2003). The decomposition of speech and noise subspaces is done through estimators, in which nulling of the noise subspace is performed. An advantage of this approach is that in some estimators the residual musical noise in the signal subspace is also mitigated, taking into consideration speech distortion (Hansen et al., 1998). This approach was also proven to be effective in noise reduction; however, it is characterized by high computations, which restricts its applicability. Additionally, total removal of musical noise using this approach results in high speech distortion (Hermus et al., 2006; Hansen et al., 1998).

2.3 Modern Speech Enhancement Approach

Modern speech enhancement approach is based on Artificial Neural Network (ANN)s, which are networks that mimic the way our brain works. These networks generate the target output, by learning a mapping function that maps the input to the target output using computational nodes similar to the neurons in our brain (Jain et al., 1996). This biological analogy between the neurons in our brain and the nodes in ANNs is due to the fact that the inputs are gathered and processed in the computational nodes using linear and nonlinear operations that generate another form of the inputs. Afterwards, these processed data are sent to another connected node for further computation. This models the way that the dendrites in the neuron receive inputs to be integrated and combined by the cell body to generate spikes, and now these spikes are sent to another neuron via the axon. Despite the fact that this analogy is loose because biological neurons do much more complex computations than nodes, all neuroscientists believe that the nodes in an ANN approximate biological neurons in a crude way (Eluyode and Akomolafe, 2013).

There are three basic components for any ANN: input features, hidden layers, and an output layer (Jain et al., 1996). The ANN is fed by a series of input features, which are the variables that could be used to predict the output. The importance of these features is that they carry more relevant and meaningful information about the input data, such as harmonics for audio data, that helps in directing the network towards the needed

functionality. Any ANN consists of at least one hidden layer with a number of hidden nodes, which apply some nonlinearity to the input in order to learn more advanced features, so as to be able to give a good estimate of the required output. The final layer of any ANN is called the output layer and this layer is responsible for producing the final estimated output of the network by integrating and scaling the output from the hidden layers to the desired target range (LeCun et al., 2015). The network maps an input to a specific target output by adjusting its parameters through a procedure known as the learning process.

The learning procedure of ANNs is based on two fundamental processes: forward propagation and backward propagation. In forward propagation, the input features are fed to the nodes of the hidden layers that learn more advanced features of the input to better predict the output. At the end of the forward propagation process, the error, or what is called the loss function, is computed, which is a function that measures the difference between the estimated output and real output (Glorot and Bengio, 2010). The loss function measures how well the network parameters are doing on learning the mapping function, and then based on its value the network adjusts its parameters through the back propagation step. During back propagation, the network updates the values of its parameter so as to minimize the loss function, and this is done by calculating the gradient of the loss function with respect to the network's weights. The network goes through forward and backward propagation recursively, until learning the mapping function that gives the best prediction of the target output (LeCun et al., 2015).

ANNs have parameters and hyperparameters that should be tuned to obtain a correct prediction. The weight (W) and bias (b) are called the network parameters, and we must initialize them at the beginning of the learning process, then the network fine tunes these parameters during the backward propagation process. These parameters are randomly initialized, and it is preferred to set these random values to very small values using a normalization technique, as large value initialization has been proven to slow down the learning process (Ioffe and Szegedy, 2015). Besides the network parameters W and b, there are also hyperparameters, such as the learning rate α, which controls the speed of the training process; the number of iterations, which is the value that defines how many times the network will go through the forward/backward propagation process; and the number of hidden layers and hidden units in the network, which affect the network performance and measure its complexity (LeCun et al., 2015).

The choice of activation functions, which are the non-linear functions applied in the hidden layers of the network, is another hyperparameter of the network. The non-linearity of those functions helps the network to learn complex features of the input data, and hence be able to better predict the output (Karlik and Olgac, 2011). There are many types of activation function used in ANNs, Linear, Sigmoid, TanH, Rectified Linear Unit (ReLU) and its edited versions: Leaky Rectified Linear Unit (LReLU), Ex-

ponential Linear Unit (ELU), and Parametric Rectified Linear Unit (PReLU) are the most popular ones. Linear is a function that simply produces an output proportional to the input, and it is normally used in the output layer as it does not add any nonlinearity. Sigmoid is an always positive function between 0 and 1, while TanH gives an output between -1 and 1. ReLU outputs zero if the input is negative and gives the same value for a non-negative input. LReLU, ELU, and PReLU are edited versions of ReLU that give a small value output for a negative input instead of zero, so as to overcome the dying ReLU problem (Pedamonti, 2018), which will be discussed later in Chapter 4. It should be noted that ReLU and its edited versions are the most common activation functions used in today's research because they are found to be the most similar functions to the non-linear computations done in biological neurons and proven to produce better performance (Grossberg, 1988; Maas et al., 2013), while Sigmoid and TanH are less commonly used. Moreover, ReLU is proven to solve the vanishing and exploding gradient problem for DNNs (Glorot and Bengio, 2010) that will be explained later.

All these hyperparameters and many others not mentioned, should be set to a value that depends on the problem the network is trying to solve. Many architectures have been reported in the literature, however, selecting between different deep learning models has been mainly empirical. This is because the difficulty of predicting the best values for these hyper parameters, so it is most common for researchers to get these values using practical trials (Arel et al., 2010). A comprehensive experimental analysis and discussion about these hyperparameters will presented in Chapter 4, to fill this gap in the literature.

In order to obtain a significant improvement in the performance of ANNs, a sufficient training data is required for the learning procedure. There are two common problems that may arise when training an ANN, known as the variance and bias problem (Schmidhuber, 2015). Variance is the problem of overfitting to the training dataset, which means the network is performing very well on the training data, but unable to generalize this good performance on unseen test data. A technique called regularization is used to overcome this problem, and the most common one used nowadays is called dropout regularization (Srivastava et al., 2014). In dropout, the network randomly drops a certain percentage of the hidden units in the hidden layers during the training process. In this way, the learning process becomes more efficient, because dropout prevents network dependence on only some specific features during the training, and in turn makes the network more robust to the changes in the test set. Although this technique negatively affects network performance on the training set, it improves the network generalization capability (Park and Kwak, 2016).

On the other hand, bias is the problem of underfitting to the training dataset, which means the network is unable to perform the task it is required to do. This problem could happen when the network architecture is not appropriate to the task the network

is doing, or because of the insufficient input data to the network (Schmidhuber, 2015). Increasing the size of the input dataset is proven to have a positive impact on network performance; however, this solution works better for networks with many hidden layers, i.e. DNNs (LeCun et al., 2015).

A DNN is simply a neural network with more than one hidden layer. By increasing the number of layers, the network is expected to give better performance, as each layer will give more information to the network that would help through the learning process (Deng et al., 2014). Deep networks were not used widely due to the vanishing and exploding gradient issue, which arise in deep architectures, as the mathematical derivative terms that are less than 1 become smaller and smaller when going deeper into the network due to the multiplication operations, until the gradient tends towards zero, and hence vanishes. On the other hand, values bigger than 1 become bigger and bigger until they tend to infinity, leading to the gradient exploding (Ioffe and Szegedy, 2015). In order to solve this problem, all gradient values should be limited to 1 or 0, so as to not be affected by the multiplication process through the hidden layers; this is exactly what the activation function ReLU and its edited versions do (Pedamonti, 2018). Deep learning is the most common technique used in today's research, and this is because of two main reasons: the first is the huge amount of data available nowadays (Dytman-Stasienko and Weglinska, 2018), the second is the complexity of the problems the network is presented with, which require a deeper neural network to solve (Deng et al., 2014).

Considering the speech enhancement task as a problem of mapping noisy speech to clean speech, this problem could be solved by training a neural network to learn this mapping function, and due to the fact that this mapping function is very complex and highly nonlinear, using a deep neural network will be a better choice than shallow neural networks (Wang and Chen, 2018). Recently, DNN-based speech enhancement has made a breakthrough in the speech de-noising process, and many architectures have been proposed (Saleem and Khattak, 2019). In the following subsections, a review on these architectures will be presented.

2.3.1 Multi-Layer Perceptron (MLP)

The Multilayer Perceptron (MLP) is one of the most basic types of neural network, in which the nodes of the hidden layers are fully connected, that is why it is sometimes called a fully connected DNN. Supervised deep learning-based speech enhancement research was first based on the MLP, as it achieved a significant improvement in noisy speech quality and intelligibility, compared to the classical approaches.

An MLP with three hidden layers for speech enhancement was first proposed by (Xu et al., 2014*b*); afterwards, the work presented in (Zhao et al., 2016) added one

more hidden layer and considered training the architecture using reverberant speech as well as noisy speech, which led to further improvement in speech intelligibility in both noisy and reverberant conditions. Another learning approach for the MLP is proposed in (Wang, 2017), which is based on 84 speech features used as an input to the MLP. The initialization of the network's weights was performed through an unsupervised learning scheme; while the speech enhancement training process was performed using supervised learning. The use of this approach results in better speech enhancement performance for the MLP. The effect of large-scale training on network generalization ability was investigated by (Chen et al., 2016), where an MLP was trained using a huge number of noise environments, in comparison to the number of noises used in previous speech enhancement research. The outcome of this investigation showed that increasing the number of noise environments used in the training process will significantly improve the generalization of the MLP for mismatched noise conditions. Many other speech enhancement research were also found to be based on the MLP architecture (Kumar and Florencio, 2016; Tu and Zhang, 2017).

One of the main advantages of the MLP is its powerful ability to learn speech features through the fully connected hidden layers, which use a huge number of connections between their nodes. However, this fully connected architecture results in increased network complexity due to the large number of computations inside the hidden layers, which in turn increase the computational cost and processing time of the network (Liu et al., 2022). Consequently, the use of Graphical Processing Unit (GPU)s is essential when training MLPs, to speed up processing time. GPUs are freely available on the cloud, such as Google Colab (Nair and Kumar, 2021), and they generally improve the training process of DNNs (Pal et al., 2019).

Model size is another aspect that should be considered when implementing an MLP for real time applications. Most MLPs have a large number of parameters, leading to a big model size, which may not fit onto the hardware of some speech enhancement applications, such as hearing aids and mobile communication (Sze et al., 2017). Finally, the performance of MLPs was proven to be highly affected by the representation of the input noisy speech. Speech signal is originally represented as a time series data, which is known as time domain speech features. Another representation of the speech signal is when being decomposed into harmonics, to give more coherent structure using Time-Frequency (T-F) representation, which is known as frequency domain speech features. The performance of MLP was shown to be significantly degraded when using time domain-based speech features. This limits the application of MLPs, especially in time domain-based speech enhancement (Fu et al., 2017).

2.3.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is another architecture used to solve the computation problem of MLPs by using the convolution operation in both forward and backward propagation steps, so as to reduce the network parameters. CNNs were first made for image related tasks in order to be able to work with the huge amount of the images' parameters, but then it is proven to be very effective in audio processing as well (Deng et al., 2014). The advantage of the CNN is its dependence on the idea of convolution, which results in fewer network parameters because of two reasons: parameter sharing and the sparsity of connections. Parameter sharing means that units within a convolution layer take advantage of a feature map generated from each unit within the same layer plane through weight sharing, while sparsity of connections means that the output value in each layer does not depend on all the inputs of the previous layer (Gonzalez, 2018). Most CNN architectures consist of three main layers: convolution layer, pooling layer, and fully connected layer. The convolution layer is the layer in which the convolution operation is done by sliding a matrix called the kernel over the input features matrix, so as to finally output a feature map matrix. Equation 2.1 defines the convolution operation:

$$S(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n), \tag{2.1}$$

where S is the feature map matrix, I is the input matrix, and K is the kernel, while (i, j) represents the pixel position in the feature map, and (m, n) is the position of the input matrix. It should be mentioned here that CNN-based networks use multiple kernels.

The pooling layer is responsible for reducing the dimensionality of the convolution layer output by only keeping the speech features with important information and discarding the rest. There are two types of pooling: max pooling and average pooling, the difference is that max pooling takes the maximum value of the selected speech features, while average pooling takes the average value (Boureau et al., 2010). A CNN typically has one or more fully connected layers at the end. Here, the neurons are fully connected to all activations of the previous layer, as in the case of the MLP. The function of this layer is to adjust the size of the output, by integrating all the learned features from the previous layers, so as to generate the final prediction of the network (Schmidhuber, 2015).

The above discussed convolution type, expressed in Equation 2.1, is the traditional type of convolution performed in the past, known as a two dimensional (2D) convolution. Another type of convolution that is used nowadays is the one dimensional (1D) convolution, in which the convolution operation works along one axis (Kiranyaz et al., 2021). The same concept of 2D convolution is applied to 1D, but the difference is in

the kernel used and the structure of the input data. The kernel in 1D convolution is applied horizontally across one feature axis, not along two axes as in the case of 2D convolution. At the same time, 1D convolution takes a 1D vector, unlike 2D convolution which accepts a 2D matrix. These differences make 1D convolution more suitable for sequence data, such as audio processing. Moreover, 1D convolution results in a lower computational cost, which helps in developing architectures suitable for real time applications (Kiranyaz et al., 2019).

The convolution operation also has many hyperparameters, such as padding size, stride size, and dilation rate. Changing these hyperparameters will lead to different types of convolution that impact the performance of the CNN (Li et al., 2021).

Padding is the process of stacking zero vectors to the borders of each feature matrix before performing the convolution operation, and padding size defines how many zero vectors should be added to the input feature matrix. This technique overcomes a drawback of the convolution operation, which is that the features along the borders of the input feature matrix are not highly considered compared to the features in the center of the matrix, because they contribute to the resulting feature map one time only. This issue results in ignoring important features as the processing proceeds into deeper layers, which negatively affects the network performance (Simard et al., 2003). Consequently, this zero-padding process will help in preserving important information by increasing the size of the input feature matrix to center the features in the borders without any other change in the input characteristics. Moreover, this size increase allows for using kernels with larger size, which enhances CNN learning. Furthermore, zero padding keeps the size of the input the same as the output after the convolution operation, which also improves the training process (Li et al., 2021). However, it has been proven with some practical experiments that in some applications zero padding may result in a slight enhancement in the performance of CNNs (Lecun, 2012).

Stride size is another hyperparameter for the convolution operation, which defines another type of convolution, the strided convolution. This hyperparameter identifies the length of column and row step for the kernel used in the convolution operation. Increasing the stride size will lead to a process similar to the downsampling operation to the output by foregoing some of the information, which will also decrease the amount of computations and the processing time. However, using a large stride size may negatively affect the network performance in reconstructing the input back to its original size correctly, so there is a tradeoff here between resource consumption and network efficiency (He and Sun, 2015).

Dilated convolution is a type of convolution that compromises between kernel size and the number of network parameters. In this type of convolution, a hyperparameter called the dilation rate is defined that introduces spacing between the kernel used, in order to increase the receptive field, which is defined as the region in the input space that

a particular CNN's feature is affected by, while keeping the total number of network's parameters the same, and hence have the same computational cost (Wei et al., 2018).

CNNs have been widely used in speech enhancement. The work in (Kounovsky and Malek, 2017) used a CNN-based speech enhancement architecture of two convolution layers and two fully connected layers to predict the log power spectra of the clean speech. Another work, Chakrabarty et al. (2018), proposes an improved CNN architecture trained using a masking-based training target (Wang et al., 2014); masking and mapping targets explanation will be provided in Chapter 3. While in (Fu et al., 2017), another version of CNN is proposed, named Fully Convolutional Neural Network (FCNN), in which the fully connected layers are replaced with convolution layers in an attempt to decrease the computational cost added by the fully connected layers. A comparison was also conducted in the same work between the basic CNN architecture and the FCNN for speech enhancement, and the results showed that FCNNs perform better and faster than CNNs. Recently, the work shown in (Ouyang et al., 2019) and (Pirhosseinloo and Brumberg, 2019) used a combination of 1D and 2D dilated convolutions in order to implement a FCNN and reported a further improvement.

2.3.3 Denoising Autoencoders (DAE)

The autoencoder is a special type of DNN, which aims to output a similar representation to the input using two separate networks: an encoder and decoder. The encoder compresses an input, *X*, by removing any unimportant information so as to finally generate a compact form of the input data, *Z*, and then the decoder reconstructs an estimated form of the input, $\tilde{X}$ (Schmidhuber, 2015). Based on this fact, the autoencoder is considered as an unsupervised learning scheme, because it relies only on the input data, with no target. This type of architecture has many applications, such as data compression (Hinton and Salakhutdinov, 2006) and visualization (Hosseini-Asl et al., 2015; Petridis and Pantic, 2016). In order to compute the loss function of the whole network, suppose that *Z* and $\tilde{X}$ are functions of their inputs, as given in Equations 2.2 and 2.3, respectively:

$$Z^i = f_1(W_1 X^i + b1), \tag{2.2}$$

$$\tilde{X}^i = f_2(W_2 Z^i + b2), \tag{2.3}$$

where W_1 and b_1 are the weight and bias parameters of the encoder network, respectively, while W_2 and b_2 are the weight and bias parameters of the decoder network, respectively. f_1 and f_2 are the nonlinear functions applied in the encoder and decoder networks, respectively, and *i* is the input index. If the Minimum Mean Square Error (MMSE) is the loss function used in the training process, then the cost function, *J*, that the network is trying to minimize in this case can be expressed as a function of the in-

put, *X*, and the encoder and decoder parameters, *W* and *b*, as given below in Equations 2.4 to 2.6:

$$J = \sum_{i=1}^{m} (\tilde{X}^{(i)} - X^{(i)})^2, \tag{2.4}$$

$$= \sum_{i=1}^{m} (f_2(W_2 Z^{(i)} + b_2) - X^{(i)})^2, \tag{2.5}$$

$$= \sum_{i=1}^{m} (f_2(W_2 f_1(W_1 X^{(i)} + b1) + b_2) - X^{(i)})^2. \tag{2.6}$$

Taking advantage of the compression process on the input data in the encoder network, Denoising Autoencoder (DAE)s have been widely used most recently in speech enhancement. The idea of DAEs is based on the fact that noise is considered as unimportant information when trying to map from noisy to clean speech, so it is reduced significantly during the compression process to produce clean speech bottleneck features, which are compact form of the input data, and then the decoder reconstructs the clean audio (Vincent et al., 2010). Based on this, training DAEs for speech enhancement can be considered as supervised learning for DNNs with a target output *Y*, representing the real clean speech. Bottleneck features lead to significant improvement in many research areas, such as speech recognition (Yu and Seltzer, 2011; Sainath et al., 2012; Grézl et al., 2007), audio classification (Zhang et al., 2016; Mun et al., 2016), speech synthesis (Wu and King, 2016) and speaker recognition (Yaman et al., 2012).

DAEs can be implemented using an MLP network, known as Deep Denoising Autoencoder (DDAE), and this is achieved by reducing the number of the hidden nodes through the hidden layers. Another type of DAE is the Convolutional Denoising Autoencoder (CDAE), based on the use of convolution layers in both the encoder and the decoder. DAEs are widely used for speech enhancement; for example, the work done in (Lu et al., 2013) presents a DDAE speech enhancement architecture, while a CDAE architecture was proposed in (Grais and Plumbley, 2017). The CDAE is the most commonly used speech enhancement architecture in recent research (Park and Lee, 2016; Pandey and Wang, 2019; Ouyang et al., 2019), because of its lower number of parameters. However, autoencoders in general cannot perfectly construct a similar representation of the input, which means the output will experience a loss, and this is the main issue of this type of DNN architecture (Coşkun et al., 2017).

2.3.4 Recurrent Neural Network (RNN)

The previously discussed DNNs belong to a category called feedforward (FF) neural networks, as the signal flows in one direction from the input to output. Another category of neural networks is known as the Recurrent Neural Network (RNN), in which the

output of the hidden node is fed back to the same node while also being an input to the next node. When making a prediction, these feedback nodes makes this architecture takes into consideration the current input and also what was learned from the previously received inputs (LeCun et al., 2015). These feedback connections are also useful when working with sequence data that change over time, and in the case of sequence-to-sequence mapping as in the speech enhancement task (El Hihi and Bengio, 1996).

Some speech enhancement architectures in the literature are based on RNNs, such as the work in (Sun et al., 2017) that used an RNN architecture with multiple mapping targets, and then a comparison was made to a basic MLP architecture. The work in (Huang et al., 2014) also compared RNNs with an MLP architecture after adding an extra time frequency masking layer that enforces some reconstruction constraints when converting from the frequency domain back to the time domain. According to the reported results, the RNN proved to be a powerful architecture for speech enhancement; however, the disadvantage of the RNN is its instability when the ReLU activation function is used, so the training of this network is difficult because it may suffer from the gradient vanishing and exploding problems (Bengio et al., 1994; Pascanu et al., 2013). Recently, the use of other activation functions facilitates the training of RNNs for speech enhancement (Strake et al., 2019); moreover, RNN-based architectures such as Long Short-Term Memory (LSTM)s and Gated Recurrent Unit (GRU) have shown good performance for real-time speech enhancement (Braun and Tashev, 2020; Weninger et al., 2015).

2.3.5 Generative Adversarial Network (GAN)

The Generative Adversarial Network (GAN) is another DNN architecture for speech enhancement. This architecture is a combination of two networks: the discriminator network and the generator network. The generator network works in the same way as an autoencoder, as its role is to generate a similar representation of the input data, while the discriminator network acts as a binary classifier trained to discriminate between a real and fake input representation. The generator output is fed to the discriminator as an input, and then based on the decision of the discriminator, the generator network adjusts its parameters to produce a better representation of the input data (Creswell et al., 2018). The advantage of this network over DAEs is that it is not only trying to remove the noise using a bottleneck representation, but also takes into consideration another important parameter, which is the correlation between the input and the output, and this improves the performance of DNNs for speech enhancement (Pascual et al., 2017).

Much research in the literature employed GANs for speech enhancement. The work in (Pascual et al., 2017) first proposes the development of a GAN network for speech enhancement using the Speech Enhancement Generative Adversarial Network (SEGAN)

architecture and reported better performance in comparison to other speech enhancement techniques available at the same time. Improved versions of SEGAN were then developed to further improve the performance (Li et al., 2022; Baby, 2020; Donahue et al., 2018). A GAN network was also presented in (Fu et al., 2019) that aims to maximize one of a two speech evaluation metric during the training process, based on the target application. It can improve the Perceptual Evaluation of Speech Quality (PESQ) score, which evaluates speech quality; or the Short Time Objective Intelligibility (STOI) score, which assesses speech intelligibility, . Recently, the work in (Phan et al., 2020) shows that cascaded GANs can significantly improves the performance of GANs for speech enhancement.

A disadvantage of GANs is the difficulty of adjusting the hyperparamters of the generator and discriminator to work successfully together, in order to finally generate an estimate to the clean speech (Mao et al., 2017). Training DNNs, in general, is challenging; and here a two DNNs are trained simultaneously to learn the mapping function that maps noisy speech to clean speech, which increases the difficulty of the training process. It was also reported that GANs are sometimes not very effective for speech enhancement, and specific adjustments are needed in order to obtain good results (Pandey and Wang, 2018*b*).

2.3.6 Other Speech Enhancement Architectures

Other speech enhancement approaches use a combination of two types of architecture, such as combining a CNN with an RNN, as presented in (Zhao, Zarar, Tashev and Lee, 2018; Tan and Wang, 2018), this architecture is known as the Convolutional Recurrent Network (CRN). The role of the CNN is to extract more advanced features from the input data; these features are then concatenated and fed to the RNN for further processing and to estimate the target clean speech. Moreover, other research are based on integrating deep learning-based speech enhancement techniques with the classical techniques (Nicolson and Paliwal, 2019), or with other learning techniques such as reinforcement learning (Koizumi et al., 2017). These approaches have been proven to be promising as well (Yuliani et al., 2021); however, the complexity that may arise from integrating different techniques is a drawback, which may restrict some speech enhancement real-time applications (Angelov and Sperduti, 2016).

Although strengths and weaknesses of different approaches are covered in various publications, a critical analysis of using different deep learning based speech enhancement approaches is not clearly reported in the literature. This analysis will be presented in this thesis in Chapter 7.

2.4 Deep Learning Based Speech Enhancement Evolution

Deep learning-based speech enhancement has shown a massive progression over the last decade. Many DNNs have been proposed for speech enhancement, starting from the simple architectures with few hidden layers (Xu et al., 2013; Wang and Wang, 2015) to the latest more complex and deeper networks (Wang, Wang and Wang, 2020; Nair and Koishida, 2021). A summary of these architectures is presented in Tables 2.1 to 2.4 and discussed below.

In 2013, the use of MLPs for speech enhancement was the first proposed approach, where an MLP network with few hidden layers was trained to map noisy speech to clean speech (Xu et al., 2013; Narayanan and Wang, 2013; Xu et al., 2014*b*). The DAE-based MLP (Yu and Seltzer, 2011) was then introduced in 2014, which was shown to be promising for speech enhancement (Liu et al., 2014). Afterwards, the use of masking-based targets was proven to improve the performance of DNNs for speech enhancement, especially speech intelligibility (Wang et al., 2014), and MLP networks for speech enhancement were trained using masking targets to generate better estimated clean speech (Wang and Wang, 2015).

In 2016, more MLP-based architectures were proposed for speech enhancement, aiming to improve the performance using different masking training targets (Chen et al., 2016). Furthermore, the use of complex masking targets was proposed in (Williamson et al., 2016) that enhances both the magnitude and phase spectrum, unlike previous speech enhancement architectures that enhance only the magnitude spectrum, assuming that the phase is less affected by the noise compared to the magnitude spectrum (Wang and Lim, 1982). However, phase enhancement was proven to be essential, especially in intrusive noise environments (Shi et al., 2006). A deeper MLP network was then presented by (Tu and Zhang, 2017) in 2017, where the use of skip connections was proven to be essential for deep architectures. The MLP was then shown to improve speech intelligibility for hearing aids, as a speech enhancement application (Chen et al., 2016; Wang, 2017; Goehring et al., 2017). In 2017, the use of CNNs (Kounovsky and Malek, 2017) and RNNs (Chen et al., 2017; Sun et al., 2017) became more common, and showed better performance than the MLP. Additionally, the first FCNN (Fu et al., 2017) and GAN architecture for speech enhancement, SEGAN (Pascual et al., 2017), were proposed in the same year, which process the noisy speech in the time domain, unlike previous speech enhancement architectures that were operating in the frequency domain.

In 2018, many time domain-based speech enhancement FCNN architectures were proposed, leading to better performance (Fu et al., 2018; Rethage et al., 2018). Moreover, the implementation of the CRN was introduced, achieving further performance improvement (Zhao, Zarar, Tashev and Lee, 2018; Tan and Wang, 2018). Furthermore,

GAN architectures that perform in the frequency domain were presented (Soni et al., 2018; Baby, 2020); at the same time, many CDAE-based architectures were proposed showing outstanding speech enhancement performance in the time domain (Pandey and Wang, 2018*a*; Macartney and Weyde, 2018). Additionally, the introduction of a two-stage speech enhancement was presented in (Zhao, Wang and Wang, 2018), to perform speech denoising and reverberation using two cascaded MLP architectures. In 2019, improved FCNN and GAN-based architectures were proposed (Germain et al., 2019; Ouyang et al., 2019; Fu et al., 2019). Finally, at the end of 2019, a hybrid two-stage architecture was presented in Strake et al. (2019) that performs speech enhancement and reconstruction in two separate stages, in order to minimize speech distortion, using a combination of RNN and CDAE architectures. Different models based on time and frequency domain features are reported in the literature, however, limited work has been reported in combining both features. Chapter 4 and 5 fill this research gap by comparing the time and frequency domain approaches, and developing a two-stage speech enhancement model in the time and frequency domains.

In DNN-based speech enhancement, speech distortion is the main drawback of the speech denoising process, especially at low SNR levels, in which the DNN removes part of the speech spectrum while trying to remove the background noise. The significance of this issue appears in subjective testing, where some of the listeners prefer the noisy speech version rather than the clean one because of the distortion, which mainly affects speech intelligibility (Xia et al., 2020*b*). Many of the proposed DNNs for speech enhancement are very effective in improving the quality of noisy speech; however, it is still very challenging to avoid the distortion that accompanies the noise removal process (Wang et al., 2019; Strake et al., 2019). The generalization ability of DNNs is another issue that becomes more significant when testing the network using a mismatched test corpus (Pandey and Wang, 2020*b*), and poor network generalization also causes speech distortion.

Consequently, recent research in 2020 and 2021 is giving more attention to this distortion issue by proposing different techniques and approaches to overcome it. This can be achieved by applying a single stage architecture that is designed to minimize distortion (Koizumi et al., 2020; Xia et al., 2020*b*; Pandey and Wang, 2020*a*; Défossez et al., 2020). Another approach is to develop a two-stage architecture, with a second stage dealing with this distortion issue (Nair and Koishida, 2021; Phan et al., 2020; Wang, Wang and Wang, 2020). Chapter 5 will contribute to this research area by developing a speech enhancement model that minimizes speech distortion. Finally, other work was found to review and analyze DNNs for speech enhancement and investigate different deep learning approaches (Saleem and Khattak, 2019; Wang et al., 2014; Yuliani et al., 2021; Wang and Chen, 2018).

Table 2.1 Best performing speech enhancement DNNs in years 2013 to 2015

Reference	Representation	Network	Target	Stages	Year
Xu et al. (Xu et al., 2013)	Time-Frequency	MLP	Mapping	1	2013
Narayanan and Wang (Narayanan and Wang, 2013)	Time-Frequency	MLP	Masking	1	2013
Liu et al. (Liu et al., 2014)	Time-Frequency	DDAE	Mapping	1	2014
Xu et al. (Xu et al., 2014*b*)	Time-Frequency	MLP	Mapping	1	2014
Wang and Wang (Wang and Wang, 2015)	Time-Frequency	MLP	Masking	1	2015

Table 2.2 Best performing speech enhancement DNNs in years 2016 to 2017

Reference	Representation	Network	Target	Stages	Year
Williamson et al. (Williamson et al., 2016)	Time-Frequency	MLP	Masking	1	2016
Chen et al. (Chen et al., 2016)	Time-Frequency	MLP	Masking	1	2016
Tu and Zhang (Tu and Zhang, 2017)	Time-Frequency	MLP	Masking	1	2017
Wang (Wang, 2017)	Time-Frequency	MLP	Masking	1	2017
Goehring et al. (Goehring et al., 2017).	Time-Frequency	MLP	Masking	1	2017
Chen et al. (Chen et al., 2017)	Time-Frequency	RNN	Masking	1	2017
Sun et al. (Sun et al., 2017)	Time-Frequency	RNN	Mapping	1	2017
Kounovsky and Malek (Kounovsky and Malek, 2017)	Time-Frequency	CNN	Mapping	1	2017
Fu et al. (Fu et al., 2017)	Time-domain raw waveform	FCNN	Mapping	1	2017
Pascual et al. (Pascual et al., 2017)	Time-domain raw waveform	GAN	Mapping	1	2017

Table 2.3 Best performing speech enhancement DNNs in years 2018 to 2019

Reference	Representation	Network	Target	Stages	Year
Fu et al. (Fu et al., 2018)	Time-domain raw waveform	FCNN	Mapping	1	2018
Rethage et al. (Rethage et al., 2018)	Time-domain raw waveform	FCNN	Mapping	1	2018
Zhao et al. (Zhao, Zarar, Tashev and Lee, 2018)	Time-Frequency	CRN	Mapping	1	2018
Tan and Wang (Tan and Wang, 2018)	Time-Frequency	CRN	Mapping	1	2018
Soni et al. (Soni et al., 2018)	Time-Frequency	GAN	Masking	1	2018
Macartney and Weyde (Macartney and Weyde, 2018)	Time-domain raw waveform	CDAE	Mapping	1	2018
Pandey and Wang (Pandey and Wang, 2018*a*)	Time-domain raw waveform	CDAE	Mapping	1	2018
Zhao et al. (Zhao, Wang and Wang, 2018).	Time-Frequency in both stages	MLP - MLP	Masking - Mapping	2	2018
Germain et al. (Germain et al., 2019)	Time-domain raw waveform	FCNN	Mapping	1	2019
Ouyang et al. (Ouyang et al., 2019)	Time-Frequency	FCNN	Mapping	1	2019
Fu et al. (Fu et al., 2019)	Time-Frequency	GAN	Mapping	1	2019
Strake et al. (Strake et al., 2019)	Time-Frequency in both stages	RNN - CDAE	Masking - Mapping	2	2019

Table 2.4 Best performing speech enhancement DNNs in years 2020 to 2021

Reference	Representation	Network	Target	Stages	Year
Koizumi et al. (Koizumi et al., 2020)	Time-Frequency	CRN	Masking	1	2020
Xia et al. (Xia et al., 2020*b*)	Time-Frequency	RNN	Masking	1	2020
Pandey and Wang (Pandey and Wang, 2020*a*)	Time-domain raw waveform	CDAE	Mapping	1	2020
Defossez et al. (Défossez et al., 2020)	Time-domain raw waveform	CDAE	Mapping	1	2020
Phan et al. (Phan et al., 2020)	Time-domain raw waveform in both stages	GAN - GAN	Mapping - Mapping	2	2020
Wang et al. (Wang, Wang and Wang, 2020)	Time-Frequency in both stages	CDAE - CDAE	Mapping - Mapping	2	2020
Nair and Koishida (Nair and Koishida, 2021)	Time-domain raw waveform - Time-Frequency	CDAE - CDAE	Mapping - Mapping	2	2021

2.5 Speech Enhancement Applications

This section reviews two main speech enhancement applications: ASR and hearing aids. Each application will be discussed in a separate subsection, and this discussion will include how speech enhancement is used for these applications, the developed ideas in the literature, and the current research issues related to these speech enhancement applications.

2.5.1 Speech Enhancement for Automatic Speech Recognition (ASR)

ASR is a technique used to translate human speech into text, and it can be applied to many applications, including conversational interactive voice response, dictations, air traffic control, automated car environment, and biomedical applications (Raut and Deoghare, 2016). Speech enhancement is an important stage for ASR, as system performance is expected to be improved by removing noise from the speech signal (Blanchard

et al., 2015; Gong, 1995). However, it has been found that the performance of an ASR system, in some cases, is not enhanced when adding a denoising stage (Narayanan and Wang, 2014; Du et al., 2014). The reason for this has motivated recent research to focus on evaluating speech enhancement techniques when integrated into a whole system as a preprocessing stage (Moore et al., 2017; Iwamoto et al., 2022; Donahue et al., 2018). The results of the evaluations performed in most research mainly refer to the speech distortion issue caused by the denoising process. A main drawback of speech enhancement is the addition of artifacts by the speech enhancement technique after processing the noisy speech signal (Iwamoto et al., 2022), leading to speech distortion (Wang et al., 2019). This distortion makes changes to the speech characteristics, negatively affecting the ability of the ASR system to interpret the speech signal, leading to higher WERs (Wang et al., 2019; Heymann et al., 2016)

The first solution proposed to this distortion issue is that recent DNNs for speech enhancement are giving more attention to speech distortion by using one of two approaches when implementing a DNN to be applied as an independent preprocessing stage to ASR. The first approach is to design a single enhancement stage DNN, in which distortion is kept as minimum as possible during the training process. This is achieved by manipulating the loss function to consider speech distortion (Xia et al., 2020*b*) or applying a feedback system including the ASR model, to ensure that the processed speech reduces the WER of ASR (Shen et al., 2019). The second approach is to develop a two-stage DNN for speech enhancement, in order to apply speech denoising and reconstruction using two separate stages, leading to lower speech distortion (Strake et al., 2019; Tang et al., 2021).

Another solution to this mismatch issue between the speech enhancement ASR models is joint training of the two models, where the ASR model is trained using the processed speech by the speech enhancement model (Wang et al., 2019; Wang and Wang, 2016). However, the clear disadvantage of this solution is the need to retrain a running ASR system to add the speech enhancement model and when applying any changes to the speech enhancement network, which is not practical. In chapter 6, a solution to this mismatch issue will be presented by developing a deep learning speech enhancement architecture for ASR.

2.5.2 Speech Enhancement for Hearing Aids

Hearing is a complex process performed by the auditory system, which is divided into two subsystems: the peripheral and central auditory systems (Bronzino, 2000). The peripheral auditory system can be divided into three parts: the outer ear, the middle, and the inner ear. The outer ear localizes and collects sound waves, which propagate in the air in the form of mechanical vibrations. These sound waves first pass through

the auditory canal of the outer ear, which amplifies a range of frequencies that can be heard by the human ear. Moreover, the auditory canal filters out any tiny substances and deals with changes in the temperature, to protect the tympanic membrane, known also as the ear drum, which is the first component in the middle ear (Shaw, 1974). The middle ear is a cavity filled with air, and it acts as a leverage system that amplifies the incoming sound waves by a factor of over 30 decibels. It consists of three bones: the malleus, the incus, and the stapes, which vibrate by the varying the air pressure of the input sound waves (Shaw and Stinson, 1983). The inner ear then finally receives the sound waves through a membrane to reach the cochlea, which is the main part of the inner ear, responsible for converting the sound waves from mechanical vibrations into electrical signals (Gan et al., 2007). These electrical signals are then transmitted to the brain through the central auditory system through nerves for sound interpretation (Bronzino, 2000).

The reduced ability to hear sounds is known as hearing loss, and it is one of the most common sensory deficits affecting human beings, especially the elderly. Hearing loss can be categorized into three types: Conductive hearing loss, Sensorineural hearing loss, and Mixed hearing loss (Neumann and Stephens, 2011).

Conductive hearing loss is a hearing impairment that occurs when the sound can not successfully reach the inner ear, so it is a malfunction of the outer or the middle ear (Hartley and Moore, 2003). This hearing loss type is caused by a deficiency in the outer ear, most commonly due to a blockage in the auditory canal; in the ear drum, for example when suffering from a hole or a perforation in the eardrum (Mehta et al., 2006); or in the middle ear, which happens when having a stiffness of the ear bones or a poor connection between any of the three bones (Legouix and Tarab, 1959). Sensorineural hearing loss is a hearing loss caused by a deficiency in the inner ear. This is mainly caused due to damage to or the reduction of the cells in the inner ear that converts the sound waves into electrical signals (Schreiber et al., 2010). The common causes of Sensorineural hearing loss are aging (Johnsson and Hawkins Jr, 1972) and high level noise exposure (Nelson et al., 2005). Finally, Mixed hearing loss is defined as damage in the outer or middle ear and in the inner ear, which means having a combination of Conductive and Sensorineural hearing loss (Zwartenkot et al., 2014).

Hearing loss can also be categorized by the degree of the hearing loss (Nadol Jr, 1993; Alshuaib et al., 2015). There are five hearing loss degrees (HL1-HL5) listed below:

- HL1: Mild hearing loss, in which the person has difficulty hearing soft sounds in noisy environments.
- HL2: Moderate hearing loss, in which the person has difficulty hearing conversational speech, especially in noisy environments.

- HL3: Moderately Severe hearing loss, in which the person can hear only a raised voice in a quiet environment, and has difficulty hearing it in a noisy environment.
- HL4: Severe hearing loss, in which the person has difficulty hearing a raised voice in a quiet environment, and extreme difficulty hearing it in a noisy environment.
- HL5: Profound hearing loss, in which the person has extreme difficulty hearing a raised voice in both quiet and noisy environments.

People suffering from any of these hearing loss types and degrees usually are advised to use a hearing aid, which is an electronic device that processes the sounds to make them audible to the hearing-impaired person (Kim and Barrs, 2006). Hearing aids are available in many styles, the most common two styles are the Behind The Ear (BTE) and In The Ear (ITE) hearing aids. BTE hearing aids are more comfortable to wear, usually suggested for young children, while ITE hearing aids are much smaller, fit snugly in the ear, and can hardly be seen; so they are usually used by older children and adults (Meredith and Stephens, 1993; Brooks, 1994). There are two technologies for hearing aids: the old analog hearing aids, which only amplify the sounds and send them to the human ear; and the recent digital hearing aids, which perform several signal processing techniques on the collected sounds before conveying them to the human ear (Levitt, 2007).

The main difference between analog and digital hearing aids is the microchip in the digital hearing aids, which is responsible for sound processing. The processing applied to the sounds are mainly divided into two techniques: analog to digital conversion and noise reduction, hence speech enhancement (Hamacher et al., 2008). Digital hearing aids are essential for people suffering from a severe degree of hearing loss, especially for the Sensorineural hearing loss type, in which the human ear has less ability to distinguish between desired and undesired sounds, leading to significant hearing difficulty in noisy environments (Johnson et al., 2016).

Speech enhancement is applied in digital hearing aids in order to improve speech intelligibility and quality, which is essential for people with hearing disabilities (Loizou, 2013). Based on the fact that deep learning techniques have made a breakthrough in eliminating background noise, current hearing aids utilize DNNs to perform speech enhancement for hearing aids (Wang, 2017; Nossier et al., 2019; Schröter et al., 2020).

Currently developed DNNs for speech enhancement remove background noise regardless of its type. Consequently, a hearing-impaired person has to rely on an external alert system to ensure their safety in emergency conditions. These systems detect the emergency sound, such as fire alarms, and use flashing lights or vibrating elements to notify the user (Beritelli et al., 2006). With the spread of smart features in most electronic devices, techniques have been suggested for the development of smart hearing

aids, which has the smart feature of detecting and amplifying emergency noise (Nossier et al., 2019). However, further investigations are needed to this technique, in order to improve performance. Chapter 6 will fill this gap in the literature.

2.6 Conclusion

In this chapter, a review was presented for deep learning-based supervised speech enhancement. Different DNN architectures were demonstrated, highlighting the advantages and disadvantages of each architecture type. Afterwards, a discussion was presented for the evolution of speech enhancement DNN architectures, and the effect of different approaches in solving deep learning-based speech enhancement issues. The chapter ended with a review of two main speech enhancement applications: ASR and hearing aids, covering the developed speech enhancement techniques for these application and the current research work in this area. The discussions in this chapter identified the gaps in the literature that will be covered in this thesis. These gaps include speech distortion as the current DNN-based speech enhancement issue that most recent research in the field are trying to solve, the mismatch issue between speech enhancement and speech recognition systems, and the important emergency noise that must not be eliminated by the speech enhancement system, especially for applications such as hearing aids. In the next chapter, the procedure of developing a DNN to perform speech enhancement will be demonstrated, covering the common techniques used to prepare the training data and the well-known speech quality evaluation metrics used to assess the performance of the DNN.

CHAPTER 3

Methodology on Developing a Deep Neural Network for Speech Enhancement

3.1 Introduction

This chapter presents all the necessary procedures to develop a DNN for speech enhancement. These procedures can be categorized into six steps: Data Collection, Data Preprocessing, Feature Extraction, Artificial Neural Network Implementation, Training Target Choice, and Evaluation of the Processed Speech. The following sections explain these six steps in details.

3.2 Data Collection

In deep learning-based speech enhancement, the data used as an input to the DNN plays an important role in the learning process, due to the fact that deep learning is a data driven approach. In order to train a DNN for speech enhancement, the network must be fed with a huge amount of data containing pairs of noisy and clean speech utterances. There are many online available datasets that can be used in the learning process, Tables 3.1, 3.2, and 3.3 show the clean speech, noise, and noisy speech datasets, respectively, that are commonly used in the literature. In order to generate the pairs of noisy and clean speech utterances for training, clean speech and noise audio files from the datasets in Tables 3.1 and 3.2 are randomly mixed to synthesize noisy speech. While in testing, real noisy datasets in Table 3.3 are usually used to evaluate the performance of the DNN in real noisy conditions.

When mixing speech and noise data to create a simulated noisy data for training, two different types of simulation can be used: instantaneous addition and room acoustic simulation. Instantaneous addition is used to generate additive background noise, which is the main noise type that speech enhancement aims to eliminate; while room acoustic simulation is used to create reverberate speech, which is another noise type that can be eliminated using speech enhancement techniques or a technique called dereverberation.

The dataset used for training should also consider different microphone conditions, such as close-talk microphones and distant microphones, which affect the intensity of

the speech and background noise. In close-talk microphone condition, the sound is loud and clear; while distant microphone sound is faint and more affected by background noise. These two conditions can also be simulated using near-field and far-field speech data.

Finally, the sound recorded by speakers can be collected from one microphone or many microphones, which divides speech enhancement field into two categories: single channel and multiple channel speech enhancement. As most of the data are recorded on multiple channels, single channel speech enhancement is based on taking the average of the sounds coming from different channels. It should be noted here that the work done in this thesis contributes to the single channel speech enhancement field.

Table 3.1 A review of the available clean speech datasets

Corpus	Description
TIMIT	English speech recording for 630 speakers, 10 sentences for each speaker, sampled at 16 kHz (Zue et al., 1990)
Voice Bank	English speech recording for 500 speakers, 400 sentences for each speaker, sampled at 48 kHz (Veaux et al., 2013)
LibriSpeech	1,000 hours of read English speech, sampled at 16 kHz (Panayotov et al., 2015)
ATR	16 hours of English speech, sampled at 48 kHz (Ni et al., 2007)
TED-LIUM	118 hours of English speech recorded from TED talks, sampled at 16 kHz (Rousseau et al., 2012)
WSJCAM0	140 speakers each speaking about 110 British English utterances, all sampled at 16 kHz (Robinson et al., 1995)
Free ST	350 English utterances for 10 speakers, sampled at 16 kHz (Surfingtech, 2015)
176 Spoken Languages	12,320 different Speech Files, each containing approximately 10 seconds of speech recorded in 1 of the 176 Possible Languages Spoken, sampled at 16 kHz (Topcoder, 2017)
DNS	A total of 562 hours of clean English read speech utterances for 11,350 speakers (Xia et al., 2020*a*)
Lombard GRID	A total of 5,400 utterances, 2,700 utterances with the Lombard effect and 2,700 plain, clean speech utterances, spoken by 54 native speakers of British English (Alghamdi et al., 2018)

Table 3.2 A review of the available noise datasets

Corpus	Description
NOISEX-92	Recording of various noises, including babble, factory, HF channel, pink, white, and military noise (Varga and Steeneken, 1993)
UrbanSound8K	8,732 recordings of 10 urban noises, including air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music (Salamon et al., 2014)
Baby Cry	400 different recordings of different baby cry sounds (Gveres, 2015)
Demand	A collection of multi-channel recordings of acoustic noise in diverse environments, including park, office, cafe, and street (Thiemann et al., 2013)
ESC 50	A collection of 2,000 recordings for 50 environmental noises, 40 for each, including: animal, nature, and urban sounds (Piczak, 2015)
CHiME3	4 noise environments including, cafes, street junctions, public transport (buses) and pedestrian areas (Barker et al., 2015)
USTC	15 home noise types, including AWGN, babble, car, and musical instrument sounds (Xu, 2013)
100 Noise	100 non speech environmental sounds, including wind, bell, cough, yawn and crowd noise (Hu, 2014)
DNS	181 hours of noise for about 150 audio classes, for a total of 60,000 audio clips (Xia et al., 2020*a*)

Table 3.3 A review of the available noisy speech datasets

Corpus	Description
AMI	100 hours of real meeting recordings in three different rooms with different acoustic properties. These recordings include close-talking and far-field microphones, individual and room-view video cameras (Carletta et al., 2005)
Reverberant	An artificial reverberant speech version of the Voice Bank clean speech corpus (Valentini-Botinhao et al., 2017*a*)
Voice bank	Noisy version of the Voice Bank clean speech corpus, created by artificially adding real noise to the speech (Veaux et al., 2013)

3.3 Data Preprocessing

Before feeding the data to the DNN, some preprocessing techniques must be applied in order to prepare the data for the training process. These operations are essential for any DNN type, as they ensure that the input data is in a suitable form for the training process; moreover, they facilitate the learning of the mapping function that maps noisy speech to clean speech. The following subsections will discuss these techniques in detail.

3.3.1 Speech and Noise Mixing

There are three main noise types to deal within the speech processing field: Random noise, Interference noise, and Reverberation noise. These noises have different characteristics, and they affect the target speech signal in different ways (Loizou, 2013).

Random noise is the common form of noise generated by sounds from the surrounding environment, such as nature sounds, animal sounds, urban noise, etc. This noise type is characterized by having an intensity spectrum similar to that of the speech signal.

Interference noise is the collection of speech signals that interfere with the target speech signal. Babble noise generated by the crowd is an example of this noise category, and it is one of the most challenging noise environments, as it is similar to the target speech signal, which makes it different for the network to detect and suppress this noise type (Haykin and Chen, 2005). The presence of more than one speaker at the same time

is another example of interference noise.

Reverberation noise is generated by the reflected speech signal from the walls, ceiling, floor, tables, or any other hard surfaces when speaking inside a room (Loizou, 2013).

Speech enhancement research only considered random noise and interference noise caused by babble noise. Handling inference noise caused by more than one speaker is performed using a technique called Speaker Separation, which is different from speech enhancement (Wang and Chen, 2018). On the other hand, dereverberation is a separate technique from speech enhancement, used to eliminate room reverberation noise (Zhao, Wang and Wang, 2018). Therefore, in order to create the noisy speech for speech enhancement processing, random or babble noise environments are additively mixed with the speech signal. This can be expressed by Equation 3.1:

$$y[k] = s[k] + n[k], \tag{3.1}$$

where, *y[k]* is the noisy speech, while *s[k]* and *n[k]* represent the speech and noise signals, respectively, and *k* is the time index.

3.3.2 Amplitude Scaling and Normalization

In real time, the intensity of the background noise varies, sometimes its level is lower than the speech signal; while in other situations the noise is highly intrusive and of a much higher level than the speech signal, where you can barely hear the spoken speech. The metric that measures the ratio between the power of the desired speech and background noise is known as SNR, and it can be defined by Equation 3.2:

$$SNR(dB) = 20\log_{10}(\frac{s_{RMS}}{n_{RMS}}), \tag{3.2}$$

where, *SNR* is in dB, s_{RMS} and n_{RMS} are the speech and noise Root Mean Square (RMS) levels, respectively, which can be calculated as in Equation 3.3 below:

$$x_{RMS} = \sqrt{\frac{1}{K}\sum_{k=0}^{K-1} x[k]^2}, \tag{3.3}$$

where, x_{RMS} is the RMS value of the signal, *K* is the signal length, and k is the time index.

The SNR of the input audio to the DNN must be adjusted to match real time situations, so a wide range of SNRs are generally used during the training process, to ensure that the network can deal with different noise levels.

Normalization is another important process in deep learning, in which the data is

transformed to a common scale to facilitate the training process. In speech enhancement, the noisy and the clean utterances are usually normalized to zero mean and unit variance; otherwise, normalization can be performed using hidden layers during the training process (Garbin et al., 2020).

3.3.3 Audio Resampling

In speech enhancement, the target output is speech, where the essential frequency bands fall between 200 to 6 kHz; while our ear has the ability to hear sounds in the frequency range of 20 to 20k Hz. The higher frequency bands usually contain noise or unimportant speech features; for this reason, limiting the frequency band of the input audio to the DNN can help the network in the denoising process, as this will only keep the relevant frequency bands for speech; this process is known as Audio Resampling, or more specifically, Audio Downsampling.

In the literature, the use of 16 kHz sampling frequency was found to be suitable for generating speech with good quality. However, it is preferable in some cases to downsample the noisy speech to 8 kHz, which is used for some applications in less challenging noise environments, such as in ASR, where the quality of the output speech is not of high importance as long as the ASR system can interpret the speech and convert it to text.

3.4 Feature Extraction

Although deep learning is a data driven approach, feature extraction was proven to significantly help with the learning process of the DNN (Fu et al., 2017). In speech enhancement, the learning process can be performed in the time or the frequency domain, so speech features can be categorized into time and frequency domain features (Alías et al., 2016; van Hengel and Krijnders, 2013). The following subsections include a discussion on the most common speech features in both domains.

3.4.1 Time Domain Features

Speech is originally represented in the time domain in the form of changes in the pressure values of the input sound with time. This is known as Raw Waveform features, where no feature extraction is applied and these values are fed directly to the DNN. There are some speech enhancement research based on Raw Waveform features (Fu et al., 2017; Pascual et al., 2017), the researchers here believe that it is better to leave the DNN to decide the most useful features during the training process, so as not to discard some features that may negatively impact the learning process.

Time frames are the second most commonly used feature for speech enhancement in the time domain. In this process, the time domain raw waveform is cut into several small

parts, known as frames, using a technique called windowing. This is performed due to the fact that feeding the network with the whole utterance consumes a large amount of memory, and the efficiency of the network to process the input data also decreases with a large input size. Windowing is simply multiplying the signal with a window function so as to divide it into many small time periods; these time periods should be as small as 5 to 100 ms, for the frame to correctly represent important signal characteristics. There are many types of windowing functions; the rectangular window is the simplest function; however, as it ends abruptly, this sharp edge of the window will lead to the appearance of frequencies that are not in the original signal, which cause the spectrum to be smeared; a problem known as spectral leakage. Instead, there are another two popular windowing functions, the Hann and Hamming windows, that could be used to overcome this problem, as they ensure that the ends of the signal are close to zero (Liang and Lauterbur, 1999).

There are two other well-known features that can be extracted from the time domain utterances: Zero Crossing Rate (ZCR) and Energy Entropy. ZCR is the rate of sign-changes of the signal during each time frame; while Energy Entropy is a measure to the abrupt changes in the energy level of an audio signal. Both of these features extract important information about the input time domain audio; however, their use is not very common in the speech enhancement field, because they only provide very specific and limited information about the speech signal (Alías et al., 2016).

3.4.2 Frequency Domain Features

The time domain speech signal can be converted to the frequency domain, in order to get more meaningful information about the speech signal. This can be achieved using a technique known as Fourier Transform, which gives the representation of the input audio samples in the frequency domain. In speech enhancement, it is common to use the Short Time Fourier Transform (STFT) technique to convert the time domain signal to the frequency domain, which is the Fourier transform of a windowed signal as it changes over time, and this generates a T-F representation of the signal (Xu et al., 2014*b*; Xia et al., 2020*b*). Equation 3.4 defines the STFT operation:

$$Y(t,f) = \sum_{f=0}^{F-1} y(m+t)h(m)e^{-j2\pi fm/F}, \tag{3.4}$$

where *Y(t,f)* is the STFT of the noisy signal, f is the frequency bin index; $\{f = 0, 1, ..., F-1\}$ and F is the total number of frequency bins, t is the time frame, $\{t = 0, 1, ..., T-1\}$ and T is the total number of frames, m is the input signal time sample, h denotes the applied window function.

Chromagram is a logarithmic STFT-based feature that represents the spectrum of

the audio mapped into one octave, to take into consideration the 12 pitch classes within an octave (Bartsch and Wakefield, 2005), defined in musical theory (Shepard, 1964). Based on the fact that this feature is suitable to represent musical and harmonic signals, it is also useful in speech enhancement, because it helps the DNN in differentiating speech from noise (Kumar et al., 2022), or dealing with singing voices and emotional speech (Issa et al., 2020).

Mel Frequency Cepstral Coefficients (MFCC) is a frequency domain-based feature that represents the signal in the Cepstral domain, which is achieved by applying a second stage inverse Fourier transform, more specifically the Discrete Cosine Transform (DCT), operation on the logarithmic of the magnitude of the Mel frequency spectrum. The conversion of the Fast Fourier Transform (FFT) frequencies to the Mel scale aims to achieve frequencies that follow a scale that resembles real human voice signals. The MFCC feature extraction procedure is represented in Figure 3.1. MFCC was proven to be very useful in speech processing, because it was proven to be robust to noisy signals; consequently, it is used extensively in speech enhancement (Li et al., 2020; Wang, Li, Siniscalchi and Lee, 2020).

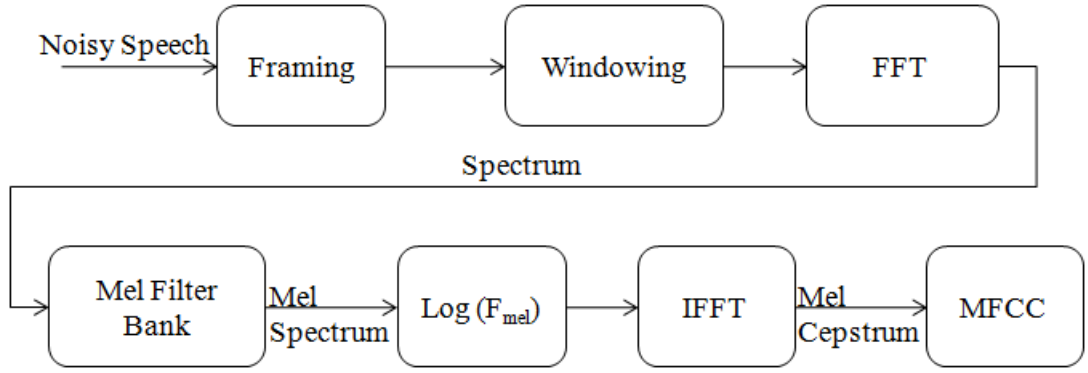


Figure 3.1 MFCC feature extraction process, IFFT refers to the Inverse Fast Fourier Transform

Another feature extraction method that represents the signal in the T-F domain is the Wavelet Transform. A Wavelet is a mathematical procedure that also analyses the signal time and frequency components, but unlike STFT, the audio here can be divided into intervals of varied sized using a filter bank that decomposes the audio into sub-bands over different regions of the frequency spectrum, without losing the time domain characterization. This allows for more precise feature extraction of high and low frequency components; moreover, it is more efficient dealing with signals that have discontinuities and sharp peaks (Mallat, 1989). Wavelet-based features were also shown to be effective in improving performance when applied in speech enhancement (Gutiérrez-Muñoz and Coto-Jiménez, 2022; Vanithalakshmi et al., 2022).

Linear Prediction Coefficients (LPC) are a widely used frequency feature in speech processing, derived from linear prediction analysis of the speech signal that accurately

represents the speech signal using few parameters (Sambur and Jayant, 1976). This feature imitates the human vocal tract as it captures the spectral envelope of the speech by identifying the vocal resonances (Atal and Hanauer, 1971), which is difficult to be determined using the FFT spectrum due to its highly varying harmonic structure. This procedure is based on dividing the audio into segments using a framing process, and then analysing the audio segments to determine voiced and unvoiced parts, the pitch of the segment, and some other speech features, in order to finally create a filter that models the vocal tract for each segment. This algorithm takes into consideration resonances while performing the analysis, resulting in a smooth spectrum with well-defined peaks corresponding to the resonances (Atal, 2003). Many speech enhancement research shows the advantage of using LPC in improving the performance (Schröter et al., 2022; Roy et al., 2021).

Another similar feature to LPC is Perceptual Linear Prediction (PLP), which is also based on linear prediction analysis of the speech signal. This feature is based on the psychophysics of human hearing, and it has an advantage over LPC as it tries to better resemble the human perception of voice by discarding any irrelevant information (Hermansky, 1990). PLP was shown to outperform LPC in speech processing (Alías et al., 2016; Mishra et al., 2010), and it is also used in speech enhancement research (Saleem et al., 2019).

Another well-known frequency based feature is the Cochleagram, which is generated by time windowing responses of a filterbank representing the frequency analysis of the cochlea. The noisy audio passes through a number of Gammatone filters to extract speech features related to different frequencies, resulting in a T-F representation to the noisy audio similar to the spectrogram. Equation 3.5 represents the impulse response of the gammatone filter in the time domain:

$$g(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi f_c t + \phi), \tag{3.5}$$

where the constant a is the amplitude that controls the gain, t is the time, n is the order of the filter, f_c is the central frequency of the filter, and ϕ is the phase. Equation 3.6 defines b, which is the decay factor determining the filter bandwidth:

$$b = 1.019 * 24.7(4.73\frac{f_c}{1000} + 1). \tag{3.6}$$

The gammatone filterbank is created by changing the center frequency f_c of the filter in the above equations. There are many features that can be extracted based on the Gammatone Frequency (GF) feature, such as Gammatone Frequency Cepstral Coefficients (GFCC) (Shao and Wang, 2008), which is calculated by applying the DCT to the GF feature. Gammatone Frequency Modulation Coefficients (GFMC) (Maganti and Matas-

soni, 2010) is another feature based on GFCC. Multiresolution Cochleagram (MRCG) (Chen et al., 2014) and Pitch-Based Feature (PITCH) (Wang and Chen, 2018) are other features that can be created based on the cochleagram. Some features are also found to integrate both cochleagram and spectrogram, such as Gabor Filterbank Feature (GFB) (Schädler et al., 2012) and Power-Normalized Cepstral Coefficients (PNCC) (Kim and Stern, 2016), to take advantage of both representations.

3.5 Artificial Neural Network Implementation

The implementation of DNN is the most important stage when developing a deep learning technique for speech enhancement. Due to the fact that this process is very complex and has many internal techniques, it can be divided into two major main processes: DNN Design and Loss Function Choice. This chapter briefly discusses DNN design, as a more detailed discussion on this process will be presented in Chapter 4.

3.5.1 DNN Design

Any of the DNNs discussed in Chapter 2 can be employed for speech enhancement. The choice of the network mainly depends on the required performance and complexity, which is defined by the application where speech enhancement is performed. Implementation of DNNs can be done using open-source frameworks, such as Keras and PyTorch (Vasilev et al., 2019). These frameworks facilitate the development of deep learning models by having built-in optimized functions to construct the network and perform training and testing. The performance of these frameworks is nearly the same; however, some frameworks were shown to be more stable when applied commercially with real data. Moreover, some frameworks offer more features and functions that help in making the training process faster and more efficient, and for this reason they are preferable than other frameworks (Dinghofer and Hartung, 2020).

The implementation of a DNN requires the adjustment of many hyperparameters that differ based on the type of the architecture. These hyperparameters highly affect the performance of DNNs, so careful consideration should be taken when choosing their values. Due to the importance and large number of these hyperparameters, a full discussion and analysis of this process will be given in the next chapter, Chapter 4.

3.5.2 Loss Function Choice

The definition of the loss function is the second important step when implementing a DNN for speech enhancement. This function defines the error that the DNN should minimize during the learning process, so it has a great impact on the final performance. MMSE is the most commonly used and default loss function for speech enhancement.

This function is based on comparing the real and estimated clean speech signal, and it can be defined in the time domain as below:

$$L_{MMSE} = \frac{1}{T} \sum_{t=0}^{T} \left[\hat{x}_2(t) - x(t) \right]^2 , \tag{3.7}$$

where L_{MMSE} is the MMSE loss function, *x(t)* and $\hat{x}(t)$ is the real and estimated clean speech, respectively. *T* is the total number of time frames, and *t* is the time index.

MMSE is the default loss function for deep learning-based speech enhancement; however, other loss functions were found to be useful for specific applications when the target is to improve a specific speech quality metric, these loss functions include: PESQ, STOI, and Scale-Invariant Signal to Distortion Ratio (SI-SDR) (Fu et al., 2019). Each of these speech quality scores is defined to measure a specific aspect of the speech quality. PESQ gives an overall quality score for the processed speech, but it highly correlates with the amount of background noise; while, STOI estimates the intelligibility of the output speech from the DNN. SI-SDR is a measure for the amount of distortion caused by the denoising process; further details about these evaluation metrics will be given in Section 3.7. When using these evaluation metrics as a loss function, the DNN is expected to generate speech that maximizes the evaluation metric used, which will be beneficial for some applications. However, the loss function and the DNN should be carefully designed in this case, in order to avoid vanishing or exploding gradient issues (Kolbæk et al., 2020).

3.6 Training Target Choice

The training target is defined as the signal that the DNN learns to generate during the training process. The deep learning-based supervised speech enhancement procedure can be seen from two perspectives: a regression or classification problem (Wang and Chen, 2018). When dealing with speech enhancement as a regression task, the network is trying to map the input noisy speech audio to clean speech. This includes the raw waveform, time frames, and T-F representation mapping, depending on the features used during the training process. Alternatively, the speech enhancement task can be processed by the DNN as a classification problem, where the network tries to generate a mask that when multiplied by the input noisy speech, outputs the clean speech. This mask works on the T-F representation of the noisy audio, and it classifies every portion of the T-F diagram as either speech or noise. It should be mentioned here that the target output is still the clean speech signal; however, the classification description is based on the fact that the network performs binary classification on every portion of the T-F representation. Consequently, training targets for supervised speech enhancement can be categorized as: mapping and masking targets (Wang et al., 2014). In the following

subsections, a discussion will be presented for these two training target types.

3.6.1 Mapping Targets

As mentioned previously, the mapping target is based on the learning domain of the DNN, as raw waveform or time frames features represent the mapping target in the time domain; while in the frequency, a spectrogram or cochleagram of the input signal can be used as a training target.

In spectrogram mapping, the network performs some processing on the noisy speech spectrogram, in order to finally predict the clean speech spectrogram. In most studies, only the magnitude spectrogram is used during the training process, while the noisy phase is retained, to be added to the output estimated clean speech spectrogram, assuming that the phase is less affected by noise in comparison to the magnitude spectrum (Xu et al., 2014*b*; Braun and Tashev, 2020). However, other studies show the importance of enhancing the phase spectrogram, so some researchers develop a DNN to perform complex spectrogram mapping, where both the magnitude and phase spectrogram are enhanced during the training process. This can be achieved as a single stage speech enhancement processing (Ouyang et al., 2019), or a two-stage based processing to achieve better performance (Wang, Wang and Wang, 2020). Phase enhancement can also be performed using separate techniques that work on retrieving the clean phase, to avoid increasing complexity that may result from developing a DNN to enhance the complex spectrogram (Zhao, Wang and Wang, 2018). As discussed in Section 3.4, some other features can be extracted from the audio spectrograms, such as MFCCs. When using these features as an input to the DNN, the network will try to map the noisy speech spectrogram-based features to the corresponding clean speech features.

In cochleagram mapping, the DNN performs speech enhancement processing on the noisy speech cochleagram to estimate the clean speech cochleagram, also known as Gammatone Frequency Target Power Spectrum (GF-TPS). However, this training target is less common than spectrogram mapping, due to the absence of the inverse procedure that converts this T-F representation back to the time domain to reconstruct the estimated clean speech audio. In order to reconstruct the time domain estimated clean speech signal from the cochleagram representation, an indirect method is used that is taken from the Computational Auditory Scene Analysis (CASA) field, dating from 1983 (Lyon, 1983). In CASA, sound source separation was performed by segmenting the cochleagram into regions belonging to each sound source. A binary matrix of 1 or 0 weights for different sound sources is then formed by grouping these regions into streams. This generated binary matrix is then multiplied by the audio containing many sounds, to finally output the target sound source, defined by the 1s weights in the binary matrix (Weintraub, 1985). Using the same idea, this binary matrix can be calculated

using the noisy and the estimated clean speech cochleagram; afterwards, this matrix can be used to reconstruct the time domain estimated clean speech signal, by weighting any T-F representation of the noisy speech with it. This matrix of weights is actually the spectrographic or T-F mask, which will be discussed in the following subsection, so when using cochleagram-based mapping targets, speech re-synthesis is done indirectly through a spectrographic mask.

3.6.2 Masking Targets

As mentioned above, masking targets are used to predict the clean speech T-F representation by multiplying the noisy speech T-F signal with a matrix (mask) that suppresses the noise regions in the noisy speech signal. When using T-F masking as a training target, the DNN deals with speech enhancement as a binary classification task, where the network learns to classify regions in the T-F representation as one of two classes: speech or noise (Wang, 2017; Chakrabarty et al., 2018; Williamson et al., 2016).

T-F masks can be categorized into two basic types: Binary Masking and Soft Masking (Samui et al., 2019). Binary Masking assumes sparseness and disjointness of the speech and noise. Sparseness means that most of the T-F bins have low energy, while disjointness means that the T-F bins of the two signals in the audio mixture do not overlap (Alberti and Ammari, 2017). Based on these assumptions, frequency bins that are likely to belong to the target signal are set to 1, while other frequency bins are set to 0, and that's why it is called binary masking. If the previously mentioned assumptions are not fulfilled in the noisy speech signal, soft masking can be used instead, which avoids the hard binary decision (1 or 0) of the binary mask, by setting each frequency bin of the noisy speech signal to a probability value between 0 and 1, based on how much it is likely to belong to the target clean speech signal.

In the following subsections, discussion and illustration will be given to the commonly used T-F masking targets in the speech enhancement field,

3.6.2.1 Ideal Binary Mask (IBM)

Ideal Binary Mask (IBM) belongs to the binary masks category, and it is considered one of the first masking targets used in the field of speech enhancement and separation. This mask generates a binary matrix with 0 values assigned to indices corresponding to portions of the spectrogram that have a high noise intensity, while 1 values are assigned to portions with higher speech amplitude (Wang, 2005). Equation 3.8 defines the IBM:

$$IBM(t,f) = \begin{cases} 1, & if\ SNR(t,f) > LC \\ 0, & otherwise \end{cases} , \tag{3.8}$$

where *t* and *f* denote time and frequency, respectively. *LC* is the local criterion or threshold that the classification to 1 or 0 is based on. This value should be chosen based on practical trials; but in the literature, it is kept to 5 dB lower than the SNR of the noisy speech mixture so as to preserve enough speech information (Wang et al., 2014).

Target Binary Mask (TBM) is an edited version of IBM, and it has been also employed in many speech enhancement research. This mask is defined using the target speech energy in each T-F unit and the average spectral energy of a reference Speech-Shaped Noise (SSN) instead of the local noise energy, defined in Equation 3.8, which means that this mask type is independent of the noise in the mixture (Kjems et al., 2009).

3.6.2.2 Ideal Ratio Mask (IRM)

Ideal Ratio Mask (IRM) belongs to the soft masking target category, which is used when the binary masking assumptions, discussed previously, are not fulfilled. IRM was proven to outperform IBM in speech enhancement research, because it outputs speech with better intelligibility (Srinivasan et al., 2006). An explanation to this is that the noise speech mixture is very complex, which makes it hard to define this noisy speech signal using the assumptions of the IBM. The IRM is presented below in Equation 3.9:

$$IRM(t,f) = \left(\frac{S(t,f)^2}{S(t,f)^2 + N(t,f)^2}\right)^{\beta}, \tag{3.9}$$

where $S(t,f)^2$ and $N(t,f)^2$ denote the speech and noise energy, respectively, in a particular T-F unit. β is a tunable parameter to scale the mask.

3.6.2.3 Complex Ideal Ratio Mask (cIRM)

Complex Ideal Ratio Mask (cIRM) is another soft mask that was proposed after many speech enhancement research pointed to the importance of enhancing the noisy phase as well as the noisy magnitude spectrogram (Williamson et al., 2016). When using this complex spectrogram-based mask, the DNN learns to give an estimate of both the clean magnitude and phase spectrogram, which leads to better clean speech reconstruction. However, this negatively affects the networkś ability to eliminate background noise in comparison to using the normal IRM (Wang et al., 2016). The STFT in this masking target type is expressed in Cartesian coordinates so as to give a meaningful phase representation that can be used in the training process. Equation 3.10 defines the cIRM which when applied to the noisy complex spectrum, produces a clean complex spectrum:

$$cIRM = \frac{Y_r S_r + Y_i S_i}{Y_r^2 + Y_i^2} + i\frac{Y_r S_i - Y_i S_r}{Y_r^2 + Y_i^2}, \tag{3.10}$$

where Y_r and Y_i are the real and imaginary parts of the noisy speech, respectively, and S_r and S_i are the real and imaginary parts of the clean speech, respectively. In practice, cIRM is expressed in a compressed format in order to be bounded to ensure training stability, and the work in (Williamson et al., 2016) and (Wang et al., 2016) defined these compression techniques. Afterwards, the estimated compressed mask is decompressed and multiplied by the noisy spectrum to produce the clean complex spectrum.

3.6.2.4 Spectral Magnitude Mask (SMM)

Spectral Magnitude Mask (SMM) is a mask type that belongs to the soft masking category; however, it takes advantage of the mapping-based targets approach as well. This is achieved by dividing the STFT magnitude of the clean speech by the STFT magnitude of the noisy speech, to generate a matrix that when multiplied with the noisy speech signal, outputs an estimate to the clean speech. This mask is also known as a Fast Fourier Transform Mask (FFT-Mask), and it is not bounded by 0 and 1, such as cIRM, so high values that are present in the mask matrix are truncated, to guarantee the stability of the training process (Wang and Chen, 2018; Wang et al., 2014). The definition of SMM is expressed in Equation 3.11:

$$SMM(t,f) = \frac{|S(t,f)|}{|N(t,f)|}, \tag{3.11}$$

where $|S(t, f)|$ and $|N(t, f)|$ are the magnitude spectrum of the clean and noisy speech, respectively. There is another version of this type of mask named Phase-Sensitive Mask (PSM) (Erdogan et al., 2015), in which the SMM mask is multiplied by the cosine of the phase difference between the noisy and clean speech, and this was reported to have a positive impact on the overall performance (Wang and Chen, 2018).

3.7 Evaluation of the Processed Speech

The last step in developing a deep learning-based speech enhancement DNN is to evaluate the quality of the output speech from the network. This evaluation can be performed using real listeners or with the aid of computer algorithms. In other words, this evaluation can be divided into two types: subjective and objective evaluation. The following subsections discuss these two evaluation types, where the most common subjective method, Mean Opinion Score (MOS), will be presented; while five well-known objective evaluation metrics will be then demonstrated.

3.7.1 Mean Opinion Score (MOS)

MOS (Itu, 1996) is a subjective evaluation of the quality of the processed speech. This method is performed using real listeners who are asked to give a score to the output

speech that ranges between 1 to 5, where the higher the score the better the quality, this score is known as the MOS. The listeners are also asked to score the intrusiveness of the remaining background noise, this score also falls between 1 and 5. An illustration of the speech quality and noise intrusiveness scores is given in Tables 3.4 and 3.5, respectively.

Table 3.4 Speech signal scale for MOS evaluation

Rating	Description
5	Very natural, no degradation
4	Fairly natural, little degradation
3	Somewhat natural, somewhat degraded
2	Fairly unnatural, fairly degraded
1	Very unnatural, very degraded

Table 3.5 Noise intrusiveness scale for MOS evaluation

Rating	Description
5	Not noticeable
4	Somewhat noticeable
3	Noticeable but not intrusive
2	Fairly conspicuous, somewhat intrusive
1	Very conspicuous, very intrusive

The MOS is finally taken based on the average scores of all listeners. Careful consideration should be taken when performing the MOS evaluation, such as the environment where the test takes place and the device used to listen to the processed speech. This is to ensure that the scores given by the listeners are as accurate as possible. This judgment is often performed by experienced people from the speech analysis field; however, some research is based on conducting this evaluation in a survey-like manner using a large number of general listeners, to indicate their opinion on the quality of the speech processed by different speech enhancement techniques (Xu et al., 2014*b*).

3.7.2 Signal to Distortion Ratio Measures

There are many objective evaluations to measure the performance of DNNs for speech enhancement, and Signal to Distortion Ratio (SDR) is one of the classical objective measures for speech quality. As mentioned in Section 3.3.2, SNR is defined as the ratio between the power of the desired speech and background noise. The same definition of

SNR can also be used as an evaluation metric for the quality of the processed speech, where Equation 3.2 is redefined and the enhanced speech is used to calculate the background noise. In this case, the metric is considered as the ratio between the power of the real speech signal and the difference between the real and estimated clean speech signal, known as SDR (Févotte et al., 2005) and it is illustrated by Equation 3.12.

$$SDR(dB) = 10\log_{10}(\frac{\sum_{k=1}^{K} s^2(k)}{\sum_{k=1}^{K}\{s(k) - \hat{s}(k)\}^2}), \tag{3.12}$$

where, *s(k)* and $\hat{s}$*(k)* are the real and estimated clean speech signals, respectively. *K* is the total number of samples and *k* is the time index.

However, SDR is not always an accurate metric for assessing the quality of the processed speech due to the non-stationary nature of the speech signal, which fluctuates over time, so calculating the power over the entire speech signal is not suitable. Segmental Signal to Noise Ratio (SNR_{seg}) (Hansen and Pellom, 1998) is an edited version of the classical SNR that calculates the power of the speech signal over short frames, and then the average is taken. This is expressed in Equation 3.13.

$$SNR_{seg}(dB) = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}(\frac{\sum_{k=Lm}^{Lm+L-1} s^2(k)}{\sum_{k=Lm}^{Lm+L-1}\{s(k) - \hat{s}(k)\}^2}), \tag{3.13}$$

Another measure that was formulated based on the SNR is the SI-SDR (Le Roux et al., 2019). This metric differs from the common SDR as it is invariant to the scale of the processed signal, which is useful for speech enhancement techniques that result in improper scaling to the generated enhanced speech. The mathematical formula of SI-SDR is expressed in Equation 3.14:

$$SI - SDR(dB) = 10\log_{10}(\frac{\|\alpha s\|^2}{\|\alpha s - \hat{s}\|^2}), \tag{3.14}$$

where the scaling factor α guarantees that the SI-SDR is invariant to the scale of $\hat{s}$, and it is expressed as given below in Equation 3.15:

$$\alpha = \frac{\hat{s}^T s}{\|s\|^2} = argmin_\alpha\|\alpha s - \hat{s}\|^2, \tag{3.15}$$

SDR, SNR_{seg}, and SI-SDR have no range, but higher values of these evaluation metrics refer to better speech quality.

3.7.3 Log Spectral Distortion (LSD)

Log Spectral Distortion (LSD) (Du and Huo, 2008) is another objective speech quality evaluation metric that measures the distortion caused by speech enhancement process-

ing, and it is widely used in the research field (Xu et al., 2014*b*, 2015, 2013). The calculation of the LSD is given in Equation 3.16:

$$LSD = \frac{1}{T}\sum_{t=0}^{T-1}\left\{\frac{1}{F/2+1}\sum_{f=0}^{F/2}\left[10log\frac{P(X(f))}{P(\hat{X}(f))}\right]^2\right\}^{\frac{1}{2}}, \tag{3.16}$$

where, *P* is the clipped power spectrum such that the dynamic range of the log-spectrum is limited to about 50 dB. The function *P* for a signal z can be expressed as:

$$P(z(f)) = max\left[|z(f)|^2, 10^{-50/10}\right]. \tag{3.17}$$

LSD measures speech distortion based on the frequency domain representation using the power spectrum, and it has no range, but lower values indicate low distortion.

3.7.4 Perceptual Evaluation of Speech Quality (PESQ)

PESQ is one of the most commonly used objective speech quality metrics in the speech enhancement field, as it is considered as the most accurate measure available nowadays (Rix et al., 2001). PESQ is an international speech quality objective measure that was officially standardized by the International Union—Telecommunication Standardization Sector (ITU-T) in February 2001. It gives an estimate to the subjective MOS metric using the original clean speech signal and the predicted clean speech generated by the DNN (Kondo, 2012). The algorithm is based on two models: a perceptual model to generate a representation to the real and estimated clean speech, and a cognitive model that estimates the MOS value based on the difference between the real and estimated clean speech representations. An illustration of the PESQ algorithm is shown in Figure 3.2. PESQ score ranges from 0.5 to 4.5, where the higher the score the better the speech quality.

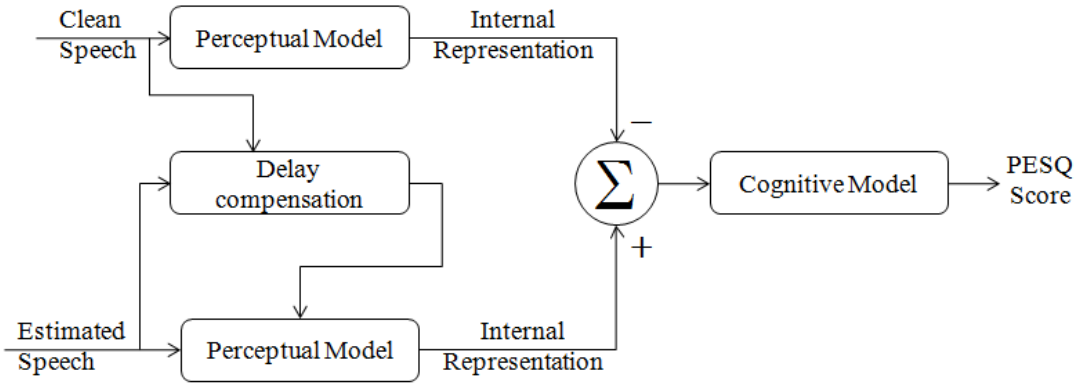


Figure 3.2 PESQ algorithm

3.7.5 Short-Time Objective Intelligibility (STOI)

STOI (Taal et al., 2011) is an objective measure for speech intelligibility, and it is the most commonly used speech intelligibility estimator in the speech enhancement field as it highly correlates with the actual speech intelligibility of the enhanced speech generated by the DNN (Healy et al., 2017). The STOI metric can be calculated using the approximation given in Equation 3.18, further details about the algorithm are demonstrated in (Taal et al., 2011).

$$d = \frac{1}{J(M-N+1)} \sum_{j=1}^{j} \sum_{m=N}^{M} \mathcal{L}(\underline{a}_{j,m}, \underline{\hat{a}}_{j,m}), \tag{3.18}$$

where *d* is the STOI estimator, a and $\hat{a}$ are the clean and enhanced short-time temporal envelope vectors, respectively; while, $\mathcal{L}$ represents the sample envelope linear correlation. *N* denotes the length of the temporal envelope, *J* is the number of one-third octave bands and *(M - N +1)* is the total number of short-time temporal envelope vectors. *M* represents the total number of time frames, and *m* and *j* are the indices for the time frames and octave bands, respectively. STOI is usually expressed as a percentage, and the higher the percentage the better the speech intelligibility.

3.7.6 The Composite MOS Estimator

A recent and widely used objective speech quality measure is the composite MOS estimator proposed in (Hu and Loizou, 2007*a*). This evaluation metric consists of three measures: Csig, Cbak, and Covl; where each score predicts the quality of the processed speech from a certain aspect, and they are obtained by combining different objective evaluation metrics that highly correlate with speech/noise distortions and the overall quality of the processed speech. Csig is a measure for signal distortion, Cbak is a measure for noise intrusiveness, while Covl measures the overall speech quality. Based on the fact that these metrics give an estimate for the MOS score, the value of each of them falls between 1 and 5, and high scores refer to better speech quality.

3.8 Conclusion

This chapter demonstrated the necessary procedures to be performed when implementing a DNN for supervised speech enhancement. A discussion was presented for the different manipulation and prepossessing techniques applied to the noisy speech before being processed by the DNN. Moreover, demonstration was given to different speech features that help the network in the learning process. We also covered different speech enhancement training targets that are essential for the supervised training procedure. Highlights were given for DNN implementation and training, as detailed illustration

with experiments will be presented in the next chapter, Chapter 4. Finally, the chapter ended with a discussion on the well-known subjective and objective speech quality evaluation metrics used in this work. In the next chapter, an experimental analysis will be given to different speech enhancement architectures. A comparison will be conducted to evaluate the performance of each network type using subjective and objective speech evaluation metrics. Moreover, the analysis will answer some questions, in order to fill gaps in the literature, such as revealing the effect of some factors on the performance, such as the training target type, the preprocessing techniques used, and the learning domain.

CHAPTER 4

An Experimental Analysis of Deep Learning Architectures for Speech Enhancement

4.1 Introduction

In comparison to other speech enhancement approaches, DNN based speech enhancement has made a breakthrough in the denoising process, and most of the proposed DNN architectures were proven to generate speech with much better quality and intelligibility. However, the implementation and setup of DNNs have been mainly empirical, due to the large number of factors affecting the learning process. In an attempt to facilitate the development of better DNNs for speech enhancement, this chapter presents a detailed experimental analysis of three well-established DNNs major categories for speech enhancement: MLP, CNN, and DAE. This analysis compares the performance of seven DNNs belonging to these three categories, by evaluating them in terms of the overall quality of the generated processed speech using five objective evaluation metrics and a subjective evaluation with 23 listeners. Moreover, this comparison covers the performance of each network in challenging noise environments; evaluating network generalization, complexity, and processing time. Afterwards, answers to some research questions have been covered, such as how to choose the learning domain and mapping target, and how network performance is impacted by changing network hyperparameters and the composition of the data, including the Lombard effect. The investigation carried out in this chapter depends on two different approaches. The first approach numerically shows and compares the results; while, the second approach interprets the results using spectrogram visualization, where the spectrograms are presented for the generated speech from all the investigated architectures. Additionally, interpretation is provided for the spectrograms of the hidden layers for CNN based models, to help in understanding how CNNs perform speech enhancement.

4.1.1 Research Contributions

The work in this chapter makes the following research contributions:

- provides a comprehensive comparison of seven different DNNs for speech en-

hancement,

- interprets the processing of DNNs to perform the denoising process,
- identifying factors that affect the performance of different DNNs for speech enhancement, and
- provides recommendations to improve the performance of DNNs for speech enhancement.

In the following sections, the general factors affecting the performance of DNNs for speech enhancement will be first presented; afterwards, implementations of the seven models will be discussed. Details about the experimental setup used to perform the analysis will be then provided. Finally, discussion and analysis of the obtained results will be given.

4.2 Factors Affecting the Learning of DNNs for Speech Enhancement

Training a DNN for speech enhancement is a complicated process that is affected by many factors. These factors can be divided into three categories: the model setup, data structure, and learning hyperparameters; this is summarized in Figure 4.1. The following subsections will discuss these three factors in details.

4.2.1 Model Setup

Before training a DNN model for speech enhancement, there are two main points to consider: the choice DNN architecture and the adjustment of the chosen architecture hyperparameters. There are many DNNs that can be used to perform speech enhancement, including the MLP, CNN, DAE, RNN, GAN, and hybrid architectures; as discussed in Chapter 2. Due to the fact that these architectures have a huge number of hyperparamters to tune, it is hard to determine how the mathematical operations specific to each architecture contribute to the denoising process. These hyperparameters are divided into: layer-specific parameters, network depth, and the used activation function. There are many layer-specific parameters, such as the number of units, convolution type, number of filters, kernel size, padding type, and dilation rate. Network depth is defined as the number of layers used to implement the DNN, and it is a very important factor that must be chosen in a way to compromise between performance and complexity. Finally, activation functions are responsible for the nonlinear operation performed by the DNN, which is essential for the learning process. For speech enhancement, the most commonly used activation functions in the DNN hidden layers are: ReLU, LReLU, ELU, PReLU; while, Linear, TanH, and Sigmoid are common activation functions in

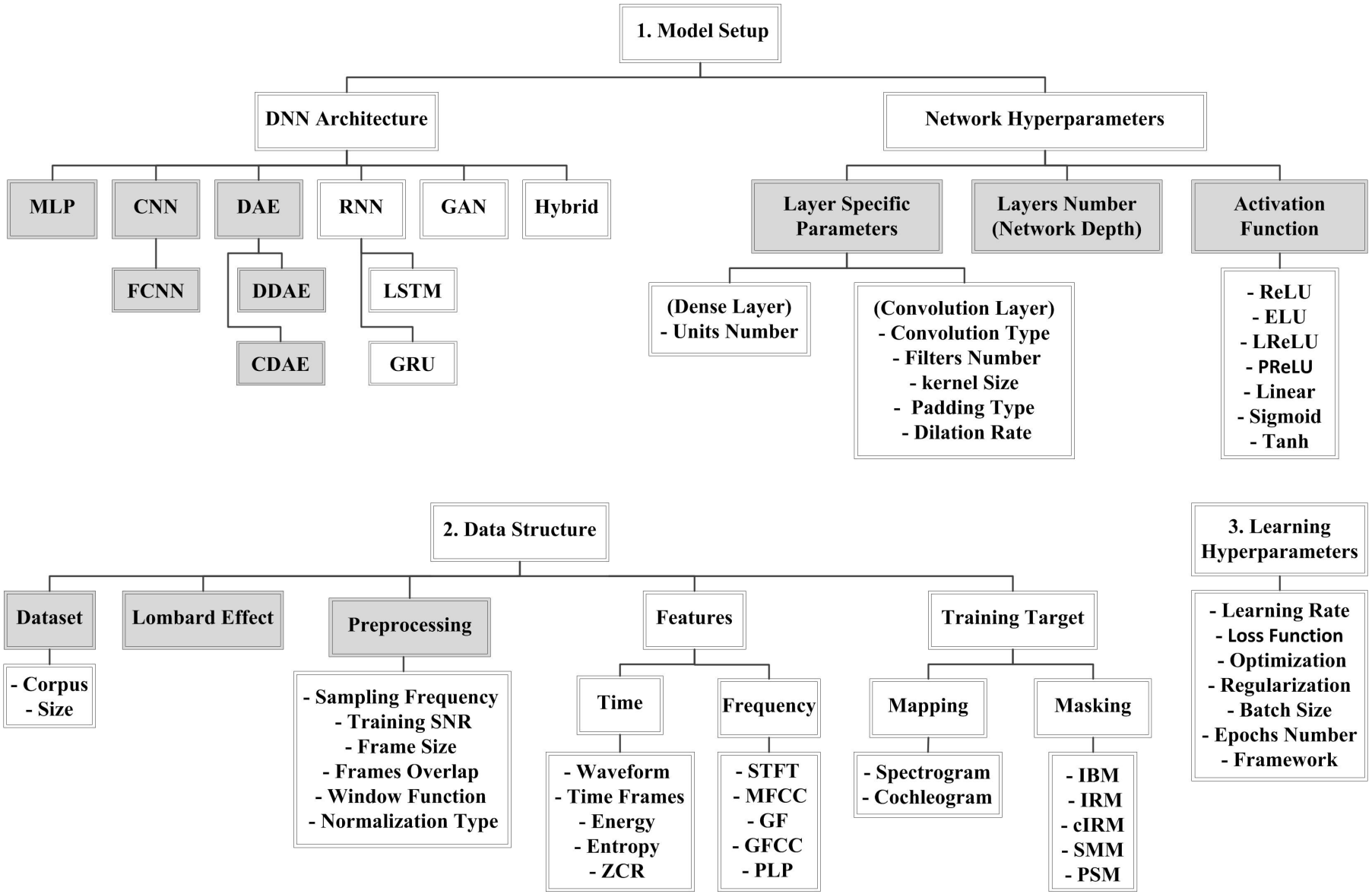


Figure 4.1 The three main factors affecting the performance of Deep Neural Networks (DNNs) for speech enhancement: Model Setup, Data Structure, and Learning Hyperparameters. The parts investigated in this chapter are shaded in grey. All acronyms are defined in Subsections 4.2.1 - 4.2.3

the output layer. Therefore, model setup step must be done carefully, in order to achieve good performance.

4.2.2 Data Structure

The size and structure of the dataset used in the training process are important factors that impact the learning process, due to the fact that deep learning is a data driven approach. The quality of the speech audio files and the variety of noise environments are crucial to guarantee a good training process, and to avoid network overfitting. Moreover, data preprocessing is mandatory before feeding the data to the DNNs. Preprocessing techniques include: audio resampling; 8 kHz and 16 kHz are the commonly used sampling frequency for speech, audio framing and windowing; which are important to ensure the efficiency of the training process, and normalization; a preprocessing technique used to ensure the stability and generalization of the training process. The input noisy audio should also be adjusted to be at specific SNR, which sets the intensity of the background noise.

Another factor that was proven to highly impact the performance is the input features to the DNN, which change the input signal representation and in turn have a great impact on the ability of the network to differentiate between speech and noise. In the time domain, it is common to use the original representation of the waveform, or to use the short time frames and extract some features, such as energy, entropy, and the ZCR (Alías et al., 2016). While, in the frequency domain, many more meaningful features can be extracted, including STFT, MFCC (Pirhosseinloo and Brumberg, 2018), Gammatone Filterbank (GF), GFCC (Shao and Wang, 2008), and PLP (Dave, 2013).

Training a DNN for speech enhancement is also greatly affected by the training target, which can be one of two types: mapping or masking (Wang et al., 2014; Odelowo and Anderson, 2018). In the case of using a mapping target, the speech enhancement problem is considered as a regression task, where the network is trying to map noisy speech to clean speech time frames, spectrogram, or cochleagram; depending on the used domain during training. While in the case of a masking target, the speech enhancement task is seen as a classification problem, where the network aims to output a mask that classifies every portion of the signal as either speech or noise, and then the enhanced speech signal can be generated by multiplying the noisy speech with this mask. There are many masking targets used in speech enhancement, such as IBM (Wang, 2005), IRM (Srinivasan et al., 2006), and SMM; also known as FFT-Mask (Wang et al., 2014), cIRM (Williamson et al., 2016), and PSM (Erdogan et al., 2015), as mentioned in Chapter 3.

Consequently, manipulation of the input data plays an important role in improving the learning process and affects the overall perception of the processed speech.

4.2.3 Learning Hyperparamters

The learning process of a DNN also has some hyperparameters, such as the learning rate, loss function, optimization technique, regularization technique, batch size, number of epochs, and the framework that was chosen for implementation (Bengio, 2012). The setup of all these hyperparameters is the third factor that impacts the performance of DNNs for speech enhancement.

4.3 The Seven Implemented DNNs

Implementation of seven DNNs has been carried out to perform the analysis presented in this chapter. These networks belong to the three broad categories: MLP, CNN, and DAE, and they are based on architectures existing in the literature. To ensure fair comparison between the seven models, some modifications were applied to the DNNs presented in literature, which lead to better performance; these modifications will be discussed below. Figure 4.2 represents the seven implemented architectures and Table 4.1 describes their configuration.

From the first category, MLP, the basic MLP architecture with three fully connected hidden layers was implemented (Xu et al., 2014*b*; Wang, 2017). Each hidden layer has 2,048 units with ReLU activation functions, followed by a batch normalization layer in order to improve training performance and stability. Three dropout layers of 20% dropout rate were used to prevent network overfitting. This network is shown in Figure 4.2(a), and its configuration is given in Table 4.1, Architecture (a).

From the second category, CNN, two architectures were implemented. The first is the basic CNN architecture with three 2D convolutional layers (Kounovsky and Malek, 2017; Chakrabarty et al., 2018). Each convolution layer has ReLU activation, 64 filters, and (3×3) kernel size. Max pooling layers were not added, to minimize information loss that may happen because of the absence of a speech reconstruction step in this configuration. Moreover, better performance was reported in the literature when removing max pooling layers for the speech enhancement process (Fu et al., 2016). Two fully connected layers were added after the convolution layers to generate the enhanced speech. The first has 512 hidden units and ReLU activation functions; while the second layer has one output unit with a linear activation function, to give the final prediction. This architecture is shown in Figure 4.2(b), and its configuration is given in Table 4.1, Architecture (b).

The second implemented CNN architecture is a FCNN model with six 1D convolution layers (Fu et al., 2017). Each convolution layer has a PReLU activation function, 64 filters, and kernel of size 20. The final convolution layer has a linear activation function and one filter, which is used to generate the output processed speech. This architecture is shown in Figure 4.2(c), and its configuration is given in Table 4.1, Architecture (c).

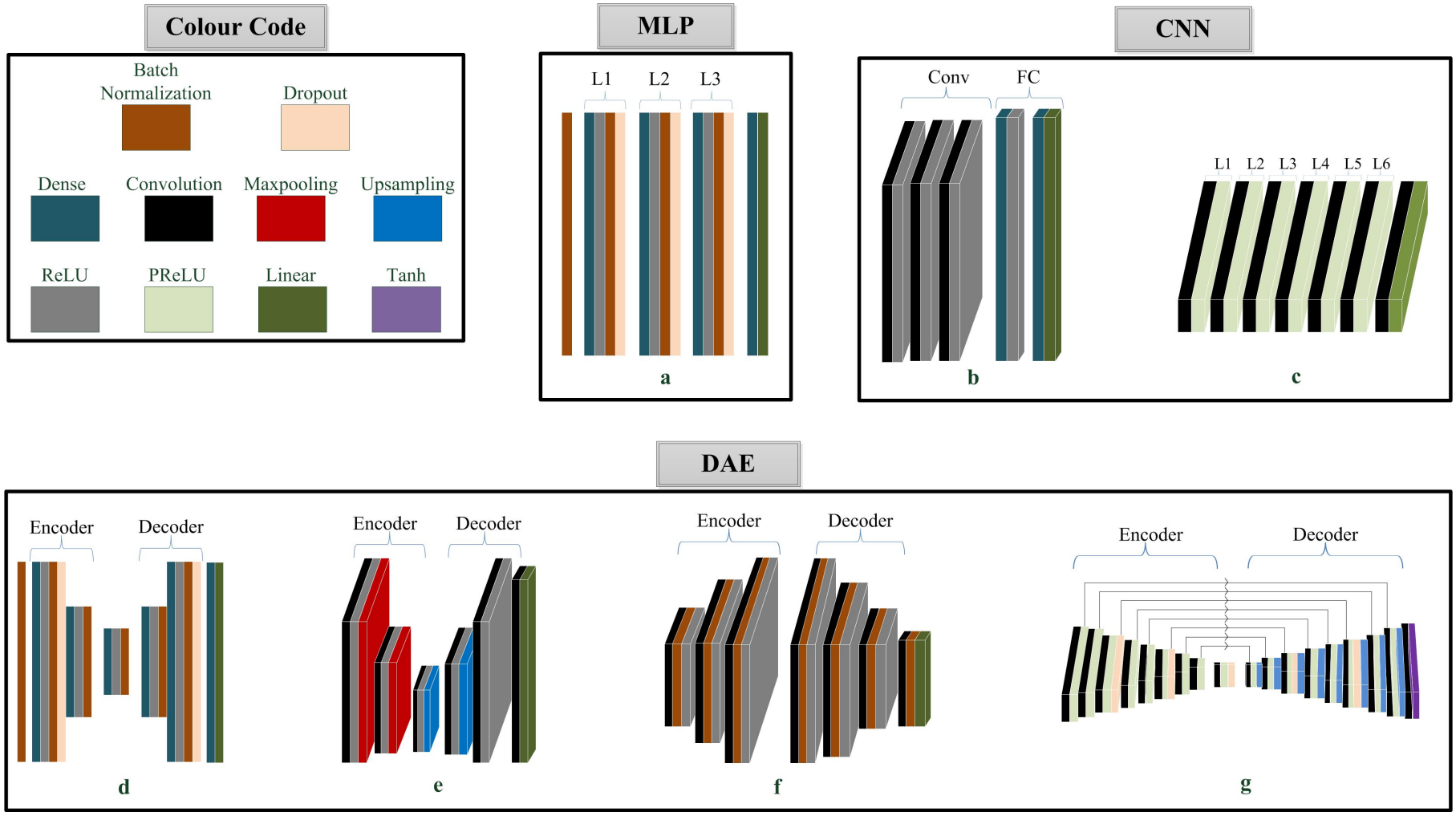


Figure 4.2 The seven DNN-based speech enhancement architectures: (a) MLP, (b) the basic CNN, (c) the FCNN, (d) the DDAE, (e) the basic CDAE, (f) a special type of CDAE, and (g) the deep CDAE

Table 4.1 The configuration of the seven implemented DNNs. This table represents the different types of layers used: Batch Normalization (BN), Fully Connected (FC), and Convolution (Conv). It also represents the number of units (Units), activation function (Act.), dropout ratio (D), kernel size (K), number of filters (F), and the sizes of max pooling (MP), upsampling (US), and stride (S)

Architecture *(a)*					**Architecture *(d)***				
Type	**Units**	**Act.**	**D**	**BN**	**Type**	**Units**	**Act.**	**D**	**BN**
BN	-	-	-	-	BN	-	-	-	-
FC	2,048	ReLU	0.2	yes	FC	2,048	ReLU	0.2	yes
FC	2,048	ReLU	0.2	yes	FC	500	ReLU	-	yes
FC	2,048	ReLU	0.2	yes	FC	180	ReLU	-	yes
					FC	500	ReLU	-	yes
					FC	2,048	ReLU	0.2	yes
FC [o/p]	129	Linear	-	no	FC [o/p]	129	Linear	-	no

Architecture *(b)*					**Architecture *(c)***				
Type	**K**	**Act.**	**F**	**Units**	**Type**	**K**	**Act.**	**F**	**Units**
2D-Conv	(3x3)	ReLU	64	-	1D-Conv	20	PReLU	64	-
2D-Conv	(3x3)	ReLU	64	-	1D-Conv	20	PReLU	64	-
2D-Conv	(3x3)	ReLU	64	-	1D-Conv	20	PReLU	64	-
FC	-	ReLU	-	512	1D-Conv	20	PReLU	64	-
					1D-Conv	20	PReLU	64	-
					1D-Conv	20	PReLU	64	-
FC [o/p]	-	Linear	-	129	1D-Conv [o/p]	20	Linear	1	-

Architecture *(e)*					**Architecture *(f)***				
Type	**K**	**Act.**	**F**	**MP/US**	**Type**	**K**	**Act.**	**F**	**BN**
2D-Conv	(3x3)	ReLU	64	MP(2x2)	2D-Conv	(7x7)	ReLU	64	yes
2D-Conv	(3x3)	ReLU	64	MP(2x2)	2D-Conv	(5x5)	ReLU	128	yes
2D-Conv	(3x3)	ReLU	64	MP(2x2)	2D-Conv	(3x3)	ReLU	256	yes
2D-Conv	(3x3)	ReLU	64	US(2x2)	2D-Conv	(3x3)	ReLU	256	yes
2D-Conv	(3x3)	ReLU	64	US(2x2)	2D-Conv	(5x5)	ReLU	128	yes
2D-Conv	(3x3)	ReLU	64	US(2x2)	2D-Conv	(7x7)	ReLU	64	yes
2D-Conv [o/p]	(3x3)	Linear	1	-	2D-Conv [o/p]	(7x7)	Linear	1	yes

Architecture *(g)*

Encoder						**Decoder**					
Type	**K**	**Act.**	**F**	**S**	**D**	**Type**	**K**	**Act.**	**F**	**US**	**D**
1D-Conv	(7x7)	PReLU	64	2	-	1D-Conv	(3x3)	PReLU	256	2	-
1D-Conv	(7x7)	PReLU	64	2	-	1D-Conv	(3x3)	PReLU	256	2	-
1D-Conv	(7x7)	PReLU	64	2	0.2	1D-Conv	(3x3)	PReLU	256	2	0.2
1D-Conv	(5x5)	PReLU	128	2	-	1D-Conv	(5x5)	PReLU	128	2	-
1D-Conv	(5x5)	PReLU	128	2	-	1D-Conv	(5x5)	PReLU	128	2	-
1D-Conv	(5x5)	PReLU	128	2	0.2	1D-Conv	(5x5)	PReLU	128	2	0.2
1D-Conv	(3x3)	PReLU	256	2	-	1D-Conv	(7x7)	PReLU	64	2	-
1D-Conv	(3x3)	PReLU	256	2	-	1D-Conv	(7x7)	PReLU	64	2	-
1D-Conv	(3x3)	PReLU	256	2	0.2	1D-Conv	(7x7)	PReLU	64	2	0.2
1D-Conv [o/p]	(7x7)	TanH	1	-	-	-	-	-	-		

From the third category, DAE, four architectures were implemented; one DDAE architecture and three CDAE architectures. The first chosen architecture is a DDAE (Lu et al., 2013) that has two fully connected layers in each of the encoder and the decoder networks. One of these layers has 2,048 hidden units and the other has 500 hidden units. Between the encoder and the decoder, a bottleneck fully-connected layer of 180 hidden units was used. All layers apply ReLU activation functions and batch normalization; moreover, a dropout of rate 20% was used in the first layer of the encoder and the last layer of the decoder. This architecture is shown in Figure 4.2(d), and its configuration is given in Table 4.1, Architecture (d).

The second architecture is the basic CDAE network (Grais and Plumbley, 2017). Three 2D convolution layers were added to each of the encoder and decoder networks, and ReLU activations were applied in each convolution layer, except the final convolution output layer, which has linear activation. In the encoder, (2×2) max pooling layer was added after each convolution layer to compress the data; while in the decoder, (2×2) upsampling layers were used to reconstruct the data. Filters of size 64 and (3×3) kernels were used in all convolution layers. This architecture is shown in Figure 4.2(e), and its configuration is given in Table 4.1, Architecture (e).

The third architecture is a special type of CDAE (Park and Lee, 2016), which has three 2D convolution layers in each of the encoder and decoder with no max pooling and upsampling layers. Therefore, this architecture does not perform data compression, but it belongs to the CDAE category as the filter size is increasing across the encoder network, and decreasing across the decoder network. This process applies a different feature extraction method than that performed by the max pooling layer, used in the previous architecture, as here feature extraction is based on changing the filter sizes across the encoder and decoder without affecting data size. This is a unique process for this architecture type, and will be the main factor affecting the performance in comparison to other architectures. The filter sizes used are 64, 128, and 256; and kernels of sizes seven, five, and three were used in both the encoder and the decoder. Batch normalization is used in all layers for training stability, and the ReLU activation was applied in all layers, except the output layer which has linear activation. This architecture is shown in Figure 4.2(f), and its configuration is given in Table 4.1, Architecture (f).

The fourth architecture is a deep CDAE that uses strided convolution (Pandey and Wang, 2019; Pascual et al., 2017). This architecture consists of nine 1D convolutional layers with PReLU activation function in the encoder and decoder, and a final convolution output layer of TanH activation. In the encoder network, strided convolution is performed with stride size 2; while upsampling of size 2 is applied in the decoder network. The filter and kernel sizes change after every three layers; 64, 128, and 256 filter sizes were used, and seven, five, and three kernel sizes were used. Consequently, this architecture combines the two feature extraction techniques: compression and increas-

ing filter size, used in the second and third CDAE types discussed above. To avoid the network overfitting to the training data, a dropout of rate 20% was applied after every three convolution layers. As this architecture is deep, skip connections were added in this implementation, to avoid information loss that might occur as the processing proceeds deeper through the network. This architecture is shown in Figure 4.2(g), and its configuration is given in Table 4.1, Architecture (g).

The choice of these architectures is based on the fact that these are from the best performing models belonging to the three main categories under investigation. Moreover, the setup used for these models, referring to Figure 4.2 and Table 4.1, was chosen in order to fairly compare specific features that are unique to each architecture type.

For the fully connected architectures, *a* and *d*, it is clear that the configuration of both architectures is the same, the difference in architecture d is a decrease in the number of hidden nodes and the addition of a decoder network for audio reconstruction. Therefore, architecture *d* is an autoencoder version of architecture *a*, and it will show the effect of autoencoder related operations when compared to architecture *a*. The same applies to the convolution-based architectures, *b* and *e*. Architecture *e* is an autoencoder version of *b*, by removing the fully-connected layers and using max pooling layers, for dimensionality reduction, and a decoder network for audio reconstruction.

For the CNN architectures, *b* and *c*, architecture *c* is an FCNN version of *b*. The main differences between these architectures are: replacing the fully connected layers with convolutional layers, processing the audio while using 1D convolutions instead of 2D, and using PReLU activations instead of ReLU. The effect of these three factors will be separately discussed in the Results section, Section 4.5.

Regarding the CDAE based architectures, the difference between architectures *e* and *f* is the feature extraction method, because architecture *e* is based on max pooling layers, while architecture *f* is based on increasing the number of filters through the hidden layers without having max pooling layers. Consequently, feature extraction is the point of comparison here. Finally, architecture *g* addresses the use of 1D strided convolutions for DAEs and the effect of increasing the depth with the use of skip connections.

4.4 Experimental Setup

This section presents the speech and noise datasets used, and how this data was prepared to conduct the experiments. Furthermore, details of the training setup and the chosen networks' hyperparamters will be discussed.

4.4.1 Dataset Selection

In the training process, five hours of clean English speech was randomly selected from the online available Voice Bank corpus (Veaux et al., 2013). This speech data was

corrupted with a total of 105 different noise environments, selected from two corpora: 90 from the 100 Environmental Noise corpus (Hu, 2014) and 15 from the NOISEX-92 corpus (Varga and Steeneken, 1993). While when showing the effect of increasing the number of noise environments on the performance, a total of 1,250 different noise environments were used, taken from the Environmental Sound Classification (ESC) 50 dataset (Thiemann et al., 2013), Urban Sound dataset (Salamon et al., 2014), and DEMAND Dataset (Thiemann et al., 2013). It should be mentioned that five hours of noisy speech was found to be enough for all the architectures to converge, based on practical trials.

In the testing process, different speech corpora were used to evaluate the performance of the architectures under different conditions. First, the performance was tested on matched dataset by using 30 minutes of clean speech from the Voice Bank corpus, not seen in the training process; this will be denoted by *matched test set*. When testing network generalization ability for mismatched dataset, 30 minutes of clean speech was selected from the LibriSpeech corpus (Panayotov et al., 2015); denoted by *mismatched test set*. While another 30 minutes of clean speech was taken from the 176 Possible Languages corpus (Topcoder, 2017), where the selected audio files contain 90 different languages; this will be denoted by *Languages test set*. The Lombard GRID corpus (Alghamdi et al., 2018) was used while investigating the effect of the Lombard phenomena; this will be denoted by *Lombard test set*.

In all cases, the speech data was corrupted with 20 noise environments, half-seen and half-unseen in the training process. These noise environments are a mixture of human-generated noise, such as crying, yawning, and human crowd sounds; and, other non-human generated noise, such as Additive White Gaussian Noise (AWGN), phone dialling, shower noise, tooth brushing, and wood creaks. Figure 4.3 represents the spectrograms of the noise environments that were used in the testing process. This figure shows how the seen noise environments are challenging, considering the fact that they cover most of the spectrum; while the unseen noise environments are different in nature to the seen noise environments, considering that they are varying across the spectrum; moreover, they include human-generated noise, which is a very challenging noise type for the network as it is similar in nature to the speech signal. This means that the evaluation and obtained results in this work are non-biased.

To evaluate the performance of the network in challenging conditions, an online noisy dataset for reverberant speech was used (Valentini-Botinhao et al., 2017*a*); this will be denoted by *reverberant test set*. Finally, the speech audios of the *matched test set* were corrupted with babble noise audio files taken from this online available noise dataset (Reddy et al., 2019); denoted by *babble noise test set*. These test sets are summarized in Table 4.2.

Table 4.2 Testing datasets

Test set	Description
Matched test set	30 minutes of clean speech from the Voice Bank corpus (Veaux et al., 2013), not seen in the training process
Mismatched test set	30 minutes of clean speech was selected from the LibriSpeech corpus (Panayotov et al., 2015).
Languages test set	30 minutes of clean speech was taken from the 176 Possible Languages corpus (Topcoder, 2017).
Lombard test set	Data from The Lombard GRID corpus (Alghamdi et al., 2018).
Reverberant test set	Data from reverberant speech dataset (Valentini-Botinhao et al., 2017*a*).
Babble noise test set	speech audios of the *matched test set* were corrupted with babble noise, taken from (Reddy et al., 2019).

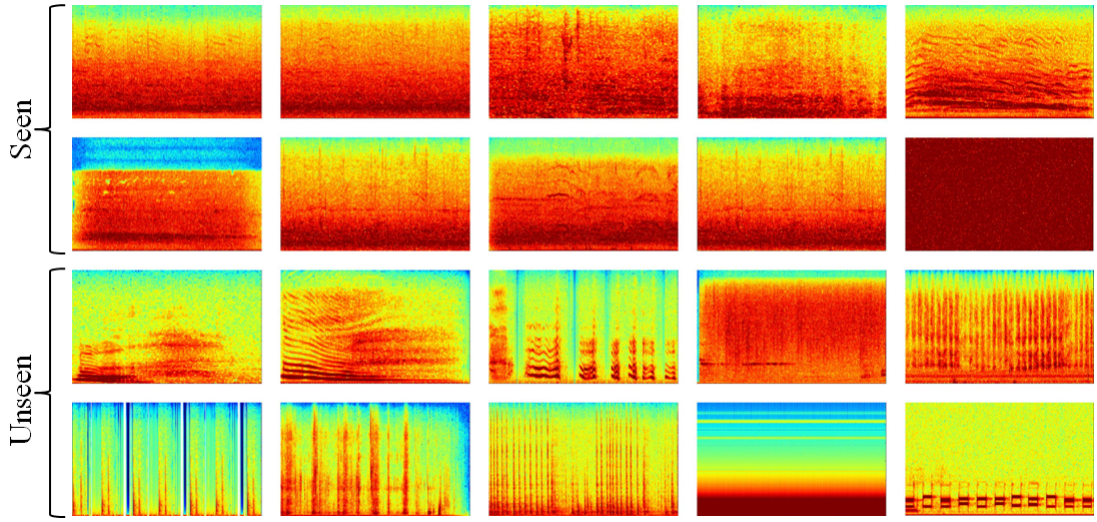


Figure 4.3 Spectrograms of the noise environments used in the testing process

4.4.2 Training Setup

The speech and noise data were mixed at the default 0 dB SNR to create the training noisy speech data. The training data was then normalized to zero mean and unit variance to improve the learning process. The sampling frequency was set to 8 kHz, to provide the most relevant speech frequency band to the DNN. Framing and windowing were applied to the data, a Hamming window was used of frame length 32 ms (256 samples) with 50% overlap. The magnitude power spectrum of the signal was then extracted with 256 FFT size, and the noisy phase was kept to be added to the estimated clean speech, assuming that the phase is less affected by the noise (Wang and Lim, 1982). In all experiments, except for the comparison between different training targets and domains,

magnitude spectrogram mapping is the training target used, in order to ensure a good generalization for all architecture types (Nossier et al., 2020). In the case of comparing mapping and masking approaches, different masking targets were used to perform the comparison; while, in time versus frequency domain learning, time frames of 2,048 length were used as a training target.

The framework used to implement the seven DNN architectures is Keras library with Tensorflow backend. MMSE is the default choice of loss function used in the training process, because our goal here is to improve all of the evaluation metrics, not a specific one (Kolbæk et al., 2020). The Adam optimizer was used; learning rate = 0.001, b1 = 0.1, b2 = 0.999. A batch size of 128 was used, and 10% of the training data was used in validation in order to monitor the performance of the networks, to avoid network overfitting. For all DNNs, no improvement in the performance was detected after 40 epochs, so the training process of all architectures is based on 50 epochs.

4.5 Results and Discussion

In this section, the outcome of the performed comparison and analysis will be presented and discussed. Subsections 4.5.1 to 4.5.5 presents the comparison between the seven DNN architectures with respect to the overall quality of the output speech in matched and challenging conditions, generalization to mismatched data, and network's complexity. While, Subsections 4.5.6 to 4.5.10 shows the effect of changing the training factors, discussed in Section 4.2, on the performance of these architectures.

4.5.1 Objective Evaluation

The seven DNN models were evaluated using the five standard, commonly used speech enhancement objective measures: PESQ (Rix et al., 2001), STOI (Taal et al., 2011), LSD (Du and Huo, 2008), SDR (Hu and Loizou, 2007*a*), ΔSegmental Signal to Noise Ratio (SSNR) (Hansen and Pellom, 1998). The DNNs were evaluated using the *matched test set* on three high SNR levels: 20 dB, 15 dB, and 10 dB; and three low SNR levels: 5 dB, 0 dB, -5 dB. The average of high and low SNRs was then calculated, denoted as *high* and *low*, respectively. This is shown in Table 4.3 and Figure 4.4.

At high SNR levels, the basic CNN network, *b*, shows better performance than the MLP network, *a*. Conversely, at low SNR levels, the MLP network, *a*, performs better than the CNN network in terms of all the evaluation metrics. Moreover, the DDAE, *d*, which is the autoencoder version of the MLP network, results in further improvements over the MLP network. This is due to the effect of bottleneck features, the unique characteristic of this network type. However, the FCNN, *c*, generates speech with better quality and intelligibility scores in comparison to the two fully-connected networks, *a* and *d*. The basic CDAE model, *e*, generates speech with the poorest overall per-

formance. However, a clear improvement in the performance is shown, when applying modifications to this network type by increasing the number of filters through the hidden layers and removing max pooling layers, to develop the CDAE network, *f*. Additionally, increasing the depth of this network with the use of 1D strided convolutions results in further improvement, and results in the network outperforming other models, as in the case of the deep CDAE network,*g*.

It should be mentioned that minimal improvement is shown at high SNR levels for most of the models, and some models generate speech with worse overall perception in this case compared to the noisy version, such as networks *a*, *b*, *e*, and *f*. The reason for this is the processing applied by the DNNs to perform the denoising process, which negatively affects speech quality, and this negative effect overrides the positive effect of the denoising process at high SNRs.

The spectrograms in Figure 4.5 show the clean, noisy, and estimated speech from the seven DNNs when tested using noisy speech with unseen tooth brushing noise at 0 dB SNR. It is clear that most of the networks managed to eliminate most of the background noise; the remaining noise is highlighted with the dashed black line. However, a main drawback of the denoising process is speech distortion, which is highlighted with the solid black line. The amount of distortion and residual noise are the main factors affecting the performance of each model, for example, network *a* and *e* suffer from very high distortion, and this explains why they have poor performance. Moreover, the output from network *e* experiences high-intensity noise and some distortion affecting the fundamental frequencies; for this reason, it has the poorest performance compared to other models. Network *b*, *c*, *d*, and *f* have some remaining high intensity noise affecting the fundamental frequencies of speech; however, they have less distortion compared to network *a* and *e*; consequently, they outperformed them. While network *g* is the only one that managed to mitigate the noise affecting the fundamental frequencies with a good reconstruction of the speech signal as well. Although network *f* managed to remove more noise compared to *g*, the fact that it has some residual high-intensity noise affecting the fundamental speech frequencies makes it perform worse than *g*.

4.5.2 Subjective Evaluation

In order to validate the results from the subjective evaluation, a subjective speech quality test was performed using 23 volunteer listeners with no hearing issues. None of the participants is in a dependent relationship with the researcher of this PhD or any member of the supervision team. The evaluation was performed online using Google form secured by a password, by sharing the form with the University of East London Computer Science masters students through the University of East London Email platform. The listeners were asked to listen in a quite environment to speech audios, and

Table 4.3 Average PESQ, STOI, LSD, and, ΔSSNR results of high SNR levels: 20, 15, and 10 dB; low SNR levels: -5, 0, 5 dB; and the average of high and low SNRs (ave)

Metric		Noisy	*a*	*b*	*c*	*d*	*e*	*f*	*g*
PESQ	**high**	2.620	2.334	2.540	2.828	2.706	1.923	2.392	**2.804**
	low	1.818	1.959	1.869	2.178	2.142	1.647	1.886	**2.282**
	ave	2.219	2.147	2.205	2.503	2.424	1.785	2.139	**2.543**
STOI	**high**	0.871	0.805	0.840	0.860	0.831	0.636	0.799	**0.868**
	low	0.715	0.715	0.688	0.751	0.739	0.569	0.704	**0.772**
	ave	0.793	0.760	0.764	0.805	0.785	0.602	0.751	**0.820**
LSD	**high**	1.633	**1.115**	1.564	1.236	1.277	1.918	1.305	1.408
	low	2.430	**1.408**	2.142	1.676	1.597	2.125	1.586	1.650
	ave	2.032	**1.261**	1.853	1.456	1.437	2.021	1.445	1.529
ΔSSNR	**high**	0.000	7.041	5.519	7.609	7.340	2.955	7.036	**7.689**
	low	0.000	7.483	5.273	7.474	**7.503**	4.181	7.146	6.888
	ave	0.000	7.262	5.396	**7.542**	7.422	3.568	7.091	7.288
SDR	**high**	0.732	3.457	2.957	4.523	4.569	1.064	4.229	**4.596**
	low	-0.555	3.019	2.494	3.957	**4.016**	1.061	3.600	3.989
	ave	0.089	3.238	2.726	4.240	**4.293**	1.062	3.914	4.292

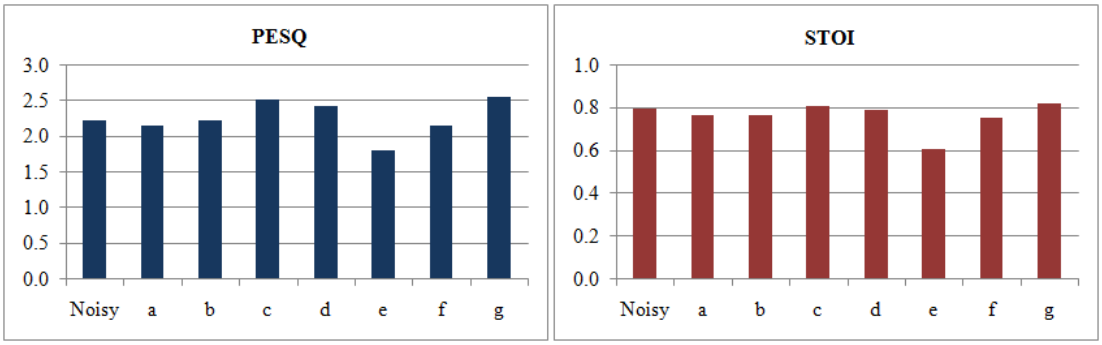


Figure 4.4 Average PESQ and STOI results at six SNR levels for the seven DNNs (a: MLP, b: CNN, c: FCNN, d: DDAE, e: shallow CDAE with max pooling, f: CDAE without maxpooling, and g: deep CDAE with strided convolution)

then evaluate the quality of the processed speech in comparison to the noisy speech by listening to the enhanced speech audios, generated by the the seven DNN models, and the original noisy speech. They scored each speech audio file between 1 and 5, where higher scores indicate better noise removal with understandable speech.

The speech used in this test was corrupted to consider a variety of challenging conditions. The noisy audio is 6 seconds audio file, consists of two English speakers, one male and then one female, with two different background noise, one seen and one unseen by the networks during training. The noises used are human-generated non-periodic crowd noise and then non-human generated periodic phone dialling noise. The noise and speech intensity are kept the same, so this evaluation is based on 0 dB SNR. All the listeners were instructed to listen to the noisy speech first, and then to the processed speech by all networks in the same order as presented in the Google form. The

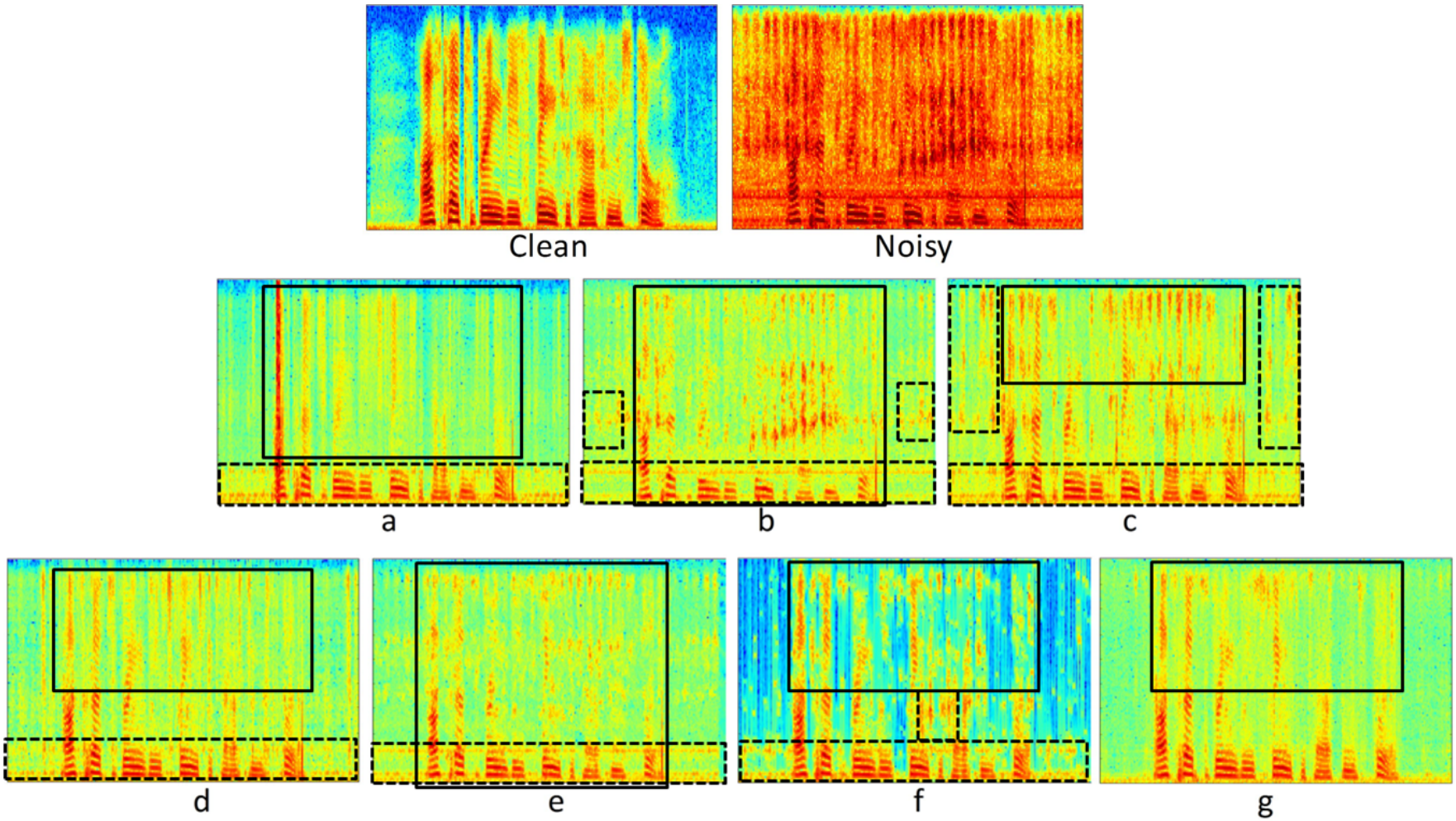


Figure 4.5 The spectrograms of clean speech, its noisy version with toothbrush noise, and the output estimated clean speech from the seven DNNs. Solid black and dashed lines highlight high distortion and high intensity residual noise, respectively.

listeners have no information about the type of the network that generated the processed speech audio files, as the files were given a random character as a name.

The statistical analysis of the obtained results from the subjective evaluation is shown in Table 4.4. The average (Ave) and the Standard Deviation (SD) were first calculated. It is noticed that network *c* is the best performing based on the human listeners' opinion, not *g* as shown before by the objective evaluation. The reason for this mismatch is the different preferences of listeners, because some listeners may prefer the existence of some remaining noise with a clearer speech, such as in the case of network *c* rather than removing most of the background noise with non-perfect speech reconstruction as in the case of network *g*, while a computer algorithms output is negatively affected by residual noise. As a result, although the compression process in DAEs and the depth of the architecture help in removing the noise, it may have a negative impact on the quality of the heard speech. The listeners' different preferences are also proven by the high SD in the case of the noisy speech, some listeners seem to find the noisy speech version better than the processed clean speech because the enhanced speech from any DNN experiences a level of distortion, which affects speech intelligibility. The mode was then calculated to show the score value with the highest occurrence among listeners for each architecture, and the percentage of occurrence of this score was also calculated. This also shows that most of the listeners preferred the processed speech by network *c*. Moreover, the original noisy speech and network *e* have the lowest score, the same as reported by the objective evaluation. Finally, the P-value was calculated to show the significance of the results compared to the noisy speech, the two-tailed T-test was performed with 95% confidence level. It was found that there is no significant difference between the average scores of network *a* and *e* when compared to the noisy speech, and this is due to the high distortion of these networks, as shown in Figure 5. The same test was also performed between all combinations of architectures, and the results show that there is no significant difference between network *d* and *g*, and network *b* and *f*.

Table 4.4 Subjective evaluation results

Metric	**Noisy**	***a***	***b***	***c***	***d***	***e***	***f***	***g***
Ave	2.13	2.57	2.70	**3.70**	3.09	2.09	2.78	2.96
SD	1.36	0.95	1.06	0.93	1.00	1.08	1.13	1.02
Mode	1	3	3	**4**	3	1	2	3
Mode%	43%	48%	35%	43%	43%	39%	39%	39%
P-value	-	**0.13**	0.03	0.00	0.002	**0.87**	0.03	0.02

4.5.3 Evaluation in Challenging Conditions

This experiment aims to show the effect of some very challenging noise environments on the architectures' performance. Although DNNs were proven to effectively eliminate different noise environments, some types of noise are still known to be difficult to separate from speech signals, such as speech babble noise (N1), having two noises in the background instead of one (N2), and reverberation (N3). The evaluation of the networks' performance in the case of these three challenging noise conditions is given in Table 4.5 and shown in Figure 4.6. The noisy speech audios used in this evaluation to create the three conditions N1, N2, and N3 are generated using the *babble noise test set*, *matched test set* after corrupting the speech with two noise environments from the noises shown in Figure 4.3, and *reverberant test set*, respectively. The results are based on testing the seven architectures at six SNRs from -5 to 20 with a step of 5, and then the average was calculated.

The results show that there is a clear degradation in the performance of all the architectures for speech babble noise (N1) and having two noise environments (N2). However, the generated speech from all architectures is still of a good overall perception, and architecture *g* remains the best performing. Architecture *e* is the only DNN that fails to produce speech with acceptable quality and intelligibility in these conditions, and the reason for this result is the original bad speech enhancement performance for this network type, shown in Table 4.4, even for less challenging noise environments. It should also be mentioned that architecture *b* shows good generalization in the case of speech babble noise environment concerning all evaluation metrics, excluding STOI. Another point is that all architectures have high ΔSSNR for two noise environments, as the networks are removing more noise in this case, which increases the difference in SSNR between the noisy and processed speech.

Regarding reverberant speech (N3), it was found that it causes a significant negative impact on the overall performance of all architectures in terms of all evaluation metrics, especially the intelligibility of the output speech (STOI). This proves that reverberation is a very specific type of noise that the DNN fails to deal with using speech enhancement processing; consequently, reverberation can be considered as a second task for the DNN besides the de-noising task, which needs different processing or a second enhancement stage to properly deal with (Zhao et al., 2016).

4.5.4 Evaluation of the Generalization Ability

Overfitting or variance is a common problem of the deep learning approach to speech enhancement. As deep learning is a data driven approach, the network can overfit to the training data during learning, which makes the network performs very well on this data but fails to maintain the same good performance for unseen data after training. As a

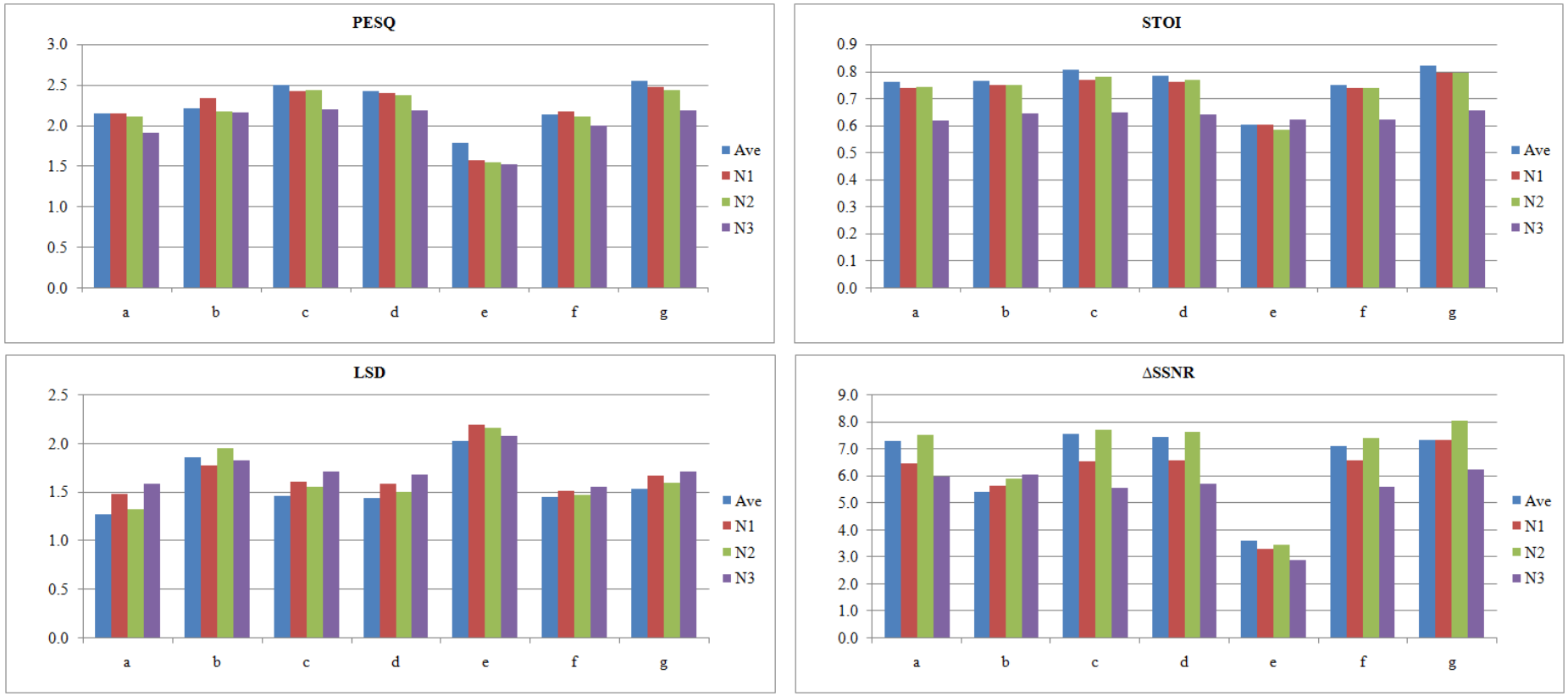


Figure 4.6 PESQ, STOI, LSD, and ΔSSNR results for three challenging noise environments: babble noise, N1; two background noises, N2; and reverberation, N3; compared to the average of seen and unseen noise shown in Table 4.3, Ave

Table 4.5 Evaluation in challenging conditions

Metric		*a*	*b*	*c*	*d*	*e*	*f*	*g*
PESQ	**N1**	2.145	2.337	2.428	2.402	1.566	2.166	**2.479**
	N2	2.103	2.174	2.429	2.368	1.546	2.110	**2.437**
	N3	1.907	2.162	**2.198**	2.184	1.522	1.999	2.180
STOI	**N1**	0.739	0.751	0.770	0.761	0.603	0.739	**0.793**
	N2	0.744	0.749	0.780	0.767	0.583	0.737	**0.794**
	N3	0.617	0.642	0.647	0.641	0.623	0.623	**0.657**
LSD	**N1**	**1.471**	1.772	1.604	1.582	2.186	1.503	1.665
	N2	**1.316**	1.951	1.544	1.492	2.153	1.469	1.594
	N3	1.578	1.818	1.708	1.674	2.066	**1.545**	1.702
ΔSSNR	**N1**	6.453	5.605	6.523	6.563	3.262	6.536	**7.301**
	N2	7.505	5.857	7.683	7.612	3.438	7.381	**8.012**
	N3	5.938	6.021	5.534	5.704	2.876	5.584	**6.208**

result, testing the generalization ability using mismatched or unseen test data is crucial to make a fair comparison between different network types.

The generalization ability of the seven implemented DNNs was evaluated by testing the performance of the network under three mismatched conditions: unseen noise environments (C1), the unseen LibriSpeech English speech dataset (C2), and unseen 90 different languages (C3). To create the test set for the unseen noise environments condition (C1), the *matched test set* speech was used and it was corrupted with only the unseen noise environments in Figure 4.3. Regarding the unseen dataset condition (C2), the *mismatched test set* was used to perform this analysis. Finally, the *Languages test set* was used to test the models' generalization for the unseen languages condition (C3). The results of this experiment are shown in Table 4.6 and Figure 4.7. These results were generated by testing the DNNs on six SNRs ranging from -5 to 20 with a step of 5, and then the average was calculated.

Most of the architectures maintained good performance in the case of unseen noise and speech from the same training dataset, C1. However, a remarkable deterioration in the performance happened for the other two mismatched conditions, unseen dataset, C2, and unseen language, C3, concerning all the evaluation metrics, except STOI. However, architecture *f* shows a very good generalization ability in the case of using different languages (2.232 and 0.798 PESQ and STOI scores, respectively), and this proves the power of extracting speech features by increasing the number of filters through the convolutional layers, which is the specific property of this architecture. An explanation of the increase in the STOI score in the case of these mismatched conditions is that the network denoising ability decreases and it does not harshly remove noise, as shown

in Δ SSNR results, so this results in more intelligible speech. This shows a tradeoff between noise removal and speech intelligibility and gives a reason why DNNs output speech with lowerSTOI than the noisy version at high SNRs, as discussed in Subsection *4.5.1* and shown in Table 4.3.

Table 4.6 Evaluation of the generalization ability

Metric		*a*	*b*	*c*	*d*	*e*	*f*	*g*
PESQ	**C1**	2.094	2.258	2.508	2.415	1.523	2.135	**2.615**
	C2	1.887	1.927	2.334	2.215	1.259	1.986	**2.352**
	C3	1.807	2.081	2.367	2.278	1.491	2.232	**2.375**
STOI	**C1**	0.760	0.775	0.815	0.786	0.618	0.753	**0.836**
	C2	0.772	0.794	0.847	0.821	0.588	0.797	**0.859**
	C3	0.764	0.797	0.844	0.822	0.551	0.798	**0.854**
LSD	**C1**	**1.312**	1.665	1.371	1.461	2.142	1.455	1.508
	C2	**1.302**	1.702	1.358	1.378	2.615	1.461	1.443
	C3	1.812	1.702	1.607	1.607	2.717	1.530	**1.475**
ΔSSNR	**C1**	6.041	4.714	6.375	6.182	3.075	5.998	**6.736**
	C2	3.098	4.202	3.184	3.186	1.523	**4.585**	3.388
	C3	4.454	3.634	4.834	4.911	2.727	**5.384**	5.302

4.5.5 Complexity Comparison

In order to fairly compare different DNNs for speech enhancement, evaluating the complexity of the network is crucial, because DNNs are generally complex and have huge computational costs. Network complexity can affect its applicability in a real-time implementation, because some devices in which speech enhancement is applied, such as mobile devices and hearing aids, have hardware limitations, and the DNN architecture might not fit onto the device hardware. Another issue related to network complexity is the increased processing time, which also limits network applicability. Therefore, the performance analysis of DNNs must include network complexity, and this is done by showing three factors related to complexity: number of parameters, number of layers, and inference processing time. This complexity comparison is shown in Table 4.7.

The results show that fully connected architectures (*a* and *d*) have the highest number of parameters. On the other hand, convolutional-based architectures: *b*, *c*, and *e* have a much lower number of parameters. However, increasing the number of filters in the hidden layers and network depth result in increased number of parameters for CNN architectures, such as in the case of architectures *f* and *g*. The processing time was calculated by processing 224 speech audio files of about 15 minutes duration in total. The algorithm was running on an NVIDIA Quadro M3000M GPU with clock 1,050 MHz and 160 GB/s memory bandwidth. The processing time is inversely proportional to the depth of the architecture, which is represented by the number of layers. It also depends

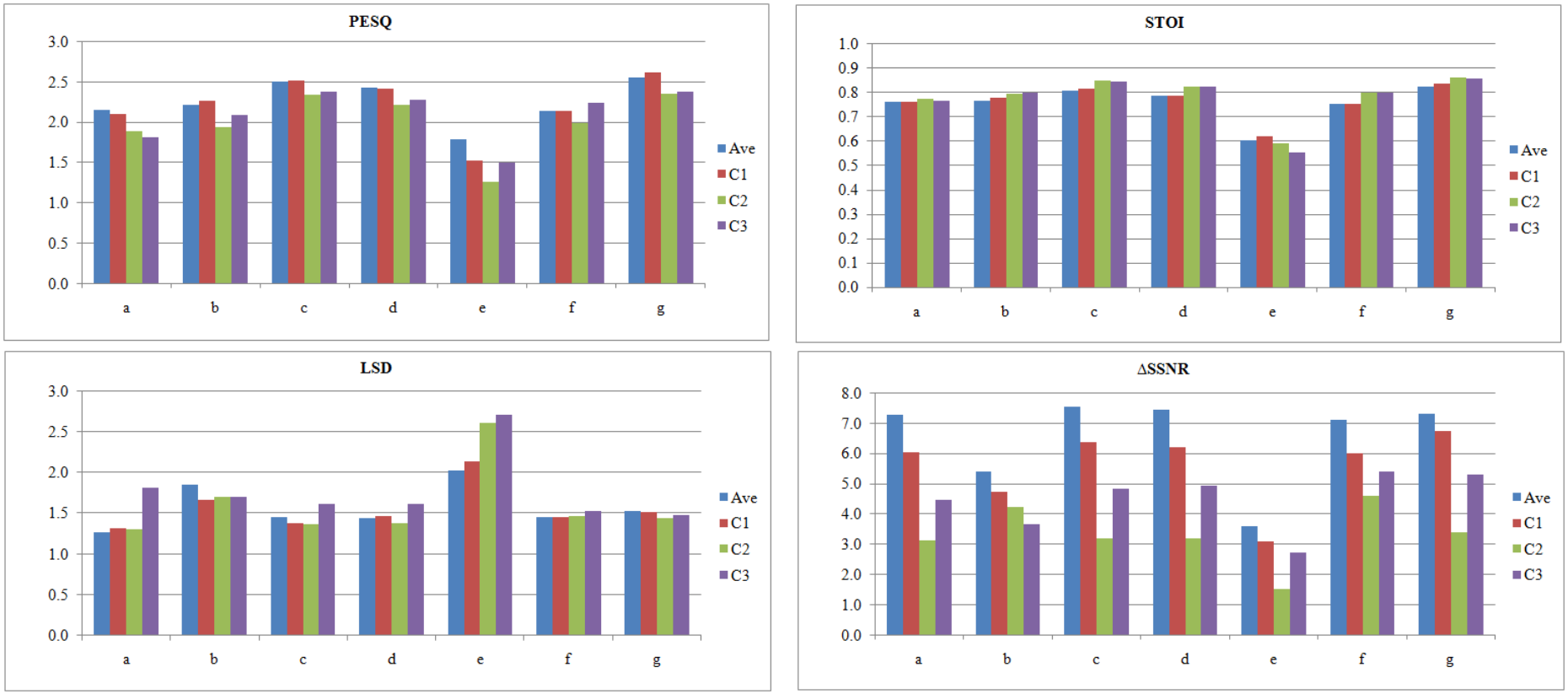


Figure 4.7 PESQ, STOI, LSD, and ΔSSNR results for processed unseen noisy speech from the same training dataset, C1; from unseen dataset, C2; and speech from 90 different languages, C3, compared to the average results shown in Table 4.3, Ave

on the architecture type, as convolutional-based DNNs are faster. Overall, architecture *b* is the least complex concerning the metrics in Table 4.7, since it is a CNN shallow network.

The complexity of the network also affects the speed of the training process, and this can be seen in Figure 4.8, which shows the loss curves of the training and validation data for the seven architectures during the training process. It is clear that complex architectures with the highest number of parameters, such as *a* and *d*, converge the fastest. The dense connections between the hidden nodes of these architecture types allow for faster learning of the mapping function that maps noisy speech to clean speech. Similarly, the complex convolution-based architecture *f* has the same fast converging behaviour, and this shows the advantage of increasing the number of filters through the hidden layer on the learning process. The other convolution-based DNNs *b*, *c*, *e*, and *g* show a more smoothly decreasing loss curve, and take longer time to converge. However, some of these architectures, such as *c* and *g*, end up with better performance than other more complex architectures, although their loss curves take a longer time to converge. As a result, complexity mainly increases the speed of the learning process; however, architecture type and design affect the performance.

Table 4.7 Comparing different networks' parameters: number of network parameters (Parm.) and layers (Layers), and testing processing time (time)

Metric	***a***	***b***	***c***	***d***	***e***	***f***	***g***
Parm.	8,948,357	**50,497**	462,081	2,784,677	112,001	1,075,717	3,078,081
Layers	15	**10**	14	20	16	21	49
Time(s)	21.5	**14.7**	24	15.5	16.7	18.4	34.5

4.5.6 Network Hyperparameters Effect

In this subsection, the effect of changing some network related hyperparameters, shown in Figure 4.1, will be investigated. Each of the three DNN main categories will be discussed separately, due to the presence of different hyperparameter for each architecture type.

4.5.6.1 MLP Architectures

The MLP is one of the first DNN architectures employed for speech enhancement, and many experiments were conducted in the literature to show the effect of different hyperparameters on architecture performance. For this reason, the effect of these hyperparameters will be only discussed based on what is reported in the literature, without repeating these experiments.

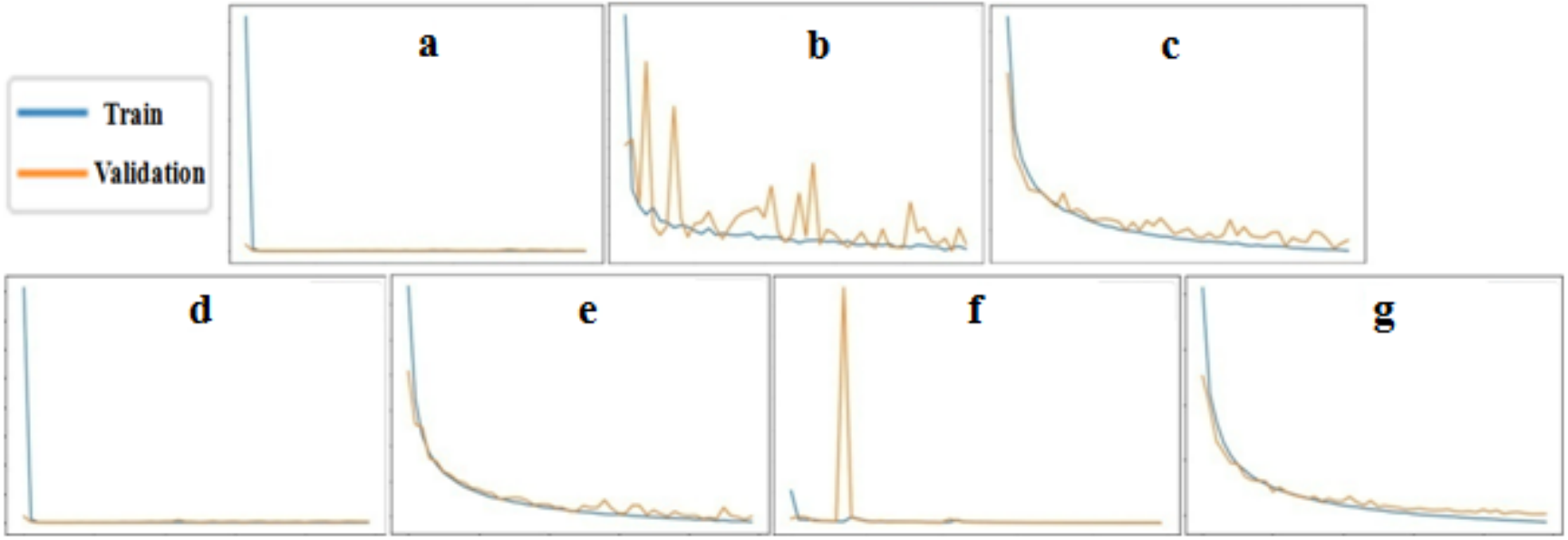


Figure 4.8 The training loss curves of the seven DNNs for the training and validation data. The x-axis represents the training epochs and the y-axis represents the mean squared error.

One of the important hyperparameters to investigate is the effect of increasing the network's depth on the performance. This was investigated in (Xu et al., 2014*b*), where the study show improvement in the performance when increasing the network's depth. However, this improvement is limited, because network's overfitting to the training data starts to happen when the architecture becomes too deep. In (Hunter et al., 2012), increasing the number of hidden units was shown to enhance the quality of the output speech; however, this highly increases the network's complexity. Therefore, the number of hidden units should be selected in a way to decrease computational cost and complexity whilst maintaining reasonable performance, and this can be achieved using the trial-and-error approach.

4.5.6.2 CNN Architectures

Based on the fact that the layers of CNN architectures are sparsely connected rather than fully connected as in the case of The MLP architecture, the activation function used has a greater impact on the network's performance. The effect of changing the activation function from ReLU to its edited versions for the CNN architecture *b* is given in Table 4.8, resulting in PReLU being the best performing activation function concerning all evaluation metrics.

Table 4.8 Effect of CNN related hyperparameters: activation functions, ReLU(CNN(b)), LReLU, ELU, and PReLU; increasing filters and kernel sizes in hidden layers; the use of 1D convolutions

Metric	CNN(b)	LReLU	ELU	PReLU	filters	$K_{(5x5)}$	CNN_{1D}
PESQ	2.205	2.188	2.274	2.342	2.371	2.413	**2.537**
STOI	0.764	0.764	0.752	0.771	0.784	0.773	**0.795**
LSD	1.853	1.891	1.700	1.534	1.569	1.455	**1.438**
ΔSSNR	5.396	6.071	6.043	6.649	6.698	6.917	**7.388**

To understand how CNNs deal with the speech enhancement problem and to show the effect of changing the activation function, a visualization of the spectrograms from the hidden layers is shown in Figures 4.9-4.12. Figure 4.10 represents 32 filters and their activations for the first hidden layer of network *b*, where the ReLU was used. The figure shows the output of the network tested using noisy speech (N) and its corresponding clean one (C), to show the behaviour of the network in both cases. It was noticed that CNNs manage to solve the speech enhancement task by applying a set of filters; these filters are represented separately in Figure 4.9 and described in Table 4.9. Some of the filters are responsible for the de-noising process, such as f_1 which mitigates the noise and outputs enhanced speech. f_2 is also a de-noising filter; however, this filter attempts to enhance the speech signal by smoothing the noise intensity to highlight speech and

then outputs enhanced speech with the same noise intensity. Another interesting filter is f_3, which works the same way as f_2; however, the output of this filter is noise, so it acts as a noise detector. Other types of filters are responsible for extracting speech features, such as f_4 which acts as a bandpass filter that outputs high and low speech frequency components. It was also found that there is a kind of filter that acts as a buffer, such as f_5, which does not affect the original input signal. It is suggested that this filter helps the network in reconstructing the clean speech and to avoid the loss of essential information. Figure 4.11 shows randomly selected filters and their activations from the second and third hidden layers of the same network, it was noticed that the same set of filters exists in these layers as well, with an extra filter f_6 that acts as a high pass filter that outputs the high-frequency speech components.

The dying ReLU problem is clear in Figures 4.11-4.12, as ReLU is turning off many filters, producing empty (white) diagrams. However, this problem was not detected when visualizing the network hidden layers when using PReLU, in Figure 4.12. This is a reason why PReLU outperforms ReLU, it can be seen from this visualization that the output after PReLU is either an enhanced speech signal or noise.

Referring to Table 4.8, "**filters**" and "$\mathbf{K_{5x5}}$" columns, the effect of increasing the filters through the hidden layers is also addressed by using 64, 128, and 256 filters in the first, second, and third layers, respectively, instead of fixing the number of filters to 64. This has a positive impact on the overall performance of the network. Moreover, a kernel of size (5x5) was used instead of (3x3) to show the effect of increasing the kernel size, and it can be seen that this also has a positive impact on the performance. Finally, 1D convolutions with PReLU were used, instead of 2D with ReLU, with a kernel size of 20. A remarkable enhancement is shown in this case, compared to the original CNN network, *b*. The implemented network after applying these modifications, ($\mathrm{CNN_{1D}}$) shown in Table 4.8, reached a performance closer to network *c* and *g*. Moreover, this network was included in the subjective testing, in subsection *4.5.2*, and it got an average score of 3.87, with 0.81 SD. Additionally, the output of the T-test shows that there is no significant difference between the average of this model and network *c*.

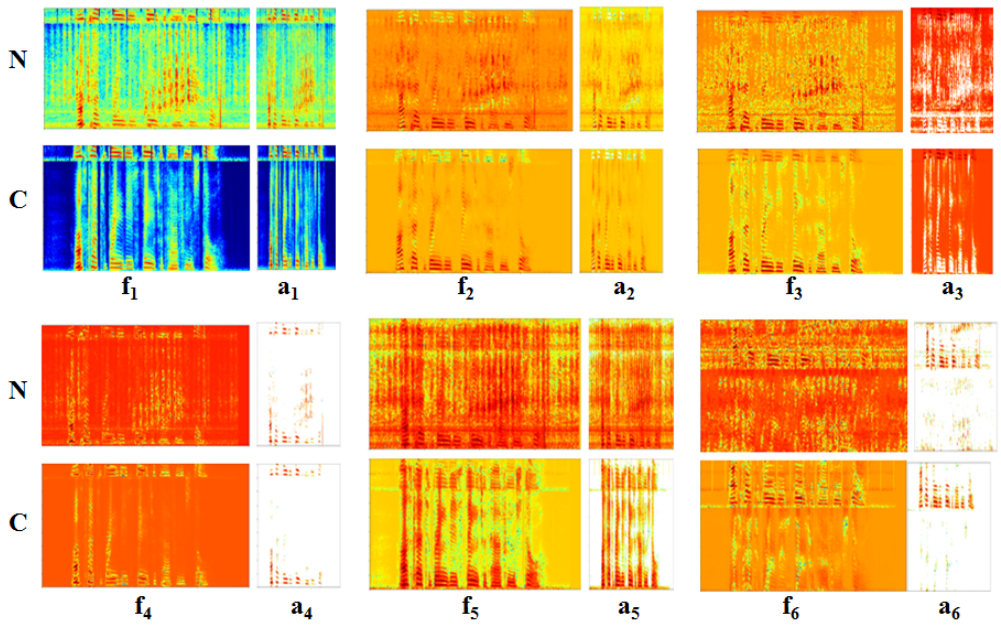


Figure 4.9 Six spectrograms randomly selected from the hidden layers of network *b*, explaining the different CNN filters for speech enhancement, for a processed noisy speech (N) and its clean version (C), *f* and *a* represent convolution filters and activation functions, respectively

Table 4.9 Description of CNN filters for the speech enhancement task

Filter	Description	Activation Output
f_1(Denoising)	Mitigate the noise	De-noised Speech
f_2(Smoothing)	Mitigate noise by smoothing its intensity to highlight speech	Speech with same intensity noise
f_3(Noise Detector)	Smoothing noise intensity and highlight speech	Noise
f_4(Band Pass)	Passes only high and low frequency bands	High and low frequency speech components
f_5(Buffer)	Gives output same as input	Original noisy speech
f_6(High Pass)	Passes only high frequency bands	High frequency speech components

4.5.6.3 DAE Architecture

Table 4.10 shows the results of the experiments for DAEs. The effect of depth was investigated; moreover, the function used for dimensionality reduction and the factors

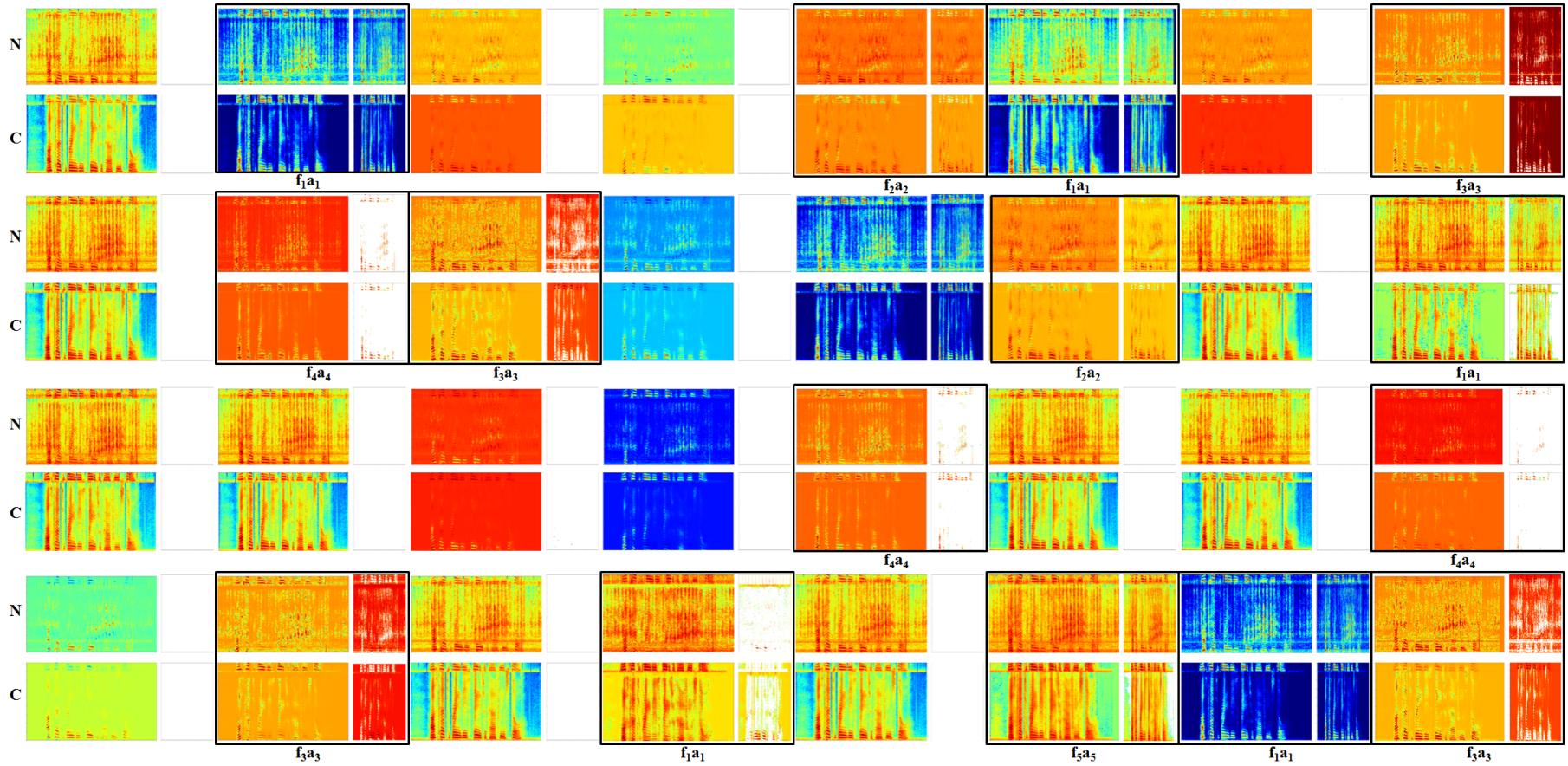


Figure 4.10 The spectrograms of 32 randomly selected convolution filters (wider spectrogram) and ReLU activation outputs from the first hidden layer of the CNN architecture *b*, for a processed noisy speech (N) and its clean version (C), *f* represents convolution filters, *a* represents activations

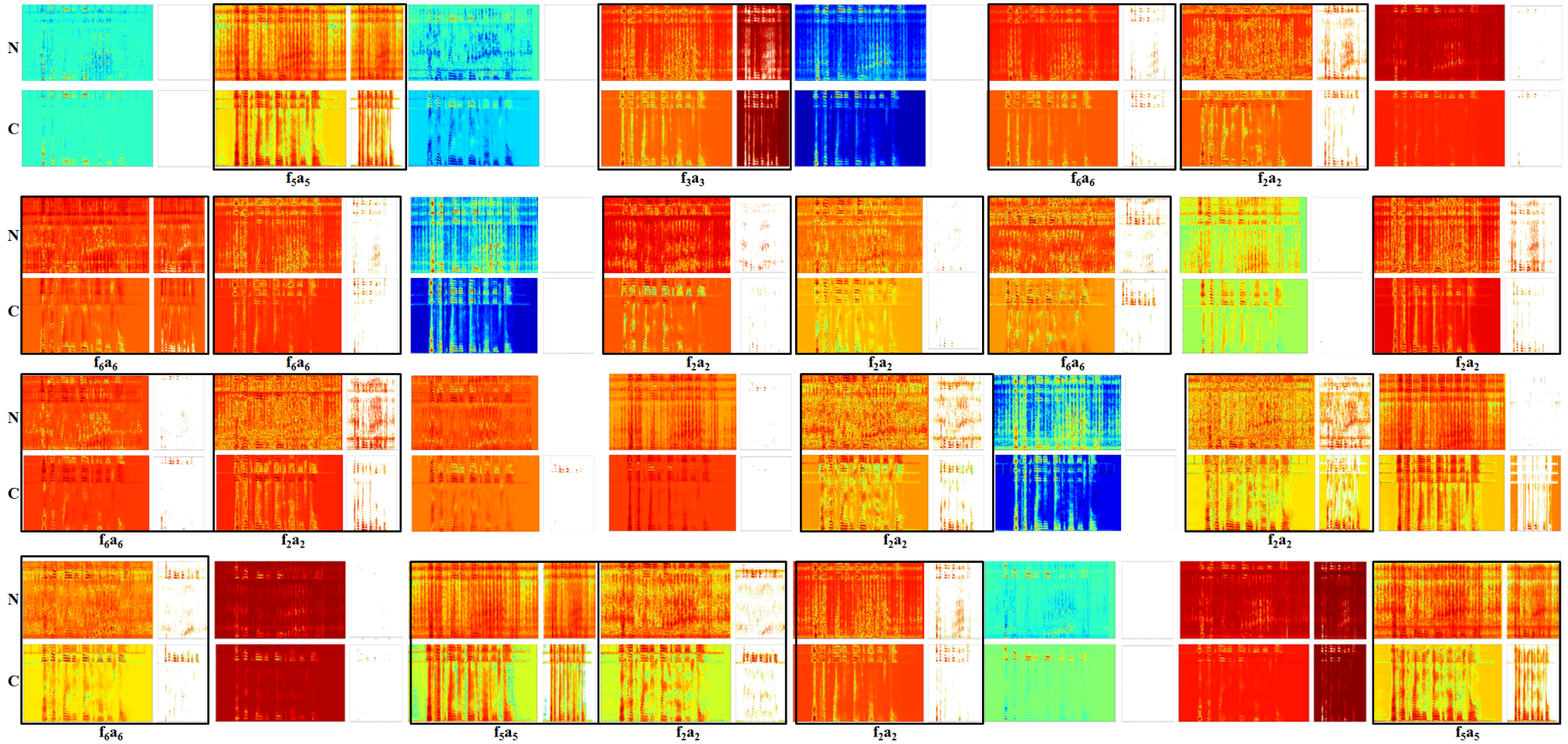


Figure 4.11 The spectrograms of 32 randomly selected convolution filters (wider spectrogram) and ReLU activation outputs from the second and third hidden layers of the CNN architecture *b*, for a processed noisy speech (N) and its clean version (C), *f* represents convolution filters, *a* represents activations

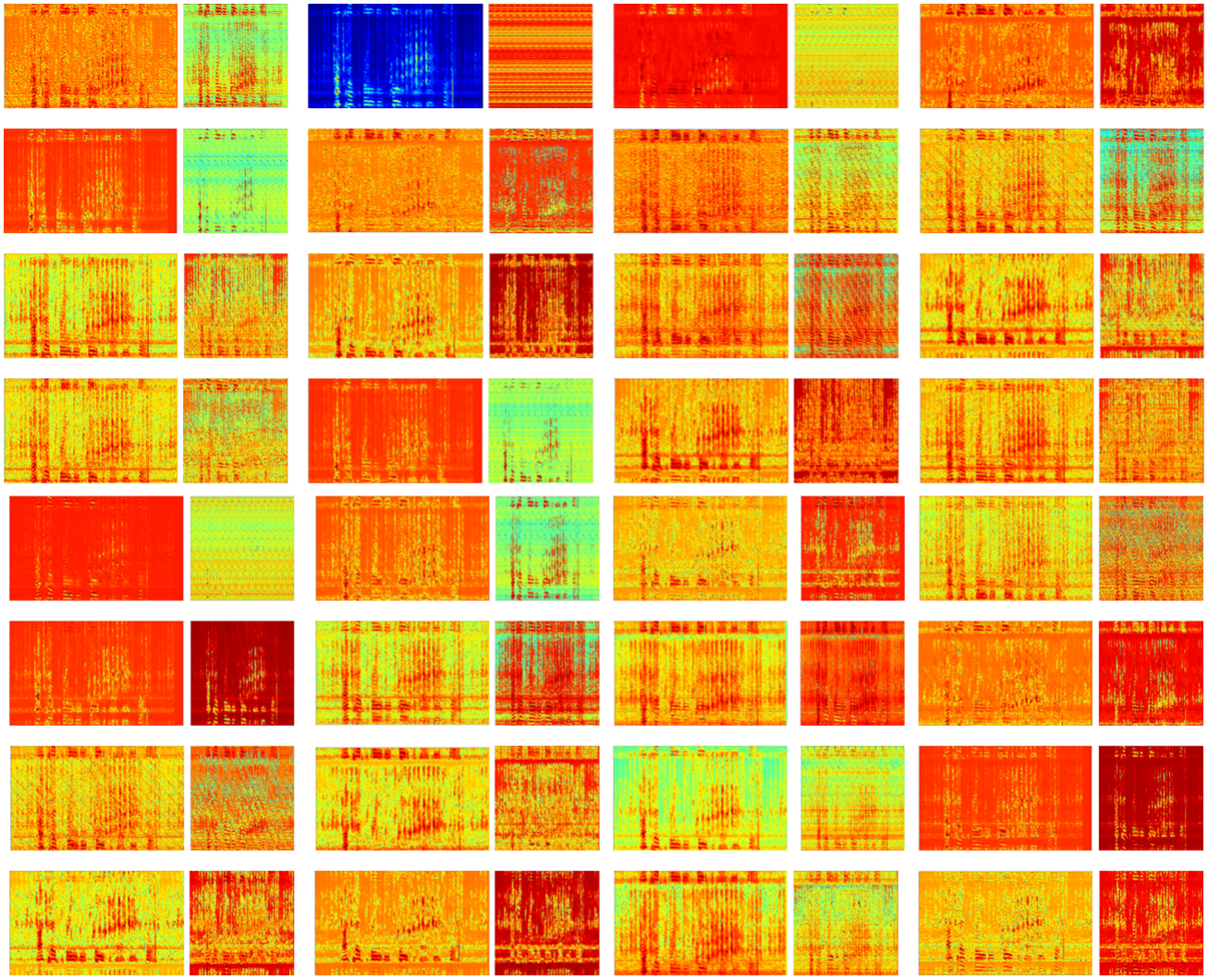


Figure 4.12 The spectrograms of 32 randomly selected convolution filters (wider spectrogram) and the PReLU activation outputs for the first hidden layer of the CNN architecture *b*, for a processed noisy speech

that affect CNN architectures, discussed above, were investigated. The results refer to the DDAE (*d*), and a deeper version of it, d_{deep}, with two more layers in each of the encoder and the decoder. The number of hidden nodes used are: 2,049, 1,024, 500, 250, and 180. Increasing the depth of DDAE was found to degrade the performance due to network overfitting, as in the case of the MLP. However, another reason for this degraded performance is the compression in the bottleneck layer, which may result in a loss of information for deep networks. The use of skip connections is a solution to this issue, although the effect of them was not investigated in our work for the DDAE, it was proven to improve the performance in (Tu and Zhang, 2017).

The basic 2D CDAE network, *e*, was edited by using strided convolutions instead of max pooling, $e_{strided}$. It can be noticed that strided convolutions lead to better results. Afterwards, the use of strided 1D convolutions with PReLU and increasing the number of filters through the hidden layers were considered, network e_{edited}, which results in further enhancement in the performance as proved in the previous subsection. Finally, one more layer was added to each of the encoder and the decoder to show the effect of increasing the depth, shown in e_{deep}. It can be concluded that increasing the depth of CDAE models results in a significant gain in the performance.

Table 4.10 Effect of DAE related hyperparameters: increasing the depth $\mathbf{d_{deep}}$ and e_{deep}; the use of strided convolutions, $\mathbf{e_{strided}}$; and the use of 1D strided convolutions with PReLU $\mathbf{e_{edited}}$

Metric	d	$\mathbf{d_{deep}}$	e	$\mathbf{e_{strided}}$	$\mathbf{e_{edited}}$	$\mathbf{e_{deep}}$
PESQ	2.424	2.310	1.785	1.802	1.887	2.457
STOI	0.785	0.773	0.602	0.637	0.695	0.774
LSD	1.437	1.548	2.021	1.983	1.938	1.472
ΔSSNR	7.422	7.335	3.568	3.549	3.779	7.310

4.5.7 Lombard Effect

It is essential to investigate how the performance of DNNs will be affected by the change of the properties of the speech signal in real noisy conditions. It is normal that people raise their voices to improve speech intelligibility in noisy environments, the phenomenon known as the Lombard Effect (Garnier and Henrich, 2014). In this experiment, all seven implemented architectures were tested using Lombard speech, to address the effect of this phenomena. An audio-visual Lombard speech corpus was used (Alghamdi et al., 2018), containing 5,400 utterances, 2,700 Lombard, and 2,700 plain reference utterances, spoken by 54 native speakers of British English. The results shown in Table 4.11 are based on testing noisy speech utterances using the *Lombard test set*, which is of 30 minutes duration, the same duration as the one used in the pre-

vious evaluation. These utterances were randomly selected from each of the Lombard and plain speech audios, and then corrupted by the same 10 unseen noisy environments used before, shown in Figure 4.3, and at the same six SNR levels. The results shown in Table 4.11 are the average scores of the used SNR levels.

The results show that the intelligibility of the processed speech is better in the case of the Lombard effect simulated speech for all the tested DNNs; moreover, an improved overall performance was found for most of the architectures. This behaviour is unexpected from DNNs, because normally DNNs have a worse performance for unseen data during the training process, such as the Lombard effect simulated speech in this case. Based on this outcome, it can be concluded that the learned features during the training process made the network robust to the change in the speech features resulting from this phenomenon. These results also support what was reported in (Michelsanti et al., 2019); however, here the authors trained a DNN using Lombard simulated speech, and it was proved to result in a better performance than training the network with normal speech.

Table 4.11 Average results for PESQ, STOI, LSD, and ΔSSNR when testing the seven DNNs using plain (P) and Lombard effect simulated speech (L) at six SNR levels, from -5 to 20 with a step of 5

Metric		a	b	c	d	e	f	g
PESQ	**P**	**1.337**	1.530	1.753	**1.635**	**1.105**	1.530	**1.880**
	L	1.315	**1.554**	**1.772**	1.626	1.071	**1.554**	1.841
STOI	**P**	0.592	0.637	0.706	0.664	0.517	0.637	0.728
	L	**0.604**	**0.663**	**0.733**	**0.684**	**0.518**	**0.663**	**0.729**
LSD	**P**	**1.606**	**1.569**	1.465	1.557	2.107	**1.569**	1.540
	L	1.607	1.579	**1.395**	**1.486**	**2.039**	1.579	**1.476**
ΔSSNR	**P**	6.686	5.709	5.442	5.623	4.171	5.709	6.066
	L	**8.552**	**7.882**	**8.174**	**7.997**	**5.183**	**7.882**	**8.394**

4.5.8 Dataset Preprocessing Effect

Based on the fact that DNN-based speech enhancement is a data-driven approach, the preprocessing and manipulation applied to the data before feeding it to the DNN is an important factor to consider. The type and setup of the architecture together with the properties of the used data highly affect the final network output. There are many techniques that can be used to prepare the data for the training process; this experiment investigates three commonly used techniques: sampling, amplitude scaling, and speech and noise mixing. The effect of increasing the sampling frequency from 8 kHz to 16

kHz is experimented, and how the intensity of the background noise affects the training process using 0 dB, -5 dB, 5 dB, and a range of different SNR levels. Finally, an experiment was conducted to show the effect of increasing the number of noise environments to generate the speech and noise mixture for training. These factors were investigated using the four best performing DNN speech enhancement networks: *a*, *d*, *g*, and the modified better performing architecture CNN_{1D}, discussed in subsection 4.5.6. The results of these experiments are given in Table 4.12 and shown in Figure 4.13.

Regarding the effect of the training SNR, training the DNN at 0 dB SNR leads to the best performance concerning all the evaluation metrics at the tested SNR levels (-5 to 20 with a step of 5). However, architecture *a* shows a higher PESQ and STOI score in the case of training the network with high SNR (5 dB), but the other metrics are negatively affected. Therefore, the noise and speech intensity level is an important feature that the DNN looks at in the training process, so it is recommended to work at 0 dB as the default training SNR, or try a range of SNRs and choose the best, depending on the evaluation metric with the highest priority to improve, and the real-time testing conditions.

Concerning the effect of the down-sampling operation, it can be noticed that all architectures output speech with better quality and higher ΔSSNR when trained using 8 kHz audio. Furthermore, the fully-connected based DNNs (MLP_a, $DDAE_d$) perform better when using the 8 kHz sampling frequency with respect to all metrics. However, convolution-based architectures (CNN_{1D}, $CDAE_g$) output speech with a slightly higher intelligibility score and lower distortion when operating in the 16 kHz sampling frequency. It should be mentioned that 8 kHz processing outperforms in terms of the de-noising task; however, when listening to the enhanced audios, although the noise in the enhanced 16 kHz speech is more audible, the quality of the speech signal is better.

In the final experiment, the DNNs were trained with 1,250 noise environments instead of 105. Increasing the number of noise environments has a positive impact on output speech quality and intelligibility. However, the results also show that exposing the network to a larger number of noise environments during the training process may have a negative impact on speech distortion (LSD) and the network's ability to remove noise (ΔSSNR). This is due to increasing the network's generalization ability to a large range of noise environments, which decreases its ability to remove noise for matched conditions. However, this helps the network to better learn clean speech features, and hence output speech with better PESQ and STOI scores.

4.5.9 Effect of Training Target

In this section, several experiments will be presented to show the effect of the training target used on the performance of the implemented architectures. As discussed in Chap-

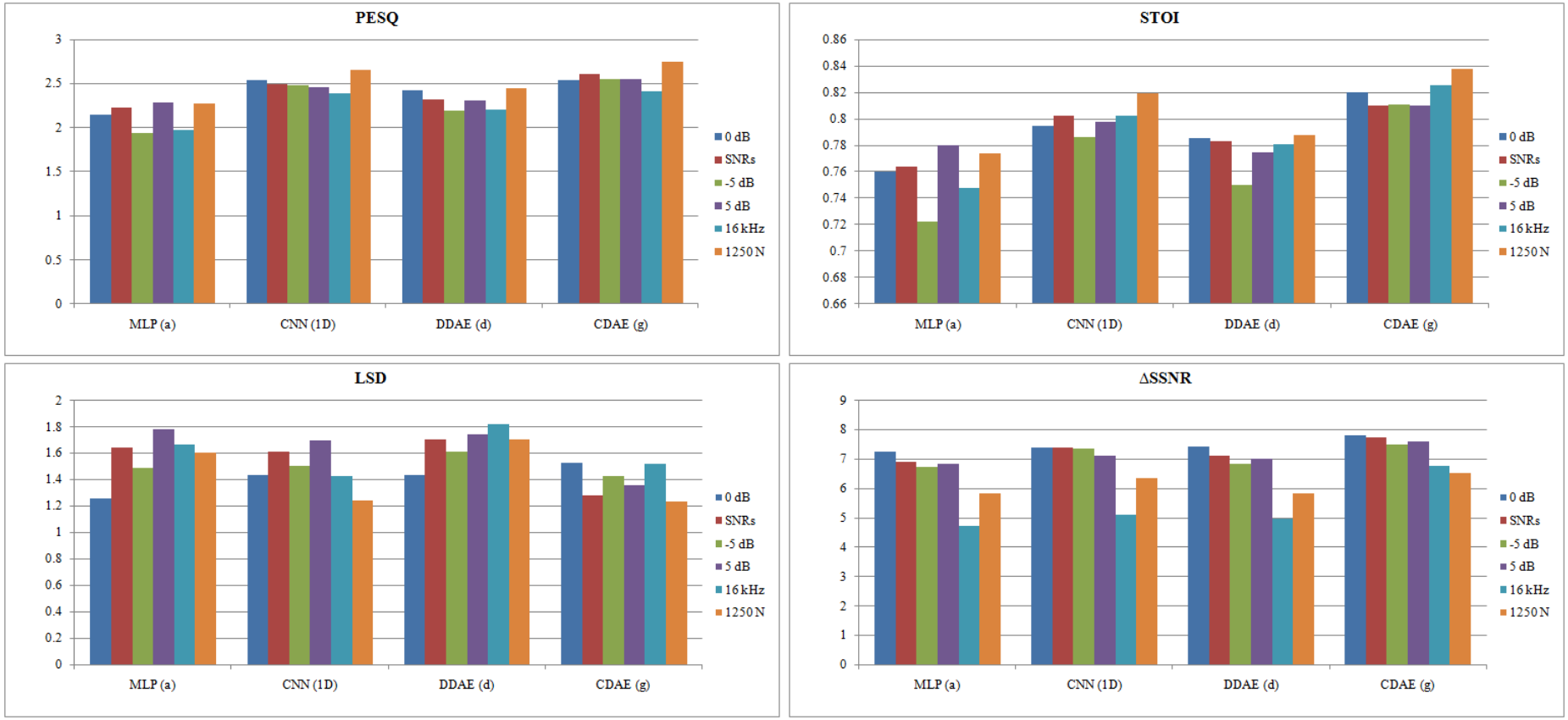


Figure 4.13 PESQ, STOI, LSD, and ΔSSNR results when training the network at 0 dB, -5 dB, 5 dB, and a range of different SNR levels (red bar), when using 16 kHz sampling frequency instead of 8 kHz, and when increasing the number of noise environments from 105 (0 dB blue bar) to 1,250.

Table 4.12 Effect of dataset preprocessing when training the network at 0 dB, -5 dB, 5 dB, and a range of different SNR levels (SNRs column), when using 16 kHz sampling frequency instead of 8 kHz (16 kHz column), and when increasing the number of noise environments from 105 to 1,250 (1250 N column).

Metric		0 dB	SNRs	-5 dB	5 dB	16 kHz	1250 N
MLP_a	**PESQ**	2.147	2.226	1.935	**2.288**	1.968	2.271
	STOI	0.760	0.764	0.722	**0.780**	0.748	0.774
	LSD	**1.261**	1.645	1.491	1.787	1.670	1.605
	ΔSSNR	**7.262**	6.931	6.750	6.854	4.728	5.854
CNN_{1D}	**PESQ**	2.537	2.493	2.486	2.461	2.384	**2.654**
	STOI	0.795	0.802	0.786	0.798	0.802	**0.819**
	LSD	1.438	1.616	1.505	1.696	1.432	**1.246**
	ΔSSNR	7.388	**7.391**	7.382	7.110	5.119	6.347
$DDAE_d$	**PESQ**	2.424	2.324	2.188	2.312	2.200	**2.450**
	STOI	0.785	0.783	0.750	0.775	0.781	**0.788**
	LSD	**1.437**	1.709	1.612	1.742	1.824	1.704
	ΔSSNR	**7.422**	7.124	6.860	7.014	4.984	5.838
$CDAE_g$	**PESQ**	2.543	2.605	2.553	2.545	2.410	**2.741**
	STOI	0.820	0.810	0.811	0.810	0.825	**0.838**
	LSD	1.529	1.279	1.429	1.361	1.518	**1.236**
	ΔSSNR	**7.814**	7.755	7.502	7.626	6.764	6.530

ter 3, mapping or masking targets can be used in order to perform DNN-based speech enhancement, as the process can be seen as a regression or a classification operation. A comparison between these two training target types will be presented in the following subsections, followed by showing the effect of improving both the phase and magnitude spectrograms using the complex spectrogram-based versions of these training targets.

4.5.9.1 Mapping Versus Masking Targets

The results of the experiment that compare the mapping and masking targets are shown in Table 4.13 to 4.16. The commonly used spectrogram mapping approach was used and compared to two commonly used masking targets: IRM and SMM. The results show that the masking-based approach generates speech with better quality (PESQ score) at very high SNRs, 20 dB and 15 dB. Conversely, at low SNR the mapping-based approach outperforms, and this is significant in the $DDAE_d$ architecture. Furthermore, the mapping approach has a lower standard deviation (SD) for the testing SNR levels, which means that this approach is more sustainable.

A clear advantage of masking based targets over mapping targets is that they man-

aged to output more intelligible speech (STOI score) for all architectures. Furthermore, the SMM, specifically, generates enhanced speech with the least distortion; while, the increase in SSNR is relatively high for both approaches.

When comparing the architectures' output with the input noisy speech, the fully connected networks (MLP_a and $DDAE_d$) failed to enhance speech quality at high SNRs (20 dB and 15 dB) in the case of the mapping target. At the same time, both mapping and masking approaches did not improve speech intelligibility at high SNRs for all architectures, except the $CDAE_g$. Finally, these results prove that no specific training target outperforms with respect to all evaluation metrics, and this is due to the high sensitivity of the speech quality evaluation metrics to any change.

Table 4.13 PESQ results for mapping and masking targets (The higher the score, the better the speech quality)

SNR		20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	AVG	SD
Noisy		2.92	2.62	2.32	2.04	1.81	1.60	2.219	0.498
MLP_a	**MAP**	2.41	2.34	2.25	2.16	2.02	1.70	2.147	0.258
	SMM	3.04	2.79	2.57	2.35	2.09	1.75	**2.433**	0.469
	IRM	2.97	2.75	2.54	2.33	2.05	1.70	2.388	0.465
CNN_{1D}	**MAP**	3.09	2.90	2.68	2.46	2.21	1.87	2.537	0.449
	SMM	3.12	2.92	2.71	2.47	2.19	1.87	2.546	0.469
	IRM	3.15	2.94	2.72	2.48	2.21	1.88	**2.564**	0.470
$DDAE_d$	**MAP**	2.82	2.72	2.58	2.41	2.19	1.83	2.424	0.368
	SMM	3.03	2.78	2.55	2.32	2.05	1.73	2.411	0.477
	IRM	3.06	2.80	2.56	2.34	2.08	1.75	**2.430**	0.478
$CDAE_g$	**MAP**	2.93	2.81	2.68	2.52	2.32	2.01	2.543	0.339
	SMM	3.19	3.01	2.83	2.62	2.38	2.04	**2.680**	0.422
	IRM	3.19	3.00	2.80	2.61	2.38	2.03	2.667	0.424

Table 4.14 STOI results for mapping and masking targets (The higher the score, the better the speech intelligibility)

SNR		20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	AVG	SD
Noisy		0.91	0.88	0.83	0.78	0.71	0.64	0.790	0.101
MLP_a	**MAP**	0.82	0.81	0.79	0.77	0.73	0.65	0.760	0.063
	SMM	0.89	0.87	0.83	0.80	0.75	0.68	**0.804**	0.078
	IRM	0.89	0.86	0.83	0.80	0.75	0.68	0.801	0.078
CNN_{1D}	**MAP**	0.88	0.86	0.83	0.79	0.74	0.67	0.795	0.078
	SMM	0.89	0.86	0.83	0.80	0.75	0.67	0.800	0.079
	IRM	0.89	0.87	0.84	0.80	0.76	0.68	**0.808**	0.077
$DDAE_d$	**MAP**	0.85	0.83	0.81	0.79	0.75	0.68	0.785	0.062
	SMM	0.90	0.87	0.83	0.80	0.75	0.68	0.804	0.080
	IRM	0.90	0.88	0.85	0.81	0.76	0.69	**0.814**	0.078
$CDAE_g$	**MAP**	0.89	0.87	0.85	0.82	0.78	0.72	0.820	0.064
	SMM	0.91	0.89	0.86	0.83	0.79	0.72	0.832	0.071
	IRM	0.91	0.89	0.87	0.83	0.79	0.72	**0.834**	0.071

Table 4.15 LSD results for mapping and masking targets (Low value indicates low distortion)

SNR		20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	AVG	SD
Noisy		1.36	1.62	1.92	2.21	2.46	2.62	2.032	0.489
MLP_a	**MAP**	1.05	1.12	1.18	1.22	1.32	1.68	**1.261**	0.225
	SMM	0.96	1.10	1.20	1.30	1.49	1.82	1.312	0.306
	IRM	1.05	1.18	1.26	1.33	1.51	1.85	1.362	0.285
CNN_{1D}	**MAP**	1.09	1.18	1.30	1.44	1.64	1.98	1.438	0.330
	SMM	0.97	1.10	1.25	1.42	1.64	1.95	**1.389**	0.363
	IRM	0.97	1.11	1.27	1.44	1.67	2.00	1.411	0.378
$DDAE_d$	**MAP**	1.23	1.28	1.32	1.40	1.54	1.85	1.437	0.230
	SMM	1.01	1.15	1.26	1.35	1.53	1.82	**1.354**	0.288
	IRM	1.04	1.21	1.34	1.46	1.64	1.93	1.437	0.316
$CDAE_g$	**MAP**	1.37	1.41	1.44	1.51	1.62	1.82	1.529	0.168
	SMM	0.86	0.93	1.02	1.13	1.30	1.54	**1.129**	0.252
	IRM	0.87	0.94	1.03	1.14	1.29	1.53	1.133	0.242

Table 4.16 ΔSSNR results for mapping and masking targets (High values show better noise removal ability)

SNR		20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	AVG	SD
$\mathbf{MLP_a}$	**MAP**	6.44	7.12	7.56	7.77	7.65	7.03	7.262	0.502
	SMM	6.91	7.48	7.80	7.81	7.42	6.68	**7.350**	0.465
	IRM	6.12	6.79	7.22	7.36	7.15	6.72	6.894	0.452
$\mathbf{CNN_{1D}}$	**MAP**	6.98	7.60	7.92	7.94	7.46	6.43	7.388	0.586
	SMM	7.08	7.70	8.03	7.96	7.37	6.49	**7.437**	0.588
	IRM	6.20	6.91	7.38	7.52	7.17	6.53	6.952	0.509
$\mathbf{DDAE_d}$	**MAP**	6.73	7.44	7.85	7.86	7.63	7.02	**7.422**	0.459
	SMM	6.96	7.53	7.85	7.85	7.41	6.80	7.400	0.442
	IRM	6.08	6.76	7.17	7.29	7.06	6.58	6.823	0.448
$\mathbf{CDAE_g}$	**MAP**	7.07	7.77	8.23	8.33	7.98	7.51	**7.814**	0.473
	SMM	7.10	7.77	8.19	8.29	7.93	7.37	7.773	0.463
	IRM	6.22	6.98	7.50	7.73	7.58	7.31	7.222	0.554

Figure 4.14 shows a visual comparison of the spectrograms from a noisy speech signal and the output speech processed by the four implemented DNNs. Each row represents a different architecture, using the two approaches. For the two fully connected architectures, MLP_a and $DDAE_d$ shown in sub figures a and c, it is clear that the mapping approach results in higher denoising ability, but inefficient speech reconstruction, especially with the high frequency components. On the other hand, masking based approaches are better at representing the clean speech signal at the expense of the ability to remove the noise. This explains the reason for the more intelligible speech produced by the masking approach. Consequently, the choice between a masking and mapping target, in this case, is a speech denoising and speech intelligibility tradeoff.

It is also clear in Figure 4.14 that the convolutional based architectures, CNN_{1D} and $CDAE_g$ shown in sub figures b and d, give a nearly similar performance for the masking and mapping approaches, which means that the output speech from this architecture type is not highly affected by the training target used. This introduces architecture design as a factor that when adjusted it can compensate the negative effects of the chosen target. Overall, it can noticed that again the $CDAE_g$ architecture, sub figure (d), is the best performing one, regardless the used training target.

Another important point to mention is that the intensity of the output speech is lower compared to the original clean and noisy speech. This drawback is shown in Figure 4.14, and it is mainly due to the DNN processing applied to the speech in the frequency domain, where the intensity of the speech is affected during the enhancement process,

and does not return to the original intensity after transferring back to the time domain.

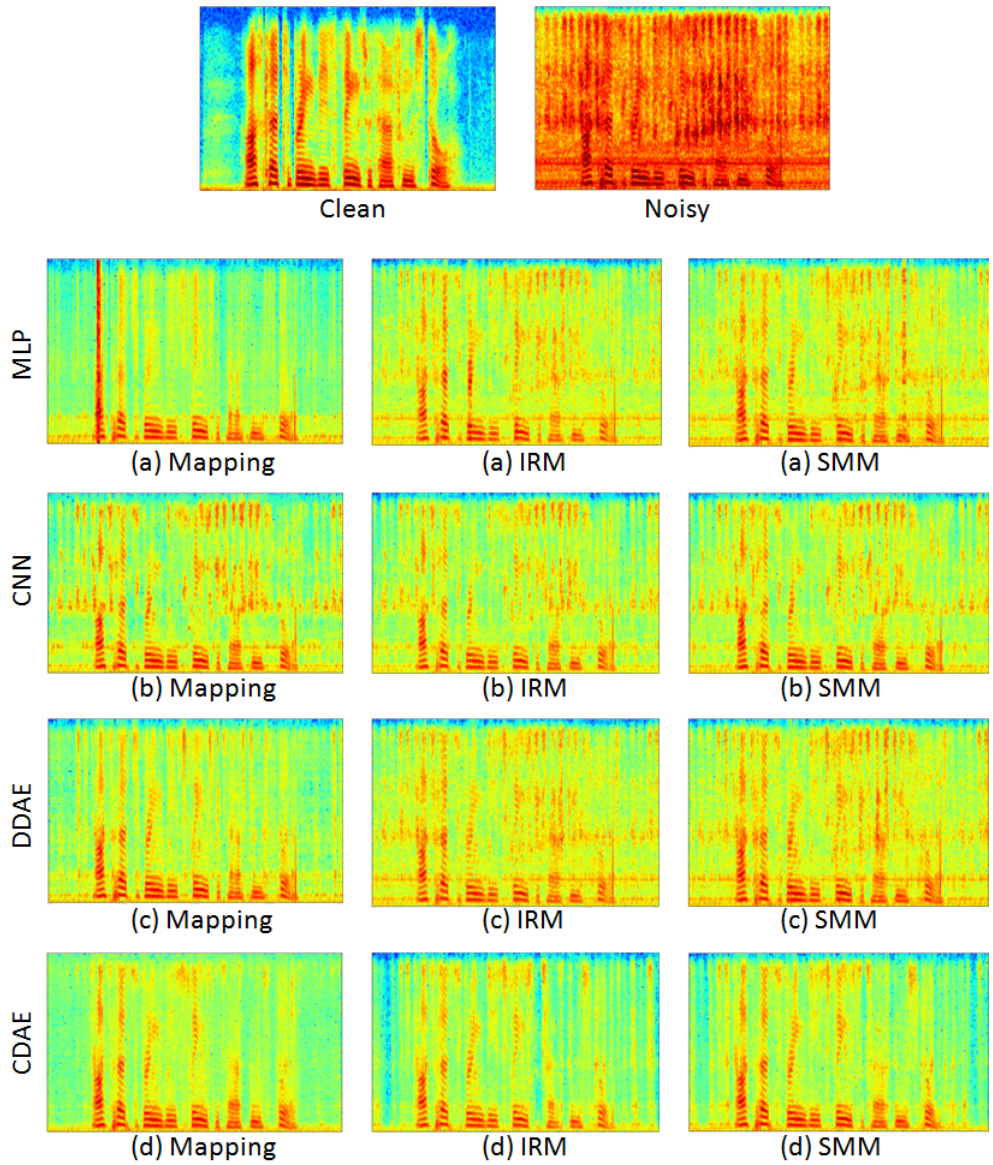


Figure 4.14 Spectrograms of the clean speech, noisy speech with tooth brushing noise at 0dB, and output speech processed by the four DNNs using spectrogram mapping, IRM, and SMM.

4.5.9.2 Targets Generalization Ability

In this experiment, the effect of the chosen training target on network generalization was tested using the *mismatched test set*. 30 minutes of clean speech audios from the LibriSpeech corpus were randomly selected, an unseen dataset during the training process, and these audios were mixed with the same seen and unseen noise environments used in the previous evaluations, shown in Figure 4.3. Table 4.17 and 4.18 show the results of the PESQ and STOI scores for the generated speech from the four architectures. A graphical representation is also shown in Figure 4.15 that compares these results with

the previous results obtained when using the *matched test set*.

Overall, there is a degradation in the overall performance for all architectures, which is expected for this highly mismatched and challenging test data. The autoencoder based architectures ($DDAE_d$ and $CDAE_g$) output speech with better quality in the case of a mapping target, although masking targets showed better performance previously when the networks' generalization ability was not considered. Additionally, there is a significant negative effect on the performance of the $CDAE_g$ architecture when using masking targets (SMM and IRM), and the output speech is unintelligible at low SNR. These results show that architecture design and type restrict the choice of the training targets, and that some architectures may fail to generalize when using a specific target, such as the convolution autoencoder-based architectures.

4.5.9.3 Complex Training Targets

In the previous comparison, the noisy phase was used to reconstruct the time domain signal, assuming that the phase is not highly affected by the noisy environment compared to the magnitude spectrum (Wang and Lim, 1982). However, with the emergence of research that show the importance of enhancing the phase in improving the performance (Shi et al., 2006; Paliwal et al., 2011), some complex spectrogram-based training targets were introduced, enhancing both the noisy magnitude and phase spectrogram during the learning process of the DNN for speech enhancement (Williamson et al., 2016; Ouyang et al., 2019). In this subsection, these complex training targets will be investigated.

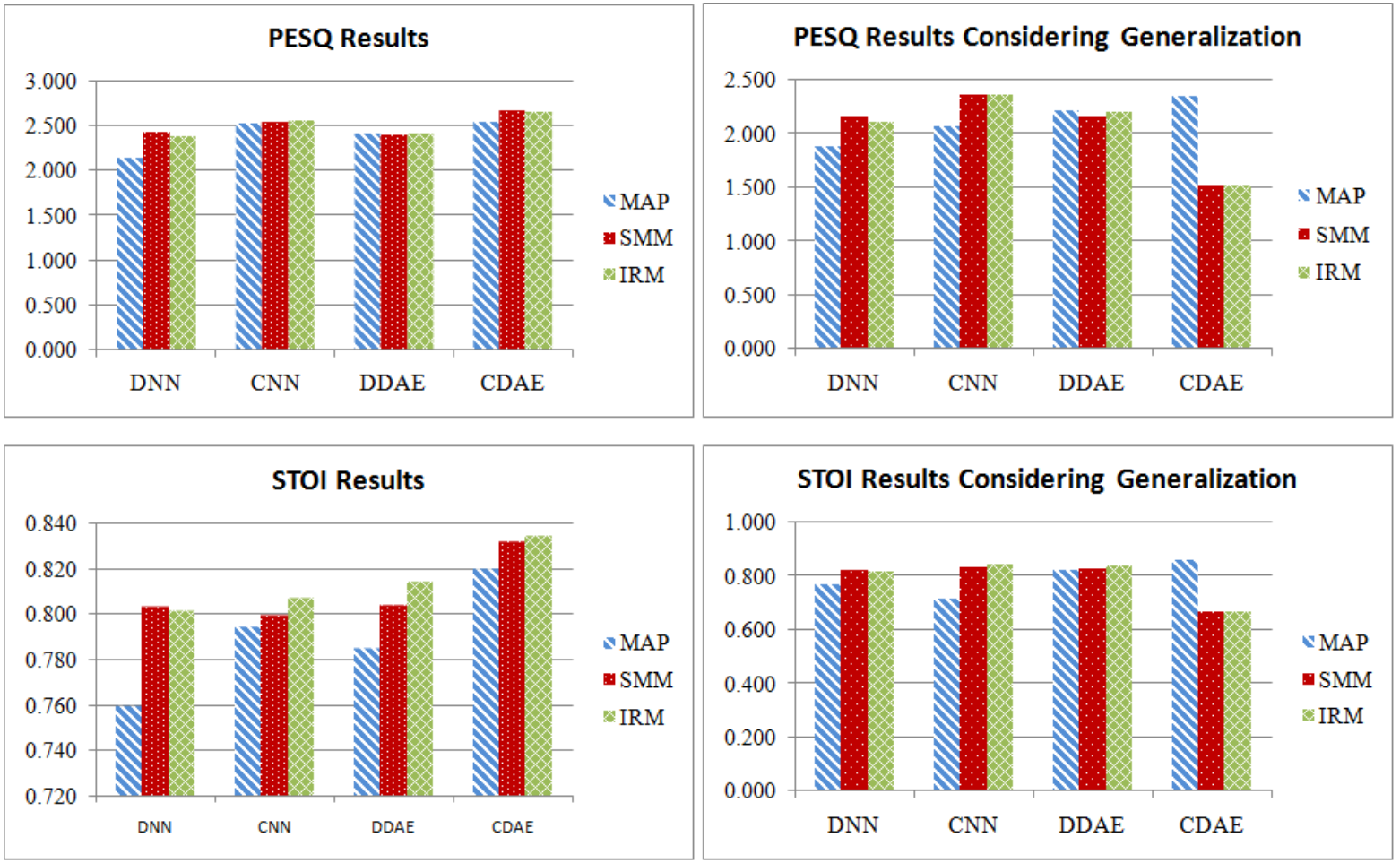


Figure 4.15 PESQ and STOI results for unseen speech from the training dataset and from the unseen LibriSpeech corpus dataset.

Table 4.17 PESQ results considering generalization ability

SNR		20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	AVG	SD
Noisy		2.69	2.36	2.03	1.75	1.51	1.27	1.935	0.532
MLP_a	**MAP**	2.16	2.09	2.02	1.93	1.75	1.37	1.887	0.293
	SMM	2.79	2.54	2.33	2.11	1.80	1.42	**2.166**	0.501
	IRM	2.74	2.52	2.30	2.07	1.73	1.36	2.119	0.513
CNN_{1D}	**MAP**	2.36	2.34	2.28	2.16	1.89	1.43	2.074	0.361
	SMM	3.01	2.80	2.57	2.30	1.95	1.55	**2.364**	0.545
	IRM	3.03	2.80	2.56	2.28	1.94	1.56	2.361	0.548
$DDAE_d$	**MAP**	2.68	2.57	2.42	2.22	1.91	1.48	**2.215**	0.451
	SMM	2.85	2.60	2.34	2.08	1.76	1.40	2.172	0.539
	IRM	2.89	2.63	2.38	2.12	1.79	1.41	2.201	0.545
$CDAE_g$	**MAP**	2.81	2.68	2.54	2.34	2.06	1.68	**2.352**	0.424
	SMM	1.64	1.62	1.60	1.54	1.45	1.30	1.526	0.130
	IRM	1.65	1.62	1.59	1.53	1.45	1.29	1.523	0.133

Table 4.18 STOI results considering generalization ability

SNR		20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB	AVG	SD
Noisy		0.95	0.92	0.87	0.81	0.73	0.65	0.823	0.117
MLP_a	**MAP**	0.84	0.82	0.81	0.79	0.74	0.63	0.772	0.076
	SMM	0.92	0.89	0.86	0.83	0.77	0.68	**0.825**	0.088
	IRM	0.91	0.89	0.86	0.82	0.76	0.67	0.817	0.091
CNN_{1D}	**MAP**	0.77	0.77	0.76	0.73	0.69	0.60	0.719	0.067
	SMM	0.93	0.91	0.88	0.84	0.78	0.69	0.836	0.090
	IRM	0.93	0.92	0.89	0.85	0.79	0.70	**0.844**	0.090
$DDAE_d$	**MAP**	0.89	0.88	0.86	0.83	0.78	0.68	0.821	0.080
	SMM	0.93	0.91	0.87	0.82	0.76	0.68	0.829	0.094
	IRM	0.94	0.92	0.88	0.84	0.78	0.69	**0.842**	0.093
$CDAE_g$	**MAP**	0.94	0.92	0.90	0.86	0.81	0.73	**0.859**	0.080
	SMM	0.70	0.70	0.69	0.67	0.65	0.60	0.667	0.040
	IRM	0.70	0.70	0.69	0.67	0.65	0.59	0.667	0.041

In order to first show the improvement that can be added by enhancing the phase,

Table 4.19 shows how the overall performance of the best performing architecture, $CDAE_g$, will change if the clean phase is used in the reconstruction process. Here, the phase of the original clean speech signal was added to the processed magnitude spectrogram by the DNN at different SNR levels. Magnitude spectrogram mapping was used in this experiment as the more generalized training target, based on the results of previous experiments. It is clear that for all testing SNRs the use of a clean phase has a considerable positive impact on the overall performance, and the improvement becomes very remarkable as the SNR decreases.

Table 4.19 Comparing processed speech using noisy and clean phase

SNR	PESQ		STOI		LSD		ΔSSNR	
	$\Theta_{\mathbf{Noisy}}$	$\Theta_{\mathbf{Clean}}$	$\Theta_{\mathbf{Noisy}}$	$\Theta_{\mathbf{Clean}}$	$\Theta_{\mathbf{Noisy}}$	$\Theta_{\mathbf{Clean}}$	$\Theta_{\mathbf{Noisy}}$	$\Theta_{\mathbf{Clean}}$
20 dB	2.93	**3.05**	0.89	**0.91**	1.37	**1.31**	7.07	**7.23**
15 dB	2.81	**2.94**	0.87	**0.90**	1.41	**1.34**	7.77	**8.01**
10 dB	2.68	**2.82**	0.85	**0.88**	1.44	**1.38**	8.23	**8.58**
5 dB	2.52	**2.67**	0.82	**0.85**	1.51	**1.44**	8.33	**8.86**
0 dB	2.32	**2.47**	0.78	**0.82**	1.62	**1.56**	7.98	**8.74**
-5 dB	2.01	**2.16**	0.72	**0.76**	1.82	**1.77**	7.51	**8.49**
Ave	2.543	**2.687**	0.820	**0.852**	1.529	**1.469**	7.814	**8.318**

Based on the above results, Table 4.19, that show the importance of phase enhancement, a similar comparison as in the previous subsection was conducted between mapping and masking approaches for the same four architectures used before; however, the complex spectrum was used in the training process. Complex spectrogram (cSpec) mapping is performed by concatenating both the real and imaginary parts of the spectrogram and feeding them to the DNNs. On the other hand, cIRM is the complex masking target used in this comparison. Table 4.20 shows the average of the obtained results, which is based on testing the DNNs at six SNRs, from -5 dB to 20 dB with step of 5 using the same *matched test set* used in the previous mapping and masking comparison. Fully connected architectures (MLP_a, $DDAE_d$) were found to perform nearly the same for both complex mapping and masking targets. MLP_a shows a little improvement for cSpec over cIRM; conversely, $DDAE_d$ performs better when cIRM is applied. However, it can be noticed that the fully connected architectures (MLP_a, $DDAE_d$) perform nearly the same for complex mapping and masking target. On the other hand, convolution-based architectures, CNN_{1D} and $CDAE_g$, generated speech with much better overall perception in the case of cSpec mapping.

When comparing these results to the average results in Tables 4.13 to 4.16, where the training is based only on the magnitude spectrogram, it can be concluded that all the architectures give better performance when only the magnitude spectrogram is considered in the training process. An explanation to this is that the learning process becomes more challenging for the DNN to estimate both the clean magnitude and phase, which ends up with worse overall performance. As a conclusion, these results suggest that improving the phase and magnitude should not be done simultaneously, in order to obtain better performance.

Table 4.20 Comparing mapping and masking targets using a complex spectrogram

Metric	**MLP_a**		**CNN_{1D}**		**$DDAE_d$**		**$CDAE_g$**	
	cSpec	**cIRM**	**cSpec**	**cIRM**	**cSpec**	**cIRM**	**cSpec**	**cIRM**
PESQ	2.042	1.959	2.448	2.226	2.135	2.158	2.425	2.258
STOI	0.690	0.688	0.777	0.701	0.692	0.713	0.788	0.735
LSD	1.633	1.439	1.506	1.424	1.702	1.300	1.539	1.223
ΔSSNR	6.753	6.316	7.201	6.461	6.495	6.531	7.506	6.598

4.5.10 Effect of Training Domain

This subsection investigates the effect of the chosen training domain on the performance of DNNs for speech enhancement. The four best performing architectures were trained again but in the time domain, to compare the performance with the previous frequency domain based implementations. The architectures were then tested the using the same *matched test set* of the previous experiments. The results of this experiment are given in Tables 4.21 to 4.25.

The results show that fully connected architectures, MLP_a and $DDAE_d$, are unable to perform speech enhancement in the time domain. The performance of these architectures is very poor when trying to learn the mapping function that maps from noisy to clean speech using time domain speech features. The CNN-based architecture, CNN_{1D}, generates speech with an acceptable overall perception in the time domain; however, the corresponding frequency domain based implementation performs better. While the CNN-based autoencoder architecture, $CDAE_g$, is the only network that outperforms in the time domain in terms of all the evaluation metrics, except speech distortion. The reason for this is the processing nature of this architecture type, which aims to output a similar representation of the input, regardless of its domain, while removing unimportant background noise in the bottleneck layer. This working principle seems to be more efficient in giving an estimate representation to clean speech in the time domain

than the frequency domain. However, the high denoising ability of this implementation results in higher distortion to the processed speech, in comparison to the frequency domain based implementation. Consequently, the choice of the working domain mainly depends on the architecture type and the evaluation metric with the highest importance to improve based on the speech enhancement application.

Another point that these results reveal is the importance of feature extraction in achieving good performance. Although deep learning is a data driven approach, the feature extraction stage plays a crucial role in learning the mapping function that maps noisy to clean speech. That is why frequency domain based implementations always manage to predict clean speech, regardless of the architecture type; while, time domain-based learning is not successful for all architecture types. Regarding the CDAE_{g} architecture, in which the time domain implementation is better, this network performs feature extraction implicitly during the training process. The compression performed by the bottleneck layer results in a nonlinear transformation of the input to another compact form, which acts as unique features that represent the input. For this reason, this architecture type managed to efficiently perform speech enhancement in the time domain, without the need for an additional feature extraction stage.

Table 4.21 PESQ scores for time and frequency domain-based learning (The higher the score, the better the speech quality).

SNR	**Noisy**	**MLP_{a}**		**CNN_{1D}**		**DDAE_{d}**		**CDAE_{g}**	
		Freq.	**Time**	**Freq.**	**Time**	**Freq.**	**Time**	**Freq.**	**Time**
20 dB	2.92	2.41	2.12	3.09	2.53	2.82	1.84	2.93	3.12
15 dB	2.62	2.34	2.12	2.90	2.44	2.72	1.82	2.81	2.97
10 dB	2.32	2.25	2.11	2.68	2.30	2.58	1.75	2.68	2.82
5 dB	2.04	2.16	2.08	2.46	2.13	2.41	1.63	2.52	2.67
0 dB	1.81	2.02	1.72	2.21	1.89	2.19	1.47	2.32	2.49
-5 dB	1.60	1.70	1.55	1.87	1.59	1.83	1.32	2.01	2.24
AVG	2.219	**2.147**	1.949	**2.537**	2.146	**2.424**	1.639	2.543	**2.716**
SD	0.498	0.258	0.250	0.449	0.355	0.368	0.207	0.339	0.322

Table 4.22 STOI scores for time and frequency domain-based learning (The higher the score, the better the speech intelligibility).

SNR	Noisy	MLP_a		CNN_{1D}		$DDAE_d$		$CDAE_g$	
		Freq.	**Time**	**Freq.**	**Time**	**Freq.**	**Time**	**Freq.**	**Time**
20 dB	0.91	0.82	0.52	0.88	0.86	0.85	0.67	0.89	0.93
15 dB	0.88	0.81	0.52	0.86	0.84	0.83	0.67	0.87	0.92
10 dB	0.83	0.79	0.52	0.83	0.79	0.81	0.67	0.85	0.90
5 dB	0.78	0.77	0.52	0.79	0.75	0.79	0.62	0.82	0.87
0 dB	0.71	0.73	0.52	0.74	0.71	0.75	0.52	0.78	0.84
-5 dB	0.64	0.65	0.48	0.67	0.64	0.68	0.47	0.72	0.77
AVG	0.790	**0.760**	0.512	**0.795**	0.765	**0.785**	0.604	0.820	**0.872**
SD	0.101	0.063	0.017	0.078	0.084	0.062	0.088	0.064	0.059

Table 4.23 LSD scores for time and frequency domain-based learning (Low value indicates low distortion).

SNR	Noisy	MLP_a		CNN_{1D}		$DDAE_d$		$CDAE_g$	
		Freq.	**Time**	**Freq.**	**Time**	**Freq.**	**Time**	**Freq.**	**Time**
20 dB	1.36	1.05	1.74	1.09	2.01	1.23	2.39	1.37	1.87
15 dB	1.62	1.12	1.74	1.18	2.03	1.28	2.43	1.41	1.89
10 dB	1.92	1.18	1.76	1.30	2.06	1.32	2.51	1.44	1.91
5 dB	2.21	1.22	1.80	1.44	2.12	1.40	2.64	1.51	1.94
0 dB	2.46	1.32	1.88	1.64	2.25	1.54	2.83	1.62	1.96
-5 dB	2.62	1.68	2.01	1.98	2.45	1.85	2.99	1.82	1.99
AVG	2.032	**1.261**	1.823	**1.438**	2.155	**1.437**	2.631	**1.529**	1.926
SD	0.489	0.225	0.105	0.330	0.170	0.230	0.237	0.168	0.046

Table 4.24 ΔSSNR scores for time and frequency domain-based learning (High values show better noise removal ability)

SNR	MLP_a		CNN_{1D}		$DDAE_d$		$CDAE_g$	
	Freq.	**Time**	**Freq.**	**Time**	**Freq.**	**Time**	**Freq.**	**Time**
20 dB	6.44	0.59	6.98	3.67	6.73	1.98	7.07	6.82
15 dB	7.12	0.79	7.60	3.04	7.44	2.10	7.77	8.53
10 dB	7.56	0.73	7.92	2.53	7.85	2.36	8.23	8.91
5 dB	7.77	1.62	7.94	2.32	7.86	3.33	8.33	8.82
0 dB	7.65	1.41	7.46	2.50	7.63	3.75	7.98	8.25
-5 dB	7.03	1.52	6.43	2.51	7.02	3.07	7.51	7.94
AVG	**7.262**	1.110	**7.388**	2.762	**7.422**	2.764	7.814	**8.212**
SD	0.502	0.457	0.586	0.507	0.459	0.721	0.473	0.771

Table 4.25 Average PESQ, STOI, LSD, and ΔSSNR results for time and frequency domain-based learning

Metric	MLP_a		CNN_{1D}		$DDAE_d$		$CDAE_g$	
	Freq.	**Time**	**Freq.**	**Time**	**Freq.**	**Time**	**Freq.**	**Time**
PESQ	**2.147**	1.949	**2.537**	2.146	**2.424**	1.639	2.543	**2.716**
STOI	**0.760**	0.512	**0.795**	0.765	**0.785**	0.604	0.820	**0.872**
LSD	**1.261**	1.823	**1.438**	2.155	**1.437**	2.631	**1.529**	1.926
ΔSSNR	**7.262**	1.110	**7.388**	2.762	**7.422**	2.764	7.814	**8.212**

4.5.10.1 Networks' Complexity Comparison

Table 4.26 shows the comparison between the used parameters in each implementation and the testing processing time. These results are based on running the algorithm on an NVIDIA Quadro M3000M GPU with clock 1,050 MHz and 160 GB/s memory bandwidth. It is clear that the number of parameters in all the time domain implementations is much higher, which leads to increased model size. Except for the $CDAE_g$, as zero padding is performed to the input frequency feature so as to keep the input size of 2,048, so the network is able to decrease the input through the 8 layers of the encoder. Convolutional-based architectures also have a lower number of parameters than fully connected architectures. This is because of the sparse connections of CNNs, and more

specifically due to the use of 1D convolution in both time and frequency implementations, which leads to a decreased number of parameters.

The processing time is calculated based on processing 224 speech audio files of about 15 minutes duration. The operation was done 6 times, then the average time was taken so as to consider any error caused by processing freezing. All frequency domain implementations take a longer time to process because of the transformation operation. The number of layers is also shown in the Table 4.26. The $CDAE_g$ architecture is the deepest architecture, 49 layers, so this is another possibility why this architecture outperforms in the time domain. Very deep neural networks are proved to be better at extracting more advanced features through the layers (Yu et al., 2013), especially in the case of convolutional-based architectures (Zhang et al., 2017). It is also clear that the depth of the architecture increases the processing time.

Table 4.26 Comparing different networks' parameters: number of parameters (P), processing time (T), and number of layers (L), for frequency (Freq.) and time (Time) domain based implementation of the four best performing DNNs

Metric	**MLP_a**		**CNN_{1D}**		**$DDAE_d$**		**$CDAE_g$**	
	Freq.	**Time**	**Freq.**	**Time**	**Freq.**	**Time**	**Freq.**	**Time**
P(10^6)	**8**	16	**0.2**	0.4	**2**	3	3	3
T(s)	21.5	**11.1**	14.1	**12.8**	15.5	**14.6**	34.5	**24**
Layers	15		10		21		49	

4.5.10.2 Factors Affecting Time Domain Learning

In the previous investigation, fully connected architectures were found to be unable to perform speech enhancement in the time domain, as the output speech has unacceptable overall perception. In this section, some experiments were conducted to show the effect of three factors on the performance of these architectures in the time domain, in an attempt to improve the training process. The three factors used in this investigation are: time frame size, architecture depth, and training dataset size. The outcome of these experiments is represented in Table 4.27, and shown in Figure 4.16. These are the average results of the six SNR levels for the previously used *matched test set*. The original output speech scores based on the first time domain experiment, Table 4.25, is also shown in Figure 4.16 for comparison.

Time Frame Size

In this experiment, a smaller frame was used of size 256 instead of the previously used 2,048 frame size. In Table 4.27, the PESQ score was found to improve by using a

smaller time frame for both the MLP_a and the $DDAE_d$ architectures; however, they still fail to give good performance for frequency-based implementations. Moreover, a significant degradation is shown in the intelligibility of the output speech (STOI score) from the $DDAE_d$ network. This is due to the compression process applied in this architecture type, which may result in severe distortion and inaccurate speech reconstruction when using an input time frame with small size, especially as this implementation lacks the use of skip connections that help in retaining the information as the processing proceeds deeper from the encoder to the decoder network.

Architecture Depth

Due to the fact that the fully connected implementations are shallow compared to the other implemented networks, an investigation was carried out to show the effect of increasing the depth of these architectures, as this might help in improving the performance in the time domain. Two more layers were added for the MLP_a architecture, for the network to have 5 layers instead of 3. Two more layers were also added to each of the encoder and decoder networks for the $DDAE_d$ architecture in order to have 4 layers in each of them. The number of hidden units were decreased through the encoder layers, 2,049, 1,024, 500, 250, and 150 units were used; and increased in reverse order through the decoder layers.

The results in Table 4.27 show that increasing the depth of the architecture has a positive impact on the overall network performance in the time domain, as both the PESQ and STOI scores are higher for the deeper version of the networks (2.026 and 2.262 PESQ scores for MLP_a and $DDAE_d$, respectively, and 0.565 and 0.622 STOI scores for MLP_a and $DDAE_d$, respectively. The improvement is very significant for the $DDAE_d$ network, which shows the importance of the depth for this architecture type; moreover, it is worth mentioning here that the addition of skip connections to the $DDAE_d$ is proven to result in further performance improvement, because of their ability to prevent information loss in deep architectures (Tu and Zhang, 2017).

Training Dataset Size

Due to the absence of the feature extraction stage in time domain learning, the dataset size may play a more important role in the training process compared to the frequency domain approach, as the learning process is mainly based on the training data in this case. As a result, this subsection investigates the effect of doubling the dataset size on the performance of fully connected architectures, MLP_a and $DDAE_d$, by retraining them using 10 hours of speech instead of the previously used 5 hours. The outcome of this experiment, given in Table 4.27, shows that speech intelligibility (STOI score) improves for both networks when using more data in the training process; however, the MLP_a architecture generates speech with lower quality (PESQ score). An explanation of this

lower PESQ score is that the network may overfit to the larger speech training data, considering that the number of the noise environments used is not high, and this will decrease the network denoising ability to unseen test data. This negative effect was not detected in the $DDAE_d$ architecture, due to the compression performed in this network type using decreased number of hidden units through the encoder hidden layers, which help the network to overcome overfitting.

Although some of the investigated factors result in some improvement in the performance of fully connected architectures in the time domain, the output speech from these networks is still of relatively low overall quality in comparison to frequency domain based implementations. Consequently, there is a need for a remarkable change in the architecture design, or the addition of techniques that will help in audio reconstruction, for these architectures to be able to perform speech enhancement in the time domain.

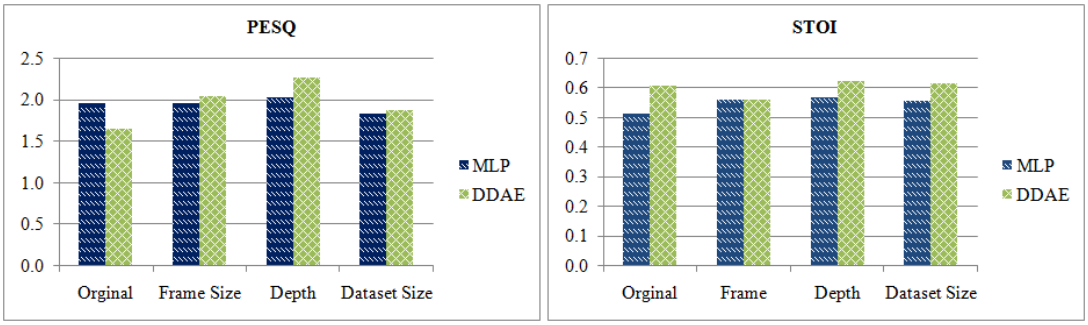


Figure 4.16 Factors affecting time domain-based learning: using smaller frame size of 256 instead of 2,048 (Frame Size column), increasing the depth of the networks (Depth column), and doubling the size of the training dataset (Dataset size column)

Table 4.27 Factors affecting time domain-based learning: using smaller frame size of 256 instead of 2,048 (Frame Size column), increasing the depth of the networks (Depth column), and doubling the size of the training dataset (Dataset size column)

Metric	**MLP_a**			**$DDAE_d$**		
	Frame Size	**Depth**	**Dataset size**	**Frame Size**	**Depth**	**Dataset size**
PESQ	1.956	2.026	1.831	2.040	2.262	1.871
STOI	0.557	0.565	0.556	0.560	0.622	0.614

4.6 Conclusion

In this chapter, an investigation has been carried out to evaluate the performance of three well-established DNNs for speech enhancement: MLP, CNN, DAE. Seven best

performing models were re-implemented, belonging to these three main categories, and their performance was analysed using the well-known speech enhancement evaluation metrics and through spectrogram representation. Based on the results of the conducted experiments, the following conclusions were reached.

Concerning the objective and subjective evaluation of the seven architectures using the speech quality measures, it was found that the deep CDAE is the best performing architecture. However, due to the lossy nature of this architecture type, most of the real listeners preferred the enhanced speech from the FCNN architecture, although it is more noisy than the output from the deep CDAE. Moreover, both subjective and objective evaluations show that the shallow versions of the CDAEs have less denoising ability than the basic CNN and FCNN, which means that increasing the architecture depth is essential for CDAE networks to be able to efficiently and effectively perform speech enhancement. Similarly, the DDAE was proven to perform better than the basic fully connected MLP. The output spectrograms from the seven model also supports the objective and subjective scores.

Regarding the effect of network-related hyperparameters, activation functions comparison of CNN architectures shows that the PReLU is the best for speech enhancement among ReLU, LReLU, and ELU activations. The application of 1D convolution instead of 2D convolution was proven to remarkably improve the performance of CNN networks for speech enhancement. Additionally, architecture depth was proven to be the main factor affecting the performance of the CDAE for speech enhancement.

Spectrograms of the internal layers of the CNN architecture with ReLU activation showed that CNNs deal with the speech enhancement task by applying filters with different functionalities. Some are de-noising, while others extract different speech features, such as the high and low-frequency components. Additionally, some filters were found to keep the original noisy speech, and they are supposed to help in the reconstruction of the estimated clean speech and avoid the loss of important information. However, the dying ReLU problem was detected in this case, which results in turning off many of these filters, and the use of PReLU instead was shown to solve this issue.

In real scenarios, speakers raise their voice in noisy environments, this known as the Lombard phenomenon. Analysis of the Lombard effect on the performance of DNNs for speech enhancement shows that the DNNs not only managed to deal with this mismatched pattern, but also show improved performance in comparison to testing the network using plain speech with no Lombard effect. Consequently, the learned speech features enable the DNN to be robust to the Lombard speech.

Data manipulation through different preprocessing techniques was proven to improve the learning process of DNNs for speech enhancement. Using noisy speech utterances at 0 dB SNR during training was shown to be the default choice, because using the same speech and noise power level makes them indistinguishable to the DNN by

level during training, which force the DNN to learn something more fundamental and guides it to learn the mapping function that maps noisy speech to clean speech. Audio sampling was proven to affect the network's denoising and reconstruction abilities, as downsampling to 8 kHz improves the denoising process, because it keeps only the essential speech frequency bands, resulting in more noise removal. However, this negatively impacts the overall quality of the processed speech, as the 16 kHz enhanced speech is of better quality, although more noise is present in the output speech.

A comparison of the training targets reveals that mapping targets are less affected by SNR changes, as making targets show high variance. For fully connected architectures, mapping targets show better denoising ability; while masking targets were proven to have better reconstruction, leading to more intelligible processed speech. On the other hand, convolution-based architectures proved to be less affected by the training target when tested using unseen noise environments and unseen speech from the same training dataset. However, when considering the generalization ability of the networks using a different dataset from the one used in the training process, the results show that masking targets are not recommended for autoencoder architectures, because there is a significant performance degradation in the case of using masking targets, especially for the CDAE architecture.

When investigating different learning domains, fully connected architectures experience a significant degradation in the performance when the learning process is performed in the time domain, and the networks failed to output speech with acceptable quality and intelligibility. Although changing the depth, frame size, and dataset size was shown to improve the overall performance of fully connected architectures learning in the time domain, a careful design and extra techniques are needed for this type of DNN when operating in the time domain, in order to achieve a good performance. Conversely, convolution-based architectures managed to perform speech enhancement in the time domain; moreover, the CDAE gives better performance in the time domain than that of the frequency domain-based implementation.

The next chapter will present a new deep learning based speech enhancement DNN that outperforms SOTA architectures in the literature. Additionally, the chapter proposes a new two-stage deep learning speech enhancement approach, which was proven to improve the performance.

CHAPTER 5

A Two-Stage Speech Enhancement Architecture using Time and Frequency Domain Approach

5.1 Introduction

This chapter presents a deep CDAE based speech enhancement architecture, which aims to compromise between speech denoising and speech distortion. The developed architecture is an asymmetric CDAE with several strided 1D convolution layers and dilated convolution blocks. The encoder network is designed to be deeper than the decoder network, in order to improve the performed feature extraction process through the hidden layers of the encoder while minimizing architecture complexity by passing this information to the decoder using skip connections, instead of adding more layers to the decoder network. In order to improve speech reconstruction and minimize speech distortion, a two-stage deep learning approach is proposed for speech enhancement that takes advantage of both the frequency and time domain speech features. The first stage applies speech denoising by running the developed architecture in the frequency domain using magnitude spectrogram mapping as a training target. Due to the deep nature of the architecture, the background noise is aggressively removed by the first stage; however, the output speech experiences high distortion. The second stage deals with this distortion issue by trying to reconstruct the removed speech from the first stage using time domain speech features. Moreover, the noisy phase is enhanced in the second stage, which leads to further denoising.

5.1.1 Relation to Prior Work

The developed architecture is an improved and deeper version of the U-Net (Ronneberger et al., 2015; Jansson et al., 2017) and its improved version Wave-U-Net (Stoller et al., 2018), using asymmetric encoder/decoder design, which improves performance while keeping complexity to a minimum. Compared to the U-Net and Wave-U-Net that both have 20 million parameters, the proposed architecture has lower number of parameters, only 6.3 million parameter. The developed architecture takes advantage of strided and dilated convolution; moreover, connections were added between encoder

layers to combine fine and cruse features, extracted during the training process. Furthermore, PReLU activations were used in both the encoder and decoder networks instead of LReLU activations used in previous work, because PReLU was found to improve the performance. The proposed architecture also uses shortcut between noisy input and enhanced output, which was found to decrease speech distortion during training, compared to previous work. The comparison between the performance of the developed architecture in this PhD work and SOTA speech enhancement models, including the Wave-U-Net, will be presented later in this chapter in Table 5.1.

5.1.2 Research Contributions

The work in this chapter makes the following research contributions:

- develops a new asymmetric CDAE based speech enhancement network that outperforms other models in the literature, and
- proposes a two-stage deep learning approach for speech enhancement that applies speech denoising while minimizing speech distortion.

In the following sections, details about the developed speech enhancement architecture will be presented in Section 5.2. Section 5.3 provides information about the data used and the setup to train and test the architecture. In Section 5.4, a comparison is presented between the frequency and time domain implementation of the proposed architecture. The proposed two-stage speech enhancement approach will be discussed in Section 5.5 and the results achieved and comparison to baselines will be presented in Section 5.6. Finally, Section 5.7 concludes this chapter.

5.2 The Proposed Speech Enhancement Architecture

The developed architecture is a fully 1D CDAE-based implementation, see Figure 5.1. The network accepts noisy speech time frame of size 2,048 as an input, which is denoised by the compression applied through the hidden layers of the encoder network, until it reaches a size of 8 in the middle bottleneck layer. Signal reconstruction is then performed by the decoder network, to decompress the enhanced speech to its original size. Based on the fact that the developed architecture is very deep, the use of skip connections between the encoder and decoder networks is essential, so as to prevent information loss when processing the speech signal through the hidden layers (Rethage et al., 2018). These connections are represented by the red lines in Figure 5.1, and they send the features extracted by the encoder network to the decoder network. Additionally, the unprocessed noisy speech input is fed to the last layer of the decoder, to help the network in the reconstruction process. This shortcut between the input and output was

found to decrease distortion and leads to better network generalization. This shortcut connection is represented by the blue line in Figure 5.1.

The encoder network has several 1D strided convolution layers and strided-dilated causal convolution blocks. Each 1D strided convolution layer has stride size 2, kernel size of 9, and PReLU activation. This layer is then followed by a strided-dilated causal convolution block, which consists of 5 dilated 1D causal convolution layers of increasing dilation rates; 1, 2, 4, 8, and 16 dilation rates used, and a final PReLU activation. The combination of strided convolution and dilated convolution techniques is proven to enhance the denoising process and improve the overall speech perception (Pandey and Wang, 2020*a*), as it allows exponential expansion of the receptive field, which decreases speech distortion without increasing the network's complexity (Yu and Koltun, 2016). We also used increasing kernel sizes as the dilation rate increases to decrease sparsity. The strided-dilated convolution block ends with a concatenation layer to combine both the fine and coarse features extracted by these techniques. The noisy speech is processed by the encoder network, to form a compact form representing the predicted clean speech signal in the bottleneck layer.

Speech reconstruction is performed by the decoder network through several 1D deconvolution layers of upsampling size 2 and PReLU activation. Each layer takes a concatenation of two inputs: the output of the previous layer and the output of the corresponding concatenation layer in the encoder network, received by the skip connections. The final convolution layer of the decoder is responsible for predicting the enhanced speech, and it has a kernel size of 7 and linear activation function. The input to this layer is a concatenation of the output of the previous layer and the original noisy speech input.

The full architecture has about 6.3 million parameters; however, the encoder is deeper than the decoder network, because the strided-dilated causal convolution blocks were not applied in the decoder network. The reason for this is that no significant improvement in the performance was detected when repeating these blocks in the decoder network, as the information gained by the skip connections provides the necessary information for the reconstruction process. Therefore, these blocks were removed from the decoder, to decrease network complexity and processing time. This results in having an encoder with 74 layers, making a total of 4.2 million parameters; while the decoder has 36 layers, making a total of 2.1 million parameters. Hence, we named this architecture Deep Encoder - Convolutional Autoencoder DEnoiser (DE-CADE), and this name will refer to this architecture throughout the rest of the thesis.

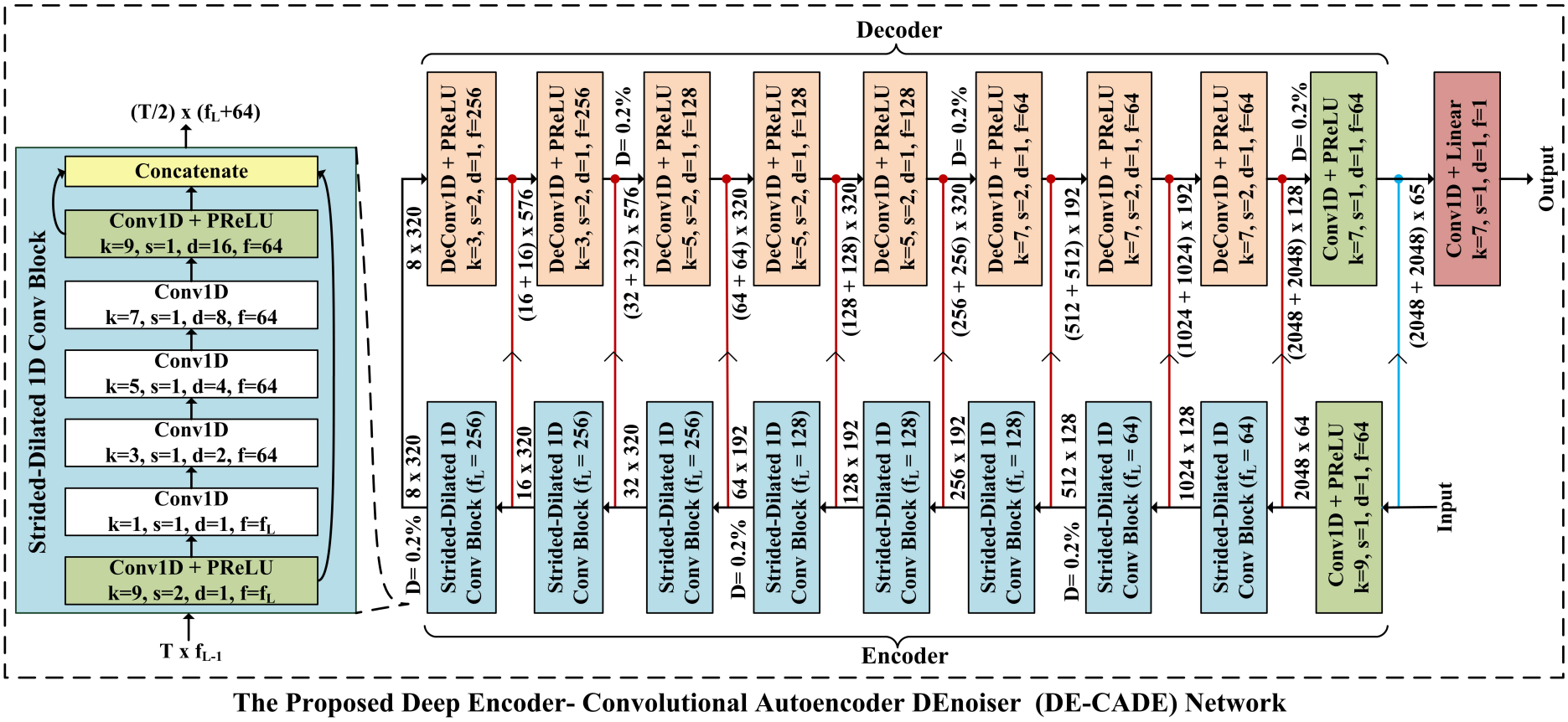


Figure 5.1 The proposed Deep Encoder - Convolutional Autoencoder DEnoiser (DE-CADE) speech enhancement architecture; k, d, f, and L represent kernel size, dilation rate, number of convolution channels and layer number respectively; s represents stride size in the encoder, and upsampling size in the decoder. T is the time samples. The red lines represent skip connections and the blue line shows the shortcut between the input and output.

5.3 Experimental Setup

In this section, the setup used to train and test the proposed DNN will be demonstrated. This section presents the speech and noise datasets used, preprocessing techniques, and the learning hyperparameters. These will be discussed separately in the following subsections.

5.3.1 Datasets

The DE-CADE architecture was trained and tested using two datasets for speech enhancement: a small-scale dataset for baseline comparison, and a large-scale dataset for measuring overall network performance. These two datasets are described in the following subsections.

5.3.1.1 Small-Scale Dataset

In the speech enhancement field, it is common to first train and test the architecture using a benchmark dataset that is used to compare with the SOTA models in the literature, and then to show the network's performance when a using large dataset. The benchmark dataset used to verify and compare our architecture performance is the Valentini dataset (Valentini-Botinhao et al., 2017*b*). This dataset is a subset of the Voice Bank corpus (Veaux et al., 2013), with a total of 30 speakers, 28 for training and 2 for testing. The speakers are native English, reading about 400 English sentences.

The training set contains noisy speech audios created by mixing the training speech utterances with 10 noise environments: 8 from the Diverse Environments Multichannel Acoustic Noise Database (DEMAND) dataset and two artificial noises, at four SNRs: 0, 5, 10 and 15dB, to make 11,572 training samples, about 18 hours of noisy speech data. The proposed DE-CADE architecture was trained using 90% of this noisy data, and 10% was used for validation during training. The Valentini test set is formed by corrupting the test speech utterances with 5 unseen noise environments from the DEMAND dataset, to make 824 test samples. This data was used to test and compare the architecture performance against SOTA speech enhancement networks. This training and testing data will be denoted by *"Small-Scale Train Set"* and *"Small-Scale Test Set"*, respectively.

5.3.1.2 Large-Scale Dataset

The architecture was also trained using a very large noisy speech dataset of 1,000 hours. The clean speech data includes 800 hours of English speech, and 200 hours of an additional 175 languages (Topcoder, 2017). The 800 hours of English speech was collected from the Microsoft Deep Noise Suppression (DNS) challenge dataset (Reddy et al.,

2020), the CSTR VCTK Corpus (Yamagishi et al., 2019) and the Reverberant speech dataset (Valentini-Botinhao et al., 2017*a*). Together make a total of 267,841 different speech utterances. The noise environments were taken from the DNS noise dataset, which is about 181 hours of noise data, and makes a total of 60,000 different noise clips (Reddy et al., 2020). The speech and noise datasets were divided into 90% for training and 10% for validation. The noisy speech training and validation data was created through random mixing of the clean speech utterances with the noise environments at a wide range of SNRs from -5 to 15 with a step of 1. This training data will be denoted by *"Large-Scale Train Set"*

In order to test architecture performance in challenging conditions that were not previously seen in the training process, a mismatched noisy test data was created using speech utterances from the Librispeech corpus (Panayotov et al., 2015). 200 randomly selected speech audio files were corrupted using 20 unseen noise environments, shown in Figure 5.2. The selected speech utterances are for 20 male and 20 female speakers, not seen by the network during training. The 20 noise environments (N1:N20 in Figure 5.2) were taken from 100 Nonspeech Environmental Sounds (Hu, 2014): 9 crowd noise, an AWGN, 2 human yawn noises, a human cry, a shower, tooth brushing, 2 footsteps, a door moving, and 2 phone dialling. Six test SNRs were considered to create the mixture, from -5 dB to 20 dB with a step of 5 dB. This test data will be denoted by *"Large-Scale Mismatched Test Set"*. We also mixed these speech files with unseen Babble, Factory, Engine, HF radio channel and Operating Room noises from the NOISEX-92 dataset (Varga and Steeneken, 1993), to perform the analysis that will be presented in Section 5.4.

To assess the network's generalization, the architecture was also tested using a matched test data that is seen during the training process. This data was used to compare the network's performance for seen and unseen noisy speech, to evaluate the network's generalization ability. We used 200 speech audios from the DNS dataset, seen in the training, and of similar length as the mismatched Librispeech speech audios. These audios were corrupted with 20 seen noise environments, randomly selected from the training DNS noise dataset. These noises include: church bell, sweeping sound, motorcycle, train, music, cry, water, crowd, wind, sea waves, siren, hummer, kitchen machine, piano, and birds; spectrograms of these matched noise environments are presented in Figure 5.3. We mixed them at the same 6 SNRs of the mismatched test set, to obtain similar conditions. This test data will be denoted by *"Large-Scale Matched Test Set"*.

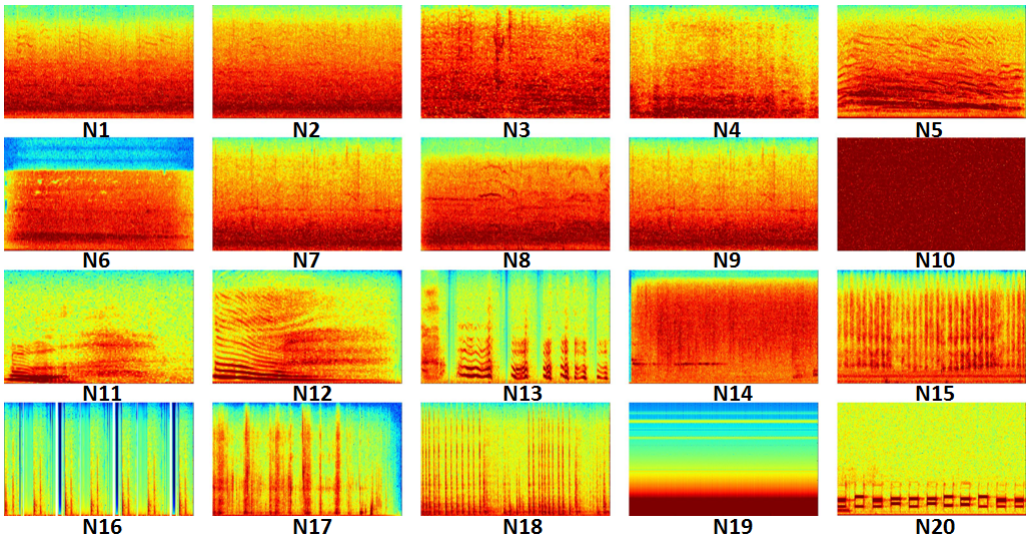


Figure 5.2 The 20 mismatched noise environments used in the testing process. N1-N9: crowd noise; N10: AWGN; N11-N12: yawn sound; N13: Cry; N14: Shower; N15: Toothbrushing; N16-N17: Footsteps; N18: Door moving; N19-N20: Phone dialing

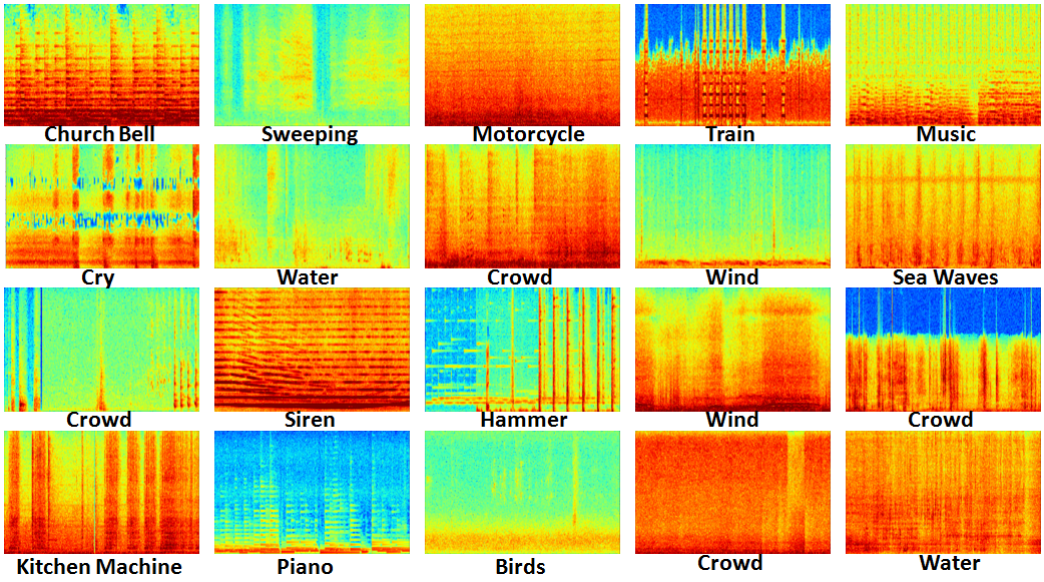


Figure 5.3 The 20 matched noise environments used in the testing process.

5.3.2 Data Preprocessing

The input noisy audio to DE-CADE is resampled to 16 kHz sampling frequency, and it is normalized to zero mean and unit variance. This sampling frequency was used as the common one for speech enhancement, in order to be able to compare DE-CADE with other speech enhancement models in the literature, as the baseline comparison is based on 16 kHz sampling frequency.

The architecture was trained twice: once in the frequency and once in the time domain, because CDAEs shows good performance in both domain, so the best practice is

to evaluate DE-CADE performance in both time and frequency, and then use the better implementation based on the scores of the speech enhancement evaluation metrics. On the one hand, T-F features were extracted for frequency domain-based implementation. The STFT was performed on the noisy speech audio, using a Hamming window with time frame of size 256 and 50% overlap. Magnitude spectrogram mapping is the used training target, where the noisy phase was not used in the processing. The noisy phase was retained and added to the final predicted clean magnitude spectrogram, and then transforming of the signal back to the time domain was performed using Inverse Short Time Fourier Transform (ISTFT). This implementation will be denoted as DE-CADE(F).

On the other hand, a Hamming window with time frames of size 2,048 and 50% overlap was the only extracted features for the time domain based implementation, and time frame mapping is the used training target in this case. The traditional overlap-add method was applied to the enhanced time frames (Griffin and Lim, 1984) to reconstruct the speech utterance. This implementation will be denoted as DE-CADE(T)

5.3.3 Learning Hyperparameters

For both time and frequency domain implementations, the DE-CADE was developed and trained using the Keras framework with Tensorflow backend. The MMSE is the loss function used with the Adam optimizer, learning rate = 0.0001, β_1 = 0.1, β_2 = 0.999. We used a training batch size = 2, and the networks were trained until convergence for 50 epochs, and then the best weights were taken based on the validation data.

5.4 Time Versus Frequency Domain Learning

When training the DE-CADE architecture in the frequency domain, DE-CADE(F), it shows great ability in removing background noise. However, the enhanced speech is highly distorted, especially the high frequency components and at low SNR levels. While processing the architecture in the time domain, DE-CADE(T), shows an opposite effect, where the denoising ability of the network is less than that of the frequency domain-based implementation; however, much better speech reconstruction was found in this case. This can be proven using the Cbak and LSD speech evaluation metrics, shown in Figure 5.4, and the spectrogram representation of the output enhanced speech from each implementation, shown in Figure 5.5.

In Figure 5.4, DE-CADE(F) shows much better Cbak scores, which proves the better denoisng ability of this implementation. At a very low SNR level, such as -5 dB, the time and frequency domain based implementations give nearly the same Cbak scores. An explanation to this is that the negative effect of using the noisy phase in the frequency domain based implementation becomes very clear at a very low SNR, and this

degrades the network's denoising ability. On the other hand, DE-CADE(T) outperforms in terms of the LSD, which shows the better speech reconstruction for this implementation, especially at low SNRs; -5, 0, 5 dB, where aggressive noise removal of the frequency network results in high distortion.

The trade-off between speech denoising and reconstruction can also be justified using spectrograms, shown in Figure 5.5, which represent clean and noisy speech at three SNRs: -5 dB crowd noise, 0 dB tooth brushing noise and 5 dB shower noise, and their corresponding estimated output from DE-CADE(F) and DE-CADE(T). It is clear that at all SNR levels, DE-CADE(F) can effectively remove background noise. However, the output speech experiences high distortion due to the spectrum representation, which gives more attention to the fundamental frequencies when reconstructing the estimated speech. On the other hand, DE-CADE(T) shows less denoising ability, but with better speech reconstruction, especially for the high-frequency components.

5.5 The Proposed Two-Stage Speech Enhancement Approach

Based on the results of the time versus frequency implementation analysis from the previous section, a two-stage speech enhancement approach will be presented in this section that combines the advantages of both time and frequency domain learning. An illustration of this approach is provided in Figure 5.6. In this approach, the noisy speech is first processed by a frequency domain based DE-CADE network, DE-CADE(F); afterwards, the output from this network along with the original noisy speech are fed to second stage DE-CADE running in the time domain, DE-CADE(T). The first stage DE-CADE(F) performs magnitude spectrogram denoising, due to the greater denoising ability of frequency domain based implementations. While, the second stage DE-CADE(F-T) mainly helps in speech reconstruction by minimizing the speech distortion caused by the noise removal processing of the first stage; additionally, phase enhancement is considered in this second stage, which results in further denoising. Mathematical analysis of this two-stage approach will be provided in the following subsection.

5.5.1 Problem Definition

The input noisy speech to the first stage DE-CADE can be represented as follows:

$$y(m) = s(m) + n(m), \tag{5.1}$$

where y represents the noisy speech, s and n are the speech and additive noise signals, respectively, and $\{y, s, n\} \in \mathrm{R}^{M \times 1}$, where M is the total number of samples in the signals, and m is the time sample index. In the case of having additional reverberate

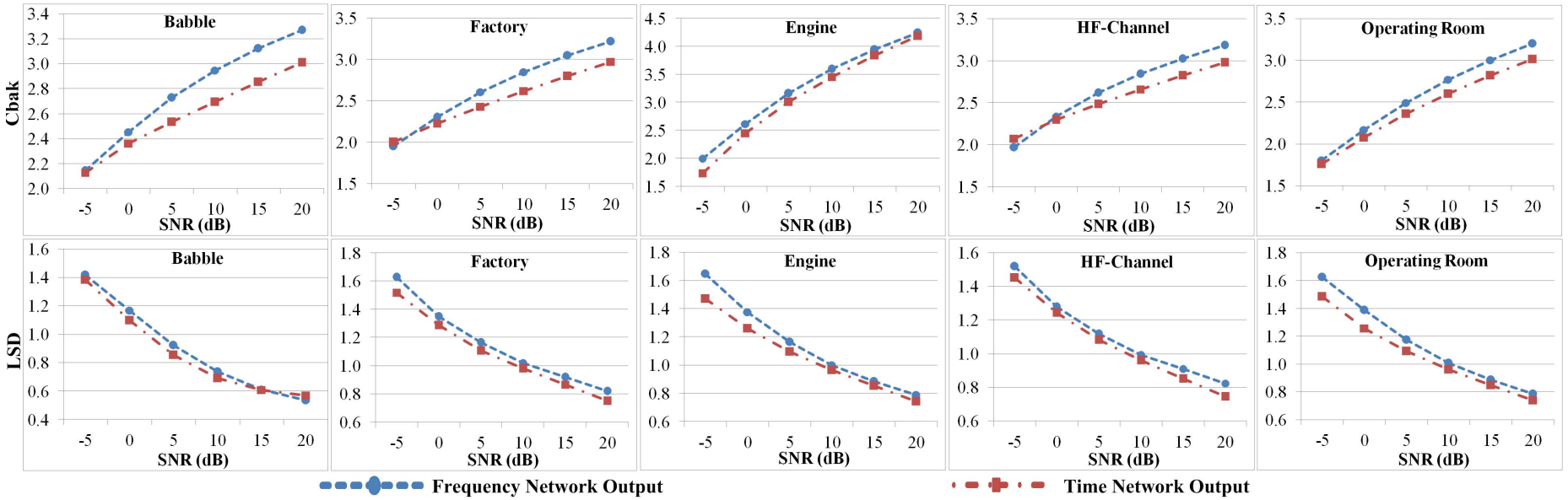


Figure 5.4 The Cbak and LSD results for the proposed network, DE-CADE, when operating in the frequency and time domain, tested on mismatched babble, factory, engine, HF-channel, and operating room noises.

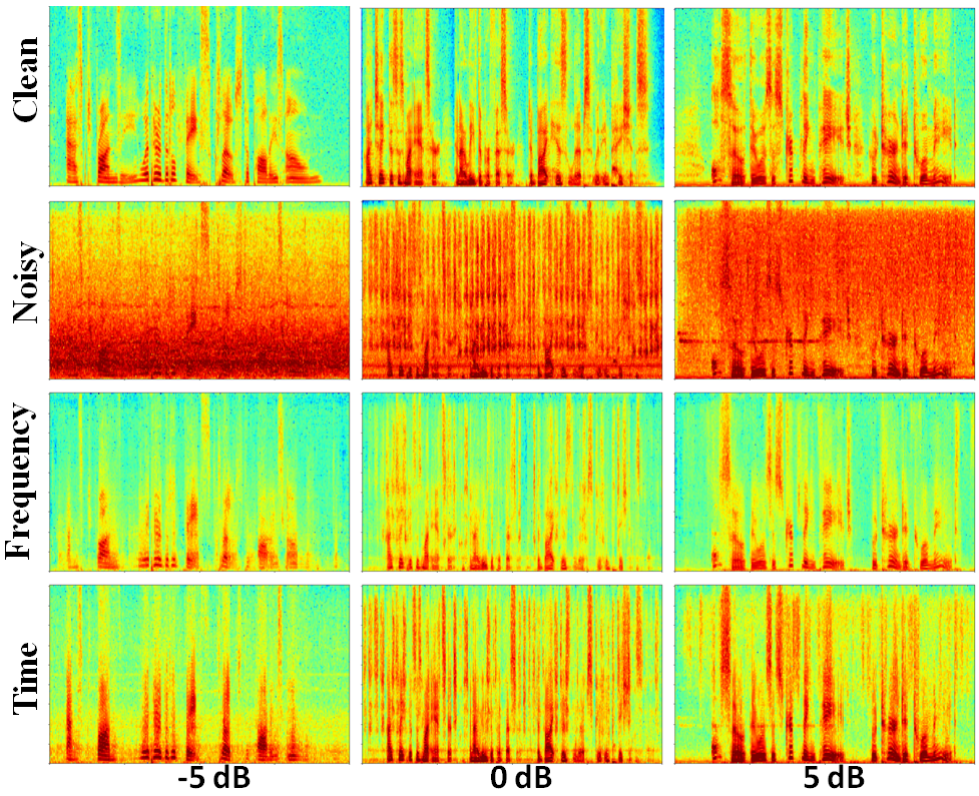


Figure 5.5 The spectrograms of the clean, noisy, and estimated speech from the frequency and time domain DE-CADE at -5 dB crowd noise, 0 dB tooth brushing noise and 5 dB shower noise.

noise in the noisy speech signal, this equation can be redefined as follows:

$$y(m) = x(m) + n(m), \tag{5.2}$$

where,

$$x(m) = s(m) * r(m) = \sum_{j=0}^{M-1} r[j]s[m-j], \tag{5.3}$$

where $*$ denotes the convolution operator, *x* denotes the reverberant speech, *r* represents the Room Impulse Response (RIR), *j* is the discrete RIR sample, and *m* is the sample point of the discrete signals, *x* and *s*.

As discussed in Chapter 4, reverberation is a special noise type that needs additional processing different from denoising. For this reason, dereverberation is currently considered as an extra task in speech enhancement, and it is applied as a separate stage to improve performance (Zhao, Wang and Wang, 2018). Consequently, DE-CADE was trained to only suppress additive noise, without applying dereverberation, which means that DE-CADE will map noisy, reverberant speech to reverberant speech as a target. Applying denoising only without dereverberation was also proven to improve the intelligibility of the output speech from speech enhancement DNNs (Zhao et al., 2016).

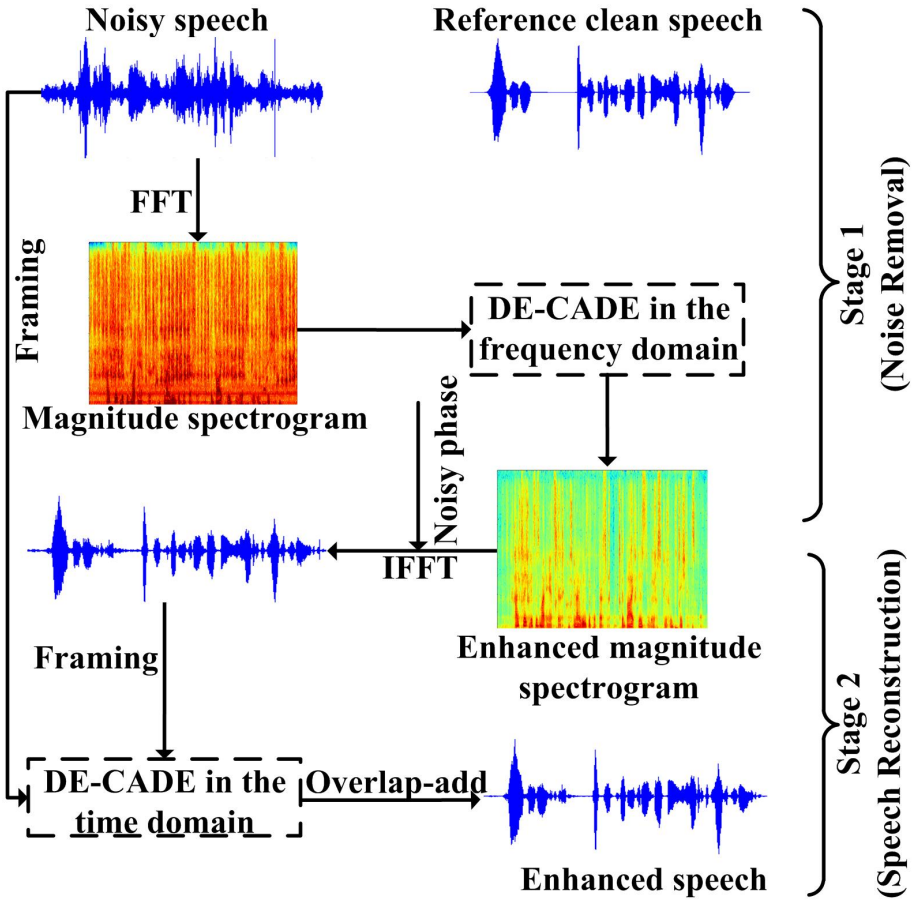


Figure 5.6 The proposed two-stage speech enhancement approach in the frequency and time domain.

For a DNN to be able to solve Equation 5.2, in order to output a good estimate of the clean speech, it is required to have additional information about the relationship between speech and additive noise. When processing noisy speech using a DNN for speech enhancement, the network performs some linear and non-linear operations to finally give an estimate of the clean speech. Based on the fact that most recent DNNs in the literature managed to generate a good prediction of the clean speech, we hypothesize that feeding the output of a DNN for speech enhancement, $\hat{x}$, with the original noisy speech, y, to another second stage speech enhancement DNN will provide the second stage DNN with the needed information about the speech and noise relationship, and this will result in a better learning process and prediction for the second stage DNN.

For the above idea to work, the information provided by the first stage DNN about the estimated clean speech must be different from that of the second stage. In deep learning-based speech enhancement, this can be achieved by either implementing two different DNNs for each stage or apply two different processing approaches using the same architecture for both stages. The proposed two-stage approach is based on the latter idea, as it is based on performing speech enhancement using DE-CADE in the frequency domain, DE-CADE(F), as a first stage to estimate the magnitude spectrogram

of the clean speech. Afterwards, this estimation is fed with the original noisy speech to the second stage DE-CADE, DE-CADE(F-T), to be processed in the time domain, where both magnitude and phase denoising are considered. In this way, a different estimation of the clean speech will be performed using the time domain features.

In the first stage, the noisy speech signal is transformed to the frequency domain to extract time-frequency features using STFT, which can be calculated as described below:

$$Y(t,f) = \sum_{f=0}^{F-1} y(m+t)h(m)e^{-j2\pi fm/F}, \tag{5.4}$$

where *Y(t,f)* is the STFT of the noisy signal, *f* is the frequency bin index; {*f = 0, 1, ..., F-1*} and *F* is the total number of frequency bins, *t* is the time frame, {*t = 0, 1, ..., T-1*} and *T* is the total number of frames, *m* is the input signal time sample, *h* denotes the applied window function, which is a Hamming window in our implementation. The time frame size was set to 256 with 50% overlap. The magnitude of the T-F features was taken to obtain the input spectrograms to the DNN, so the frequency domain representation of Equation 5.2 can be expressed as:

$$|Y(t,f)| = |X(t,f)| + |N(t,f)|, \tag{5.5}$$

where, $|Y(t,f)|$, $|N(t,f)|$, and $|X(t,f)|$ are the magnitude spectrograms of the noisy speech, noise and speech signals, respectively. The first stage DE-CADE was then trained to estimate the clean speech magnitude spectrogram, $|X(t,f)|$. The choice of this mapping approach-based target is based on the fact that CDAEs have bad generalization ability when using masking targets, as discussed in Chapter 4. In this stage, we assumed that the phase is not highly affected by noise compared to the magnitude spectrogram (Wang and Lim, 1982), so the noisy phase was not considered in the first stage processing, and was kept to be added to the final estimated clean magnitude spectrogram.

The processing applied by each layer of DE-CADE is based on a 1D dilated causal convolution operation, which can be expressed as follows:

$$B(u,v) = \sum_{c} \sum_{w+d*q=v} A(c,w) * weight(u,c,q), \tag{5.6}$$

where, *B(u,v)* is the output of the 1D dilated causal convolution layer, *A(c,w)* is the layer input, *weight(u,c,q)* is the filter applied to the input, *u* is the number of applied convolution channels, *v* is the output width, *c* is the number of input channels, *w* is the input width, *q* is the filter width and *d* is the dilation rate.

Each convolution layer is followed by a nonlinear function, PReLU in our case, so

the output, *G*, from the non-linearity layer will be:

$$G(u,v) = PReLU(B(u,v)), \tag{5.7}$$

where,

$$PReLU(B(u,v)) = \begin{cases} \text{B(u,v)}, & \text{if } B > 0, \\ \alpha \text{ B(u,v)}, & \text{otherwise}, \end{cases} \tag{5.8}$$

where α is a variable parameter that changes based on the model during training. Mean Square Error (MSE) is the loss function used with the Adam optimizer, learning rate = 0.0001, β_1 = 0.1, β_2 = 0.999. DE-CADE(F) will minimize the frequency domain MSE loss, given below, to estimate the speech magnitude spectrum.

$$L_F = \frac{1}{TM} \sum_{t=0}^{T} \sum_{f=0}^{F} \left[|\hat{X}(t,f)| - |X(t,f)| \right]^2, \tag{5.9}$$

where L_F is the loss function for the frequency network, DE-CADE(F), *T* is the total number of frames, and *F* is the number of frequency bins. After processing the noisy speech using several convolution and non-linearity functions, the estimated clean speech STFT, $\hat{X}$*(t,f)*, can be reconstructed using the estimated speech magnitude spectrogram, $|\hat{X}(t,f)|$, and the STFT phase of the noisy speech, $\angle$*Y(t,f)*. This can be expressed as follows:

$$\hat{X}(t,f) = \sqrt{|\hat{X}(t,f)|} \otimes e^{j\angle Y(t,f)}, \tag{5.10}$$

where $\otimes$ denotes element-wise multiplication. Finally, the time domain estimated speech signal from the first stage, $\hat{x}$, can be generated using the ISTFT.

$$\hat{x}_1(m) = ISTFT(\hat{X}(t,f)). \tag{5.11}$$

In the second time domain-based stage, DE-CADE(F-T), both the noisy speech, *y*, and the estimated clean speech by the first stage, $\hat{x}_1$, are concatenated on two different channels, and then fed to a similar second stage network but operating in the time domain. Framing is the only preprocessing operation applied to the inputs using a frame size of 2,048 with 50% overlap. The input concatenated time frames, y_2*(t)*, to the second stage network, DE-CADE(F-T), can be represented as follows:

$$y_2(t) = (y(t), \hat{x}_1(t)), \tag{5.12}$$

where, *t* is the time frame, *y(t)* and $\hat{x}_1$*(t)* are the framed noisy and estimated speech, respectively. The network here will try to enhance both the magnitude and phase, given the time-domain representation of the noisy speech and the denoised speech from the

first stage. This will allow different learning and enhancement processes from that of the first stage. MSE is the loss function used for the second stage, as an optimum choice to reduce the time domain prediction error (Kolbæk et al., 2020). This can be expressed as given below.

$$L_T = \frac{1}{T} \sum_{t=0}^{T} \left[\hat{x}_2(t) - x(t) \right]^2, \tag{5.13}$$

where L_T is the loss function of the second enhancement stage and $\hat{x}_2(t)$ is the estimated clean speech frame from the second stage. We finally apply overlap-add procedure to obtain the final estimated clean speech, $\hat{x}_2(m)$.

5.6 Results and Discussion

In this section, evaluation and analysis of the proposed architecture and two-stage approach will be presented. This evaluation covers the performance of the architecture in improving the noisy speech signal, a comparison to other best performing speech enhancement models in the literature, and the complexity of the architecture. The following subsections demonstrate these points in detail.

5.6.1 Baseline Comparison

The architecture performance was first verified by training it using the *"Small-Scale Train Set"*, and then it was tested using *"Small-Scale Test Set"* to be compared with SOTA models in the literature. For comparison, we used a combination of classical and SOTA deep learning-based speech enhancement models, listed below:

- classical Wiener filter approach (Scalart et al., 1996)
- SEGAN (Pascual et al., 2017)
- Wave U-Net (Macartney and Weyde, 2018)
- WaveNet (Rethage et al., 2018)
- MMSE-GAN (Soni et al., 2018)
- Deep Feature Loss (Germain et al., 2019)
- Deep Xi-ResLSTM (Nicolson and Paliwal, 2019)
- Metric-GAN (Fu et al., 2019)
- SEGAN-D (Phan et al., 2020)
- DEMUCS (Défossez et al., 2020)

- Koizumi et al. (Koizumi et al., 2020)
- T-GSA (Kim et al., 2020)
- Deep MMSE (Zhang et al., 2020)

The results of this evaluation are given in Table 5.1, where the models are listed in ascending order based on the overall predicted MOS score, Covl. The training and evaluation of the SOTA architecture was performed by the original authors, and they are just presented in this thesis for comparison based on the table presented in this website (Zhang et al., 2021), so all the SOTA networks were not trained or tested.

The results show that the proposed two-stage architecture, DE-CADE(F-T), outperforms in terms of both Csig and Covl scores. In comparison to other models, our architecture shows a good compromise between noise removal and speech reconstruction, because although the Cbak score is lower for our architecture when compared to some other networks, our implementation managed to decrease speech distortion, leading to the best Csig score, which is the target of this design, and finally this results in an improved overall performance; the highest Covl score. The single stage version of the proposed architecture, DE-CADE(F) also performs better than most of the models; however, the two-stage implementation leads to significant improvement.

5.6.2 Complexity Analysis

Figure 5.7 shows the number of parameters of the first stage DE-CADE(F) and the two-stage version, DE-CADE(F-T), of the proposed architecture, highlighted in red, in comparison with other SOTA speech enhancement models. It should be noted that in this analysis, we only included architectures whose number of parameters were reported by the authors. The single-stage frequency domain-based network, DE-CADE(F), shows a comparable number of parameters to other architectures, such as Wavnet and CDAE-T, but it shows better performance based on the evaluation in Table 5.1. The two-stage architecture, DE-CADE(F-T), is more complex, but it significantly improves speech quality and overall performance as shown in Table 5.1. Moreover, it is of remarkably less complexity compared to the GAN architectures, see Figure 5.7.

5.6.3 Large-Scale Training Performance

In this section, we evaluated the DE-CADE architecture when trained using the *"Large-Scale Train Set"*, and tested using the *"Large-Scale Mismatched Test Set"* and the babble noise corrupted speech used in the evaluation of Section 5.4; these datasets were defined in Section 5.3. Moreover, a comparison was performed with the standard CDAE-based implementations that were trained and tested using the same dataset. The results of these experiments will be presented in the following subsections.

Table 5.1 Performance comparison of SOTA speech enhancement models using the Valentini Voice Bank dataset benchmark (Valentini-Botinhao et al., 2017*b*), based on the results presented online by authors (Zhang et al., 2021).

Metric	**Csig**	**Cbak**	**Covl**
Noisy	3.35	2.44	2.63
Wiener (Scalart et al., 1996)	3.23	2.68	2.67
SEGAN (Pascual et al., 2017)	3.48	2.94	2.80
Wave U-Net (Macartney and Weyde, 2018)	3.52	3.24	2.96
WaveNet (Rethage et al., 2018)	3.62	3.23	2.98
MMSE-GAN (Soni et al., 2018)	3.80	3.12	3.14
Deep Feature Loss (Germain et al., 2019)	3.86	3.33	3.22
Deep Xi-ResLSTM (Nicolson and Paliwal, 2019)	4.01	3.25	3.34
Metric-GAN (Fu et al., 2019)	3.99	3.18	3.42
SEGAN-D (Phan et al., 2020)	3.46	3.11	3.50
DEMUCS (Défossez et al., 2020)	4.14	3.21	3.54
Koizumi et al. (Koizumi et al., 2020)	4.15	3.42	3.57
DE-CADE(F)	4.00	3.11	3.60
T-GSA (Kim et al., 2020)	4.18	**3.59**	3.62
Deep MMSE (Zhang et al., 2020)	4.28	3.46	3.64
DE-CADE(F-T)	**4.36**	3.01	**3.86**

5.6.3.1 Architecture Performance

Figure 5.8 presents five speech quality scores for the output speech from DE-CADE in the frequency domain, DE-CADE(F), DE-CADE in the time domain, DE-CADE(T), and the two-stage DE-CADE, DE-CADE(F-T). It can be noticed from the results that DE-CADE(F) outperforms in terms of the Cbak scores, which proves the higher denoising ability of the frequency domain-based implementation. On the other hand, DE-CADE(T) managed to estimate clean speech with better intelligibility and lower distortion. This is very clear in the case of babble noise, where speech reconstruction is very challenging, as the noise here is similar to the target speech signal, and this results in severe distortion in the case of DE-CADE(F) while the network was trying to eliminate background noise. The proposed two-stage approach DE-CADE(F-T) achieves the best compromise between speech denoising and reconstruction. This is shown in the Cbak graphs, which show an increase in the noise level in some cases compared to DE-CADE(F). This increase in the noise level avoids significant distortion and improves performance with respect to all the other evaluation metrics.

Table 5.2 presents the average of the numerical results presented in Figure 5.8 in the case of the 20 mismatched noise environments. Table 5.2 also shows the performance

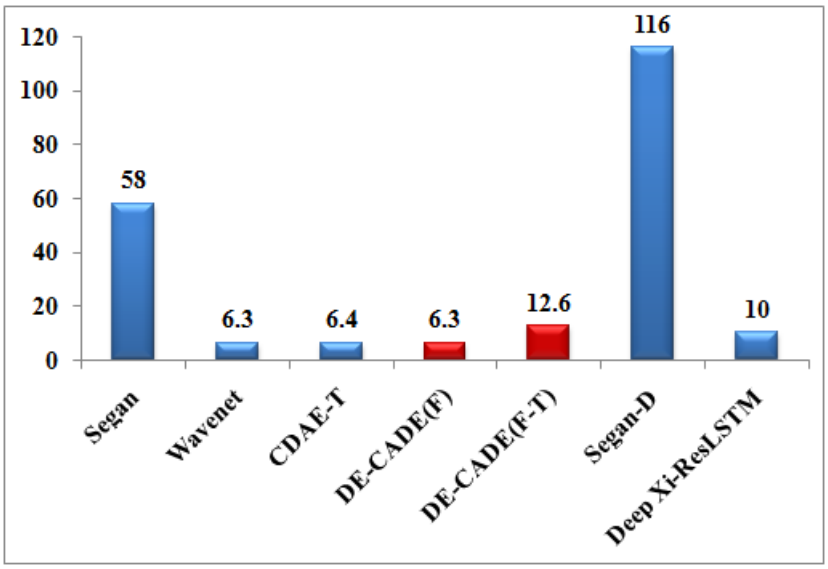


Figure 5.7 A comparison between the number of parameters for the first stage of our architecture, DE-CADE(F), its two-stage version, DE-CADE (F-T), and SOTA speech enhancement models.

of the two-stage model DE-CADE(F-T) after 18 and 50 training epochs, DE-CADE(F-T)(18^{th}) and DE-CADE(F-T)(50^{th}). It can be seen that the performance gain after 18 epochs is not significant, which means that 50 epochs were more than enough for the architecture to converge. Large-scale training and validation curves are shown in Figure 5.9.

Table 5.2 Performance comparison of the two-stage approach to single stage implementations in the frequency and time domains, using the *Large-Scale Mismatched Test Set*. The results are averaged over 6 SNRs, from -5 to 20 with 5 dB step.

Metric	PESQ	STOI	LSD	Csig	Cbak	Covl
Noisy	2.086	83.28	1.422	2.856	2.037	2.421
DE-CADE(F)	2.623	88.21	0.862	3.658	**2.777**	3.120
DE-CADE(T)	2.630	88.32	0.853	3.673	2.596	3.042
DE-CADE(F-T)(18^{th})	2.782	89.63	**0.791**	3.862	2.744	3.305
DE-CADE(F-T)(50^{th})	**2.797**	**89.69**	0.790	**3.865**	2.762	**3.315**

5.6.3.2 Comparison to the Standard CDAE

The single and two-stage DE-CADE performance was compared to the standard CDAE, proposed in (Pandey and Wang, 2018*a*). Table 5.3 shows the outcome of this comparison, where all the models were trained and tested using the *Large-Scale Train Set* and the *Large-Scale Mismatched Test Set*, respectively. The results shows that regardless of the training domain used, both single stage and two-stage DE-CADE based implementations; DE-CADE(F), DE-CADE(T), and DE-CADE(F-T), perform better

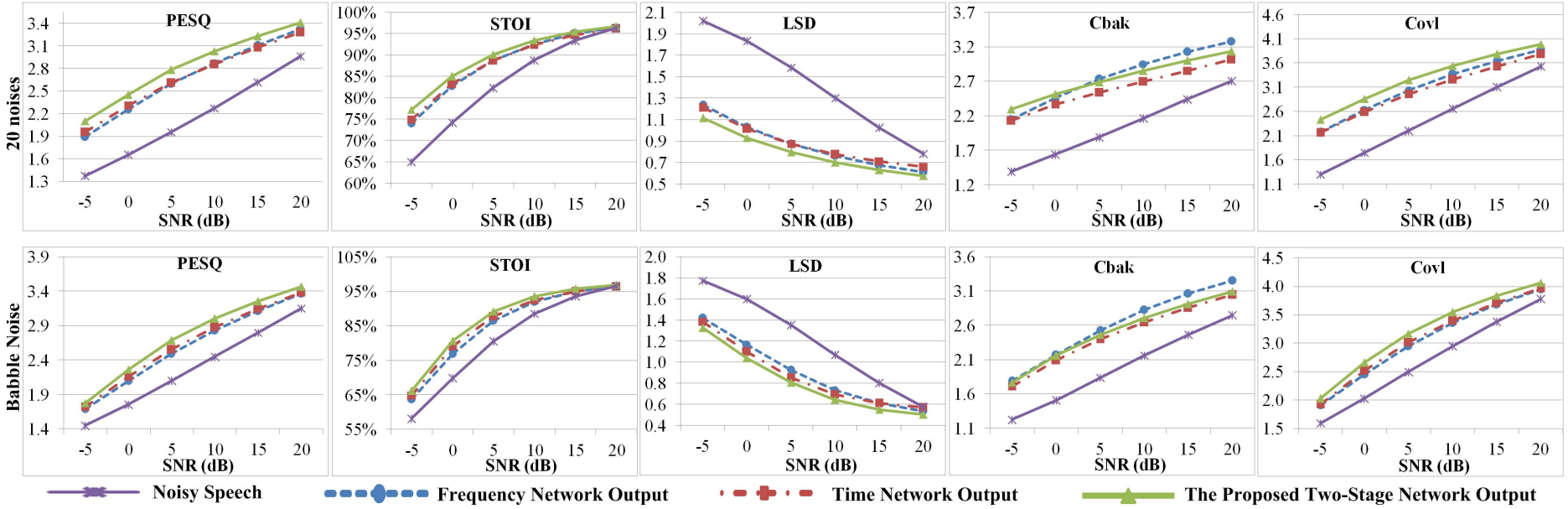


Figure 5.8 The PESQ, STOI, LSD, Cbak, and Covl of the proposed DE-CADE architecture, trained in the frequency domain (DE-CADE(F)), blue line; time domain (DE-CADE(T)), red line; the proposed two-stage approach (DE-CADE(F-T)), green line; and the reference noisy speech, purple line, for the 20 mismatched noise environments in Figure 5.2 and babble noise.

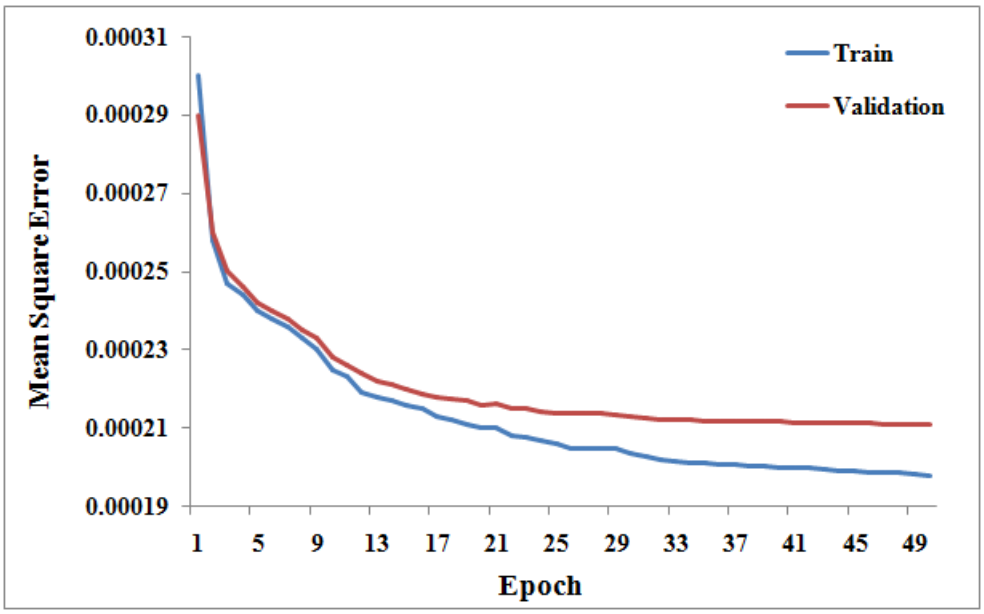


Figure 5.9 The training and validation curves for large-scale training with 1,000 hours of noisy speech.

than the standard CDAE based models, CDAE(F) and CDAE(T). The two-stage DE-CADE(F-T)(50th) outperforms all implementations with respect to all the evaluation metrics, except the Cbak score, where the single stage frequency domain-based network DE-CADE(F) performs better; however, this improvement is at the expense of all the other evaluation metrics.

Table 5.3 Performance comparison of the architecture to standard CDAE speech enhancement networks, using the *Large-Scale Mismatched Test Set*. The results are averaged over 6 SNRs, from -5 to 20 with 5 dB step.

Metric	PESQ	STOI	LSD	Csig	Cbak	Covl
Noisy	2.086	83.28	1.422	2.856	2.037	2.421
CDAE(F) (Pandey and Wang, 2018*a*)	2.622	86.78	1.285	3.438	2.687	3.009
CDAE(T) (Pandey and Wang, 2018*a*)	2.556	87.33	0.936	3.543	2.588	3.016
DE-CADE(F)	2.623	88.21	0.862	3.658	**2.777**	3.120
DE-CADE(T)	2.630	88.32	0.853	3.673	2.596	3.042
DE-CADE(F-T)(50th)	**2.797**	**89.69**	**0.790**	**3.865**	2.762	**3.315**

5.6.4 Architecture Generalization

In this section, the architecture generalization will be evaluated by comparing the difference between the performance when using a matched test set that is seen during the training process, *"Large-Scale Matched Test Set"*, and a mismatched test set not previously seen by the network, *"Large-Scale Mismatched Test Set"*; these test sets were defined in Section 5.3. This will show the network's variance, which is an important

measure to assess the generalization ability of DNNs. Moreover, an evaluation was performed for the generalization of the proposed two-stage approach, by replacing the first stage DE-CADE(F) with two DNNs from the literature. These speech enhancement networks are frequency domain-based implementations, trained with huge speech and noise datasets, and their models are available online. The output from these DNNs, which is not previously seen in the training process, was fed to the second stage network of the two-stage DE-CADE architecture. Details about these experiments are presented in the following two subsections.

5.6.4.1 Generalization to Mismatched Test set

The results in Table 5.4 show that the difference between the performance of DE-CADE(F-T)(18^{th}) in the case of matched and mismatched datasets is acceptable for all evaluation metrics, as it is normal for DNNs to perform better for data seen during training. One of the interesting results in this table is the Cbak score, where the architecture outputs speech with a lower Cbak score in the case of matched test data, which is not common as the DNN should be able to better remove seen noise environments. This proves the ability of this implementation to compromise between noise removal and speech reconstruction, because this decrease in the Cbak score results in a better overall performance in comparison to the mismatched test set, as shown in the Covl score.

Table 5.4 The performance of the proposed two-stage network for matched and mismatched test data. The results are averaged over 6 SNRs, from -5 to 20 with 5 dB step.

Metric	PESQ	STOI	LSD	Csig	Cbak	Covl
Matched	**2.848**	**91.44**	**0.773**	**3.865**	2.635	**3.341**
Mismatched	2.782	89.64	0.791	3.862	**2.744**	3.305

5.6.4.2 Two-Stage Approach Generalization

In this subsection, an important experiment was conducted to show the effect of the proposed two-stage frequency then time approach using different speech enhancement architectures in the first stage. This experiment is based on taking the processed speech from pre-trained frequency domain based DNNs for speech enhancement in the literature, and then feed this output as an input to the trained second stage DE-CADE, DE-CADE(F-T). The used frequency domain networks are: an RNN model (Braun and Tashev, 2020) and an MLP model (Xu et al., 2015). These models were not previously seen in the training process. An illustration of this experiment is shown in Figure 5.10.

The alternative first stage models are available online in (Xia et al., 2020*a*) and (Yong et al., 2015). The RNN architecture is the baseline DNN, provided by Microsoft as a part of the DNS challenge, and it is based on GRUs (Cho et al., 2014) and FF layers. The MLP network is only based on FF layers, to form a highly dense network. The training target in both networks is based on the masking approach. When comparing these networks to the first stage DE-CADE(F) that the second stage is trained on, it is clear that different setup and approaches were applied. This will increase the mismatch between the first stage DNN used in testing and the second stage DE-CADE(F-T), to fairly assess the performance and generalization ability of the DE-CADE(F-T).

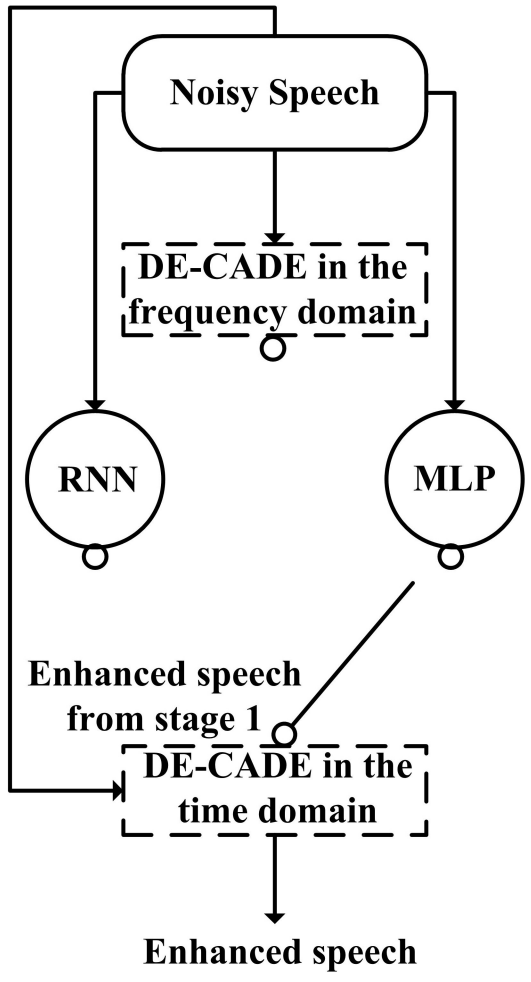


Figure 5.10 The testing process of the two-stage approach generalization, where RNN and MLP are unseen first stage networks during the training.

Additionally, the test speech and noise environments used were not seen by the first stage RNN and the second stage DE-CADE(F-T) architectures during training; however, the first stage MLP network was trained on these noise environments. Based on this, the effect of the second stage will be evaluated on two conditions for the first stage network: seen and unseen test data during the training process, which will show the effect of adding the DE-CADE(F-T) when the noise environments are matched or

mismatched for first stage DNNs. The output from the first stage architecture is concatenated with the noisy speech as input to the second stage DE-CADE, then we evaluated the performance of these architectures using the same *Large-Scale Mismatched Test Set*, described in Section 5.3.

The results of this evaluation are presented in Table 5.5, where the subscripts 1 and 2 denote running the architecture as a single stage and after adding the second stage, DE-CADE(F-T), of our architecture, respectively. The results show that the overall performance of these networks improved when adding the second stage, especially the intelligibility score, which is a difficult factor to improve (Xia et al., 2020*b*). It should also be noted that although the first stage networks were trained to estimate a masking target, which is proven to improve the intelligibility score (Wang et al., 2014), adding the second stage DE-CADE(F-T) results in further significant improvement in speech intelligibility. Moreover, the DE-CADE(F-T) improves the performance of the MLP network, although the noise environments used were seen by the MLP during training. In conclusion, the proposed second stage DE-CADE(F-T) in the time domain can be used as an independent reconstruction stage to other DNN based speech enhancement frequency networks in the literature, to improve their overall performance.

Table 5.5 The performance of other DNNs after adding the time domain second stage of our architecture, using the mismatched test data. The results are averaged over 6 SNRs, from -5 to 20 with 5 dB step.

Metric	**PESQ**	**STOI**	**LSD**	**Csig**	**Cbak**	**Covl**
RNN_1	2.653	88.45	0.845	3.71	2.697	3.141
RNN_2	**2.782**	**89.57**	**0.818**	**3.826**	**2.759**	**3.28**
MLP_1	2.992	88.42	0.835	3.611	**2.986**	3.287
MLP_2	**3.019**	**90.38**	**0.808**	**3.998**	2.962	**3.495**

5.6.5 Comparison to Cascaded Approach

Several experiments were conducted using the DE-CADE architecture, described in Section 5.2, to compare the proposed two-stage frequency then time approach to other cascaded approaches. In all approaches, the estimated output of the first stage, $\hat{x}$, is fed to the second stage; while, in the proposed approach, both the noisy speech, y, and the estimation of the first stage $\hat{x}$ are concatenated and fed to the second stage. Table 5.6 shows this comparison, and the description of each approach is defined below:

- T(y)-T($\hat{x}$): two-stage DE-CADE, in which the first and second stages are operating in the time domain.
- F(y)-F($\hat{x}$): two-stage DE-CADE, in which the first and second stages are operating in the frequency domain.

- T(y)-F($\hat{x}$): two-stage DE-CADE, in which the first stage is operating in the time domain and the second stage in the frequency domain.
- F(y)-T($\hat{x}$): two-stage DE-CADE, in which the first stage is operating in the frequency domain and the second stage in the time domain.
- F(y)-T($\hat{x}$,y): the proposed two-stage DE-CADE with first frequency domain stage and second time domain stage, and the noisy speech is taken through to the second stage along with the output of the first stage.

The evaluations show that the proposed approach, (F(y)-T($\hat{x}$,y)), outperforms other cascaded approaches for all evaluation metrics, except the Cbak results that measure the denoising ability. The cascaded frequency-frequency approach, (F(n)-F($\hat{x}$)), shows the best noise removal performance; however, all the other evaluation metrics are negatively affected. This is more evidence that the frequency network has better denoising ability, as discussed in Section 5.4.

Table 5.6 Performance comparison of the proposed two-stage approach to the cascaded approach, using the mismatched test data. The results are averaged over 6 SNRs, from -5 to 20 with 5 dB step.

Metric	PESQ	STOI	LSD	Csig	Cbak	Covl
Noisy	2.086	83.28	1.422	2.856	2.037	2.421
T(n)-T($\hat{x}$)	2.591	88.01	0.981	3.566	2.581	3.050
F(n)-F($\hat{x}$)	2.609	87.52	0.892	3.557	**2.794**	3.066
T(n)-F($\hat{x}$)	2.643	87.63	0.906	3.580	2.785	3.094
F(n)-T($\hat{x}$)	2.680	87.92	0.884	3.712	2.665	3.176
F(n)-T($\hat{x}$,n)	**2.797**	**89.69**	**0.790**	**3.865**	2.762	**3.315**

5.7 Conclusion

This chapter presents a new CDAE model for speech enhancement, named DE-CADE. The architecture is designed in a way to benefit from the good feature extraction capability of deep networks while keeping the complexity as minimum as possible. Moreover, a two-stage deep learning approach was proposed that deals with speech distortion caused by speech enhancement processing. This approach is based on performing consecutive speech enhancement using DE-CADE but in two different learning domains, frequency then time. The proposed architecture and two-stage approach were proven to outperform SOTA speech enhancement models in the literature. The evaluations also show that the second stage approach generalizes to other speech enhancement DNNs, as it was found that the second stage DE-CADE running in the time domain can be

integrated into other frequency domain-based speech enhancement networks in the literature, to improve their performance. Finally, the presented two-stage approach was compared to other cascaded approaches, and the results show that this approach is the best to compromise between speech denoising and reconstruction, which leads to highest scores for the overall speech quality evaluation metrics.

The next chapter will present edited versions of the proposed speech enhancement architecture in this chapter, in order to apply and test the architecture for real time speech enhancement applications.

CHAPTER 6

Application-based Speech Enhancement

6.1 Introduction

This chapter discusses two speech enhancement applications: ASR and Hearing Aids, and how the developed architecture in Chapter 5 can be optimized for these applications. The discussion of each application will be presented separately in two sections, by presenting the results that were obtained by tuning the DE-CADE architecture, presented in Chapter 5, to perform speech enhancement for the target application.

6.1.1 Research Contribution

The contributions of the work in this chapter are as follows:

- optimizes the second stage network of the two-stage DE-CADE architecture to solve the mismatch issue between single stage speech enhancement DNNs and ASR models,
- the development of a full speech enhancement architecture, designed specifically to improve the performance of ASR systems,
- the development of an edited version of the DE-CADE using a deep CRN-based architecture for speech enhancement of lower complexity, and
- proposes an integrated hearing aid and alert system architecture to improve the functionality of currently available hearing aids.

6.2 Speech Enhancement for Automatic Speech Recognition

The speech enhancement architecture proposed in this section for ASR is used to solve the mismatch issue between the speech enhancement ASR models, discussed in Subsection 2.5.1 (page 26). The proposed architecture is a two-stage speech enhancement model in which speech denoising is applied in the first stage using the DE-CADE network, described in Chapter 5, which performs speech enhancement in the frequency domain. While the second stage is a Least Square Generative Adversarial Network

(LSGAN) model that uses a GAN-based speech enhancement network running in the time domain, to deal with speech distortion caused by the first denoising stage. Moreover, a noise SNR classifier was added as a first processing stage before applying speech enhancement, in order to perform speech enhancement only when necessary, which decreases network complexity and improves the overall performance. The full architecture will be described in details later in Section 6.2.2 and it is represented in Figure 6.1.

The following subsection will present a full description of the developed architecture, including the mathematical explanation of the distortion issue under investigation. Afterwards, results and discussion will be given, showing how this architecture improves the performance of a real ASR system in noisy conditions.

6.2.1 Speech Distortion

The addition of a speech enhancement model provides important preprocessing for an ASR system in intrusive noise environments, where the noisy speech signal has low SNR value. The time domain noisy speech *y* affected by a noise environment *n* can be expressed as in Equation 6.1:

$$y(k) = s(k) + n(k), \tag{6.1}$$

where *s* is the clean speech signal and *k* is the time index. The added frontend speech enhancement module generates an estimate to the clean speech signal $\hat{s}$ by applying a nonlinear mapping function that maps noisy speech to clean speech. Considering the case when the speech enhancement model only performs denoising to the input noisy speech, which is the common case of most available single-stage DNNs for speech enhancement, the output estimated clean speech signal will have a higher SNR compared to the input noisy speech. However, a new noise form will accompany $\hat{s}$, caused by the distortion that occurred during the denoising process, as proved in (Wang et al., 2019), and this noise is known as distortion noise. Based on this fact, the estimated clean speech by the frontend DNN speech enhancement model can be defined by Equation 6.2:

$$\hat{s}(k) = s(k) + \alpha n(k) + n_d(k), \tag{6.2}$$

where α is a scaling factor that describes the decrease in the noise intensity, and n_d is the added distortion noise. Although the output speech from most DNNs has better quality and intelligibility than the noisy unprocessed speech, a mismatch problem between the backend ASR system and the frontend speech enhancement model occurs when the distortion noise, n_d, is significant; in other words, n_d is greater than αn, leading to major changes in the characteristics of the estimated speech compared to the clean and noisy speech signals. Consequently, the negative effect of this distortion noise

outweigh the positive denoising effect of the speech enhancement model, leading to ASR performance degradation.

In order to deal with this distortion noise, a second stage speech enhancement architecture is proposed in this chapter, which is trained to minimize this dominant distortion noise using a GAN-based model with a MSE loss function, known as LSGAN. As discussed in Chapter 2, the GAN architecture consists of two DNNs: a generator and a discriminator network. The generator (*D*) in the proposed architecture performs speech reconstruction by reducing distortion noise through the feedback of the discriminator network (*G*), which is a binary classifier that differentiates between clean and distorted speech. Both the generator and discriminator have a MSE loss function, which can be expressed as in Equations 6.4 and 6.3, respectively:

$$min_D L_{LSGAN}(D) = \frac{1}{2}E_{s\sim P_{data}(s)}[(D(s,y)-b)^2]+ \\ \frac{1}{2}E_{\hat{s}\sim P_{\hat{s}}(\hat{s})}[(D(G(\hat{s},y),y)-a)^2], \quad (6.3)$$

$$min_G L_{LSGAN}(G) = \frac{1}{2}E_{\hat{s}\sim P_{\hat{s}}(\hat{s})}[(D(G(\hat{s},y),y)-b)^2, \quad (6.4)$$

where *b* is an all-one vector representing the label for real clean speech, while *a* is an all-zero vector that represents the label for estimated clean speech. *D(s,y)* is the output of the discriminator with concatenated real clean speech and noisy speech as an input, and *D(G(ŝ,y),y)* is the output of the discriminator with concatenated noisy speech and the second stage estimated clean speech from the generator network as an input; this is clarified in Figure 6.1. The noisy speech is fed to both the generator and the discriminator, as it was found that this improves the learning process, because when the noisy signal is seen as a different signal from the clean speech, noise reconstruction will be avoided during the training process. By applying this second reconstruction stage, the architecture will improve the quality of the estimated speech from the first denoising stage, focusing on the distortion noise, which will ideally overcome the mismatch problem between the speech enhancement model and the ASR model.

In order to further improve the performance of the integrated speech enhancement and ASR system, additional processing is applied to the input noisy speech before applying speech enhancement. This processing aims to test the SNR level of the input noisy speech, to decide whether a speech enhancement stage is required or not before performing ASR. This will improve the overall performance by decreasing system complexity and processing time when speech enhancement is not needed. Moreover, better WERs can be achieved for clean and high SNR speech, by preventing the speech distortion caused by the speech enhancement model. To apply this technique, a CNN-based binary classifier was developed to differentiate between high and low SNR speech sig-

nals. The classifier will activate the speech enhancement model only if a low SNR input speech is detected.

As most ASR systems are trained to deal with some noisy speech signal, the choice of the decision boundary of the classifier between high and low SNR speech will be based on the performance of the backend ASR system in noisy conditions. In the proposed implementation, the classifier was designed to run the speech enhancement model for input noisy speech with 15 dB SNR value or less, so 15 dB is the threshold SNR used to differentiate between high and low SNR speech. The choice of this threshold SNR value is based on evaluations performed to the testing ASR system in noisy environments, where the ASR model was found to be able to deal with noisy speech with SNR value greater than 15 dB without the need to add a speech enhancement stage.

The CNN classifier decision is made based on the average of five audio features that are concatenated together and fed to the classifier network to output a prediction, further details of this classification stage will be given in Subsection 6.2.2. The used input feature vector to the classifier, C_i, can be represented as in Equation 6.5:

$$C_i = \overline{y}_{MFCC} \oplus \overline{y}_{Mel} \oplus \overline{y}_{SC} \oplus \overline{y}_{Chroma} \oplus \overline{y}_{T}, \tag{6.5}$$

where y_{Mel} is the Mel-Spectrogram, y_{MFCC} is the MFCCs, Y_{SC} is the Spectral Contrast, Y_{Chroma} is the Chromagram, and Y_T is the Tonnetz (Alías et al., 2016).

6.2.2 The Developed Architecture

The developed speech enhancement architecture for ASR is shown in Figure 6.1. The architecture is composed of three DNNs: the CNN binary SNR classifier, the first-stage denoising DE-CADE, and the speech reconstruction LSGAN network. The following subsections will demonstrate the structure and function of each DNN.

6.2.2.1 The SNR Classifier

A CNN-based network was developed in order to perform binary classification to the SNR value of the input noisy speech, to decide whether speech enhancement processing is needed or not. The network consists of three 1D convolution layers with PReLU activation functions. In order to improve network generalization and overcome overfitting to the training data, the three convolution layers were followed by a dropout layer of 0.2% rate. Two dense layers were added after the convolution layers, one with ReLU activation for further processing the advanced features extracted by the convolution layers, and the final one with Sigmoid activation to perform the prediction. The classifier generates an output based on the detected SNR value of the noisy speech, where 15 dB SNR was chosen as the threshold that differentiates between the low SNR noisy speech (less than or equal 15 dB) that is required for speech enhancement processing, and the

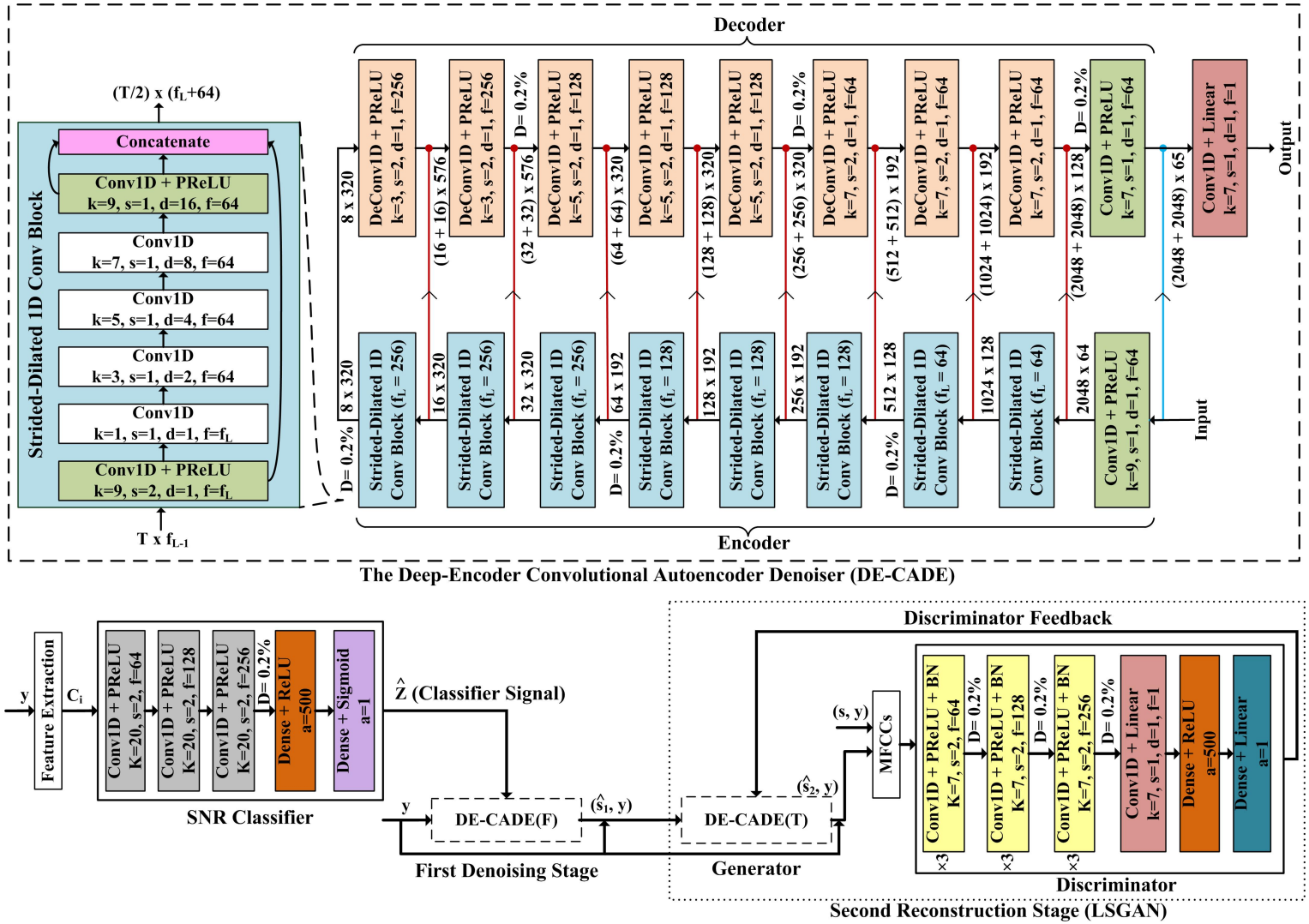


Figure 6.1 The proposed speech enhancement architecture. k, d, f, and L represent kernel size, dilation rate, number of convolution channels, and layer number respectively; s represents stride size in the encoder, and upsampling size in the decoder. T is the time samples, and a is the number of units. y is the noisy speech, C_i is the input feature vector to the SNR classifier, and $\hat{z}$ is the predicted label by the SNR classifier. s is the clean speech, and $\hat{s}_1$ and $\hat{s}_2$ are the enhanced speech by the first and second stage, respectively.

high SNR and clean speech (greater than 15 dB) that is better fed directly to the ASR system without applying speech enhancement.

6.2.2.2 The First Denoising Stage

In case that the input noisy speech is classified as low SNR by the CNN network, the noisy speech will be first processed by the frequency domain-based DE-CADE architecture, DE-CADE(F), to eliminate background noise. The noisy speech is converted from the time to frequency domain by applying STFT using a Hamming window of size 256 and 50% overlap. The T-F features will be then fed to the DE-CADE(F) architecture, to generate an estimate to the clean speech spectrogram using several layers of strided-dilated convolutions in the encoder and deconvolution and upsampling in the decoder. Phase denoising is not applied in this stage, so the noisy phase is used in the reconstruction of the time domain estimated clean speech by this first enhancement stage.

6.2.2.3 The Second Reconstruction Stage

The applied second enhancement stage DE-CADE was modified, in order to improve the performance of a baseline ASR model (Peddinti et al., 2015) provided by Intelligent Voice for research purposes. A LSGAN-based architecture was used to perform speech reconstruction, to minimize distortion noise caused by the first denoising stage, which is the main reason for the mismatch problem between speech enhancement models and ASR systems, as discussed in Subsection 6.2.1. The generator of the LSGAN architecture is a DE-CADE network, described in Chapter 5, operating in the time domain, DE-CADE(T). Therefore, framing is performed to the output speech from the first enhancement stage using a Hamming window with frames of size 2,048 and 50% overlap. During training, magnitude and phase enhancement were applied in this second enhancement stage, and the output of the generator is fed to the discriminator network, which evaluates the quality of the processed speech in terms of speech distortion.

The discriminator network is a CNN-based binary classifier network that differentiates between distorted and clean speech. The network performs classification based on the frequency domain-based MFCC features, because these are the main features used by the ASR system being tested. This will ensure that the processed speech by the second enhancement stage will keep the most important speech information, which will help the ASR to successfully transcribe the processed speech. The discriminator consists of nine 1D strided convolution layers with stride size of 2 and PReLU activations. These layers extract advanced features from the MFCCs of the input audio, and then the final decision is performed using two dense layers. The first dense layer performs further processing using ReLU activation, and the final dense layer gives the prediction

using a Linear activation function. It should be noted here that Linear activation was used in the final layer of the discriminator network because this is a LSGAN-based implementation, which uses the MSE loss function in both the generator and discriminator, as illustrated in Subsection 6.2.1.

6.2.3 Experimental Setup

All three DNN models in the architecture shown in Figure 6.1 were trained using speech and noise data from the DNS challenge dataset (Xia et al., 2020*a*), provided by Microsoft. The dataset has more than 500 hours of speech and 181 hours of noise data. The data was divided into 90% for training and 10% for validation. Afterwards, speech utterances were mixed with noise utterances to form the training and validation noisy/clean pairs, required for the training of the first and second stage speech enhancement DNNs. While for the SNR classifier training, the data was categorized by the SNR value, where noisy speech with a SNR value of 15 dB or less was labeled as low SNR speech (binary 1), and noisy speech with a SNR value higher than 15 dB and clean speech data were labeled as high SNR speech (binary 0). A wide range of SNRs was used during the training of the three DNNs, as training includes SNR values from 0 to 20 dB in steps of 1 dB.

Two challenging test sets were used to evaluate the performance of the proposed architecture, one with extremely intrusive noise environments and the other with less intrusive noise environments. For both test sets, 224 clean audio samples were used, for 56 speakers and 224 different speech utterances. These audio samples were randomly selected from the Voice Bank Corpus (Veaux et al., 2013), which is a different dataset than the one used in the training process. To create the first test set, these clean speech audio samples were corrupted with 9 highly intrusive crowd noise environments (N1:N9) and AWGN, taken from the 100 Nonspeech Environmental Sounds dataset (Hu, 2014); this will be denoted by *Test Set (1)*. On the other hand, the second test set was formed by corrupting the clean speech audio samples with 10 different less intrusive noise environments (N91:N100) from the 100 Nonspeech Environmental Sounds dataset; this will be denoted by *Test Set (2)*. These test sets are very challenging to the proposed architecture, based on the fact that the clean speech utterances were taken from a dataset unseen during training, the number of speakers is high, and the noise environments are highly mismatched and challenging. This ensures fair assessment of the architecture for real situations (Pandey and Wang, 2020*b*).

As the ASR model used for testing was trained using narrow band speech with 8 KHz sampling frequency, the speech enhancement architecture was trained using the same sampling frequency, so all input speech audios were downsampled to 8 KHz. As discussed in Subsection 6.2.1, MSE is the loss function used by the first and second

stage speech enhancement models, with Adam optimizer, learning rate = 0.0001, β_1 = 0.1 for the first enhancement stage DE-CADE network and β_1 = 0.5 for the second enhancement stage LSGAN model. On the other hand, Binary Cross Entropy (BCE) is the loss function used by the SNR classifier. A batch size of 2 was used in training, and the first and second stage speech enhancement DNNs were trained for 100 and 20 epochs, respectively, which were enough for both models to converge. While the SNR classifier was trained for 300 epochs.

6.2.4 Results and Discussion

Three experiments were conducted to evaluate the performance of the proposed speech enhancement architecture for ASR. Speech enhancement performance was first assessed in comparison to similar SOTA speech enhancement models in the literature. Moreover, another evaluation was performed to show the effect of adding the architecture as a preprocessing stage to the ASR model being tested. A final experiment was then conducted to evaluate the generalization ability of the second stage LSGAN model, used to avoid the mismatch problem between the speech enhancement model and the ASR model. This was performed by processing the noisy speech by first stage speech enhancement DNNs in the literature, different from the DE-CADE(F) model used in the training process, and then the output from these DNNs was fed to the LSGAN, to show the improvement gained by the LSGAN.

The PESQ (Rix et al., 2001), STOI (Taal et al., 2011), and SI-SDR (Le Roux et al., 2019) are three measures used to evaluate speech quality, intelligibility, and distortion, respectively. On the other hand, the standard WER was used to evaluate the performance of the ASR model. The experiments were performed using four testing SNR values: 0 dB, 5 dB, 15 dB, and 20 dB for both test sets. For speech enhancement evaluations, shown in Tables 6.1 and 6.2, the average results are given. While second stage LSGAN generalization ability evaluations, shown in Tables 6.5 and 6.6, were performed using 0 dB SNR only, which is the most challenging test SNR value, enough to prove network generalization.

6.2.4.1 Speech Enhancement Performance

The results in Tables 6.1 and 6.2 show the performance of the proposed two-stage architecture for ASR, *DE-CADE*$_{ASR}$, for *Test Set(1)* and *Test Set(2)*, respectively, against two similar two-stage speech enhancement models: a cascaded GAN model (Phan et al., 2020), *GAN*$_2$, and the two-stage DE-CADE model presented in Chapter 5, *DE-CADE(F-T)*. The results also include the performance of the first enhancement stage by the frequency domain-based DE-CADE, *DE-CADE*$_{s1}$, and another single stage best performing GAN model (Fu et al., 2019), *GAN*$_1$, developed to optimize the PESQ score.

It is clear from the results that proposed architecture for ASR outperforms with respect to all the evaluation metrics for both test sets. The architecture shows better performance than the previously proposed DE-CADE(F-T) architecture, which shows the positive effect of replacing the second stage DE-CADE model with the LSGAN-based model.

Table 6.1 Performance comparison of the architecture to other speech enhancement networks using *Test Set (1)*.

Metric	Noisy	GAN_1	DE-CADE_{s1}	GAN_2	DE-CADE(F-T)	DE-CADE_{ASR}
PESQ	2.20	2.49	2.68	2.73	2.87	**3.02**
STOI (%)	80	81.5	81.8	82.2	84.1	**85.3**
SI-SDR	4.13	9.82	10.77	11.05	11.36	**12.64**

Table 6.2 Performance comparison of the architecture to other speech enhancement networks using *Test Set (2)*

Metric	Noisy	GAN_1	DE-CADE_{s1}	GAN_2	DE-CADE(F-T)	DE-CADE_{ASR}
PESQ	2.50	2.81	2.95	3.11	3.20	**3.30**
STOI (%)	83.7	84.8	86.4	87.8	88.2	**88.6**
SI-SDR	6.10	11.16	12.64	12.81	13.98	**15.06**

6.2.4.2 Automatic Speech Recognition Performance

Tables 6.3 and 6.4 show the performance of the ASR model after adding the proposed speech enhancement architecture to process noisy speech for *Test Set(1)* and *Test Set(2)*, respectively. The tables present the WERs of unprocessed speech, $\text{WER}_{Unproc.}$, after the first enhancement stage, WER_{SE1}, after the second enhancement stage, WER_{SE2}, and for the full speech enhancement architecture with the SNR classifier network, WER_{C+SE}. WERs are presented when processing clean speech and noisy speech at the four testing SNRs, the average of the results is also given in the tables. It should be mentioned here that the WER of the clean speech utterances is high compared to the WER of the Valentini test set, reported in this work (Giri et al., 2019), because here 224 speech utterances for 56 speakers were randomly selected from the training set, compared to the 824 test speech utterances for only two speakers in the Valentini test set. This more challenging test conditions were used to fairly show the effect of adding the speech enhancement network on the performance of ASR; while the performance of the ASR as an independent system is outside the scope of the work done in this thesis.

The classification accuracy of the SNR classifier for both *Test Set(1)* and *Test Set(2)* is 100% at very low SNR values (0 dB and 5dB), where it is not challenging for the classifier to detect these highly intrusive noise levels. However, the classifier accuracy

at 15 dB SNR is 82% for *Test Set(1)* and 80% for *Test Set(2)*. A degradation in the classification accuracy was detected at 15 dB SNR, because it is the threshold SNR value used to differentiate between low and high SNR speech, which means it is the most challenging SNR for the classifier to output the correct decision. While the accuracy of the classifier improves again as the SNR level increases, 89% and 90% accuracy at 20 dB SNR for *Test Set(1)* and *Test Set(2)*, respectively. Finally, the classifier accuracy for clean speech data is 94%. The classification accuracy for *Test Set(1)* and *Test Set(2)* is summarized in Figure 6.2.

The results in Tables 6.3 and 6.4 show that the full speech enhancement architecture with the classifier leads to the lowest average WER for both test sets. The mismatch problem between speech enhancement and the ASR model can be verified by the WER of the processed speech by the first stage speech enhancement, WER_{SE1}, where the WER is higher than that of the unprocessed noisy speech, although the first stage was proven to output speech with better quality and intelligibility. It can be clearly seen how the second enhancement stage LSGAN model solves this issue, leading to 8.2% reduction in the WER in comparison to the noisy speech. Another point that should be discussed is the slightly higher WER at 15 dB SNR for the full architecture with classifier, WER_{C+SE}, compared to the case when the classifier is not added, WER_{SE2}. Again, this is due to the fact that 15 dB SNR is a very challenging value, as it is the decision boundary SNR value. This slight increase in the WER was caused by noisy speech audios that were incorrectly classified by the classifier network. Finally, the unprocessed clean speech is of the lowest WER, due to the distortion added by any speech processing techniques even for clean speech. However, further improvement in classification accuracy for clean speech data can completely avoid this issue. This shows the importance of the added classifier network as an initial processing before performing speech enhancement for ASR application.

Table 6.3 Performance of the automatic speech recognition system using *Test Set (1)*

SNR	Clean	20 dB	15 dB	5 dB	0 dB	Ave
$\textbf{WER}_{Unproc.}$	31.9	33.6	40.4	71.3	88	53.04
$\textbf{WER}_{SE1}$	32.4	35.7	39.5	65.2	88.3	52.22
$\textbf{WER}_{SE2}$	32.4	35	39.2	60.8	79.8	49.44
$\textbf{WER}_{C+SE}$	32.1	33.9	39.4	60.8	79.8	**49.20**

6.2.4.3 Second Stage Generalization

A generalization ability assessment was performed on the second stage LSGAN model, similar to the one performed on the second stage DE-CADE, presented in Chapter 5. In the first enhancement stage, two pre-trained single stage DNNs were used in testing: an

Figure 6.2 SNR classifier accuracy for *Test Set (1)* and *Test Set (2)*

Table 6.4 Performance of the automatic speech recognition system using *Test Set (2)*

SNR	Clean	20 dB	15 dB	5 dB	0 dB	Ave
WER$_{Unproc.}$	31.9	33.2	39.7	53.9	65.7	44.9
WER$_{SE1}$	32.4	33.9	36.7	48.2	59.2	42.1
WER$_{SE2}$	32.4	33.8	35.5	43.5	51.9	39.4
WER$_{C+SE}$	32.1	33.4	35.6	43.5	51.9	**39.3**

MLP model (Xu et al., 2015) and an RNN model (Braun and Tashev, 2020), available in (Xia et al., 2020*a*) and (Yong et al., 2015), respectively. These models performs speech enhancement in the frequency domain using masking targets, which is a different training target from the mapping-based target of the first stage DE-CADE network used in training. This will evaluate network generalization to DNNs with different training approaches as well. Moreover, the pre-trained MLP model used was trained using the noise environments in both test sets, making the testing conditions not challenging for the MLP network. This is to show the improvement added by the proposed second stage, even when the test data is matched and seen by the first enhancement stage during training.

The results of this experiment are shown in Tables 6.5 and 6.6 for *Test Set(1)* and *Test Set(2)*, respectively. The presented results are for 0 dB SNR, and the subscripts 1 and 2 denote the model running as a single stage and after adding the second stage of our architecture, respectively. For both test sets, the results show that the second stage LSGAN improves speech quality and intelligibility for both single stage DNNs used in testing. On the other hand, the improvement in ASR performance is clear after adding the LSGAN, and it solved the mismatch problem occurred by the processing

of the MLP network. This evaluation stands as a proof that the proposed second stage LSGAN architecture can act as a standalone speech enhancement model that can be applied to other single stage speech enhancement DNNs, to allow their application as a preprocessing stage to ASR models.

Table 6.5 Generalization of the second stage network to other speech enhancement models using *Test Set (1)*.

Metric	Noisy	$\mathbf{MLP}_1$	$\mathbf{MLP}_2$	$\mathbf{RNN}_1$	$\mathbf{RNN}_2$
PESQ	1.71	2.41	**2.48**	2.35	**2.42**
STOI(%)	68.4	75.8	**75.9**	76.3	**76.6**
WER	88	88.2	**86.3**	85.4	**79.4**

Table 6.6 Second stage network generalization to other speech enhancement models using *Test Set (2)*

Metric	Noisy	$\mathbf{MLP}_1$	$\mathbf{MLP}_2$	$\mathbf{RNN}_1$	$\mathbf{RNN}_2$
PESQ	1.92	2.84	**2.92**	2.48	**2.53**
STOI(%)	73.8	82.7	**82.9**	79.9	**80.1**
WER	65.7	65.9	**59.3**	60.1	**54.2**

6.3 Speech Enhancement for Hearing Aids

This section presents a developed DNN architecture for smart speech enhancement, defined as a technique that aims to enhance the speech signal by eliminating any background noise, except emergency noises. The architecture is shown in Figure 6.3, and it consists of a CNN-based noise classifier and a Deep Convolutional Recurrent Network (DCRN). The input noisy speech is first processed by the CNN classifier to detect the noise environment. The output from the classifier determines the mode that the DCRN will apply to process the noisy speech. If emergency noise is detected by the classifier, the DCRN will run in an audio enhancement mode, to amplify both speech and emergency noise and mitigate any other undesired noise. If the classifier detects an undesired, non-emergency noise environment only accompanying the speech signal, the DCRN will run in speech enhancement mode, to perform regular speech enhancement processing by trying to eliminate all background noise. This architecture can be considered as an integrated hearing aid and alert system in one electronic device, which aims to develop currently available hearing aids while maintaining the same performance of the speech enhancement module.

The following subsections will cover all the technical details of this smart speech enhancement architecture, and the mathematical explanation of the developed smart speech enhancement. Afterwards, results and discussion will be provided.

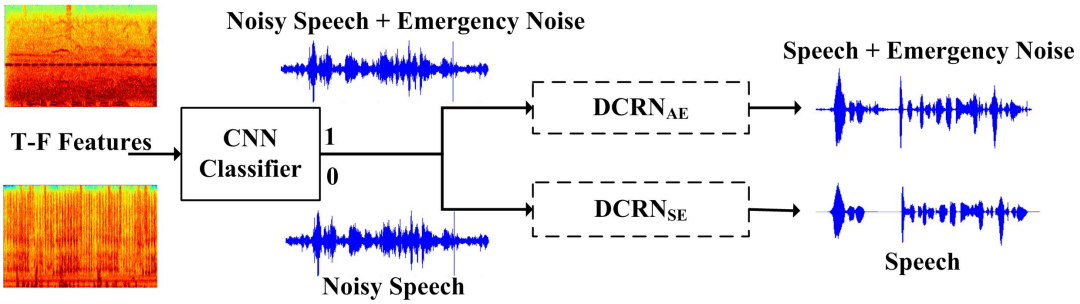


Figure 6.3 Smart speech enhancement architecture: an integrated speech enhancement and alert system for hearing aids.

6.3.1 Smart Speech Enhancement

The input noisy speech to any speech enhancement DNN can be represented as follows:

$$y(k) = s(k) + n(k), \tag{6.6}$$

where *y*, *s*, and *n* are the noisy speech, clean speech and noise, respectively, $\{y, s, n\} \in \mathrm{R}^{K \times 1}$, where K is the total number of samples, and k is the sample index.

The developed DCRN, shown in Figure 6.4, processes audio frames in the time domain, where the encoder compresses the input using strided convolutions which will help in the denoising process. It also performs several 1D dilated causal convolutions to increase the depth and extract more features. Dilated convolution can be defined as in Equation 6.7:

$$G(u,v) = PReLU(\sum_c \sum_{w+d*q=v} A(c,w) * weight(u,c,q)), \tag{6.7}$$

where, *G(u,v)* is the output of the 1D dilated causal convolution and PReLU activation, *A(c,w)* is the layer input, *weight(u,c,q)* is the filter applied to the input, *u* is the number of applied convolution channels, *v* is the output width, *c* is the number of input channels, *w* is the input width, *q* is the filter width and *d* is the dilation rate.

The two LSTM layers between the encoder and decoder will extract temporal information from the generated bottleneck features from the encoder. The LSTM layer operations are given below in Equations 6.8-6.13:

$$f_t = \sigma_g(W_f \times x_t + U_f \times h_{(t-1)} + b_f), \tag{6.8}$$
$$i_t = \sigma_g(W_i \times x_t + U_i \times h_{(t-1)} + b_i), \tag{6.9}$$
$$o_t = \sigma_g(W_o \times x_t + U_o \times h_{(t-1)} + b_o), \tag{6.10}$$
$$\grave{c}_t = \sigma_c(W_c \times x_t + U_c \times h_{(t-1)} + b_c), \tag{6.11}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \grave{c}_t, \tag{6.12}$$
$$h_t = o_t \cdot \sigma_c(c_t), \tag{6.13}$$

where f_t is the forget gate, i_t is the input gate, o_t is the output gate, c_t is the cell state, h_t is the hidden state, σ_g and σ_c are Sigmoid and Tanh activation functions, respectively, and $\cdot$ represents element wise multiplication. The decoder will then decompress the output to its original size using dilated 1D convolution and upsampling layers.

The DCRN for speech enhancement minimizes a MMSE loss function, L_{SE}, during the training process, in order to generate an estimate to the clean speech signal, *ŝ(t)*. This can be described by Equation 6.14:

$$L_{SE} = \frac{1}{T}\sum_{t=0}^{T}\left[\hat{s}(t) - s(t)\right]^2, \tag{6.14}$$

where, *t* is the time frame index, and *T* is the total number of frames.

In smart speech enhancement, we categorize the input noise as emergency (n_e) or unimportant noise (n_u), so in this case Equation 6.6 can be represented as in Equations 6.15 and 6.16:

$$\begin{aligned} y(k) &= s(k) + n_e(k) + n_u(k) && (6.15)\\ &= x(k) + n_u(k), && (6.16) \end{aligned}$$

where *x(k)* is the clean speech in addition to the emergency noise. In this case, the DCRN minimizes a different loss function than that of speech enhancement, because here the network runs in an audio enhancement mode to enhance both speech and emergency noise while suppressing other unimportant background noise. Therefore, the MMSE loss function for smart speech enhancement, L_{SSE}, can be defined as in Equation 6.17:

$$L_{SSE} = \frac{1}{T}\sum_{t=0}^{T}\left[\hat{x}(t) - x(t)\right]^2. \tag{6.17}$$

Although this audio enhancement mode has an important role in retaining emergency noise, it negatively affects the denoising ability of the DCRN for unimportant noise. To avoid this issue, the noisy speech will be first processed by a CNN noise classifier that acts as a switch to run the DCRN in one of two modes: speech enhancement or audio enhancement mode. If the classifier detects emergency noise, the DCRN will perform audio enhancement to enhance both speech and emergency noise while suppressing any other kind of noise. Otherwise, the DCRN will perform speech enhancement to improves the speech signal only and discard any other noise environments.

The classifier accepts five features as input that are useful for audio classification: Mel-Spectrogram (Y_{Mel}), MFCC (Y_{MFCC}), Spectral Contrast (Y_{SC}), Chromagram (Y_{Chroma}), and Tonnetz (Y_T) (Alías et al., 2016). Mel-Spectrogram and MFCC are mainly used to model human hearing perception, while Chromagram and Tonnetz

model the harmonic structure of speech and noise and shows harmonic relationships. Spectral Contrast is defined as the decibel difference between peaks and valleys in the spectrum. It measures energy variations of frequency at each timestamp and represents the relative spectral characteristics. These features were extracted, averaged and concatenated to form the input vector to the classifier C_i. This is shown below in Equation 6.18:

$$C_i = \overline{Y}_{MFCC} \oplus \overline{Y}_{Mel} \oplus \overline{Y}_{SC} \oplus \overline{Y}_{Chroma} \oplus \overline{Y}_{T}. \tag{6.18}$$

Based on the detected noise environment, the classifier will decide the DCRN mode of operation. The classifier is trained to differentiate between emergency and non-emergency noise by minimizing a BCE loss function, L_C, given in Equation 6.19:

$$L_C = \frac{1}{M} \sum_{i=1}^{M} \left[Z_i \log \hat{Z}_i + (1 - Z_i) \log (1 - \hat{Z}_i) \right], \tag{6.19}$$

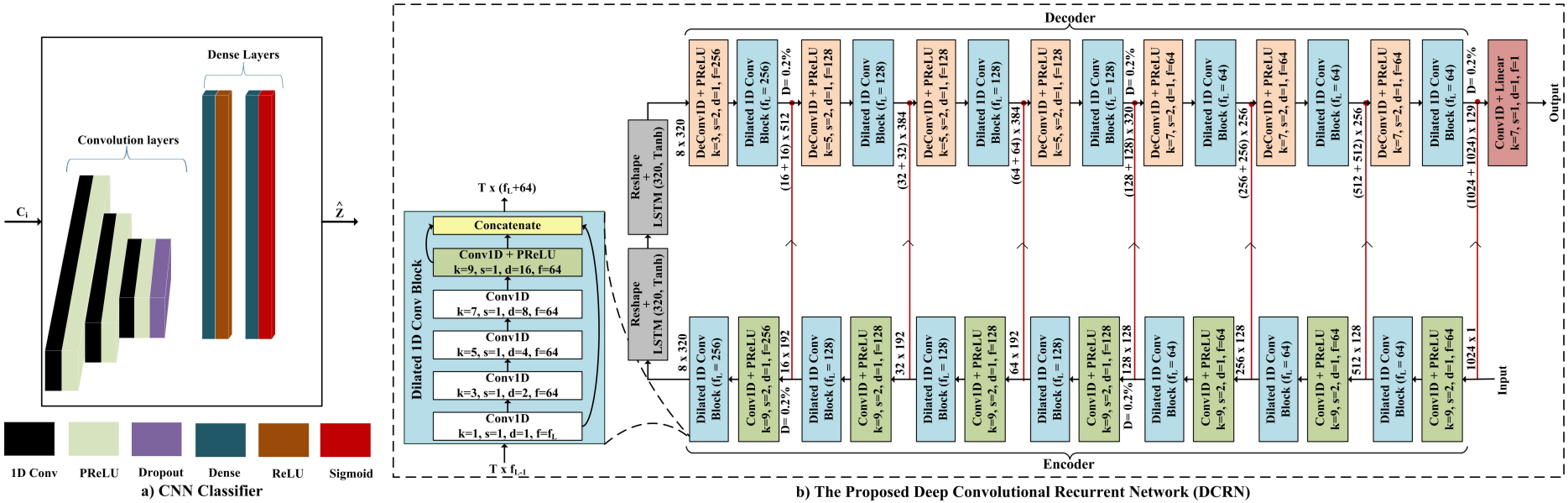
where M is the total number of input samples, i is the sample index, Z is the target binary value (1 if emergency noise is detected and 0 otherwise), and $\hat{Z}$ is the predicted probability generated by the model.

6.3.2 The Developed Smart Speech Enhancement Architecture

The developed smart speech enhancement architecture is presented in Figure 6.4, and it consists of two DNNs: the CNN-based noise classifier and the DCRN. The following subsections will illustrate each network separately.

6.3.2.1 The Convolutional Classifier

The input noisy speech is first processed using a binary noise classifier that classifies the input audio as noisy speech with undesired noise (class 0) or noisy speech with emergency noise (class 1). The classifier is shown in Figure 6.4(a), and it is a CNN-based architecture due to the proven efficiency of CNNs in noise classification (Park and Lee, 2020; Mushtaq and Su, 2020). The network extracts T-F features from the input noisy speech, which improve the classification accuracy (Mushtaq and Su, 2020). The architecture consists of three 1D strided-convolution layers with PReLU activations, a stride of size 2, and a kernel of size 10. Filter sizes of 64, 128, and 256 were used for the first, second and third layers, respectively. The input is compressed by these three layers to extract more advanced features that will help in the prediction process, which is performed using two dense layers. The first dense layer has 512 units andReLU activations, while the second dense layer is an output layer with sigmoid activation. The output from the classifier is then fed to the DCRN to perform speech enhancement if the output is 0, or audio enhancement if the output is 1.



b) The Proposed Deep Convolutional Recurrent Network (DCRN)

Figure 6.4 The proposed smart speech enhancement architecture for hearing aids, a) the CNN classifier; C_i is the input feature vector and $\hat{Z}$ is the predicted class, b)the Deep Convolutional Recurrent Network (DCRN); k, d, f, and L represent kernel size, dilation rate, number of convolution channels, and layer number respectively; s represents stride size in the encoder, and upsampling size in the decoder. T is the time samples. The red lines represent skip connections.

6.3.2.2 The Deep Convolutional Recurrent Network (DCRN)

The developed DCRN is based on the proposed DE-CADE architecture, presented in Chapter 5; however, modifications have been made to decrease network complexity for hearing aid devices. This is achieved by implementing a symmetric encoder and decoder network and applying one enhancement stage only. Moreover, two LSTM layers were added to convert the DE-CADE architecture to a DCRN. All these modifications will improve the performance of the first stage DE-CADE and decrease network complexity by applying single stage speech enhancement instead of two stages, which will be more suitable for applications such as hearing aids, where device size matters. This architecture is presented in Figure 6.4(b).

Similar to the DE-CADE network, the developed DCRN is a 1D convolution-based network that takes advantage of strided and dilated convolution to improve the process of feature extraction (Pandey and Wang, 2020*a*). The architecture operates in the time domain using input time frames of 1,024 size, considering the recent promising performance of time domain-based speech enhancement (Défossez et al., 2020). It should be mentioned here that the 10 ms maximum latency restriction of currently available hearing aids was not considered in this work, as the network accepts a time frame of 64 ms.

The architecture is divided into three components: the encoder, two LSTM layers, and the decoder. Both the encoder and decoder networks are implemented using several dilated convolution blocks, where the kernel size used for the convolution layers in each block increases across the hidden layers. This allows for better feature extraction by increasing the receptive field of each feature vector, which can be illustrated by the diagram in Figure 6.5. The use of these increased kernel size convolutions prevents information loss that might occur by the compression process in deep hidden layers of the encoder; at the same time, it decreases network complexity by avoiding the use of large kernel sizes in all the convolution layers.

The input is compressed in the encoder network using strided convolutions of size 2 and PReLU activations, to finally reach a size of 8, while deconvolution is applied in the decoder network using convolution layers with PReLU activations and upsampling layers of size 2, to reconstruct the audio back to its original size. To avoid network overfitting, a dropout of 0.2% is used after every three dilated convolution blocks.

The two LSTM layers were added in the middle before feeding the signal to the decoder network, each has 320 units with Tanh activation. The role of these layers is to process the compressed bottleneck features to consider temporal dynamics of speech. Further details for other hyperparameters used for each layer are provided in Figure 6.4.

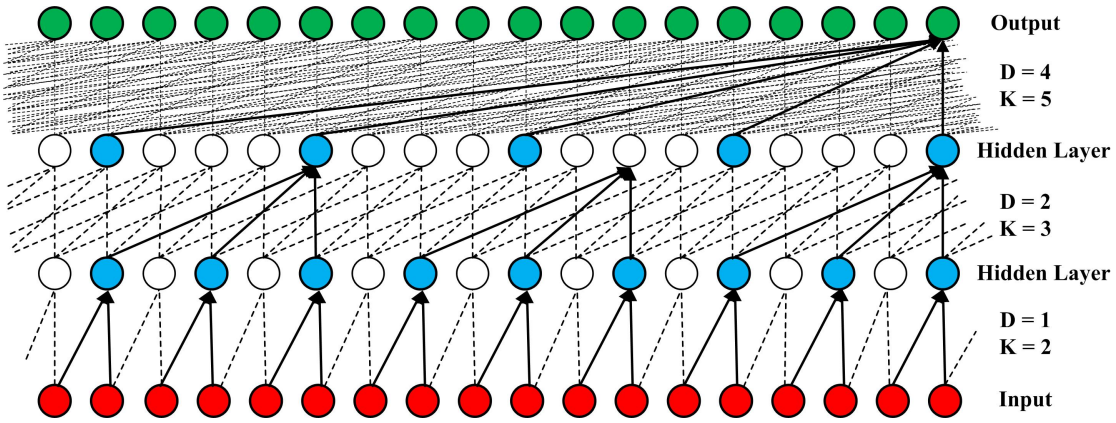


Figure 6.5 Illustration of dilated causal convolution with increased kernel size

6.3.3 Experimental Setup

The Microsoft DNS challenge dataset (Reddy et al., 2021) was used to train the DCRN. The dataset has more than 500 hours of speech and 181 hours of non-emergency noise data. The speech and noise data was first divided, 90% for training and 10% for validation; afterwards, they were randomly mixed to create the noisy speech audios with non-emergency noise at a wide range of SNRs, from 0 dB to 20 dB with a step of 1. This creates a total of 65,000 and 6,500 noisy speech utterances with non-emergency noise used for training and validation of the DCRN to perform speech enhancement, respectively. This dataset will be denoted as *Speech Enhancement Train Set*

To train the DCRN to perform smart speech enhancement using audio enhancement procedure that enhances both speech and emergency noise, a total of 1,478 emergency noise audio samples were collected for 5 emergency noise types: 118 alarm audio samples, including fire alarms, door bells, and alarm clocks; 440 car horn audio samples; 440 car siren audio samples; 440 baby crying audio samples; and 40 footstep audio samples. A total of 240 emergency noise audio samples were taken from the ESC-50 dataset (Piczak, 2015), 800 from UrbanSound8K database (Salamon et al., 2014), 400 from Donate-a-Cry corpus (Gveres, 2015), and 38 from Mixkit website (Elements, 2019). First, these emergency noise audio samples were randomly mixed with the clean speech data from the DNS dataset at 0 dB SNR, to help the network to deal with speech and emergency noise similarly during training, and then this mixture was corrupted with the non-emergency noise data from the DNS dataset at SNR levels ranging from 0 dB to 20 dB with a step of 1. Similar to the speech enhancement module training, 90% of the data was used for training and 10% was used for validation. This dataset is denoted as *Smart Speech Enhancement Train Set*

To create a challenging test set, different speech and noise corpora were used in the testing process. Clean speech data were randomly selected from the Librispeech corpus (Panayotov et al., 2015), where 100 speech utterances for 5 male and 5 female

speakers were used. To create the *Smart Speech Enhancement Test Set*, these clean speech utterances were first mixed at 0 dB SNR with five emergency noises unseen during the training process, collected from the Mixkit website. Afterwards, this mixture was corrupted with 10 mismatched non-emergency noise environments, unseen during the training process, taken from the 100 Nonspeech Environmental Sounds dataset (Hu, 2014). These noise environments are: 9 crowd noises, including babble noise, and AWGN, -5 dB, 0 dB, and 5 dB are the used test SNRs, where -5 dB is an unseen SNR during training. To create the *Speech Enhancement Test Set*, the clean speech utterances were only corrupted with the mismatched non-emergency noise environments at the same testing SNR levels (-5 dB, 0 dB, and 5 dB).

The frontend binary noise classifier was trained and tested using the same datasets as the DCRN. The *Speech Enhancement Train Set* was labeled by binary 0, which runs DCRN in the speech enhancement mode; while the *Smart Speech Enhancement Train Set* was labeled by binary 1, which runs DCRN in the audio enhancement mode.

To generate speech audios with good quality, all the speech utterances were resampled to 16 kHz sampling frequency. Normalization to zero mean and unit variance was then applied to improve the training process. T-F features were extracted for the CNN classifier and the BCE loss function was used, as explained in Subsection 6.3.1. For the DCRN, audio framing was performed with 1,024 frame size and 50% overlap. We used MSE loss function and Adam optimizer, learning rate = 0.0001, β_1 = 0.1, β_2 = 0.999. The used batch size is 4, and the number of epochs is 20, which was found to be sufficient for the network to converge. The final weights were taken based on the validation data, to avoid network overfitting.

6.3.4 Results and Discussion

Evaluation was performed for the proposed architecture for normal hearing and hearing-impaired listeners, considering that the architecture is designed specifically for hearing aids. For normal hearing listeners, the PESQ and STOI evaluation metrics were used to evaluate speech quality and intelligibility, respectively. Moreover, further evaluations were conducted using the composite speech quality measures, Csig, to measure the speech signal quality in terms of speech distortion; Cbak, to measure noise intrusiveness; and Covl, to measure the overall quality of the processed speech. All these evaluation metrics were fully described in Chapter 3.

For hearing loss, we used the Hearing-Aid Speech Quality Index (HASQI) (Kates and Arehart, 2010) (from 0 to 1) and the Hearing-Aid Speech Perception Index (HASPI) (Kates and Arehart, 2021) (from 0 to 1) to measure speech quality and intelligibility, respectively. The Hearing-Aid Audio Quality Index (HAAQI) (Kates and Arehart, 2015) (from 0 to 1) is also used to measure the quality of the output speech with emergency

noise audio. For all these evaluation metrics higher values indicate better speech quality.

6.3.4.1 Speech Enhancement Model Comparison to Baselines

An experiment was first conducted to evaluate the performance of the DCRN speech enhancement model, DCRN$_{SE}$, in comparison with SOTA speech enhancement models in the literature using the Valentini Voice Bank dataset benchmark (Valentini-Botinhao et al., 2017*b*). The results also include the performance of the first stage DE-CADE in the frequency domain, DE-CADE(F), and the two-stage DE-CADE architecture, DE-CADE(F-T), presented in Chapter 5.

The outcome of this comparison is presented in Table 6.7, in which the DCRN$_{SE}$ outperforms all models with respect to the PESQ and Covl scores, expect the previously proposed two-stage speech enhancement architecture presented in Chapter 5, DE-CADE(F-T). As mentioned before in Subsection 6.3.2, the developed DCRN was designed in this chapter for hearing aids, which is an application that requires DNNs with fewer network parameters to fit onto the device hardware. For this reason, the DCRN performs worse than the two-stage DE-CADE(F-T); however, the DCRN is less complex with only 7 million parameters in comparison to the DE-CADE(F-T), which has 12.6 million parameters. The DCRN also shows better performance than the single stage DE-CADE(F), without significant increase in network's parameters; DCRN has 0.7 million parameters more than DE-CADE(F). It should be noted that the STOI results were not reported by the authors of Wave U-Net (Macartney and Weyde, 2018), Metric-GAN (Fu et al., 2019), SEGAN-D (Phan et al., 2020), Koizumi et al. (Koizumi et al., 2020), and T-GSA (Kim et al., 2020); for this reason, they are not included in Table 6.7.

6.3.4.2 Smart Speech Enhancement Architecture Performance

Further evaluations were conducted to assess the performance of the full smart speech enhancement architecture, given in Figure 6.4, for normal and hearing-impaired listeners. The performance of the speech enhancement and audio enhancement processing will be presented and discussed in the following subsections.

Speech Enhancement Evaluation

This evaluation is to assess the performance of the speech enhancement module after adding the CNN noise classifier. In order to perform this evaluation, the smart speech enhancement architecture was tested using the *Speech Enhancement Test Set*, described in Subsection 6.3.3, in which the speech signal is only corrupted with non-emergency noise. The classifier accuracy for this test set is 90%, and the evaluation of the quality of the processed speech by the architecture is shown in Table 6.8. PESQ and STOI

Table 6.7 Performance comparison with SOTA speech enhancement models using the Valentini Voice Bank dataset benchmark (Valentini-Botinhao et al., 2017*b*).

Metric	PESQ	STOI	Csig	Cbak	Covl
Noisy	1.97	91.5	3.35	2.44	2.63
Wiener Scalart et al. (1996)	2.22	92.0	3.23	2.68	2.67
SEGAN Pascual et al. (2017)	2.16	93.0	3.48	2.94	2.80
Wave U-Net Macartney and Weyde (2018)	2.40	-	3.52	3.24	2.96
MMSE-GAN Soni et al. (2018)	2.53	93.0	3.80	3.12	3.14
Deep Xi-ResLSTM Nicolson and Paliwal (2019)	2.65	91.0	4.01	3.25	3.34
Metric-GAN Fu et al. (2019)	2.86	-	3.99	3.18	3.42
SEGAN-D Phan et al. (2020)	2.39	-	3.46	3.11	3.50
DEMUCS Défossez et al. (2020)	3.07	**95.0**	4.14	3.21	3.54
Koizumi et al. Koizumi et al. (2020)	2.99	-	4.15	3.42	3.57
T-GSA Kim et al. (2020)	3.06	-	4.18	**3.59**	3.62
Deep MMSE Zhang et al. (2020)	2.95	94.0	**4.28**	3.46	3.64
DE-CADE(F)	3.21	93.4	4.00	3.11	3.60
$DCRN_{SE}$	3.29	93.5	4.18	2.96	3.76
DE-CADE(F-T)	3.38	93.8	**4.36**	3.01	**3.86**

scores were used for normal listeners, while HASQI and HASPI were used for hearing-impaired listeners, considering two hearing loss degrees: Mild hearing loss (HL1) and Moderate hearing loss (HL2). The values of the hearing loss degree were taken from the real Occupational Hearing Loss (OHL) Worker Surveillance Data (Masterson et al., 2013), which is a dataset used to estimate the prevalence of hearing loss among U.S. industries. Hearing loss data was randomly selected for 100 workers, 50 males and 50 females, for mild and moderate hearing loss cases, 50 values for each.

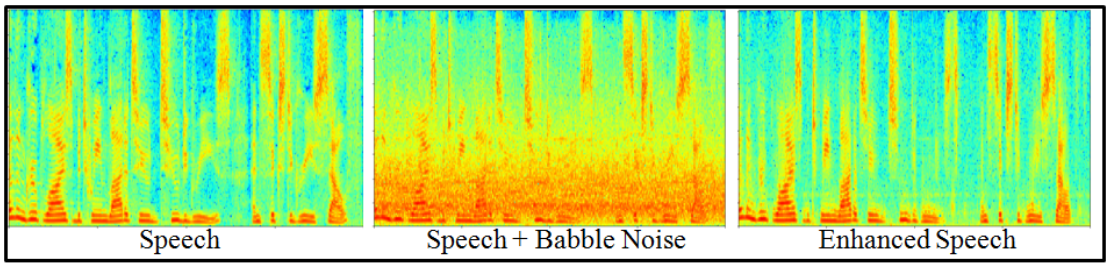
The presented results in Table 6.8 are the average of the three test SNRs: -5 dB, 0 dB, and 5 dB. This evaluation includes the scores of the unprocessed speech; the speech enhancement network, $DCRN_{SE}$; and the smart speech enhancement network, $DCRN_{SSE}$. The results show that both networks improve the quality and intelligibility of the speech for both normal and hearing-impaired listeners. The smart speech enhancement architecture, $DCRN_{SSE}$, generates estimated speech with slightly worse quality and intelligibility compared to the speech enhancement network, $DCRN_{SE}$. The reason for this degradation in performance is the failure of the classifier to classify some challenging undesired crowd noise used in testing. Consequently, the noisy speech will be processed with the audio enhancement network, based on the wrong decision of the classifier in this case. This negatively affects the denoising capability of the architecture for non-emergency noise environments, because the DCRN was trained to output emergency noise with speech, resulting in more background noise.

The spectrograms in Figure 6.6(a) show the performance of the speech enhancement module of the smart speech enhancement architecture when tested using speech

corrupted with undesired, non-emergency babble noise, not previously seen in the training process, at 0 dB SNR. The classifier outputs the correct decision for this test case, to run the architecture in speech enhancement mode. When comparing the enhanced speech to the original speech spectrogram, it is clear that the network managed to eliminate most of the challenging babble noise, proving its effective denoising ability.

Table 6.8 Speech enhancement performance of the architecture for normal and hearing-impaired listeners

Metric	Normal Hearing		Hearing Loss			
	PESQ	STOI	HASQI		HASPI	
			HL1	HL2	HL1	HL2
Unprocessed	1.57	70	0.37	0.24	70	65
$DCRN_{SE}$	2.12	77	0.57	0.38	76	70
$DCRN_{SSE}$	2.00	76	0.56	0.36	75	68



a) Speech Enhancement Model Performance

b) Audio Enhancement Model Performance

Figure 6.6 The performance of the proposed architecture, a) speech enhancement mode, b) audio enhancement mode

Audio Enhancement Evaluation

This evaluation is to assess the performance of the audio enhancement module of the smart speech enhancement architecture, which is used to generate speech and important emergency noise. The *Smart Speech Enhancement Test Set*, described in Subsection 6.3.3, was used to perform this evaluation, and the HAAQI score was used to assess the quality of the enhanced speech and emergency noise audio. The classifier accuracy for this test set is 95%. The outcome of this experiment is shown in Table 6.9, where the presented results are the average of the three test SNRs: -5 dB, 0 dB, and 5 dB.

For both mild and moderate hearing loss degrees, it can be seen that the smart speech enhancement architecture, $DCRN_{SSE}$, generates processed audio with a quality compared to the unprocessed audio, which proves the applicability of the presented architecture for hearing aids.

The spectrograms in Figure 6.6(b) show the performance of the audio enhancement model when tested using speech with fire alarm emergency noise corrupted with undesired challenging babble noise, not previously seen in the training process, at 0 dB SNR. These diagrams shows that the audio enhancement model mitigates undesired babble noise while trying to generate both speech and emergency fire alarm noise.

Table 6.9 Performance of the architecture for speech and emergency noise enhancement

Metric	**HAAQI**	
	HL1	**HL2**
Unprocessed	0.21	0.16
$DCRN_{SSE}$	0.44	0.34

6.4 Conclusion

This chapter covers two popular speech enhancement applications: hearing aids and ASR. The chapter presents two speech enhancement DNN architectures for these applications, based on the developed DE-CADE architecture, presented in Chapter 5. The DE-CADE architecture was modified and optimized in this chapter to improve its performance, in order to finally adapt to each application. For ASR, an architecture was developed to solve the mismatch issue between speech enhancement models and ASR models, which is a current research question. The architecture minimizes speech distortion, which is the main cause of this problem; moreover, a prepossessing SNR classifier was added to achieve further improvement. For hearing aids, a smart speech enhancement technique was developed, which enhances both speech and important emergency noise, in order ensure the safety of hearing aids users in emergency situations. The obtained results show promising performance for the modified architectures in the case of the two applications under investigation. This proves the applicability of the presented work in this PhD thesis.

CHAPTER 7

Conclusions and Recommendations

7.1 Introduction

In this chapter, concluding summary of thesis chapters is first presented, followed by conclusions of this PhD thesis using critical analysis for deep learning-based supervised speech enhancement through Strengths, Weaknesses, Opportunities, Challenges (SWOC) analysis. This is a general analysis that highlights the Strengths, Weaknesses, Opportunities, and Challenges of this approach, in order to finally determine its position in the signal processing field. This analysis identifies the strength of this approach and why it is a hot topic. It also presents the current issues of the technique and the future investigations needed, through weaknesses and challenges, respectively. Furthermore, it suggests opportunities for the approach to develop in the future, by discussing ideas that may lead to further improvements in the research area. Fig. 7.1 shows the SWOC matrix and a detailed explanation is presented in the following sections. This chapter also provides summary of thesis contributions, and concluding summary of thesis chapters. Finally, recommendations for future will be provided.

7.2 Concluding Summary

The main conclusions of each chapter are summarised as follows:

Chapter 1 provides introduction to the thesis, and defines thesis aims, objectives, and contributions.

Chapter 2 presented a review of speech enhancement approaches, focusing on deep learning-based speech enhancement. This review gives a brief discussion about classical speech enhancement techniques; afterwards, a detailed illustration was presented to the modern deep learning speech enhancement techniques. The chapter covers different DNN architectures, and the strengths and weaknesses for each architecture type. Moreover, it summarized the progress of deep learning-based speech enhancement, and how different approaches contribute to improving the performance.

This review was important to have a deep knowledge of the research field, understanding the current issues, and defining the research question of the thesis.

Chapter 3 showed the procedure needed to develop a DNN for speech enhancement through dividing it into six steps. Each step was described separately, and discussion was given on the different approaches that can be followed in each step. This review was essential to have an organized and systematic procedure to follow, in order to develop a new speech enhancement model.

Chapter 4 presented an experimental analysis to seven best performing DNN architectures in the literature. Moreover, it investigates the effect of different techniques and approaches on the performance. The main conclusions of this chapter are as follows. Deep CDAE is the best performing DNN architecture for speech enhancement in both the frequency and time domains, in comparison to MLPs, DDAEs, CNNs, FCNNs, and shallow CDAE. The use of 1D strided convolution layers with PReLU activations improves the performance. Although deep learning is a data driven approach, feature extraction has a great positive impact on the performance. Mapping targets also show better generalization ability to mismatched test data than masking targets.

Chapter 5 presented a new architecture and two-stage speech enhancement approach that minimizes speech distortion by applying speech denoising in the frequency domain, and then speech reconstruction in the time domain. The results showed that the two-stage speech enhancement architecture outperforms SOTA speech enhancement models in the literature.

Chapter 6 investigated two speech enhancement applications, hearing aids and ASR. The architecture presented in Chapter 5 was modified and optimized to apply smart speech enhancement for hearing aids, and to be applied as a preprocessing stage to ASR systems to reduce WER in noisy environments. The results showed promising performance for the developed architectures for both applications, proving the possibility of applying this work in the real world.

Chapter 7 presents critical SWOC analysis to deep learning-based speech enhancement, concludes the thesis, and provides recommendations for future research.

7.3 SWOC Analysis

In this section, a SWOC analysis will be presented to deep learning-based speech enhancement. Figure 7.1 shows the SWOC matrix.

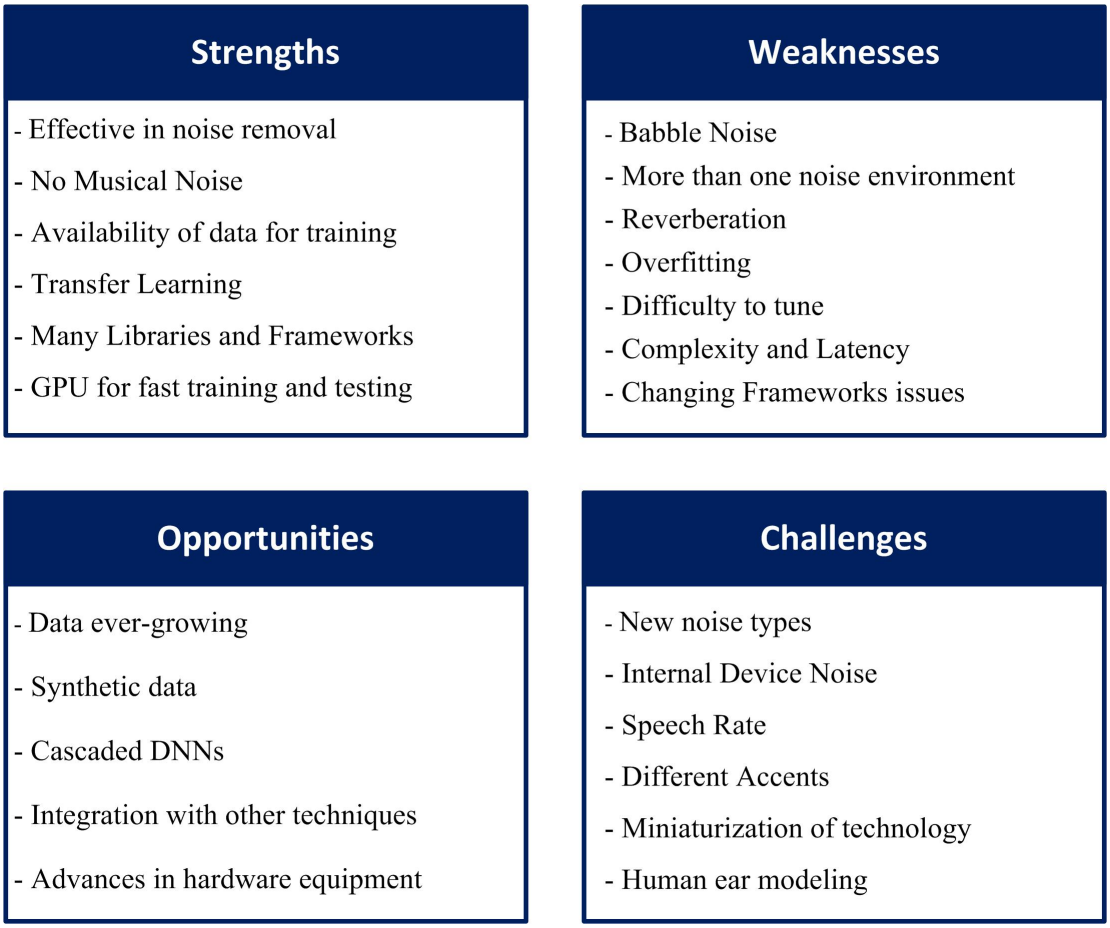


Figure 7.1 SWOC matrix

7.3.1 Strengths

In comparison to the classical techniques, deep learning-based speech enhancement is more efficient at removing background noise, especially at very low SNRs. Additionally, deep learning-based speech enhancement generates speech with better quality and intelligibility (Wang and Chen, 2018). Moreover, deep learning-based speech enhancement managed to solve the well-known problem of musical noise (Uemura et al., 2009) for the classical techniques, especially the spectral subtraction method. Musical noise, a remnant, unnatural noise that accompanies the processed speech, is the main drawback of classical speech enhancement techniques when applied in devices such as hearing aids, as it causes unsatisfactory performance for customers (McCormack and Fortnum, 2013). Therefore, solving this issue can be considered as a real advantage to the modern speech enhancement approach.

Over the previous decades, there has been a massive increase in the number of speech and noise datasets available online. This data will facilitate the development of more efficient DNNs for speech enhancement, as it is essential in the training process and in improving the generalization ability of networks. Furthermore, transfer learning (Wang and Zheng, 2015) is one of the techniques that makes the training of speech

enhancement DNNs more efficient. This technique allows the reuse of a network pre-trained with a huge amount of data for a certain task as a starting point to another task. In this case, the speech enhancement network can be fine-tuned with a small amount of data to perform the required task. This is helpful when training the network on different languages, as the collection of a huge dataset for every language is not always necessary (Xu et al., 2014*a*).

The implementation and training of DNNs is now much easier with the high availability of deep learning libraries and frameworks, which facilitate development and testing. Additionally, GPU-based hardware equipment, which is readily available now, results in faster DNN training and real-time testing.

7.3.2 Weaknesses

Interference or babble noise, which is one or more speech signals accompanying the target speech, is one of the most challenging type of background noise for speech enhancement. Although some DNNs managed to almost completely remove different noise types, the algorithm cannot effectively deal with interference or babble noise. A clear degradation in network performance happens when testing it using babble noise, this is because the DNN lacks the ability to separate different speech sources, as it is trained to differentiate between noise and speech features. Another drawback is that the efficiency of the technique is negatively affected when more than one background noise exists, which is the most common situation in the real world; while, most speech enhancement DNN evaluations in the literature are based on one background noise. This will lead to poorer performance than reported in the literature. Reverberation was also proven to have a significant negative impact on the performance of DNNs for speech enhancement. As a result, there should be specific techniques and adjustment to be added to DNN training in order to deal with these challenging noise conditions.

Another weakness point is network overfitting, which is related to the training process of DNNs. As deep learning is a data-driven approach, the network usually performs better on data similar to the one used in the training process; however, it can fail to generalize this performance on mismatched data (Lawrence and Giles, 2000). This will result in uncertain performance of DNNs in real-time and the output might be unpredictable. Furthermore, DNNs are difficult to tune, as changing the network's parameters or data structure highly affects the training process.

The complexity and huge computational cost of deep learning techniques act as a barrier when trying to implement the technique in real-time. Hardware and memory restrictions for specific applications may restrict the applicability of this approach. Testing processing time is another complexity related issue that will be an obstacle for real-time applications. Although very fast GPUs can be used to speed up processing

times, their usage will result in higher product costs, decreasing its affordability. A final weakness, related to DNN implementation, is changing the framework used in training for real-time implementation purposes (Bahrampour et al., 2015). Frameworks allow faster and easier development of DNNs; however, the performance might be affected by changing the framework.

7.3.3 Opportunities

Recently, the size of the available data has exponentially increased, and it is expected that this increase will continue (Dytman-Stasienko and Weglinska, 2018). The performance of deep learning-based speech enhancement techniques will be positively affected by this ever-growing data, and this will result in improving the quality and intelligibility of the generated speech. Furthermore, data synthesis is another approach that can be used in the case of data scarcity, and it is proven to improve networks' performance (Tremblay et al., 2018). Synthetic data is also solving the issues related to real data privacy and restricted use regulations and provides more flexibility in manipulating data and creating challenging conditions to learn in the training process, which will finally result in improved performance (Barbosa et al., 2018). The cascaded approach for speech enhancement networks was also shown to positively impact the performance and some DNN combinations have shown promising results (Zhao, Zarar, Tashev and Lee, 2018; Tan and Wang, 2018; Phan et al., 2020); while, other combinations have not been visited yet which may also improve the overall performance. The integration of deep learning and other techniques, such as reinforcement learning (Koizumi et al., 2017) and non-negative matrix factorization (Vu et al., 2016), is another field that opens the opportunity for enhancing the performance of deep learning techniques.

With the continuous advances in technology and hardware equipment, it will be possible to improve computation costs and latency, or solve long processing time problems for deep learning techniques. This will open the opportunity for deep learning-based speech enhancement to invade the marketplace (Pan et al., 2018).

7.3.4 Challenges

The continuous invention of new machines, equipment, transportation, electronic devices, etc. will lead to the introduction of new environmental noise, which may further increase noise levels. This will be very challenging for deep learning techniques to deal with, and may negatively affect performance. Moreover, electronic devices and machines have internally generated noise (Teel, 2005), which are rarely studied in the literature. These internal noises are unpredictable and vary among different devices, so it may cause significant performance degradation in real-time implementations (Cameron et al., 1992). Different speech rates for different speakers acts as another challenge

to deep learning-based speech enhancement techniques, as it has specific patterns or features that may be confusing for the DNN to process. Additionally, the wide range of accents for every language makes the speech enhancement task more challenging, because this can result in different phonemes that the network did not learn during the training process. Although our brain has a great ability to amend and understand these incorrect phonemes or pronunciation, it is not granted that a computer algorithm will have the same capability to deal with this issue.

The miniaturization of technology (Peercy, 2000) is a trend that may act as an obstacle when developing deep learning techniques. The continuous need to shrink electronic devices to be more efficient and portable is very challenging for complex techniques like deep learning-based speech enhancement because device miniaturization may restrict performance improvement and the technique's applicability. As a result, applying a deep learning approach for speech enhancement may not cope with customers' needs for smaller devices.

The mathematical modeling of phenomena or processes has been greatly advanced (Tomlin and Axelrod, 2007), where computer modelling and simulation are used to mimic certain functionality. Human ear modelling gained attention for decades (Lyon, 1982), and more advanced models are being developed to simulate the complex functions of the human ear (Givelberg and Bunn, 2003). The fact that these models are more understandable and controllable than deep learning-based speech enhancement techniques makes them real competitors, as the deep learning approach is still ambiguous. Although there is a study that shows that combining the two approaches leads to better performance (Baby and Verhulst, 2018), developing a good mathematical model for simulating human ear physiology threatens the existence of deep learning-based speech enhancement techniques.

7.4 Summary of Thesis Contributions

The contributions of this thesis can be summarized as follows.

- Investigating deep neural networks for speech enhancement through experimental analysis, to identify the factors affecting the performance, which enables the development of better speech enhancement architecture.
- Developing a new deep learning-based architecture for speech enhancement that outperforms SOTA speech enhancement models in the literature.
- Proposing a new deep learning two-stage approach for speech enhancement that takes advantage of time and frequency domain features, which minimizes speech distortion.

- Optimizing the developed architecture and applying it to improve the performance of two real time speech enhancement applications, hearing aid and ASR.
- Providing critical SWOC analysis to the deep learning approach for speech enhancement

7.5 Recommendations for Future Research

The analysis performed in Chapter 4 can be expanded by investigating recent two-stage speech enhancement complex approaches and the recently proposed loss functions that aim to minimize distortion or maximize a specific speech quality evaluation metric.

Further investigation is needed to the two-stage speech enhancement approach, proposed in Chapter 5, which preforms consecutive speech enhancement processing in the frequency then in the time domain. In this thesis, the approach utilizes the same DNN architecture in both stages. Although the approach was tested and validated using different DNNs in the first stage and the results show promising performance, the use of different DNN combinations in the training process as well may lead to further improvement. As a result, future work is needed to train and test this two-stage approach using dissimilar DNNs in the training process.

Further improvement is needed to the accuracy of the CNN binary noise classifier used in the smart speech enhancement architecture, proposed in Chapter 6. This can be achieved by increasing the dataset size of the emergency and non-emergency noise, or developing better DNN to perform the classification. On the other hand, the classification accuracy of the SNR binary classifier of the architecture proposed in the same chapter for ASR also requires improvement for SNR values near the decision boundary. A possible solution to improve this accuracy is to increase the size of noisy speech data around the boundary SNR level and manipulate this data to create challenging training conditions for the DNNs, which can improve the learning process.

REFERENCES

Abd El-Fattah, M., Dessouky, M. I., Diab, S. M. and Abd El-Samie, F. E.-S. (2008), 'Speech enhancement using an adaptive wiener filtering approach', *Progress in Electromagnetics Research* **4**, 167–184.

Alam, M. J. and O'Shaughnessy, D. (2011), 'Perceptual improvement of wiener filtering employing a post-filter', *Digital Signal Processing* **21**(1), 54–65.

Alberti, G. S. and Ammari, H. (2017), 'Disjoint sparsity for signal separation and applications to hybrid inverse problems in medical imaging', *Appl. Comput. Harmon. A.* **42**(2), 319–349.

Alghamdi, N., Maddock, S., Marxer, R., Barker, J. and Brown, G. J. (2018), 'A corpus of audio-visual lombard speech with frontal and profile views', *J. Acoust. Soc. Am.* **143**(6), EL523–EL529.

Alías, F., Socoró, J. C. and Sevillano, X. (2016), 'A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds', *Applied Sciences* **6**(5), 143.

Alshuaib, W. B., Al-Kandari, J. M. and Hasan, S. M. (2015), 'Classification of hearing loss', *Update On Hearing Loss* **4**, 29–37.

Amehraye, A., Pastor, D. and Tamtaoui, A. (2008), Perceptual improvement of wiener filtering, *in* '2008 IEEE International Conference on Acoustics, Speech and Signal Processing', IEEE, pp. 2081–2084.

Angelov, P. and Sperduti, A. (2016), 'Challenges in deep learning'.

Arel, I., Rose, D. C. and Karnowski, T. P. (2010), 'Deep machine learning-a new frontier in artificial intelligence research [research frontier]', *IEEE computational intelligence magazine* **5**(4), 13–18.

Atal, B. S. (2003), Speech synthesis based on linear prediction, *in* R. A. Meyers, ed., 'Encyclopedia of Physical Science and Technology (Third Edition)', third edition edn, Academic Press, New York, pp. 645–655.
URL: *https://www.sciencedirect.com/science/article/pii/B0122274105007201*

Atal, B. S. and Hanauer, S. L. (1971), 'Speech analysis and synthesis by linear prediction of the speech wave', *The journal of the acoustical society of America* **50**(2B), 637–655.

Baby, D. (2020), 'ISEGAN: Improved speech enhancement generative adversarial networks', *arXiv preprint arXiv:2002.08796* .

Baby, D. and Verhulst, S. (2018), Biophysically-inspired features improve the generalizability of neural network-based speech enhancement systems, *in* 'INTERSPEECH', ISCA.

Bahrampour, S., Ramakrishnan, N., Schott, L. and Shah, M. (2015), 'Comparative study of deep learning software frameworks', *arXiv* .

Barbosa, I. B., Cristani, M., Caputo, B., Rognhaugen, A. and Theoharis, T. (2018), 'Looking beyond appearances: Synthetic training data for deep CNNs in re-identification', *Comput. Vis. Image Und.* **167**, 50–62.

Barker, J., Marxer, R., Vincent, E. and Watanabe, S. (2015), The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines, *in* 'IEEE Workshop on ASR and Understanding (ASRU)', IEEE, pp. 504–511.

Bartsch, M. A. and Wakefield, G. H. (2005), 'Audio thumbnailing of popular music using chroma-based representations', *IEEE Transactions on multimedia* **7**(1), 96–104.

Bengio, Y. (2012), Practical recommendations for gradient-based training of deep architectures, *in* 'Neural networks: Tricks of the trade', Springer, pp. 437–478.

Bengio, Y., Simard, P., Frasconi, P. et al. (1994), 'Learning long-term dependencies with gradient descent is difficult', *IEEE transactions on neural networks* **5**(2), 157–166.

Beritelli, F., Casale, S., Russo, A. and Serrano, S. (2006), An automatic emergency signal recognition system for the hearing impaired, *in* '12th Digital Signal Processing Workshop and 4th IEEE Signal Processing Education Workshop', IEEE, pp. 179–182.

Blanchard, N., Brady, M., Olney, A. M., Glaus, M., Sun, X., Nystrand, M., Samei, B., Kelly, S. and D'Mello, S. (2015), A study of automatic speech recognition in noisy classroom environments for automated dialog analysis, *in* 'International conference on artificial intelligence in education', Springer, pp. 23–33.

Boll, S. (1979), 'Suppression of acoustic noise in speech using spectral subtraction', *IEEE Transactions on acoustics, speech, and signal processing* **27**(2), 113–120.

Boureau, Y.-L., Ponce, J. and LeCun, Y. (2010), A theoretical analysis of feature pooling in visual recognition, *in* 'Proceedings of the 27th international conference on machine learning (ICML-10)', pp. 111–118.

Braun, S. and Tashev, I. (2020), Data augmentation and loss normalization for deep noise suppression, *in* 'International Conference on Speech and Computer', Springer, pp. 79–86.

Bronzino, J. D. (2000), *Biomedical Engineering Handbook 2*, Vol. 2, Springer Science and Business Media.

Brooks, D. N. (1994), 'Some factors influencing choice of type of hearing aid in the UK: behind-the-ear or in-the-ear', *British Journal of Audiology* **28**(2), 91–98.

Cameron, D. E., Lang, J. H. and Umans, S. D. (1992), 'The origin and reduction of acoustic noise in doubly salient variable-reluctance motors', *IEEE Trans. Ind. Appl.* **28**(6), 1250–1255.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M. et al. (2005), The AMI meeting corpus: A pre-announcement, *in* 'International workshop on machine learning for multimodal interaction', Springer, pp. 28–39.

Chakrabarty, S., Wang, D. and Habets, E. A. (2018), Time-frequency masking based online speech enhancement with multi-channel data using convolutional neural networks, *in* 'IWAENC', IEEE, pp. 476–480.

Chen, J., Wang, Y. and Wang, D. (2014), 'A feature study for classification-based speech separation at low signal-to-noise ratios', *IEEE Trans. Audio Speech Lang. Proc.* **22**(12), 1993–2002.

Chen, J., Wang, Y., Yoho, S. E., Wang, D. and Healy, E. W. (2016), 'Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises', *The Journal of the Acoustical Society of America* **139**(5), 2604–2612.

Chen, Z., Huang, Y., Li, J. and Gong, Y. (2017), Improving mask learning based speech enhancement system with restoration layers and residual connection., *in* 'INTERSPEECH', pp. 3632–3636.

Cho, K., van Merriënboer, B., Bahdanau, D. and Bengio, Y. (2014), On the properties of neural machine translation: Encoder–decoder approaches, *in* 'Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation', pp. 103–111.

Coşkun, M., YILDIRIM, Ö., Ayşegül, U. and Demir, Y. (2017), 'An overview of popular deep learning methods', *European Journal of Technique* **7**(2), 165–176.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. and Bharath, A. A. (2018), 'Generative adversarial networks: An overview', *IEEE signal processing magazine* **35**(1), 53–65.

Dave, N. (2013), 'Feature extraction methods LPC, PLP and MFCC in speech recognition', *Int. j. Adv. Res. Eng. Tech.* **1**(6), 1–4.

Défossez, A., Synnaeve, G. and Adi, Y. (2020), 'Real time speech enhancement in the waveform domain', p. 3291–3295.

Deng, L., Yu, D. et al. (2014), 'Deep learning: methods and applications', *Foundations and trends® in signal processing* **7**(3–4), 197–387.

Dinghofer, K. and Hartung, F. (2020), Analysis of criteria for the selection of machine learning frameworks, *in* '2020 International Conference on Computing, Networking and Communications (ICNC)', IEEE, pp. 373–377.

Donahue, C., Li, B. and Prabhavalkar, R. (2018), Exploring speech enhancement with generative adversarial networks for robust speech recognition, *in* '2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 5024–5028.

Drullman, R. (1995), 'Speech intelligibility in noise: Relative contribution of speech elements above and below the noise level', *The Journal of the Acoustical Society of America* **98**(3), 1796–1798.

Du, J. and Huo, Q. (2008), A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions, *in* 'Conf. Int. Speech Comm. Assoc.'.

Du, J., Wang, Q., Gao, T., Xu, Y., Dai, L.-R. and Lee, C.-H. (2014), Robust speech recognition with speech enhanced deep neural networks, *in* 'Fifteenth annual conference of the international speech communication association'.

Dytman-Stasienko, A. and Weglinska, A. (2018), 'Big data: Digital marketing and trendwatching'.

El Hihi, S. and Bengio, Y. (1996), Hierarchical recurrent neural networks for long-term dependencies, *in* 'Advances in neural information processing systems', pp. 493–499.

Elements, E. (2019), 'Mixkit'.
URL: *https://mixkit.co/*

Eluyode, O. and Akomolafe, D. T. (2013), 'Comparative study of biological and artificial neural networks', *European Journal of Applied Engineering and Scientific Research* **2**(1), 36–46.

Ephraim, Y. and Van Trees, H. L. (1995), 'A signal subspace approach for speech enhancement', *IEEE Transactions on speech and audio processing* **3**(4), 251–266.

Erdogan, H., Hershey, J. R., Watanabe, S. and Le Roux, J. (2015), Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks, *in* 'ICASSP', IEEE, pp. 708–712.

Févotte, C., Gribonval, R. and Vincent, E. (2005), 'BSS_EVAL toolbox user guide–revision 2.0'.

Fu, S.-W., Liao, C.-F., Tsao, Y. and Lin, S.-D. (2019), MetricGAN: Generative adversarial networks based black-box metric scores optimization for speech enhancement, *in* 'Int. Conf. Mach. Learn.', PMLR, pp. 2031–2041.

Fu, S.-W., Tsao, Y. and Lu, X. (2016), SNR-aware convolutional neural network modeling for speech enhancement., *in* 'Interspeech', pp. 3768–3772.

Fu, S.-W., Tsao, Y., Lu, X. and Kawai, H. (2017), Raw waveform-based speech enhancement by fully convolutional networks, *in* 'APSIPA ASC', IEEE, pp. 006–012.

Fu, S.-W., Wang, T.-W., Tsao, Y., Lu, X. and Kawai, H. (2018), 'End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **26**(9), 1570–1584.

Gan, R. Z., Reeves, B. P. and Wang, X. (2007), 'Modeling of sound transmission from ear canal to cochlea', *Annals of biomedical engineering* **35**(12), 2180–2195.

Garbin, C., Zhu, X. and Marques, O. (2020), 'Dropout vs. batch normalization: an empirical study of their impact to deep learning', *Multimedia Tools and Applications* **79**(19), 12777–12815.

Garnier, M. and Henrich, N. (2014), 'Speaking in noise: How does the Lombard effect improve acoustic contrasts between speech and ambient noise?', *J. Comput. Speech Lang.* **28**(2), 580–597.

Germain, F. G., Chen, Q. and Koltun, V. (2019), Speech denoising with deep feature losses, *in* 'INTERSPEECH', pp. 2723–2727.

Giri, R., Isik, U. and Krishnaswamy, A. (2019), Attention wave-u-net for speech enhancement, *in* '2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)', IEEE, pp. 249–253.

Givelberg, E. and Bunn, J. (2003), 'A comprehensive three-dimensional model of the cochlea', *J. Comput. Phys.* **191**(2), 377–391.

Glorot, X. and Bengio, Y. (2010), Understanding the difficulty of training deep feedforward neural networks, *in* 'Proceedings of the thirteenth international conference on artificial intelligence and statistics', JMLR Workshop and Conference Proceedings, pp. 249–256.

Goehring, T., Bolner, F., Monaghan, J. J., Van Dijk, B., Zarowski, A. and Bleeck, S. (2017), 'Speech enhancement based on neural networks improves speech intelligibility in noise for cochlear implant users', *Hearing research* **344**, 183–194.

Goh, Z., Tan, K.-C. and Tan, T. (1998), 'Postprocessing method for suppressing musical noise generated by spectral subtraction', *IEEE Transactions on Speech and Audio Processing* **6**(3), 287–292.

Gong, Y. (1995), 'Speech recognition in noisy environments: A survey', *Speech communication* **16**(3), 261–291.

Gonzalez, R. C. (2018), 'Deep convolutional neural networks [lecture notes]', *IEEE Signal Processing Magazine* **35**(6), 79–87.

Grais, E. M. and Plumbley, M. D. (2017), Single channel audio source separation using convolutional denoising autoencoders, *in* '2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)', IEEE, pp. 1265–1269.

Grézl, F., Karafiát, M., Kontár, S. and Cernocky, J. (2007), Probabilistic and bottle-neck features for LVCSR of meetings, *in* '2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07', Vol. 4, IEEE, pp. IV–757.

Griffin, D. and Lim, J. (1984), 'Signal estimation from modified short-time fourier transform', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **32**(2), 236–243.

Grossberg, S. (1988), 'Nonlinear neural networks: Principles, mechanisms, and architectures', *Neural networks* **1**(1), 17–61.

Gutiérrez-Muñoz, M. and Coto-Jiménez, M. (2022), 'An experimental study on speech enhancement based on a combination of wavelets and deep learning', *Computation* **10**(6), 102.

Gveres (2015), 'Donate-a-cry corpus', *[Online]. Available: https://github.com/gveres/donateacry-corpus* .

Hamacher, V., Kornagel, U., Lotter, T. and Puder, H. (2008), 'Binaural signal processing in hearing aids: Technologies and algorithms', *Advances in digital speech transmission* **14**, 401–429.

Hansen, J. H. and Pellom, B. L. (1998), An effective quality evaluation protocol for speech enhancement algorithms, *in* 'ICSLP'.

Hansen, P. S. K., Hansen, S. D. and Sørensen, J. A. (1998), 'Signal subspace methods for speech enhancement'.

Hartley, D. E. and Moore, D. R. (2003), 'Effects of conductive hearing loss on temporal aspects of sound transmission through the ear', *Hearing research* **177**(1-2), 53–60.

Haykin, S. and Chen, Z. (2005), 'The cocktail party problem', *Neural Comput.* **17**(9), 1875–1902.

He, K. and Sun, J. (2015), Convolutional neural networks at constrained time cost, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 5353–5360.

Healy, E. W., Delfarah, M., Vasko, J. L., Carter, B. L. and Wang, D. (2017), 'An algorithm to increase intelligibility for hearing-impaired listeners in the presence of a competing talker', *The Journal of the Acoustical Society of America* **141**(6), 4230–4239.

Hermansky, H. (1990), 'Perceptual linear predictive (PLP) analysis of speech', *the Journal of the Acoustical Society of America* **87**(4), 1738–1752.

Hermus, K., Wambacq, P. et al. (2006), 'A review of signal subspace speech enhancement and its application to noise robust speech recognition', *EURASIP Journal on Advances in Signal Processing* **2007**(1), 045821.

Heymann, J., Drude, L. and Haeb-Umbach, R. (2016), 'Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition', *Computer Speech and Language* .

Hinton, G. E. and Salakhutdinov, R. R. (2006), 'Reducing the dimensionality of data with neural networks', *science* **313**(5786), 504–507.

Hosseini-Asl, E., Zurada, J. M. and Nasraoui, O. (2015), 'Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints', *IEEE transactions on neural networks and learning systems* **27**(12), 2486–2498.

Houtgast, T. (1981), 'The effect of ambient noise on speech intelligibility in classrooms', *Applied Acoustics* **14**(1), 15–25.

Hu, G. (2014), '100 nonspeech environmental sounds', *[Online]. Available: http://www.cse.ohiostate.edu /pnl/corpus/HuCorpus.html* .

Hu, Y. and Loizou, P. C. (2003), 'A generalized subspace approach for enhancing speech corrupted by colored noise', *IEEE Transactions on Speech and Audio Processing* **11**(4), 334–341.

Hu, Y. and Loizou, P. C. (2007*a*), 'Evaluation of objective quality measures for speech enhancement', *IEEE Trans. Audio Speech Lang. Proc.* **16**(1), 229–238.

Hu, Y. and Loizou, P. C. (2007*b*), 'Subjective comparison and evaluation of speech enhancement algorithms', *Speech communication* **49**(7-8), 588–601.

Huang, P.-S., Kim, M., Hasegawa-Johnson, M. and Smaragdis, P. (2014), Deep learning for monaural speech separation, *in* 'ICASSP', IEEE, pp. 1562–1566.

Hudspeth, A. J. (1989), 'How the ear's works work', *Nature* **341**(6241), 397–404.

Hunter, D., Yu, H., Pukish III, M. S., Kolbusz, J. and Wilamowski, B. M. (2012), 'Selection of proper neural network sizes and architectures—a comparative study', *IEEE Trans. Ind. Inform.* **8**(2), 228–240.

Ioffe, S. and Szegedy, C. (2015), Batch normalization: Accelerating deep network training by reducing internal covariate shift, *in* 'International conference on machine learning', PMLR, pp. 448–456.

Issa, D., Demirci, M. F. and Yazici, A. (2020), 'Speech emotion recognition with deep convolutional neural networks', *Biomedical Signal Processing and Control* **59**, 101894.

Itu, T. (1996), 'ITU-T recommendation p. 800. methods for objective and subjective assessment of quality'.

Iwamoto, K., Ochiai, T., Delcroix, M., Ikeshita, R., Sato, H., Araki, S. and Katagiri, S. (2022), 'How bad are artifacts?: Analyzing the impact of speech enhancement errors on ASR', *arXiv preprint arXiv:2201.06685* .

Jain, A. K., Mao, J. and Mohiuddin, K. M. (1996), 'Artificial neural networks: A tutorial', *Computer* **29**(3), 31–44.

Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A. and Weyde, T. (2017), Singing voice separation with deep u-net convolutional networks, *in* '18th International Society for Music Information Retrieval Conference', pp. 23–27.

Johnson, C. E., Danhauer, J. L., Ellis, B. B. and Jilla, A. M. (2016), 'Hearing aid benefit in patients with mild sensorineural hearing loss: a systematic review', *Journal of the American Academy of Audiology* **27**(04), 293–310.

Johnsson, L.-G. and Hawkins Jr, J. E. (1972), 'Sensory and neural degeneration with aging, as seen in microdissections of the human inner ear', *Annals of Otology, Rhinology and Laryngology* **81**(2), 179–193.

Kamath, S. and Loizou, P. (2002), A multi-band spectral subtraction method for enhancing speech corrupted by colored noise., *in* 'ICASSP', Vol. 4, Citeseer, pp. 44164–44164.

Karlik, B. and Olgac, A. V. (2011), 'Performance analysis of various activation functions in generalized MLP architectures of neural networks', *International Journal of Artificial Intelligence and Expert Systems* **1**(4), 111–122.

Kates, J. M. and Arehart, K. H. (2010), 'The hearing-aid speech quality index (HASQI)', *Journal of the Audio Engineering Society* **58**(5), 363–381.

Kates, J. M. and Arehart, K. H. (2015), 'The hearing-aid audio quality index (HAAQI)', *IEEE/ACM transactions on audio, speech, and language processing* **24**(2), 354–365.

Kates, J. M. and Arehart, K. H. (2021), 'The hearing-aid speech perception index (HASPI) version 2', *Speech Communication* **131**, 35–46.

Kim, C. and Stern, R. M. (2016), 'Power-normalized cepstral coefficients (PNCC) for robust speech recognition', *IEEE Trans. Audio Speech Lang. Proc.* **24**(7), 1315–1329.

Kim, H. H. and Barrs, D. M. (2006), 'Hearing aids: a review of what's new', *Otolaryngology—Head and Neck Surgery* **134**(6), 1043–1050.

Kim, J., El-Khamy, M. and Lee, J. (2020), T-GSA: Transformer with gaussian-weighted self-attention for speech enhancement, *in* 'ICASSP', IEEE, pp. 6649–6653.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D. J. (2021), '1D convolutional neural networks and applications: A survey', *Mechanical systems and signal processing* **151**, 107398.

Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O. and Gabbouj, M. (2019), 1-D convolutional neural networks for signal processing applications, *in* 'ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 8360–8364.

Kjems, U., Boldt, J. B., Pedersen, M. S., Lunner, T. and Wang, D. (2009), 'Role of mask pattern in intelligibility of ideal binary-masked noisy speech', *J. Acoust. Soc. Am.* **126**(3), 1415–1426.

Koizumi, Y., Niwa, K., Hioka, Y., Kobayashi, K. and Haneda, Y. (2017), DNN-based source enhancement self-optimized by reinforcement learning using sound quality measurements, *in* '2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 81–85.

Koizumi, Y., Yatabe, K., Delcroix, M., Masuyama, Y. and Takeuchi, D. (2020), Speech enhancement using self-adaptation and multi-head self-attention, *in* 'ICASSP', IEEE, pp. 181–185.

Kolbæk, M., Tan, Z.-H., Jensen, S. H. and Jensen, J. (2020), 'On loss functions for supervised monaural time-domain speech enhancement', *IEEE Trans. Audio Speech Lang. Proc.* **28**, 825–838.

Kondo, K. (2012), *Subjective quality measurement of speech: its evaluation, estimation and applications*, Springer Science and Business Media.

Kounovsky, T. and Malek, J. (2017), Single channel speech enhancement using convolutional neural network, *in* '2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)', IEEE, pp. 1–5.

Kumar, A. and Florencio, D. (2016), 'Speech enhancement in multiple-noise conditions using deep neural networks', *arXiv preprint arXiv:1605.02427* .

Kumar, A., Solanki, S. S. and Chandra, M. (2022), 'Stacked auto-encoders based visual features for speech/music classification', *Expert Systems with Applications* **208**, 118041.

Lawrence, S. and Giles, C. L. (2000), Overfitting and neural networks: conjugate gradient and backpropagation, *in* 'IJCNN', Vol. 1, IEEE, pp. 114–119.

Le Roux, J., Wisdom, S., Erdogan, H. and Hershey, J. R. (2019), SDR–half-baked or well done?, *in* 'ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 626–630.

Lecun, P. S. (2012), 'Convolutional neural networks applied to housenumbers digit classification'.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), 'Deep learning', *nature* **521**(7553), 436–444.

Legouix, J. and Tarab, S. (1959), 'Experimental study of bone conduction in ears with mechanical impairment of the ossicles', *The Journal of the Acoustical Society of America* **31**(11), 1453–1457.

Lev-Ari, H. and Ephraim, Y. (2003), 'Extension of the signal subspace speech enhancement approach to colored noise', *IEEE Signal Processing Letters* **10**(4), 104–106.

Levitt, H. (2007), 'A historical perspective on digital hearing aids: how digital technology has changed modern hearing aids', *Trends in amplification* **11**(1), 7–24.

Li, R., Sun, X., Li, T. and Zhao, F. (2020), 'A multi-objective learning speech enhancement algorithm based on IRM post-processing with joint estimation of SCNN and TCNN', *Digital Signal Processing* **101**, 102731.

Li, Y., Sun, M. and Zhang, X. (2022), 'Perception-guided generative adversarial network for end-to-end speech enhancement', *Applied Soft Computing* **128**, 109446.

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. (2021), 'A survey of convolutional neural networks: analysis, applications, and prospects', *IEEE transactions on neural networks and learning systems* .

Liang, Z.-P. and Lauterbur, P. C. (1999), 'Principles of magnetic resonance imaging: A signal processing perspective'.

Liu, D., Smaragdis, P. and Kim, M. (2014), Experiments on deep learning for speech denoising, *in* 'Fifteenth Annual Conference of the International Speech Communication Association'.

Liu, R., Li, Y., Tao, L., Liang, D. and Zheng, H.-T. (2022), 'Are we ready for a new paradigm shift? a survey on visual deep MLP', *Patterns* **3**(7), 100520.

Loizou, P. C. (2013), *Speech enhancement: theory and practice*, CRC press.

Loizou, P. C. and Kim, G. (2010), 'Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions', *IEEE transactions on audio, speech, and language processing* **19**(1), 47–56.

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

Lyon, R. (1982), A computational model of filtering, detection, and compression in the cochlea, *in* 'ICASSP', Vol. 7, IEEE, pp. 1282–1285.

Lyon, R. (1983), A computational model of binaural localization and separation, *in* 'ICASSP', Vol. 8, IEEE, pp. 1148–1151.

Maas, A. L., Hannun, A. Y., Ng, A. Y. et al. (2013), Rectifier nonlinearities improve neural network acoustic models, *in* 'Proc. icml', Vol. 30, Citeseer, p. 3.

Macartney, C. and Weyde, T. (2018), 'Improved speech enhancement with the wave-u-net', *arXiv* .

Maganti, H. K. and Matassoni, M. (2010), An auditory based modulation spectral feature for reverberant speech recognition, *in* 'Eleventh Annual Conf. Int. Speech Comm. Assoc.'.

Malca, Y. and Wulich, D. (1996), Improved spectral subtraction for speech enhancement, *in* 'EUSIPCO', IEEE, pp. 1–5.

Mallat, S. G. (1989), 'A theory for multiresolution signal decomposition: the wavelet representation', *IEEE transactions on pattern analysis and machine intelligence* **11**(7), 674–693.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z. and Paul Smolley, S. (2017), Least squares generative adversarial networks, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 2794–2802.

Masterson, E. A., Tak, S., Themann, C. L., Wall, D. K., Groenewold, M. R., Deddens, J. A. and Calvert, G. M. (2013), 'Prevalence of hearing loss in the united states by industry', *American journal of industrial medicine* **56**(6), 670–681.

McAlexander, T. P., Gershon, R. R. and Neitzel, R. L. (2015), 'Street-level noise in an urban setting: assessment and contribution to personal exposure', *Environmental Health* **14**(1), 1–10.

McCormack, A. and Fortnum, H. (2013), 'Why do people fitted with hearing aids not wear them?', *Int. J. Audiology* **52**(5), 360–368.

Mehta, R. P., Rosowski, J. J., Voss, S. E., O'Neil, E. and Merchant, S. N. (2006), 'Determinants of hearing loss in perforations of the tympanic membrane', *Otology and neurotology: official publication of the American Otological Society, American Neurotology Society [and] European Academy of Otology and Neurotology* **27**(2), 136.

Meredith, R. and Stephens, D. (1993), 'In-the-ear and behind-the-ear hearing aids in the elderly', *Scandinavian Audiology* **22**(4), 211–216.

Michelsanti, D., Tan, Z.-H., Sigurdsson, S. and Jensen, J. (2019), 'Deep-learning-based audio-visual speech enhancement in presence of lombard effect', *Speech Comm.* **115**, 38–50.

Mishra, A. N., Shrotriya, M. and Sharan, S. (2010), Comparative wavelet, PLP, and LPC speech recognition techniques on the hindi speech digits database, *in* 'Second International Conference on Digital Image Processing', Vol. 7546, SPIE, pp. 724–729.

Moore, A. H., Parada, P. P. and Naylor, P. A. (2017), 'Speech enhancement for robust automatic speech recognition: Evaluation using a baseline system and instrumental measures', *Computer Speech and Language* **46**, 574–584.

Mun, S., Shon, S., Kim, W. and Ko, H. (2016), Deep neural network bottleneck features for acoustic event recognition., *in* 'Interspeech', pp. 2954–2957.

Musa, M. K., Razak, M. S. A., Mahyeddin, M. E., Mustafa, M. S. S. and Lip, R. (2022), An investigation of noise emission level at KTM railway tracks around the community, *in* 'AIP Conference Proceedings', Vol. 2644, AIP Publishing LLC, p. 050020.

Mushtaq, Z. and Su, S.-F. (2020), 'Environmental sound classification using a regularized deep convolutional neural network with data augmentation', *Applied Acoustics* **167**, 107389.

Nadol Jr, J. B. (1993), 'Hearing loss', *New England Journal of Medicine* **329**(15), 1092–1102.

Nair, A. A. and Koishida, K. (2021), Cascaded time+ time-frequency Unet for speech enhancement: Jointly addressing clipping, codec distortions, and gaps, *in* 'ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 7153–7157.

Nair, G. G. and Kumar, C. S. (2021), Speech enhancement system for automatic speech recognition in automotive environment, *in* '2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)', IEEE, pp. 01–07.

Narayanan, A. and Wang, D. (2013), Ideal ratio mask estimation using deep neural networks for robust speech recognition, *in* '2013 IEEE International Conference on Acoustics, Speech and Signal Processing', IEEE, pp. 7092–7096.

Narayanan, A. and Wang, D. (2014), 'Investigation of speech separation as a front-end for noise robust speech recognition', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **22**(4), 826–835.

Neitzel, R., Gershon, R. R., Zeltser, M., Canton, A. and Akram, M. (2009), 'Noise levels associated with new york city's mass transit systems', *American journal of public health* **99**(8), 1393–1399.

Nelson, D. I., Nelson, R. Y., Concha-Barrientos, M. and Fingerhut, M. (2005), 'The global burden of occupational noise-induced hearing loss', *American journal of industrial medicine* **48**(6), 446–458.

Neumann, K. and Stephens, D. (2011), 'Definitions of types of hearing impairment: a discussion paper', *Folia Phoniatrica et Logopaedica* **63**(1), 43–48.

Ni, J., Hirai, T., Kawai, H., Toda, T., Tokuda, K., Tsuzaki, M., Sakai, S., Maia, R. and Nakamura, S. (2007), 'ATRECSS: ATR english speech corpus for speech synthesis'.

Nicolson, A. and Paliwal, K. K. (2019), 'Deep learning for minimum mean-square error approaches to speech enhancement', *Speech Comm.* **111**, 44–55.

Nossier, S. A., Rizk, M., Moussa, N. D. and el Shehaby, S. (2019), 'Enhanced smart hearing aid using deep neural networks', *Alexandria Engineering Journal* **58**(2), 539–550.

Nossier, S. A., Wall, J., Moniri, M., Glackin, C. and Cannings, N. (2020), Mapping and masking targets comparison using different deep learning based speech enhancement architectures, *in* 'IJCNN', IEEE, pp. 1–8.

Odelowo, B. O. and Anderson, D. V. (2018), A study of training targets for deep neural network-based speech enhancement using noise prediction, *in* 'ICASSP', IEEE, pp. 5409–5413.

Ortega-García, J. and González-Rodríguez, J. (1996), Overview of speech enhancement techniques for automatic speaker recognition, *in* 'Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96', Vol. 2, IEEE, pp. 929–932.

Ouyang, Z., Yu, H., Zhu, W. and Champagne, B. (2019), A fully convolutional neural network for complex spectrogram processing in speech enhancement, *in* 'ICASSP', IEEE, pp. 5756–5760.

Pal, S., Ebrahimi, E., Zulfiqar, A., Fu, Y., Zhang, V., Migacz, S., Nellans, D. and Gupta, P. (2019), 'Optimizing multi-GPU parallelization strategies for deep learning training', *IEEE Micro* **39**(5), 91–101.

Paliwal, K., Wójcicki, K. and Shannon, B. (2011), 'The importance of phase in speech enhancement', *Speech Comm.* **53**(4), 465–494.

Pan, W., Li, Z., Zhang, Y. and Weng, C. (2018), 'The new hardware development trend and the challenges in data management and analysis', *Data Sci. Eng.* **3**(3), 263–276.

Panayotov, V., Chen, G., Povey, D. and Khudanpur, S. (2015), Librispeech: an ASR corpus based on public domain audio books, *in* 'ICASSP', IEEE, pp. 5206–5210.

Pandey, A. and Wang, D. (2018*a*), A new framework for supervised speech enhancement in the time domain., *in* 'INTERSPEECH', pp. 1136–1140.

Pandey, A. and Wang, D. (2018*b*), On adversarial training and loss functions for speech enhancement, *in* '2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 5414–5418.

Pandey, A. and Wang, D. (2019), 'A new framework for CNN-based speech enhancement in the time domain', *IEEE Trans. Audio Speech Lang. Proc.* **27**(7), 1179–1188.

Pandey, A. and Wang, D. (2020*a*), Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain, *in* 'ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 6629–6633.

Pandey, A. and Wang, D. (2020*b*), 'On cross-corpus generalization of deep learning based speech enhancement', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 2489–2499.

Park, G. and Lee, S. (2020), 'Environmental noise classification using convolutional neural networks with input transform for hearing aids', *International journal of environmental research and public health* **17**(7), 2270.

Park, S. and Kwak, N. (2016), Analysis on the dropout effect in convolutional neural networks, *in* 'Asian conference on computer vision', Springer, pp. 189–204.

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

Pascanu, R., Mikolov, T. and Bengio, Y. (2013), On the difficulty of training recurrent neural networks, *in* 'International conference on machine learning', pp. 1310–1318.

Pascual, S., Bonafonte, A. and Serra, J. (2017), SEGAN: Speech enhancement generative adversarial network, *in* 'INTERSPEECH', pp. 3642–3646.

Pedamonti, D. (2018), 'Comparison of non-linear activation functions for deep neural networks on MNIST classification task', *arXiv preprint arXiv:1804.02763* .

Peddinti, V., Povey, D. and Khudanpur, S. (2015), A time delay neural network architecture for efficient modeling of long temporal contexts, *in* 'Sixteenth annual conference of the international speech communication association'.

Peercy, P. S. (2000), 'The drive to miniaturization', *Nature* **406**(6799), 1023.

Petridis, S. and Pantic, M. (2016), Deep complementary bottleneck features for visual speech recognition, *in* '2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 2304–2308.

Phan, H., McLoughlin, I. V., Pham, L., Chén, O. Y., Koch, P., De Vos, M. and Mertins, A. (2020), 'Improving GANs for speech enhancement', *Sig. Proc. Lett.* **27**, 1700–1704.

Piczak, K. J. (2015), ESC: Dataset for environmental sound classification, *in* 'MM–ACM', ACM, pp. 1015–1018.

Pirhosseinloo, S. and Brumberg, J. S. (2018), A new feature set for masking-based monaural speech separation, *in* 'ACSSC', IEEE, pp. 828–832.

Pirhosseinloo, S. and Brumberg, J. S. (2019), Monaural speech enhancement with dilated convolutions., *in* 'Interspeech', pp. 3143–3147.

Raut, P. C. and Deoghare, S. U. (2016), 'Automatic speech recognition and its applications', *International Research Journal of Engineering and Technology* **3**(5), 2368–2371.

Reddy, C. K., Beyrami, E., Dubey, H., Gopal, V., Cheng, R., Cutler, R., Matusevych, S., Aichner, R., Aazami, A., Braun, S. et al. (2020), 'The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework', *arXiv preprint arXiv:2001.08662* .

Reddy, C. K., Beyrami, E., Pool, J., Cutler, R., Srinivasan, S. and Gehrke, J. (2019), 'A scalable noisy speech dataset and online subjective test framework', *arXiv* .

Reddy, C. K., Dubey, H., Koishida, K., Nair, A., Gopal, V., Cutler, R., Braun, S., Gamper, H., Aichner, R. and Srinivasan, S. (2021), 'Interspeech 2021 deep noise suppression challenge', *arXiv preprint arXiv:2101.01902* .

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

Rix, A., Beerends, J., Hollier, M. and Hekstra, A. (2001), 'Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs', *ITU-T Recommendation, p. 862.* .

Robinson, T., Fransen, J., Pye, D., Foote, J. and Renals, S. (1995), WSJCAM0: a british english speech corpus for large vocabulary continuous speech recognition, *in* 'ICASSP', Vol. 1, IEEE, pp. 81–84.

Ronneberger, O., Fischer, P. and Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, *in* 'Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18', Springer, pp. 234–241.

Rousseau, A., Deléglise, P. and Esteve, Y. (2012), TED-LIUM: an automatic speech recognition dedicated corpus., *in* 'LREC', pp. 125–129.

Roy, S. K., Nicolson, A. and Paliwal, K. K. (2021), 'DeepLPC: A deep learning approach to augmented kalman filter-based single-channel speech enhancement', *IEEE Access* **9**, 64524–64538.

Sainath, T. N., Kingsbury, B. and Ramabhadran, B. (2012), Auto-encoder bottleneck features using deep belief networks, *in* '2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)', IEEE, pp. 4153–4156.

Salamon, J., Jacoby, C. and Bello, J. P. (2014), A dataset and taxonomy for urban sound research, *in* 'MM–ACM', ACM, pp. 1041–1044.

Saleem, N., Irfan Khattak, M., Ali, M. Y. and Shafi, M. (2019), 'Deep neural network for supervised single-channel speech enhancement', *Archives of Acoustics* **44**.

Saleem, N. and Khattak, M. I. (2019), 'A review of supervised learning algorithms for single channel speech enhancement', *International Journal of Speech Technology* **22**(4), 1051–1075.

Sambur, M. and Jayant, N. (1976), 'LPC analysis/synthesis from speech inputs containing quantizing noise or additive white noise', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **24**(6), 488–494.

Samui, S., Chakrabarti, I. and Ghosh, S. K. (2019), 'Time-frequency masking based supervised speech enhancement framework using fuzzy deep belief network', *Appl. Soft Comput.* **74**, 583–602.

Sarsenbayeva, Z., van Berkel, N., Velloso, E., Kostakos, V. and Goncalves, J. (2018), 'Effect of distinct ambient noise types on mobile interaction', *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **2**(2), 1–23.

Scalart, P. et al. (1996), Speech enhancement based on a priori signal to noise estimation, *in* 'ICASSP', Vol. 2, IEEE, pp. 629–632.

Schädler, M. R., Meyer, B. T. and Kollmeier, B. (2012), 'Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition', *J. Acoust. Soc. Am.* **131**(5), 4134–4151.

Schmidhuber, J. (2015), 'Deep learning in neural networks: An overview', *Neural networks* **61**, 85–117.

Schreiber, B. E., Agrup, C., Haskard, D. O. and Luxon, L. M. (2010), 'Sudden sensorineural hearing loss', *The Lancet* **375**(9721), 1203–1211.

Schröter, H., Rosenkranz, T., Escalante-B, A. N., Aubreville, M. and Maier, A. (2020), CLCNet: Deep learning-based noise reduction for hearing aids using complex linear coding, *in* 'ICASSP', IEEE, pp. 6949–6953.

Schröter, H., Rosenkranz, T., Escalante-B, A.-N. and Maier, A. (2022), 'Low latency speech enhancement for hearing aids using deep filtering', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **30**, 2716–2728.

Shao, Y. and Wang, D. (2008), Robust speaker identification using auditory features and computational auditory scene analysis, *in* 'ICASSP', IEEE, pp. 1589–1592.

Shaw, E. A. (1974), The external ear, *in* 'Auditory system', Springer, pp. 455–490.

Shaw, E. and Stinson, M. (1983), The human external and middle ear: Models and concepts, *in* 'Mechanics of hearing', Springer, pp. 3–10.

Shen, Y.-L., Huang, C.-Y., Wang, S.-S., Tsao, Y., Wang, H.-M. and Chi, T.-S. (2019), Reinforcement learning based speech enhancement for robust speech recognition, *in* 'ICASSP', IEEE, pp. 6750–6754.

Shepard, R. N. (1964), 'Circularity in judgments of relative pitch', *The journal of the acoustical society of America* **36**(12), 2346–2353.

Shi, G., Shanechi, M. and Aarabi, P. (2006), 'On the importance of phase in human speech recognition', *IEEE Trans. Audio Speech Lang. Proc.* **14**(5), 1867–1874.

Simard, P. Y., Steinkraus, D., Platt, J. C. et al. (2003), Best practices for convolutional neural networks applied to visual document analysis., *in* 'Icdar', Vol. 3.

Soni, M. H., Shah, N. and Patil, H. A. (2018), Time-frequency masking-based speech enhancement using generative adversarial network, *in* 'ICASSP', IEEE, pp. 5039–5043.

Srinivasan, S., Roman, N. and Wang, D. (2006), 'Binary and ratio time-frequency masks for robust speech recognition', *Speech Comm.* **48**(11), 1486–1501.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), 'Dropout: a simple way to prevent neural networks from overfitting', *The journal of machine learning research* **15**(1), 1929–1958.

Stoller, D., Ewert, S. and Dixon, S. (2018), 'Wave-u-net: A multi-scale neural network for end-to-end audio source separation', *arXiv preprint arXiv:1806.03185* .

Strake, M., Defraene, B., Fluyt, K., Tirry, W. and Fingscheidt, T. (2019), Separated noise suppression and speech restoration: LSTM-based speech enhancement in two stages, *in* '2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)', IEEE, pp. 239–243.

Sun, L., Du, J., Dai, L.-R. and Lee, C.-H. (2017), Multiple-target deep learning for LSTM-RNN based speech enhancement, *in* '2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)', IEEE, pp. 136–140.

Surfingtech (2015), 'Free ST american english corpus', *[Online]. Available: http://www.openslr.org/45/* .

Sze, V., Chen, Y.-H., Yang, T.-J. and Emer, J. S. (2017), 'Efficient processing of deep neural networks: A tutorial and survey', *Proceedings of the IEEE* **105**(12), 2295–2329.

Taal, C., Hendriks, R., Heusdens, R. and Jensen, J. (2011), 'An algorithm for intelligibility prediction of time-frequency weighted noisy speech', *IEEE Trans. Audio Speech Lang. Proc.* **19**(7), 2125–2136.

Tan, K. and Wang, D. (2018), A convolutional recurrent neural network for real-time speech enhancement., *in* 'Interspeech', pp. 3229–3233.

Tang, C., Luo, C., Zhao, Z., Xie, W. and Zeng, W. (2021), Joint time-frequency and time domain learning for speech enhancement, *in* 'Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence', pp. 3816–3822.

Teel, J. C. (2005), 'Understanding noise in linear regulators', *Texas Instruments Analog Applicant* .

Thiemann, J., Ito, N. and Vincent, E. (2013), DEMAND: A collection of multi-channel recordings of acoustic noise in diverse environments, *in* 'Meetings Acoust.'.

Tomlin, C. J. and Axelrod, J. D. (2007), 'Biology by numbers: mathematical modelling in developmental biology', *Nature* **8**(5), 331.

Topcoder (2017), '176 spoken languages', *[Online]. Available: http://www.topcoder.com/contest /problem/SpokenLanguages2/trainingdata.zip* .

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S. and Birchfield, S. (2018), Training deep networks with synthetic data: Bridging the reality gap by domain randomization, *in* 'IEEE Conf. CVPR', pp. 969–977.

Tu, M. and Zhang, X. (2017), Speech enhancement based on deep neural networks with skip connections, *in* 'ICASSP', IEEE, pp. 5565–5569.

Uemura, Y., Takahashi, Y., Saruwatari, H., Shikano, K. and Kondo, K. (2009), Musical noise generation analysis for noise reduction methods based on spectral subtraction and MMSE STSA estimation, *in* 'ICASSP', IEEE, pp. 4433–4436.

Upadhyay, N. and Jaiswal, R. K. (2016), 'Single channel speech enhancement: using wiener filtering with recursive noise estimation', *Procedia Computer Science* **84**, 22–30.

Upadhyay, N. and Karmakar, A. (2015), 'Speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study', *Procedia Computer Science* **54**, 574–584.

Valentini-Botinhao, C. et al. (2017*a*), 'Noisy reverberant speech database for training speech enhancement algorithms and TTS models', *[Online]. Available: https://datashare.is.ed.ac.uk/handle/10283/2791* .

Valentini-Botinhao, C. et al. (2017*b*), 'Noisy speech database for training speech enhancement algorithms and TTS models', *University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR)* .

van Hengel, P. W. and Krijnders, J. D. (2013), 'A comparison of spectro-temporal representations of audio signals', *IEEE/ACM transactions on audio, speech, and language processing* **22**(2), 303–313.

Vanithalakshmi, M., Subitha, D. and Velmurugan, S. (2022), Wavelet based speech enhancement algorithm for hearing aid application, *in* '2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)', Vol. 1, IEEE, pp. 11–15.

Varga, A. and Steeneken, H. (1993), 'Assessment for automatic speech recognition: Ii. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems', *Speech Comm.* **12**(3), 247–251.

Vasilev, I., Slater, D., Spacagna, G., Roelants, P. and Zocca, V. (2019), *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*, Packt Publishing Ltd.

Veaux, C., Yamagishi, J. and King, S. (2013), The voice bank corpus: Design, collection and data analysis of a large regional accent speech database, *in* '2013 international conference oriental COCOSDA held jointly with 2013 conference on Asian spoken language research and evaluation (O-COCOSDA/CASLRE)', IEEE, pp. 1–4.

Verteletskaya, E. and Simak, B. (2011), 'Noise reduction based on modified spectral subtraction method', *IAENG International journal of computer science* **38**(1), 82–88.

Vihari, S., Murthy, A. S., Soni, P. and Naik, D. (2016), 'Comparison of speech enhancement algorithms', *Procedia computer science* **89**, 666–676.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P.-A. (2010), 'Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion', *Journal of machine learning research* **11**(Dec), 3371–3408.

Vu, T. T., Bigot, B. and Chng, E. S. (2016), Combining non-negative matrix factorization and deep neural networks for speech enhancement and automatic speech recognition, *in* 'ICASSP', IEEE, pp. 499–503.

Wang, D. (2005), On ideal binary mask as the computational goal of auditory scene analysis, *in* 'Speech Separation by Humans and Machines', Springer, pp. 181–197.

Wang, D. (2017), 'Deep learning reinvents the hearing aid', *IEEE Spectrum* **54**(3), 32–37.

Wang, D. and Chen, J. (2018), 'Supervised speech separation based on deep learning: An overview', *IEEE Trans. Audio Speech Lang. Proc.* **26**(10).

Wang, D. and Lim, J. (1982), 'The unimportance of phase in speech enhancement', *IEEE Trans. Audio Speech Lang. Proc.* **30**(4), 679–681.

Wang, D. and Zheng, T. F. (2015), Transfer learning for speech and language processing, *in* 'APSIPA', IEEE, pp. 1225–1237.

Wang, P., Tan, K. et al. (2019), 'Bridging the gap between monaural speech enhancement and recognition with distortion-independent acoustic modeling', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 39–48.

Wang, S., Li, W., Siniscalchi, S. M. and Lee, C.-H. (2020), A cross-task transfer learning approach to adapting deep speech enhancement models to unseen background noise using paired senone classifiers, *in* 'ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 6219–6223.

Wang, Y., Narayanan, A. and Wang, D. (2014), 'On training targets for supervised speech separation', *IEEE tran. on audio, speech, and lang. proc.* **22**(12), 1849–1858.

Wang, Y. and Wang, D. (2015), A deep neural network for time-domain signal reconstruction, *in* '2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 4390–4394.

Wang, Z.-Q. and Wang, D. (2016), 'A joint training framework for robust automatic speech recognition', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**(4), 796–806.

Wang, Z.-Q., Wang, P. and Wang, D. (2020), 'Complex spectral mapping for single- and multi-channel speech enhancement and robust ASR', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 1778–1787.

Wang, Z., Wang, X., Li, X., Fu, Q. and Yan, Y. (2016), Oracle performance investigation of the ideal masks, *in* 'IWAENC', IEEE, pp. 1–5.

Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J. and Huang, T. S. (2018), Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 7268–7277.

Weintraub, M. (1985), A theory and computational model of auditory monaural sound separation, PhD thesis, Stanford University.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Roux, J. L., Hershey, J. R. and Schuller, B. (2015), Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR, *in* 'International conference on latent variable analysis and signal separation', Springer, pp. 91–99.

Williamson, D. S., Wang, Y. and Wang, D. (2016), 'Complex ratio masking for monaural speech separation', *IEEE Trans. Audio Speech Lang. Proc.* **24**(3), 483–492.

Wu, Z. and King, S. (2016), 'Improving trajectory modelling for DNN-based speech synthesis by using stacked bottleneck features and minimum generation error training', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**(7), 1255–1265.

Xia, Y., Braun, S., Reddy, C. K. A., Dubey, H., Cutler, R. and Tashev, I. (2020*a*), 'Deep Noise Suppression (DNS) Challenge', *[Online]. Available: https://github.com/microsoft/DNS-Challenge* .

Xia, Y., Braun, S., Reddy, C. K., Dubey, H., Cutler, R. and Tashev, I. (2020*b*), Weighted speech distortion losses for neural-network-based real-time speech enhancement, *in* 'ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 871–875.

Xu, Y. (2013), 'USTC-made 15 noise', *[Online]. Available: Available: https://pan.baidu.com/s/1dER6UUt* .

Xu, Y., Du, J., Dai, L.-R. and Lee, C.-H. (2013), 'An experimental study on speech enhancement based on deep neural networks', *IEEE Signal processing letters* **21**(1), 65–68.

Xu, Y., Du, J., Dai, L.-R. and Lee, C.-H. (2014*a*), Cross-language transfer learning for deep neural network based speech enhancement, *in* 'ISCSLP', IEEE, pp. 336–340.

Xu, Y., Du, J., Dai, L.-R. and Lee, C.-H. (2014*b*), 'A regression approach to speech enhancement based on deep neural networks', *IEEE Trans. Audio Speech Lang. Proc.* **23**(1), 7–19.

Xu, Y., Du, J., Huang, Z., Dai, L.-R. and Lee, C.-H. (2015), Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement, *in* 'INTERSPEECH', pp. 1508–1512.

Yamagishi, J., Veaux, C., MacDonald, K. et al. (2019), 'CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning Toolkit (version 0.92)', *University of Edinburgh. The Centre for Speech Technology Research (CSTR)* .

Yaman, S., Pelecanos, J. and Sarikaya, R. (2012), Bottleneck features for speaker recognition, *in* 'Odyssey 2012-The Speaker and Language Recognition Workshop'.

Yong, X., Jun, D., Zhen, H., Li-Rong, D. and Chin-Hui, L. (2015), 'DNN based Speech Enhancement Demo', *[Online]. Available: https://github.com/yongxuUSTC/DNN-Speech-enhancement-demo-tool* .

Yu, D. and Seltzer, M. L. (2011), Improved bottleneck features using pretrained deep neural networks, *in* 'Twelfth annual conference of the international speech communication association'.

Yu, D., Seltzer, M. L., Li, J., Huang, J.-T. and Seide, F. (2013), 'Feature learning in deep neural networks studies on speech recognition tasks', *arXiv* .

Yu, F. and Koltun, V. (2016), Multi-scale context aggregation by dilated convolutions, *in* 'Int. Conf. Learn. Representations (ICLR)', pp. 1–9.

Yuliani, A. R., Amri, M. F., Suryawati, E., Ramdan, A. and Pardede, H. F. (2021), 'Speech enhancement using deep learning methods: A review', *Jurnal Elektronika dan Telekomunikasi* **21**(1), 19–26.

Zhang, B., Xie, L., Yuan, Y., Ming, H., Huang, D. and Song, M. (2016), Deep neural network derived bottleneck features for accurate audio classification, *in* '2016 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)', IEEE, pp. 1–6.

Zhang, Q., Nicolson, A. and Horst, F. (2021), 'Objective scores obtained on the valentini voicebank test set', *[Online]. Available: https://github.com/anicolson/DeepXi* .

Zhang, Q., Nicolson, A., Wang, M., Paliwal, K. K. and Wang, C. (2020), 'DeepMMSE: A deep learning approach to MMSE-based noise power spectral density estimation', *Trans. Audio Speech Lang. Proc.* **28**, 1404–1415.

Zhang, Y., Chan, W. and Jaitly, N. (2017), Very deep convolutional networks for end-to-end speech recognition, *in* 'ICASSP', IEEE, pp. 4845–4849.

Zhao, H., Zarar, S., Tashev, I. and Lee, C.-H. (2018), Convolutional-recurrent neural networks for speech enhancement, *in* '2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 2401–2405.

Zhao, Y., Wang, D., Merks, I. and Zhang, T. (2016), DNN-based enhancement of noisy and reverberant speech, *in* 'ICASSP', IEEE, pp. 6525–6529.

Zhao, Y., Wang, Z.-Q. and Wang, D. (2018), 'Two-stage deep learning for noisy-reverberant speech enhancement', *Trans. Audio Speech Lang. Proc.* **27**(1), 53–62.

Zue, V., Seneff, S. and Glass, J. (1990), 'Speech database development at MIT: TIMIT and beyond', *Speech Comm.* **9**(4), 351–356.

Zwartenkot, J. W., Snik, A. F., Mylanus, E. A. and Mulder, J. J. (2014), 'Amplification options for patients with mixed hearing loss', *Otology and Neurotology* **35**(2), 221–226.

LIST OF PUBLICATIONS

1. S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, ”Convolutional Recurrent Smart Speech Enhancement Architecture for Hearing Aids”, in Proceedings of the 2022 INTERSPEECH, Incheon, Korea, 18-22 September. pp. 1-5.

2. S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, “Two-stage deep learning approach for speech enhancement and reconstruction in the frequency and time domains,” in Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July. IEEE, 2022, pp. 1–10.

3. S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, “An experimental analysis of deep learning architectures for supervised speech enhancement,” Electronics, vol. 10, no. 1, p. 17, 2021.

4. S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, “A comparative study of time and frequency domain approaches to deep learning based speech enhancement,” in Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July. IEEE, 2020, pp. 1–8.

5. S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings, “Mapping and masking targets comparison using different deep learning based speech enhancement architectures,” in Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July. IEEE, 2020, pp. 1–8.

Appendices

Appendix A

Ethical Approval

University of East London

Pioneering Futures Since 1898

Dear Soha,

Application ID: ETH2223-0119

Original application ID: ETH1920-0059

Project title: Deep Learning-based Speech Enhancement for Real-life Applications

Lead researcher: Miss Soha Abdallah Abdelhafiz Nossier

Your application to Arts and Creative Industries School Research Ethics Committee was considered on the 20th January 2023.

The decision is: **Approved**

The Committee's response is based on the protocol described in the application form and supporting documentation.

Your project has received ethical approval for 4 years from the approval date.

If you have any questions regarding this application please contact your supervisor or the administrator for the Arts and Creative Industries School Research Ethics Committee.

Approval has been given for the submitted application only and the research must be conducted accordingly.

Should you wish to make any changes in connection with this research/consultancy project you must complete 'An application for approval of an amendment to an existing application'.

The approval of the proposed research/consultancy project applies to the following site.

Project site: **University of East London**

Principal Investigator / Local Collaborator: Miss Soha Abdallah Abdelhafiz Nossier

Approval is given on the understanding that the UEL Code of Practice for Research and the Code of Practice for Research Ethics is adhered to.

Any adverse events or reactions that occur in connection with this research/consultancy project should be reported using the University's form for Reporting an Adverse/Serious Adverse Event/Reaction.

The University will periodically audit a random sample of approved applications for ethical approval, to ensure that the projects are conducted in compliance with the consent given by the Ethics and Integrity Sub-Committee and to the highest standards of rigour and integrity.

Please note, it is your responsibility to retain this letter for your records.

With the Committee's best wishes for the success of the project.

Yours sincerely,

Catherine Hitchens

Ethics, Integrity and Compliance Manager

University of East London

Pioneering Futures Since 1898

Dear Soha

Application ID: ETH1920-0059

Project title: Deep Learning based Speech Enhancement and Recognition in Noisy and Adverse Environments

Lead researcher: Miss Soha Abdallah Abdelhafiz Nossier

Your application to University Research Ethics Sub-Committee was considered on the 2nd of March 2020.

The decision is: **Approved**

The Committee's response is based on the protocol described in the application form and supporting documentation.

Your project has received ethical approval for 2 years from the approval date.

If you have any questions regarding this application please contact your supervisor or the secretary for the University Research Ethics Sub-Committee.

Approval has been given for the submitted application only and the research must be conducted accordingly.

Should you wish to make any changes in connection with this research project you must complete 'An application for approval of an amendment to an existing application'.

The approval of the proposed research applies to the following research site.

Research site: University of East London

Principal Investigator / Local Collaborator: Miss Soha Abdallah Abdelhafiz Nossier

Approval is given on the understanding that the UEL Code of Practice for Research and the Code of Practice for Research Ethics is adhered to.

Any adverse events or reactions that occur in connection with this research project should be reported using the University's form for Reporting an Adverse/Serious Adverse Event/Reaction.

The University will periodically audit a random sample of approved applications for ethical approval, to ensure that the research projects are conducted in compliance with the consent given by the Research Ethics Committee and to the highest standards of rigour and integrity.

Please note, it is your responsibility to retain this letter for your records.

With the Committee's best wishes for the success of the project

Yours sincerely

Fernanda Silva

Administrative Officer for Research Governance