Aversive Sound Classification and Filtration Using Deep Neural Networks

by

Behnaz Bahmei

M.Sc., K.N. Toosi University of Technology, 2014

B.Sc., Shahid Chamran University, 2006

Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in the

School of Mechatronic Systems Engineering

Faculty of Applied Sciences

SIMON FRASER UNIVERSITY

Fall 2022

Declaration of Committee

Name:	**Behnaz Bahmei**
Degree:	**Doctor of Philosophy**
Title:	**Aversive Sound Classification and Filtration Using Deep Neural Networks**
Committee:	**Chair:** **Amr Marzouk** Lecturer, Mechatronic Systems Engineering
	Siamak Arzanpour Co-supervisor Associate Professor, Mechatronic Systems Engineering
	Elina Birmingham Co-supervisor Associate Professor, Education
	Oliver Schulte Committee Member Professor, Computing Science
	Faranak Farzan Committee Member Associate Professor, Mechatronic Systems Engineering
	Ramtin Rakhsha Examiner Lecturer, Mechatronic Systems Engineering
	Kouhyar Tavakolian External Examiner Associate Professor, Biomedical Engineering University of North Dakota

Ethics Statement

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

Decreased sound tolerance (DST) is common among many children with Autism Spectrum Disorder (ASD). When children are exposed to specific aversive sounds at school, they may be very distressed and can react with behaviours such as covering their ears, yelling, screaming, or running out of the room to avoid the aversive sound. Schools' approaches for accommodating DST include letting students wear earplugs or earmuffs or allowing them to leave to take breaks in a quiet area. Most wearable devices (e.g., earmuffs, ear plugs, noise canceling headphones) tend to block or attenuate all sounds indiscriminately, including speech, and if the child leaves the classroom to escape the noise, this will disrupt learning and social interaction. Therefore, existing strategies tend to interfere with the child's full participation in class and other activities. This thesis aims to develop an intervention tool to selectively filter out aversive sounds for children with ASD. Ideally, this tool will attenuate unwanted sounds (e.g., dog barking, sirens, jackhammers) while letting other sounds (e.g., the teacher's voice) to be heard. In this thesis, Deep Neural Network (DNN) methods and signal processing techniques are employed to intelligently identify the aversive sounds in the environment, attenuate them from the ambient sound and pass the rest of the sound to users. To identify aversive sounds, a combination of a Recurrent Neural Network (RNN) and a Convolutional Neural Network (CNN) is used. After the aversive sound is identified, another part of this thesis is dedicated to filter the aversive sound. A DNN-based learning framework is proposed to address the audio denoising problem for real-time applications. The proposed method has the ability to suppress stationary noises such as engine and air conditioner, and also non-stationary, and dynamic noises such as dog barking, siren, and jackhammer. Further, a Graphical User Interface (GUI) is designed to combine the identification and filtration components of the intervention. The user-friendly GUI enables the users to initiate specific tasks in order to hear their surrounding sounds without any disturbances. In order to evaluate the performance of the proposed intervention technique, several testing sessions are conducted with autistic individuals.

Keywords: Autism Spectrum Disorder; Decreased Sound Tolerance; Deep Neural Networks; Environmental Sound Identification, Audio Denoising, Real-time Processing

This thesis is dedicated to my parents and my husband, for their love, support, and encouragement.

Acknowledgements

I would like to express my special appreciation and thanks to my supervisors Dr. Siamak Arzanpour and Dr. Elina Birmingham for their continuous support during my Ph.D. study. They have been a tremendous mentor to me. I learned valuable lessons from their vision, personality and professionalism. They were always encouraging me and available when I need their guidance.

Besides, I would like to thank my committee, Professor Oliver Schulte, and Dr. Faranak Farzan for serving as my committee members.

I would like to express my deep appreciation to my beloved husband, Peyman Sindareh Esfahani, who always support me with his endless love. I truly appreciate his support during the very stressful and challenging moments of my life.

A special thanks to my parents, and my brothers for their love, support, and encouragement. They had always believed in me and my abilities. I am grateful to have them in my life.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ASD	Autism Spectrum Disorder
DST	Decreased Sound Tolerance
ESC	Environmental Sound Classification
DNN	Deep Neural Networks
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
GUI	Graphical User Interface

Glossary

Stationary Noises	Stationary noises have constant statistical properties (for example, power spectral density) over time such as air conditioning sound.
Non-stationary Noises	Non-stationary noises have inconsistent statistical properties over time such as siren sound. In the real world, most of the noises are non-stationary.
Dynamic Noises	Dynamic noises are a type of non-stationary noises; however, they are more complex. For example, they have multiple frequency components over a short time interval such as speech and a dog barking sound.

Chapter 1.

Introduction

1.1. Background and Motivation

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder that is characterized by abnormal social interactions, verbal, and nonverbal communications, and repetitive behaviours from early childhood [1]. According to the Centre of Disease Control (CDC), the prevalence of autism is currently 1 in 44 among children aged 8 years [2] in 2018. Children with ASD demonstrate serious problems with attention, orientation, and atypical reactions to the environment and sensory stimuli (45-95% of autistic individuals [3]). Hyper-sensitivity to specific sounds, light, or touch is often reported among this population [4]. These sensory differences have been found to affect other functions, including social, and cognitive abilities [5].

Decreased sound tolerance (DST) is one of the most associated clinical features reported in children with ASD [6]. Research indicates that certain sounds or specifically certain features of sounds such as loud, sudden, high pitched, shrill, unfamiliar, harsh, and repetitive are more problematic and can cause aversive reactions for people diagnosed with ASD [7], [8]. The most common examples of aversive sounds reported by parents include sirens, toilet flush, dog barking, baby crying, fireworks, vacuum cleaner, and clapping [8]. These children find everyday sounds extremely aversive, interfering with their ability to participate in school, community, and family activities. The nature of the aversive sounds varies across individuals, although most commonly, families and caregivers describe the trigger sounds as loud, high-pitched, and sudden [8]. Aversive sounds do not always share these characteristics and can be sounds that are unlikely to be bothersome to individuals without DST [7]. The prevalence of DST is reported to range from 50% to 70% among autistic children and adults at some point in their lives [9]. Different forms of DST, including hyperacusis, misophonia, and phonophobia, have been identified in the larger audiology literature. Hyperacusis is a frequently cited condition and is described as a decreased tolerance for daily noises at levels that would not bother most people [9], [10]. Individuals with hyperacusis have normal auditory detection thresholds; however, it appears that their threshold for

loudness discomfort is lowered, enabling moderately loud noises to be perceived as too loud, or even painful [9], [11]. Misophonia, on the other hand, is a neuropsychiatric disorder in which “trigger” noises (such as chewing, lip-smacking, tapping, sniffing, and so on) evoke excessive and inappropriate emotional reactions even at modest amplitudes [12], [13]. Phonophobia is a specialized phobia of certain sounds or groups of sounds. Individuals suffering from phonophobia frequently engage in preventative behaviours such as avoiding potential sound sources [9], [14] and covering their ears [15].

Children with ASD often present their aversion to sounds through preemptive and physical behaviors which include avoiding the potential sources of the sound [14] and covering their ears [15]. The common emotional reactions reported by families occurring during their child’s negative reactions to sounds include stress, irritability, scared, and nervousness [8]. The anxiety caused by aversive sounds makes those children reluctant to participate in crowded events and noisy places such as school [8], [16]. There is also a relationship between the severity of DST and decreased mental health, quality of life, and higher autism symptoms [17]. It is not hard to realize that the limitations imposed by DST seriously affect the quality of life of the ASD community.

Few techniques are available to help autistic people with DST, their families, and caregivers, cope with the problems caused by auditory stimuli. Table 1-1 shows some of the strategies used by parents for dealing with DST [8].

Table 1-1 Parent’s use of, and satisfaction with, common coping strategies for dealing with distressing sounds and noises

		Parent Satisfaction Rating (%)				
Strategy	Used by	Very Satisfied	Somewhat Satisfied	Neutral	Somewhat Unsatisfied	Very Unsatisfied
Warning	94.3%	26.5%	32.5%	26.5%	8.4%	6.0%
Taking a Break	83.0%	35.6%	37.0%	23.3%	4.1%	0.0%
Avoid Noisy Setting	81.8%	22.2%	38.9%	22.2%	12.5%	4.2%

Headphones (with Music)	59.1%	11.5%	38.5%	21.2%	11.5%	17.3%
Earmuffs	54.5%	18.8%	35.4%	29.2%	12.5%	4.2%
Headphones (without Music)	43.2%	18.4%	10.5%	18.4%	18.4%	34.2%
Ear Plugs	42.0%	2.7%	16.2%	32.4%	18.9%	29.7%
Noise Cancelling Headphones (without Music Playing)	37.5%	36.4%	21.2%	18.2%	15.2%	9.1%
White Noise Devices	35.2%	29.0%	16.1%	29.0%	6.5%	19.4%
Noise Cancelling Headphones (with Music Playing)	30.7%	48.1%	18.5%	18.5%	11.1%	3.7%
Hearing Aids	1.1%	0.0%	0.0%	100.0%	0.0%	0.0%

Note: these are ranked based on % used by.

According to Table 1-1, the most used strategies are warning the child of the presence of the noise, taking a break, or avoiding a noisy setting. These strategies remove children from the environment and impose major barriers to children's social inclusion and learning opportunities. Another major category is using wearable devices such as earmuffs, earplugs, and noise-canceling headphones. Earplugs and earmuffs, which use a passive noise isolation strategy, can block aversive sounds to a tolerable level, however, they also block other sounds such as human speech that should not be filtered. Some headphones block auditory stimuli significantly by providing a masking sound such as music or white noise, which might be annoying for some users, and again these methods block human speech. Noise-canceling headphones employ signal processing tools to reduce unwanted ambient sound, however, they are not effective in

filtering non-stationary, highly dynamic, and sudden sounds such as sirens, toilet flush, dog barking, baby crying, fireworks, vacuum cleaner, and clapping.

Furthermore, the issue of sound sensitivity in ASD is a subjective problem meaning that the nature and type of sounds can vary from one person to another. For example, some individuals expressed that they are sensitive to certain sounds such as dog barking, however other individuals are not [8]. Therefore, a one-size-fits intervention is not an appropriate solution for these individuals. The appropriate intervention must have the ability to be customized for different users with different aversive settings. This problem can be better addressed with a smart device that selectively identifies and suppresses the aversive sound while leaving the rest of the ambient sounds intact. The smart device requires to simulate human intelligence systems for identification and suppression using Artificial Intelligence (AI) technique.

AI is applied as a powerful analysis, study, and development tool in various fields of science and technology. It has led to remarkable outcomes in image detection, speech recognition, natural language processing, and signal value estimation. In recent years, AI has been employed in different acoustic and audiology studies. For example, AI-based Environmental Sound Classification (ESC) techniques are introduced to identify environmental sounds. In the ESC problem, the goal is to recognize a specific sound source, such as dog barking, siren, and drilling. ESC AI-based techniques can be divided into two categories, i.e. supervised [18] and unsupervised learning [19]. Supervised algorithms employ labeled data consisting of training inputs and corresponding outputs while unsupervised algorithms learn from the patterns in unlabeled data using clustering techniques.

AI has also been employed in signal denoising to attenuate unwanted noises from the environment. In speech enhancement, AI is widely used as an effective solution for converting a noisy speech signal to a high-quality and clean speech signal. AI-based speech enhancement algorithms aimed to attenuate the ambient noise components from speech signals to make them more intelligible for the listeners. Several AI-based speech denoising and speech enhancement methods are presented to increase the performance of speech-related applications such as Automatic Speech Recognition (ASR) [20], hearing aids devices [21], and audio/video communications [22].

In the following, the related literature is reviewed in detail and the research thesis's contributions are presented.

1.2. Previous Work

1.2.1. Environmental sound classification

The sounds around us in the real world are divided into three categories: (i) human sounds such as speech, (ii) sounds that are created by human activities such as street sounds and music, and (iii) natural sounds that are created by nature such as wind, rain, and animals [23]. All these sounds have their own structure and characteristics such as amplitude and frequency. Over the past decades, the identification of these sounds has been a focus of research because of its applications in various fields including assistive technologies, tools for individuals with hearing impairment [24], context awareness [25], surveillance [26], urban planning [27], biology [28], monitoring [29], and multimedia information retrieval [30].

Speech and music are two categories of audio signals that have been widely investigated in the literature [31], [32]. Therefore, speech and music recognition methods were merely reflected in Environmental Sound Recognition (ESR) methods.

In machine learning-based audio signal processing, the main problem is to find effective characteristics that provide higher speed and accuracy in audio classification. Several signal processing and machine learning techniques are developed to address audio classification including matrix factorization [33], dictionary learning [34], wavelet filter banks [35], Support Vector Machine (SVM) [36], hidden Markov models (HMM) [37], gaussian mixture models (GMM) [38], k-nearest neighbor (KNN) algorithm [39]. These methods and algorithms mostly consist of two main processing steps, i.e., feature extraction and pattern learning. The feature extraction part is the most important step in audio classification system design [40]. In the feature extraction step, each input pattern is mapped to a feature vector, which represents the pattern in the feature space and distinguishes it from other patterns. Since most environmental sounds are usually non-stationary and dynamic signals and there are also overlapping background sounds (e.g., environmental sounds are typically highly correlated) in the ambient, feature selection and extraction are challenging and directly affect the performance of the classification

process. Several sound features have been proposed in the literature for environmental sound classification including time-domain and frequency-domain representation of the signal. Time-domain features include zero-crossing rate (ZCR) [41], linear prediction coefficients (LPC) [42], audio signal energy function [43], and maximum amplitude [44]. Frequency-domain features or spectral features include short Fourier transform coefficients [45], mel-frequency cepstral coefficients (MFCC) [45].

Following the feature extraction, a classifier is needed to categorize different classes of sounds. As mentioned earlier, various traditional pattern learning techniques are introduced for the classification process including K-nearest neighbor, neural networks, Gaussian mixture model, hidden Markov models, minimum distance, and Bayes classifiers. These traditional algorithms have two main disadvantages: first, since their connections between the input and the hidden units are fixed, they do not provide both time and frequency invariance. Second, they are limited to short time frames, so they are not able to model longer events such as rain and sounds correlations [43]. In recent years, deep learning techniques such as Deep Neural Networks (DNNs) [46], Convolutional Neural Networks (CNNs) [47], Long Short-Term Memory (LSTM) [48], Recurrent Neural Networks (RNNs) [49], and Deep Auto Encoders (DAE) [50] have been introduced to enhance the recognition performance of environmental sounds. Due to the learning capability of the hierarchical features from high-dimensional raw data, deep layers in deep learning are more accurate than the traditional techniques [18]. For example, CNNs have been actively used for various sound classification tasks and have shown promising performance. They have the ability to obtain energy modulation patterns through time and frequency representations such as spectrograms even for sounds like drilling, constructional noise, and engine which have noise-like structures. On the other hand, they can successfully learn spectro-temporal patterns in different sound classes even in situations that some sounds are mixed [47]. They achieved high performance in monophonic [51] and polyphonic [52] in real-world environments by capturing and processing small input frames of the spectrogram. CNN's simple architecture outperforms traditional approaches such as GMMs, HMMs, and Nonnegative Matrix Factorization (NMF) classification techniques [53].

Similarly, RNNs have been used in the literature for sound classification. They are capable of modeling sequential data such as videos and text which makes them a powerful algorithm for real-time processing. They have been successfully applied to

several audio processing applications such as ASR [54] and polyphonic Sound Event Detection (SED) [55]. RNNs use information from the previous layers and time frames and provide a memorizer net of information. Recently, RNN and CNN approaches are combined to take advantage of both their strengths. Such structure was first proposed in [56] for document classification and later applied to image classification [57] and music transcription [58]. These studies have shown that the CNN-RNN framework can better learn image features and achieves superior performance than the state-of-the-art methods.

Despite these advantages, deep learning algorithms need a huge data set for efficient performance. As a result, the main challenge in deep learning problems is gathering enough data to train the network. Generally, data collection is a time-consuming procedure and requires considerable effort and cost, especially when the network needs thousands and millions of samples. Data augmentation is a strategy to address this problem. In principle, it works by increasing the diversity of the available data for training models, without actually collecting new data. There are various data augmentation techniques including flip, rotation, scale, crop, translation, and Gaussian noise [47], [59]. Among these methods, cropping, padding, and horizontal flipping are commonly used to train large neural networks. These augmentations, however, are mostly being implemented with low-level transformations, which in general are not capable of improving the performance of conventional or advanced deep learning classifiers [60]. Those low-level augmentations also have some weaknesses including a linear nature and weak data distribution enhancement [47]. Recently, generative adversarial networks (GANs) [61] have created new opportunities for researchers to achieve better high-quality results. This technique is vastly used in image generation and augmentation. The idea of GANs is to simultaneously train two models, a generative model G, and a discriminative model D. The generative model creates photorealistic images that are similar to the original data distribution and the task of discriminative model is to determine whether a given image looks realistic (image came from the training data) or artificially created.

1.2.2. Speech denoising

The genesis of multimedia communications several decades ago and the importance of high-quality audio in that application was the main motivation for the

advancements of speech denoising methods. Since the beginning of this field, several audio denoising and speech enhancement methods and systems have been proposed. Traditional methods employed different signal processing analysis methods such as nonlocal diffusion filters [62], diffusion maps [63], and averaging energy and amplitude threshold [64]. Most of those methods are developed for continuous and steady-state interferences with limited influence on transient and complex noises including non-stationary and highly dynamic noises such as siren and dog-barking. Although some of the most powerful non-stationary noise suppression algorithms reduce transient noise, the output suffers from distortion and is unsatisfactory [65].

Machine learning approaches for speech denoising employ supervised and unsupervised techniques. Supervised data-driven approaches require prior training data such as sparse non-negative matrix factorization [66], and neural networks [67]. Unsupervised approaches, such as spectrum subtraction [68], prior signal-to-noise ratio (SNR) estimation [69], and Wiener filtering [70], however, do not require any prior training data. Most of these proposed algorithms estimate the background noise's power spectral density (PSD). However, in non-stationary noise reduction, these algorithms' performance is severely constrained. In comparison to unsupervised techniques, supervised approaches have been found to create higher-quality enhanced speech signals by providing more previous information to the system. However, in general, these traditional machine learning-based speech enhancement approaches fail to suppress non-stationary and highly dynamic noises in voice signals because of their fast-fluctuating nature.

In recent years, deep learning methods such as DNNs [71], CNNs [72], long short-term memory (LSTM) [73], RNNs [74], and deep denoising autoencoders (DDAE) [75] have achieved remarkable performance in speech enhancement and audio denoising. Deep learning models employ two approaches including (i) directly predicting the clean signal [67] and (ii) predicting a mask for filtering [68]. These algorithms address the issues in traditional methods by proposing several single and multi-channel approaches with either supervised or unsupervised learning models. The superior performance of deep learning models, among other data-driven models, is attributed to their exceptional nonlinear mapping ability, which directly transforms the noisy signal into the enhanced one [76].

One of the most difficult aspects of developing a speech enhancement algorithm for audio/video communication is preserving perceived speech quality while suppressing noise. In the past, constrained objective functions have been used to optimize such a compound objective. Alternatively, a simpler objective like the mean squared (log) error (MSE) can be optimized. In a deep learning framework, the major benefit is the relative simplicity to incorporate complex learning objectives that achieves higher quality and intelligibility [77].

Despite the success of deep learning algorithms in speech enhancement, their performance for real-time audio processing applications and complex noisy situations is not well investigated. In a real-time application, the system needs to perform all the steps, including pre-processing, feature extraction, prediction, and post-processing in a short timeframe and the output must be intelligible without any noticeable delay by human ears. Therefore, the application of noise suppression for speech enhancement using deep learning is considered a highly complex problem.

1.3. Summary of the Research Project

This thesis aims to design a framework as an intervention technique for individuals with ASD to attenuate selected aversive sounds which cause negative impacts on their social life interacting with their environment. The selected sounds are commonly reported among the ASD population as examples of aversive noises. First, a deep learning-based algorithm is proposed to improve environmental sound classification tasks by using various strategies for modeling and data augmentation. The proposed model for classification is a unified CNN-RNN network to benefit from both CNN as a powerful feature extractor and RNN as a strong sequential pattern learner. The input of the model is a frequency representation of the audio signal, and the output is the predicted class. To improve the model's accuracy, several techniques are used including batch normalization, transfer learning, and three feature representations map. As mentioned earlier, deep learning models need a huge amount of data samples to achieve their remarkable performance. In this thesis, in order to overcome the lack of data, two data augmentation strategies including traditional and intelligent augmentation are considered. In the traditional data augmentation approach, first, different background noises (crowd sound, street sound, and restaurant) are added to the data samples and then, pitch shifting is applied. For intelligent data augmentation, a deep convolutional

GAN structure is used as a high-quality augmentation method to generate further data samples and improve the performance of the classification model. The developed CNN-RNN network can work in real-time and be programmed with executable instructions to identify and classify an aversive sound in the environment.

When an aversive sound is detected, the filtering system is needed to remove it from the signal, in real-time. The main goal of the filtering part is to automatically remove the identified aversive signals from the noisy signal using the generated feature map and obtain a clean sound. In this thesis, a DNN-based learning framework is proposed to address the single-channel speech enhancement problem for real-time applications. Single channel speech enhancement is typically referred to the methods in which a filter is applied to the noisy speech to recover enhanced speech signal. In this framework, an audio file comprising a combination of speech and noise is captured using a microphone. The system performs mathematical analysis to extract the audio features and provide the input to the DNN model. The DNN model is designed and trained to construct a ratio mask from the noisy signal by employing linear and nonlinear weights in the learning process. Then, the clean speech coefficients are computed using the ratio mask. Pre-processing and post-processing steps are included to increase speech quality and integrability. The framework is trained and tested on complex noise structures such as dog barking, siren, vacuum cleaner, and engine.

These algorithms are integrated and deployed into a platform using a graphical user interface. The GUI can provide a recommended action to the user. The recommended action can be, for example, (1) suppress the signal by stopping the transfer of the signal from the microphone to the speaker, (2) attenuate the ambient sound by lowering the volume, (3) remove the aversive sound signal from the mix-signal, or (4) mask the ambient sound by playing pre-recorded sounds. The GUI can be responsible for all communications, settings, customizations, and user-defined operations. In order to make the system work in real-time, a framing strategy is proposed to show how processing steps are organized. Afterward, several pilot sessions are conducted with autistic adults to examine the noise attenuation system introduced in this thesis in real-life.

1.4. Contribution and Objectives of Research

In summary, the major contributions and objectives of this thesis are as follows:

- **An intelligent intervention techniques is proposed and designed to ameliorate decreased sound tolerance in individuals with ASD.**

In this thesis, a novel intervention technique which employs signal processing and deep learning methods is proposed to address the issue of DST among autistic individuals. The proposed techniques is able to detect a specific aversive sound in the environment and then filter the detected sound while leaving the other sounds intact.

- **A unified CNN-RNN classification framework is proposed for aversive sound classification.**

In order to classify and identify aversive sounds, a unified RNN-CNN method is proposed to take advantage of both methods. In this structure, a parallel combination of CNN and RNN networks is constructed in which the CNN branch plays the feature extractor role to extract semantic representations from the inputs. RNN branch plays the temporal summarizer role to label relationships and labels dependency. Experimental results demonstrate that the proposed approach achieves superior performance compared to the state-of-the-art methods.

- **High-quality new data samples are generated using a deep convolutional GAN model.**

The high scalability and excellent sample generation capabilities of deep convolutional GAN are employed to create new samples and improve the performance of the classification model. Deep convolutional GAN techniques is used mainly for image data augmentation, and in this thesis its performance is evaluated on environmental sound spectrogram augmentation. The experimental results show that this data augmentation method can produce spectrograms with similar structures to the training set.

- **A novel speech enhancement DNN-based framework is proposed**.

A DNN-based learning framework is proposed to address the single-channel speech enhancement problem for real-time applications. The simulation results indicate that the

proposed method can remove the background noise from the speech signal, even in the presence of complex noisy conditions (e.g., unsteady real-world environments) and difficult noise types (non-stationary and highly dynamic noises).

- **A comprehensive framework is designed to work in real-time audio processing.**

A comprehensive framework is designed to integrate the classification and filtration algorithm. A cumulative framing strategy has been considered to operate the system in real-time with unnoticeable delay. A user interface is designed and is programmed with executable instructions to obtain input data, transmit data and provide output data.

1.5. Thesis Outline

This thesis consists of five chapters which are organized as follows.

In chapter 2, the theory and experimental methodology for environmental sound classification including feature selection, feature extraction, model selection, and training process are discussed. The simulation results are provided at the end of this chapter and compared with the existing state-of-the-art methods. This chapter addresses the environmental sound classification and data generation objective.

In chapter 3, the theory and experimental methodology for filtration including audio preprocessing, model selection, and post-processing are discussed. The simulation results are provided at the end of this chapter and compared with the existing methods. This chapter addresses the sound filtration objective.

In chapter 4, the combination of classification and filtration frameworks is presented. Then, the real-time setting and implementation are explained. The details of the prepared demo and its features are also provided in this chapter. This chapter addresses the objective of designing a comprehensive framework as an intervention technique for individual with ASD.

In chapter 5, a general discussion, conclusions, and recommendations for future works are stated.

Chapter 2.
Environmental sound classification

2.1. Introduction

Deep Learning emerged in 2006 as a new area of machine learning and achieved significant success in various applications of computer vision, phonetic recognition, conversational speech recognition, natural language processing, audio processing, robotics, etc [78]. Speech, audio, and acoustics are one of the largest deep learning application areas across research labs and industry. Before deep learning, sound recognition applications were dependent on traditional signal processing tools to extract features such as phonetics concepts. These required a lot of signal processing expertise, hand engineering techniques, and efforts for system tuning and optimization. However, with deep learning, traditional audio processing techniques are no longer needed and a standard data preparation process will be sufficient [47]. The features will be extracted automatically from the prepared data and there is no need for manual or custom generation of the features. Moreover, deep learning techniques have achieved remarkable performance in the area of audio and speech processing [46], [71], [72].

Deep learning involves the acquisition of multiple-level representations and abstractions that aid in data interpretation [78]. Deep learning is implemented using a neural network architecture consisting of several layers. These layers involve linear and non-linear functions to model complicated data such as audio and image. Each layer receives data from its previous layer, imply the functions, and then passes it to the next layer. Each layer consists of several neurons, which are the network's core basic units and have various forms of connections to other neurons in other layers. Deep neural networks, in essence, are based on the architecture of the human brain. The network is trained to produce prescribed outputs given provided inputs. Figure 2-1.A shows the schematic of a learning structure known as a neural network including input, hidden, and output layers. In this figure, the input layer is where the initial data is fed to the neural network. The output layer produces the result for given inputs and hidden layers are defined as the intermediate layer between the input and output layer and place where all the computation is done. The initial layers allow the system to extract a structured and complicated understanding of the patterns in the data. The black arrows in this figure

represent the weights that are optimized during the learning process. The weights' main function is to give importance to those inputs in each layer that contribute more towards the learning. It does so by introducing the sum of the scalar multiplication between the input value and the weight matrix. This operation is shown by capital sigma in Figure 2-1 in each neuron. Afterward, the weighted sum of the inputs in each neuron is activated by an activation function. The activation function introduces non-linearity in the working of each neuron to consider varying linearity with the inputs. Without this, the output would just be a linear combination of input values and would not be able to introduce non-linearity in the network. In Figure 2-1, the activation function is introduced by capital $\mathcal{F}$, following the weighted sum. Note that the activation function does not consist of any learnable parameter to be optimized. Optimizing the parameters of the weighted sum in each layer is called cognitive analogy and is performed by using optimization algorithms such as backpropagation.

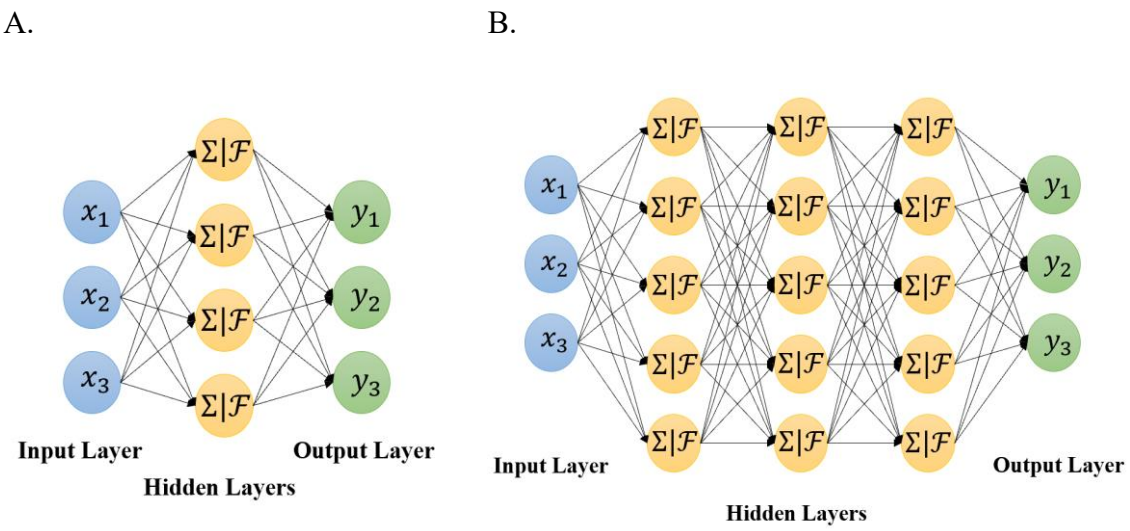


Figure 2-1 Schematic of A. neural network, and B. deep neural network

A neural network with multiple hidden layers and multiple nodes in each hidden layer is known as a deep learning system or a deep neural network. Deep learning neural networks are distinguished from neural networks on the basis of their depth or number of hidden layers. Deep learning is the development of deep learning algorithms that can be used to train and predict output from complex data. The word “deep” in deep learning refers to the number of hidden layers i.e. depth of the neural network. Essentially, every neural network with more than the three layers, that is, including the Input Layer and

Output Layer can be considered a deep learning model. Figure 2-1.B shows the structure of a deep neural network. In this chapter, a deep learning-based structure is proposed to address environmental sound classification. Several deep learning techniques are employed to improve the deep learning structure's accuracy and performance. The proposed structure is able to classify environmental sounds with a high accuracy level and achieves superior performance compared to the state-of-the-art classification models.

In the following, the theory behind the deep learning techniques is introduced. Afterward, the proposed techniques are introduced and examined. Then, the experimental results and conclusion are described.

2.2. Theory

Two sub-categories of deep learning techniques include deep discriminative models for supervised applications such as classification, recognition, and speech recognition, and generative models for unsupervised applications such as clustering, and dimensionality reduction. Supervised models need labeled data sets for training and are mainly used for classification and regression problems. On the other side, unsupervised models do not need prior labeled training datasets and are mainly used for clustering and data visualization. In this thesis, the classification techniques are used for sound classification. The details of the deep neural network structure and techniques, including Convolutional Neural Network (CNN) layers, Recurrent Neural Network (RNN) layers, feature extraction, transfer learning, and data augmentation are explained in detail in the following sections.

2.2.1. CNN

CNNs are the main and the most widely used structure for image classification, image localization, and object recognition in computer vision. CNN is similar to the presented neural network in Figure 2-1 which includes neurons, weights, and activation functions. In addition, CNN employs outstanding components that make them unique for image processing applications. These components include convolutional layers and pooling layers. Figure 2-2 shows an overall schematic of a CNN structure including

convolutional layers, pooling layers, fully connected layers, and softmax function at the end.

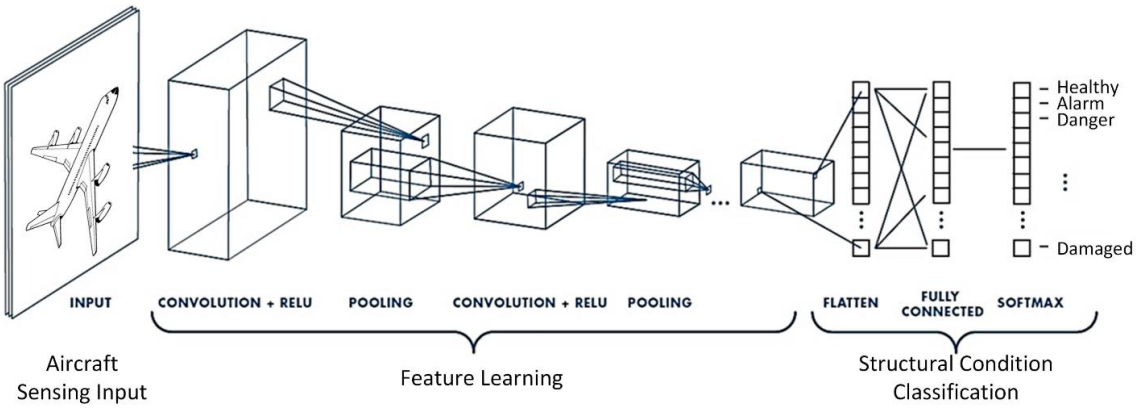


Figure 2-2 A typical convolutional neural network structure including input, conolutional layers, pooling layers and fully connected layers [79]

Convolution is the mathematical operation over the matrix representation of an input image. An image can be represented as a matrix of image elements arranged in columns and rows. The yellow matrix in Figure 2-3 shows a matrix representation of an image, e.g. the input image in Figure 2-2. The convolutional operation is defined as the convolution of the input image and the filter and results in the output also known as the feature map. Filters detect spatial patterns such as edges in an image by detecting the changes in intensity values of the image. In CNNs, filters are often represented by a 3×3 matrix. For example, $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ filter can be used for vertical lines detection of an input image. The filter weights are considered as the learnable parameters in the CNNs. In the terminology of CNN, a convolution operation is a linear operation that involves the multiplication of a filter with the input image represented by a matrix. The convolution operator takes the convolution filter and slides it over the image or input matrix. In other words, the filter moves on the image and scans the input image. By placing the filter on each part of the image, the numbers in the filter are multiplied by the corresponding pixels in the image. Then, all the numbers add up to create one cell in the output. Figure 2-3 indicates an example of convolutional operation. For example, placing the filter on the upper left of the input shown by the dashed circle in Figure 2-3, will result in 12, which is the product of each cell in filter and input, and adding up them together, $(1 \times 1) + (2 \times -1) + (3 \times 4) + (5 \times -2) + (8 \times 1) + (9 \times 0) + (5 \times 0) + (0 \times 2) +$

$(1 \times 3) = 12$. This number will be placed at the upper left of the output. Then, the filter will slide one pixel to the right and the same procedure will be applied to construct the second output which is 4.

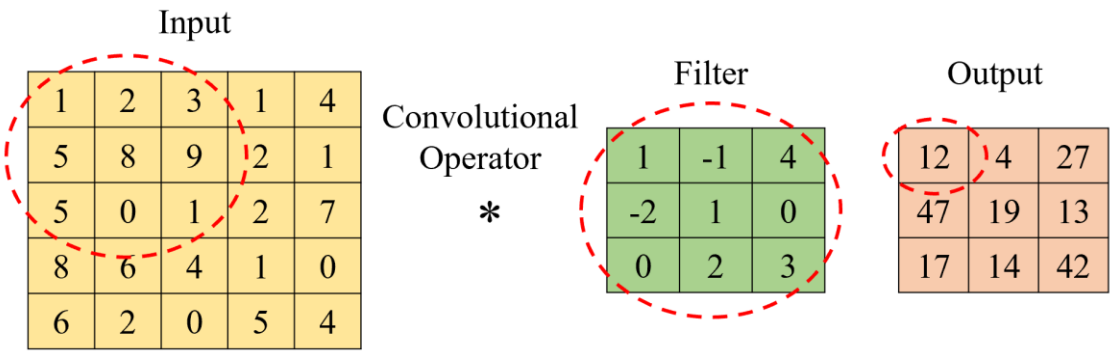


Figure 2-3 A convolutional operation

This pixel size is also known as the stride size, which in this case is 1. As another example, if the stride size was 2, the filter moves 2 pixels toward the right to calculate the second output and so on. The higher stride results in a lower dimension output. The same procedure also applies when the filter moves down to scan the entire image. The output of multiplying the filter with the input image is constructed as a reduced dimension matrix also known as a feature map which contains features of the image with respect to the filter. According to Figure 2-2, these feature maps can be considered as an input to the next filter which constructs another CNN layer in the CNN architecture.

According to Figure 2-2, the constructed feature maps are passed through an activation function in order to create nonlinear features in the neural network. In complicated data structures such as images and audios, nonlinear activation functions make it easier for a neural network to learn information from the data and differentiate between the outputs. Figure 2-4 indicates a number of nonlinear functions widely used as an activation functions. Among all nonlinear functions, the Rectified Linear Unit (ReLU) function is the most popular. ReLU function is defined as the $max(0, x)$. Therefore, it maps the negative values to zero and leaves the positive values as they are. The advantage of using ReLU compared to conventional sigmoid, and tanh is that ReLU is faster to compute than the other activation functions. This makes a significant

difference in training and inference time for neural networks. Leaky ReLU is another version of ReLU which has a small slope for negative values instead of a flat slope.

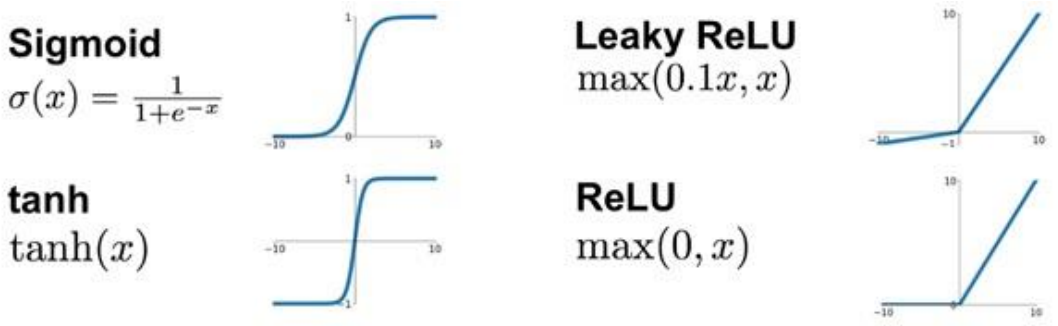


Figure 2-4 Nonlinear activation functions in deep neural networks

Figure 2-2 also contains pooling layers following each convolutional layer and its associated activation function. The purpose of the pooling layer is to reduce the spatial size of the output of the convolution layer and consequently reduce the computation amount. Pooling layers compute the summary of the nearby output of the convolution layer and they reduce the dimension of the convolution layer without sacrificing too much information. Max pooling and average pooling are the most commonly widely used pooling layers in CNNs. To illustrate, a 2×2 max-pooling layer considers a rectangle in the feature map in which contains four pixels, as the input to the max-pooling layer. The maximum of the four pixels is calculated and considered as a single pixel in the output. Unlike filter weights, the pooling layer has no wights to be trained in the CNN structure and it performs a simple and effective dimensionality reduction of the convolution layer.

Extracted features using convolutional layers are eventually transferred into a vector to create the final output vector. Fully Connected (FC) layers which are also called dense layers are used for this purpose. A CNN structure may contain several dense layers after convolutional layers as shown in Figure 2-2. The output of the last convolution and pooling layer is converted to a single row named dense layer. Then, similar to a neural network, they may pass through different dense layers with activation functions which are contained trainable parameters. At last, the softmax function is used as the activation function in the output layer of CNN to predict a probability distribution.

Softmax is used as an activation function for multi-class classification problems where the number of classes is more than two class labels.

2.2.2. RNN

An RNN is a class of artificial neural networks in which the connection lines between the nodes are sequential. It allows RNN to model the dynamic temporal behavior of sequences through directed cyclic connections between the nodes [80]. The RNN model has different structures including LSTM [81], and Gated Recurrent Unit (GRU) [82].

In principle, a standard RNN iterates over the individual input feature vectors $(x_1, x_2, \ldots, x_T)$ and computes the sequence of hidden state vectors $(h_1, h_2, \ldots, h_T)$ [45]. At a frame time t, where $1 \leq t \leq T$, h_t is computed as

$$h_t = \mathcal{H}(x_t, h_{t-1}) \tag{1}$$

where $\mathcal{H}$ denotes the hidden layer function. In the proposed network, the GRU cell is employed in which the function $\mathcal{H}$ is implemented by the compound of the following functions,

$$r_t = sigm(W_{xr}x_t + W_{hr}h_{t-1} + b_r), \tag{2}$$

$$z_t = sigm(W_{zz}x_t + W_{hz}h_{t-1} + b_z), \tag{3}$$

$$\tilde{h}_t = tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h), \tag{4}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t. \tag{5}$$

where the W variables denote the weight matrices and the b variables are the biases. The r, z, and $\tilde{h}$ variables represent the reset gate vector, the update gate vector, and the new hidden state vector candidate, respectively. The $\odot$ operator denotes the element-wise vector product. Consider A$\odot$B as the element-wise multiplication between

A and B. A and B are the same dimension matrices to compute a new matrix, C, with the same dimension as A and B. Matrix C elements are the products of the corresponding elements in A and B, $C_{i,j} = A_{i,j} \times B_{i,j}$ which i and j indicate the index number of rows and columns, respectively. RNN networks have different structures according to the nature of their input and output. The sequence-to-label structure is when the input is a sequence of data, and the output is a single label. The classification problem is considered a sequence-to-label RNN structure, and the network output is determined by the final state vector h_T as the following.

$$\hat{y} = sigmoid(W_{hy}h_T + b_y) \tag{6}$$

Figure 2-5 displays a sequence-to-label RNN structure. In this figure $H_0, H_1, H_2, \ldots$ represent the output of the previous layers or hidden states, $x_0, x_1, x_2, \ldots$ represent the input, and $\hat{y}$ is the output label.

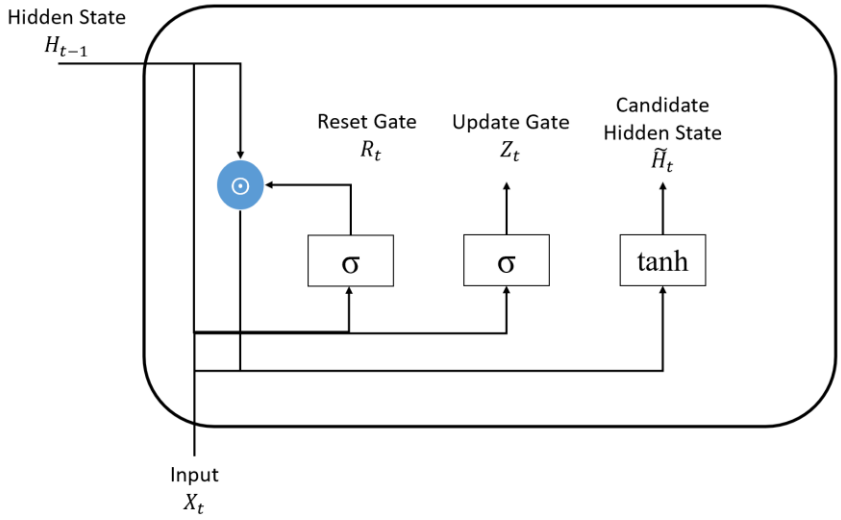


Figure 2-5 A Schematic of an GRU over time including input and hidden state vectors

2.2.3. Neural Network Training and Transfer Learning

CNNs and RNNs are the main blocks to construct a deep neural network model architecture. The constructed models learn to produce relevant output according to their input given a training dataset of examples. A training dataset includes a set of pairs of

input and output data samples which are used to train the neural network model. The training process involves finding a set of weights in the neural network, including the learnable weights in CNN, RNN, and dense layers such that the neural network output can predict the dataset outputs with acceptable performance. The training process is an iterative procedure, meaning that it progresses step by step with small updates to the model weights in each iteration to improve the performance and the accuracy of the neural network model. This procedure is a very time-consuming and costly process that needs a huge amount of memory and rich datasets to achieve an acceptable performance. Nowadays, there exist huge datasets for different applications of machine learning and deep learning. On the other hand, the cost of training neural network models on these datasets is very expensive. To overcome these problems, there are several methods that are commonly used in the area of deep learning to overcome this problem including transfer learning.

Transfer learning is often used in computer vision and natural language processing. It basically takes advantage of the knowledge of a pre-trained model for a different but related model. In other words, the weights that a network has learned for problem A are transferred to the new problem B. The basic idea is to apply the knowledge gained by a model in the problem for which a lot of data samples are available to another problem that has limited data samples. In transfer learning, some of the middle and early layers of the model are preserved, and only the last layers need to be retrained to make the model appropriate for the new task. This reduces the model's dependency on labeled data. For example, ImageNet is a large data set consisting of about 14 million images, and more than 21000 classes [83]. There exist different CNN architectures including Inception [84], VGG family [85], and ResNet [86] which are pre-trained on the ImageNet dataset, and the model structures along with their learned parameters are further used for real-world image-based classification problems with fewer amount of data. Figure 2-6 shows a schematic view of transfer learning. In this figure, dataset 1 contains a huge amount of data that is trained on a CNN-based neural network. The model structure and the learned parameters are stored as “knowledge", and further used for other tasks using a smaller dataset 1. In this thesis, the transfer learning technique is used to train the neural network and to use the knowledge from a pre-trained model on a huge data set and also to save cost and time compared to training from scratch.

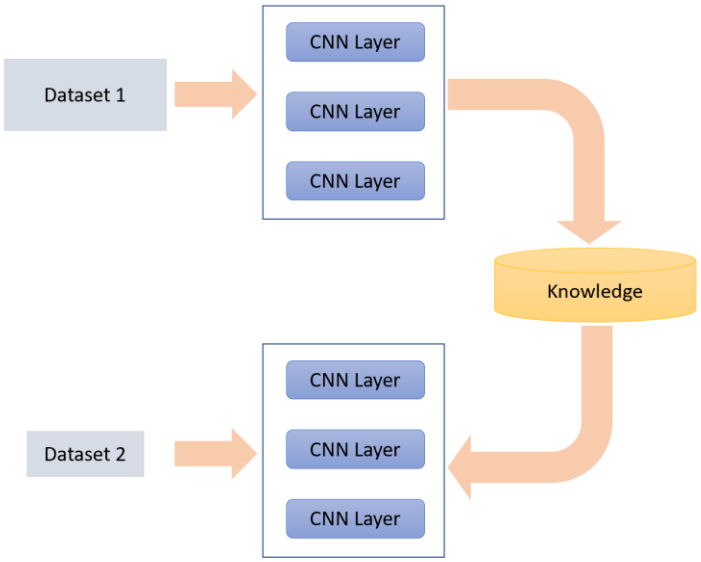


Figure 2-6 A schematic view of transfer learning

2.2.4. Input Data Preparation and Feature Engineering

Data preparation is the process of cleaning and transforming raw data for further processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data, and combining data sets to enrich data. Feature engineering or feature extraction is the process of using specific engineering knowledge to extract features (characteristics, properties, attributes) from raw data. Often these two concepts are used interchangeably, however, in this thesis, feature extraction is defined as a set of engineering operations performed on the raw data in order to produce suitable input data containing meaningful features for the classification task.

Sounds are created by variations in air pressure. An audio signal is a representation of sound, which is represented by a changing level of electrical voltage, over time. An audio signal feature can be represented in time-domain and frequency-domain representations. **Error! Reference source not found.** shows a time-domain example of a sine wave which can be considered as an audio signal for a simple explanation of the concepts related to an audio signal. In time-domain feature representation, the intensity of air pressure variations over time is called amplitude. The period is the amount of time that takes for a signal to complete one full cycle. The

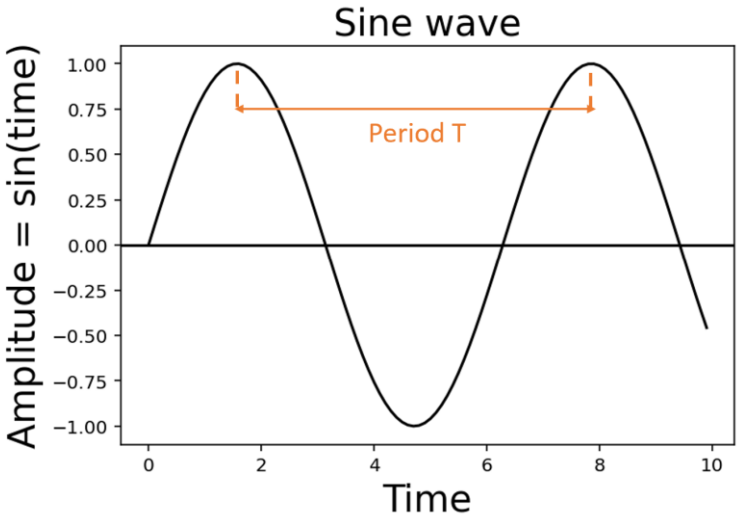


Figure 2-7 Sine wave signal

amplitude and period definitions are described using **Error! Reference source not found.**. In addition, frequency is defined as the number of waves produced by a signal in one second. The frequency is measured in Hertz.

Various sounds that we hear do not follow such basic and predictable patterns, which are introduced using a sine-wave in **Error! Reference source not found.**, and contain a constant frequency and amplitude over time. For example, Figure 2-8 shows a human speech in time-domain representation which include various frequencies and amplitudes. Further, sound signals from different sources with various frequencies can be combined to form composite sound signals with highly complicated patterns.

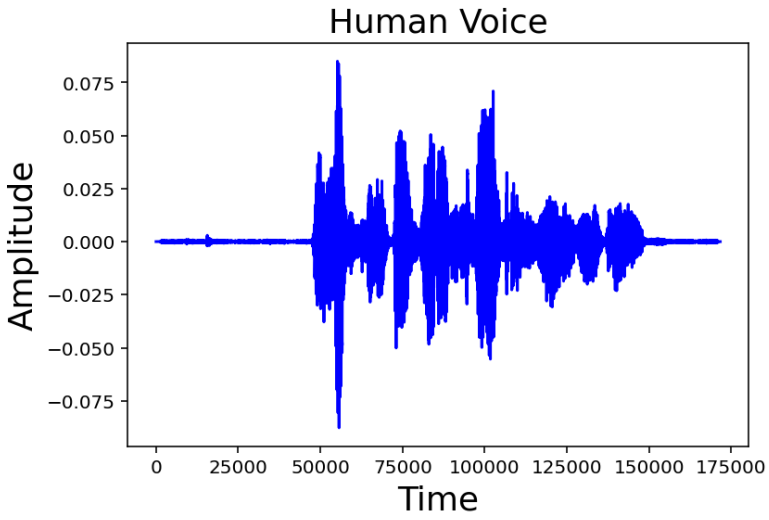


Figure 2-8 A sample of human voice signal

Frequency-domain features, including spectrogram, depict the frequency information of a signal as well as its amplitude. To transfer a signal from the time domain to the frequency domain, the Fourier transform is used. Fourier Transforms decompose a signal into its constituent frequencies. Figure 2-9 shows the spectrogram of the human voice signal shown in Figure 2-8. In this figure, the horizontal axis shows the time, the vertical axis shows the frequency, and the amplitude is shown by color.

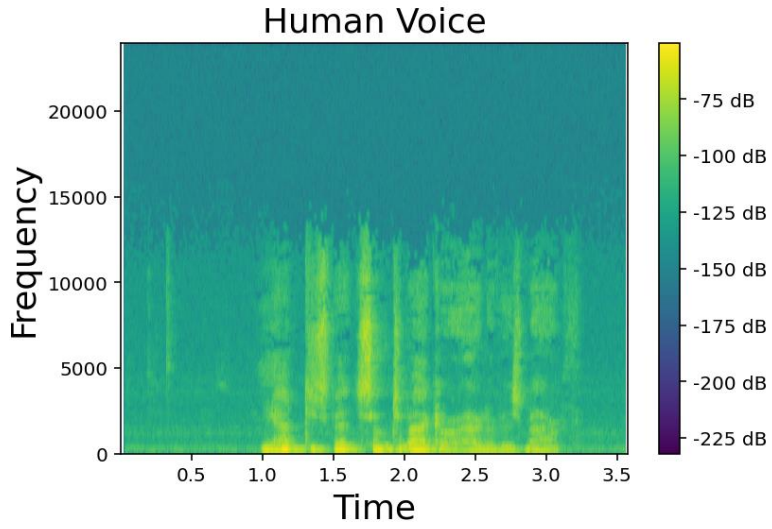


Figure 2-9 A sample of human voice spectrogram

Among spectrograms, Mel-spectrogram is a logarithmic frequency scale of the spectrogram and is considered one of the most common and effective features for speech and audio recognition [47]. The mel-scale is a logarithmic frequency scale designed to better adapt to human hearing. This feature is characterized by experimenting with human interpretation of pitch to describe the human auditory system on a linear scale [87]. The experiment shows that pitch is linearly perceived in the frequency range of 0-1000 Hz. Above 1000 Hz, the scale becomes logarithmic. Equation (7) describes the formula to convert f (hertz) into f (mel).

$$f_{mel} = 2595\ log_{10}(1 + \frac{f_{hertz}}{700}) \tag{7}$$

Figure 2-10 shows a mel-frequency spectrogram representation of the human voice signal of Figure 2-8.

The time-domain and frequency-domain features explained here are utilized to extract semantic features from the sound signal to be fed as the input to the neural network model.

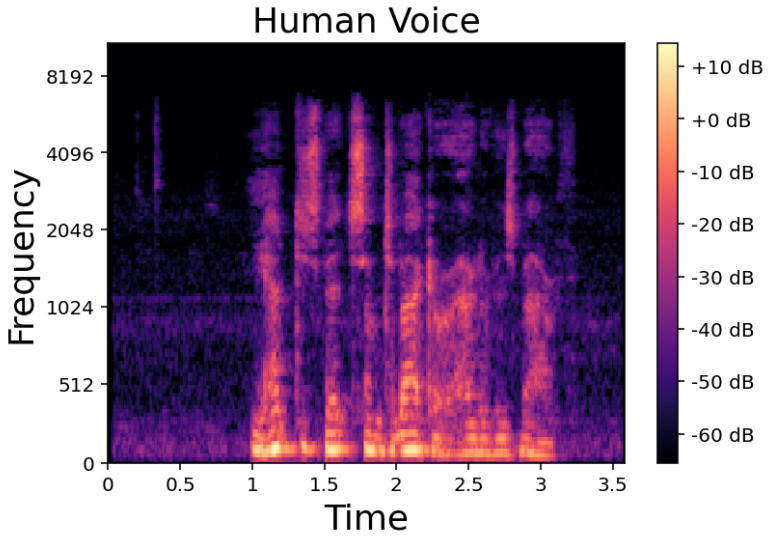


Figure 2-10 A sample of human voice mel spectrogram

2.2.5. Data Augmentation

In many applications, access to a huge dataset is expensive and sometimes not possible. In order to increase the amount of data, data augmentation techniques are used to generate new data from the existing data. Data augmentation is a powerful strategy to increase the diversity of available data and make it possible to train models without collecting new data. There are two data augmentation strategies including traditional and intelligent augmentation. In traditional data augmentation, different data deformations are used including adding noise, rotation, cropping, translation, flipping, scaling, and brightness [88]. For audio, commonly used data augmentation techniques include adding background noises like crowd sound, street sound, and restaurant to the data samples, and shifting the pitch of the audio samples to create different new sounds. In this thesis, the two well-known conventional data augmentation techniques are used to increase the number of training data. Figure 2-11 shows the new data constructed from a human speech and a background noise, in this case, a restaurant environment, which is used in this thesis. Also, Figure 2-12 shows the spectrogram of the original audio versus the shifted versions of the original audio.

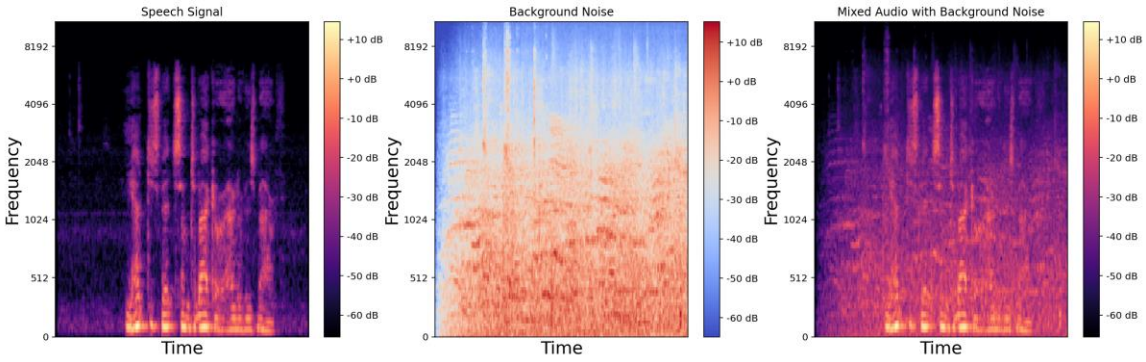


Figure 2-11 Speech signal, background noise, mixed-signal

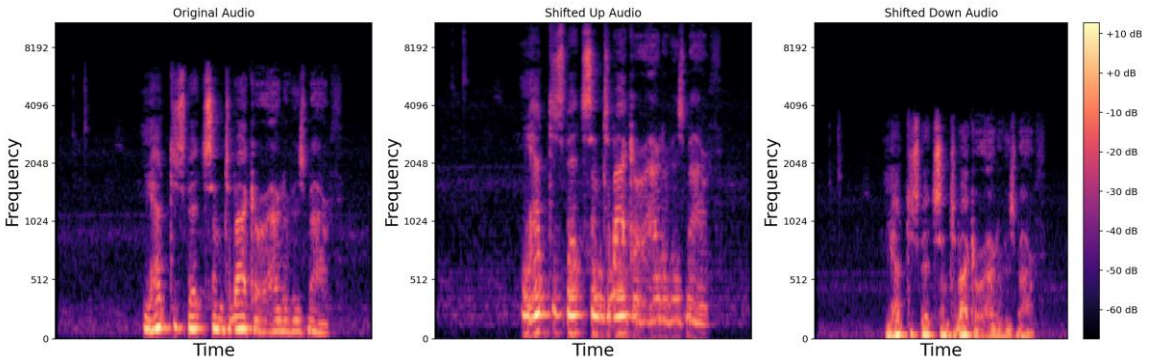


Figure 2-12 Original audio, shifted up the audio pitch by half an octave and shifted down the audio pitch by half an octave

Another category of data augmentation techniques is known as intelligent data augmentation. Generative Adversarial Networks (GANs) are an innovative intelligent data augmentation technique to create new data instances that are similar to the training data. The GAN framework consists of two models including a generative model and a discriminative model. The generator creates samples close to the training samples and feeds them to the discriminator. The discriminator then receives the images together with the real images and compares them to distinguish the real ones. This process is continued until the discriminator cannot distinguish between real and generated images. GANs are known to have several drawbacks due to the battle of the two networks. The outputs of the generative convolutions model are often inaccurate. In addition, GANs are hard to tune and get to work properly. To tackle those drawbacks, a class of GANs called Deep Convolutional Generative Adversarial Networks (DCGAN) is proposed that has a set of architectural constraints to stabilize GANs [60]. In DCGAN, the discriminator (D) is a set of convolution layers with stride, so it downsamples the input image at every convolution layer. On the other hand, a generator (G) is a set of convolution layers with transpose convolutions, so it upsamples the input image at every convolution layer [60]. The architecture of DCGAN is shown in Figure 2-13.

In this thesis, DCGANs are utilized in order to generate new training data to be fed to the neural network classifier.

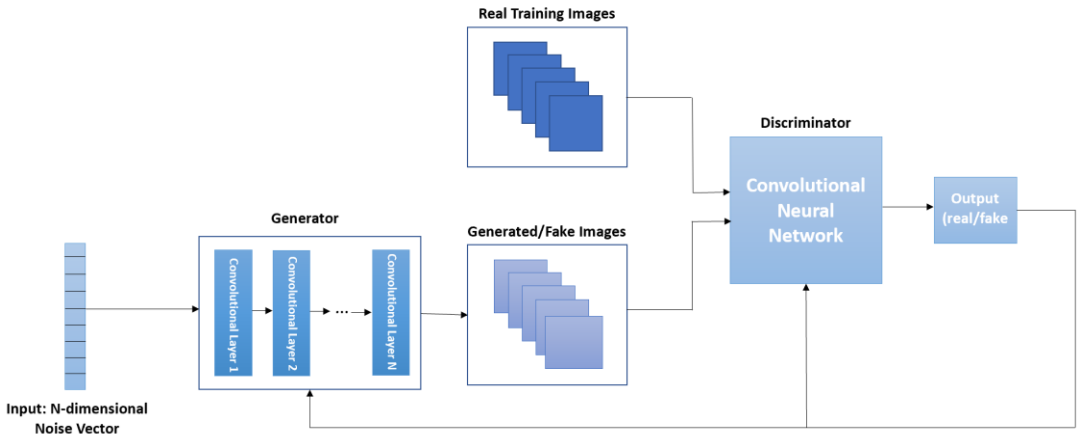


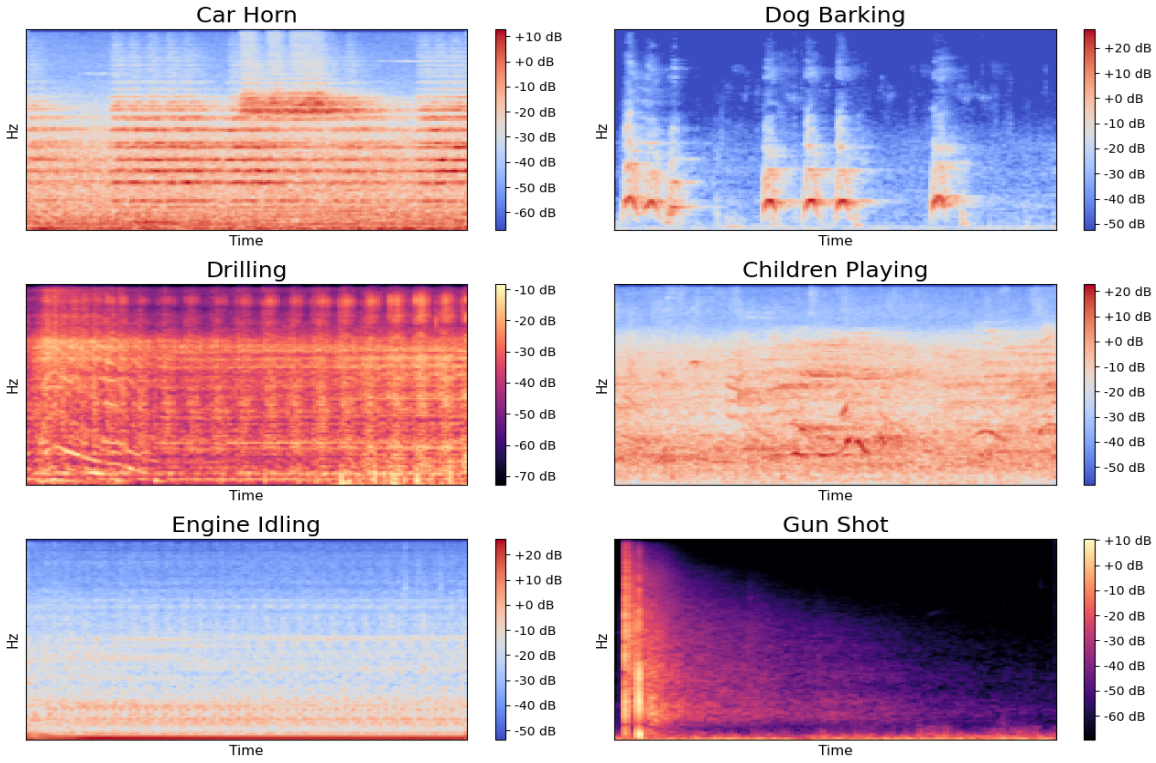
Figure 2-13 DCGAN Architecture

2.3. Experimental Methodology

In this section, the previously illustrated techniques are used to architect a deep neural network that has the ability to accurately classify different classes of environmental sounds including dog barking, siren, jackhammer, and so on. To accomplish this task, the utilized available dataset is illustrated. Then, the required features are extracted from data in order to be fed to the deep neural network for the environmental classification task.

2.3.1. Dataset

There are several available datasets for various sorts of sounds including ESC-10, ESC-50, UrbanSound8K, and FSDK. These datasets are created for different applications and contain a huge number of labeled audio samples. In this thesis, the UrbanSound8K dataset is used to evaluate the performance of the classification and data augmentation techniques. UrbanSound8K consists of 8732 ordinary sounds from daily life. These sounds are categorized into 10 classes, i.e. drilling, dog barking, siren, air conditioner, car horn, children playing, engine idling, gunshot, jackhammer, and street music. These sounds are digital audio files in .wav format and less than 4 seconds. Mel-spectrograms of some samples of this dataset are presented in Figure 2-14.


Car Horn
Hz
Time
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
Dog Barking
Hz
Time
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
Drilling
Hz
Time
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
Children Playing
Hz
Time
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
Engine Idling
Hz
Time
+20 dB
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
Gun Shot
Hz
Time
+10 dB
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB

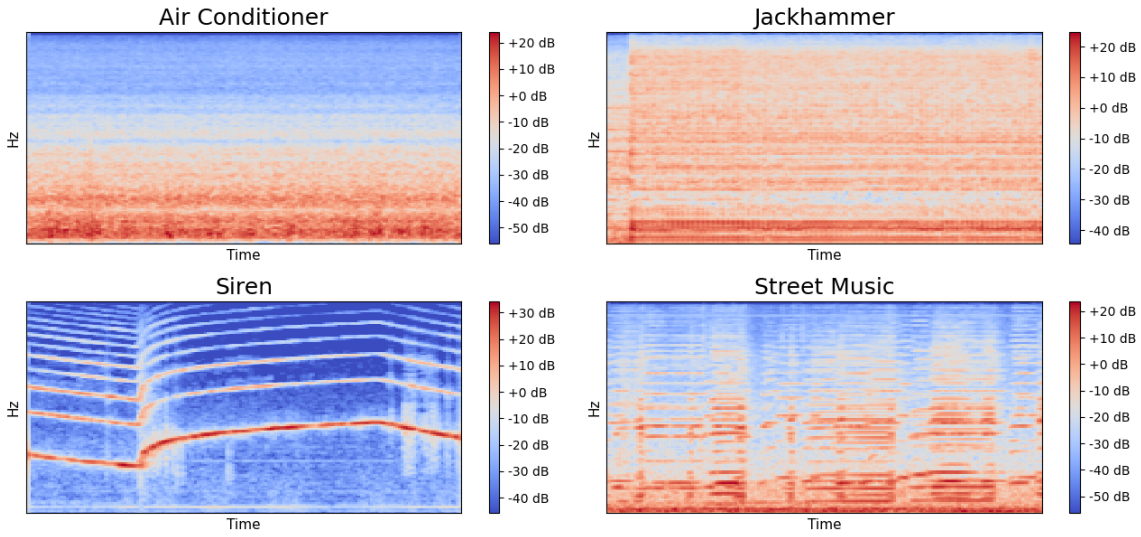


Figure 2-14 Mel Spectrograms of some audio samples from UrbandSound8K dataset

2.3.2. Feature Selection and Transfer Learning

In this study, a total of three feature engineering techniques are used to build three feature representation maps including mel-spectrogram, and decompose audio time-series data into harmonic and percussive components. The sounds which are considered in this thesis are in various ranges of frequencies. Some of them are transient, non-stationary, and have a noise-like structure. They can be categorized as a percussive sound class [89]. On the other hand, some of the sounds are consistent and stationary sounds and can be categorized as a harmonic sound class [89]. mel-spectrogram, percussive mel-spectrogram, and harmonic mel-spectrogram ultimately provide a three-dimensional image feature map for each audio. Figure 2-15 depicts the architecture for this feature map.

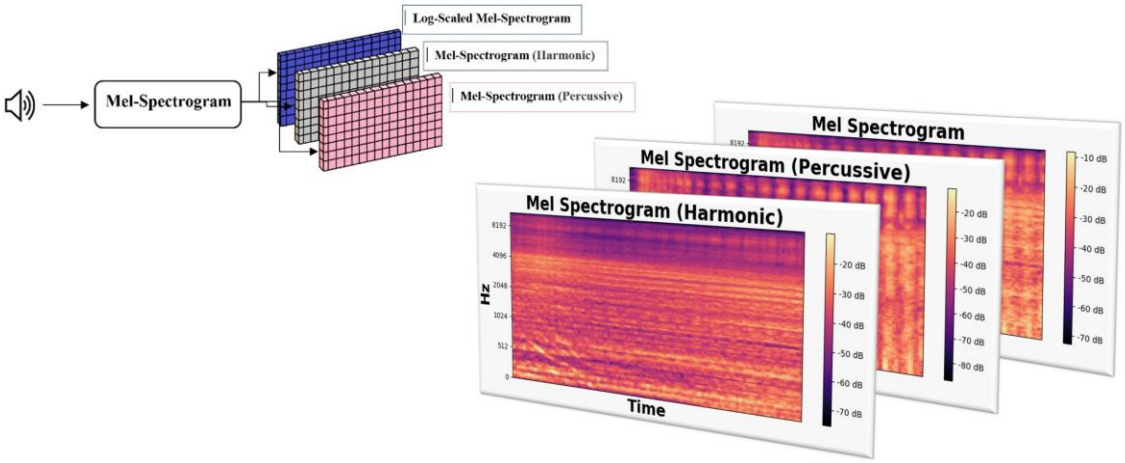


Figure 2-15 The architecture of feature map creation

2.3.3. Data Augmentation

In this section, the DCGAN data augmentation technique is used to generate further data samples from the UrbanSound8K dataset and examine the performance enhancement of the classification model for environmental sound recognition. The length of each data sample is up to 4 seconds. Mel-spectrogram feature extraction is used, and the mel-spectrograms are generated using equation (7). The dimension of the generated images is 768×384. To avoid a huge amount of training parameters, the original images are resized to 64×64. Then, the windows of audio data are extracted as sub-samples from each audio sample. As the first feature map, a log-scaled mel spectrogram is created using audio sub-samples.

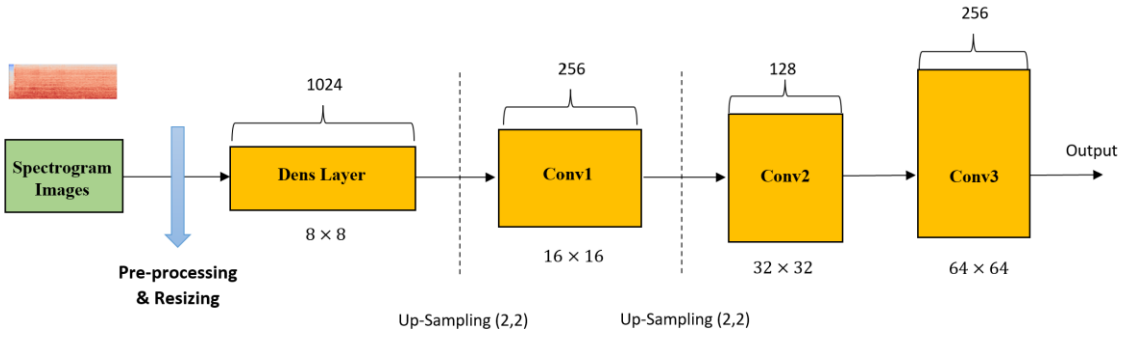


Figure 2-16 The DCGAN generator Architecture

The architectures for the generator is illustrated in Figure 2-16. The GAN generator in this study is comprised of 4 convolutional layers. Batch normalization is applied to every layer of the network to improve learning efficiency. Batch Normalization is a method used to normalize the values of each batch of output features at each layer of the deep neural network before feeding it to the next layer. Batch normalization has proved advantages which makes the deep neural network training process faster and more stable through the normalization of the inputs at each layer [90]. The DCGAN model is trained over 400 epochs using a batch size of 32. The DCGAN structure including the overall layers structures is set based on [60]. In the generator model, the ReLU activation function is used after each layer except the last one. For the last layer, the hyper tangent activation function is applied to obtain the image of 3 channels with pixel values between -1 and 1.

For the discriminator, instead of a hyper tangent, standard sigmoid activation is used on the output layer to determine the probability of the generated image. The stride size for the discriminator is 4, 4, 4, 2 respectively. The Adaptive Moment Estimation (ADAM) optimizer is used in order to update network weights [91].

2.3.4. Classification Model Architecture

In the proposed classification architecture in this thesis in order to overcome the environmental sound classification task, a parallel combination of CNN and RNN networks is constructed. Figure 2-17 shows the detailed structure of the proposed CNN-RNN network. In this architecture, the CNN branch plays the feature extractor role to extract semantic representations from the inputs. RNN branch plays the temporal summarizer role to label relationships and labels dependency. The pre-processed training data is fed to each neural branch in order to extract features and learn patterns using both CNN and RNN networks. Afterward, the extracted features are concatenated and passed through three dense layers in order to construct the final output which is the probability of each class related to the input data.

In order to utilize the knowledge from previously learned models and apply them to the classifier task in this thesis, transfer learning is used to train the CNN branch. The pre-trained VGG-19 (Visual Geometry Group) model is used as a feature extractor to extract features from all three dimension feature maps. VGG-19 is a convolutional neural

network model trained on the ImageNet dataset which contains over 14 million images belonging to 1000 classes. The pre-trained VGG model then follows with the average pooling layer and drops out in order to decrease the dimensions of the input and make use of regularization to reduce overfitting and improve generalization error, respectively. The extracted features are then flattened using a dense layer with a Leaky ReLU activation function to make them ready to be concatenated with the output of the RNN branch.

In the RNN branch, a deep GRU-based RNN architecture is used. The input to this branch is the mel-spectrogram of the constructed training data which are fed to the input layer. The RNN branch consists of two GRU layers followed by batch normalization. Afterward, a dense layer with Leaky RelU followed by batch normalization is used to make the output ready to be concatenated with the output of the CNN branch. The next step is to concatenate the output of the CNN and the RNN branches in order to make use of the extracted information in both branches. To complete the classification architecture, the concatenated data are passed through two dense layers with Leaky ReLU and batch normalization. Afterward, the last layer consists of a dense layer with the SoftMax activation function which predict a probability distribution for the 10 classes.

In order to train the explained architecture, the loss function is calculated by comparing the predicted class for each input data sample and its true label. Then the cost function is calculated using all the examples in each batch of training data. For the loss function, the categorical cross-entropy loss is used which is a well-known loss function for classification tasks. Also, the ADAM optimizer is used to update the weights of the CNN-RNN architecture.

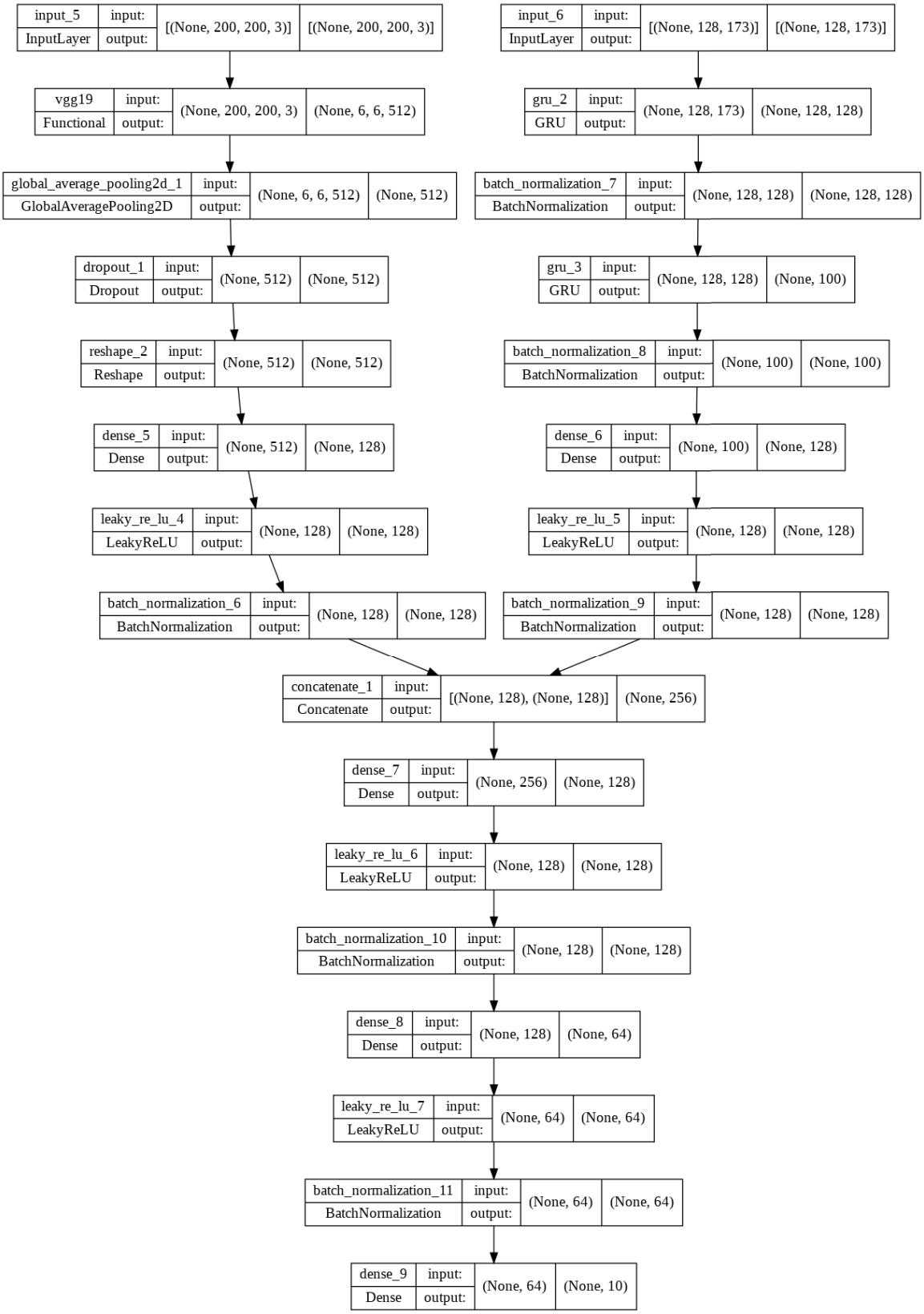


Figure 2-17 The proposed architecture of the CNN-RNN model for classification.

2.4. Results

In this section, the overall performance of the proposed CNN-RNN architecture and the DCGAN data augmentation technique is explained. In order to evaluate the performance of the augmentation techniques used in this thesis, two scenarios are considered including the evaluation of the proposed CNN-RNN model using the original training data and the performance of the model using the augmented data added to the original data. Table 2-1 illustrates the classification accuracy in both scenarios. The accuracy of the model using the original data is 93.3% while using the augmented data adding to the original data could improve the accuracy of the model by 4.7%. This result indicates that using augmentation techniques has a significant impact on the accuracy of the model, concluding the fact that adding more data to the model would improve the performance of the overall model. Moreover, another implicit conclusion from the presented result is that the generated images using DCGAN have similar features to the original data and adding them to the original data would lead to improving the accuracy of the proposed model.

To corroborate these findings, the overall accuracy and loss of the CNN-RNN model for training and validation datasets are presented in this thesis. The overall accuracy and loss of the CNN-RNN model for training and validation datasets are presented in Figure 2-18. Based on the results shown in Figure 2-18, the model loss and accuracy between the training and validation process are quite consistent. Although the accuracy and loss overfit slightly in the beginning, after epoch 40, as the training epochs go through, the training and validation loss are become closer, resulting in resolving the overfitting problem in the beginning. The fluctuations that happened in the accuracy and loss plots in Figure 2-18 are due to the batch ADAM optimizer which means that the training and validation data are batched and then feed to the model.

Table 2-2 illustrates a comprehensive comparison of the proposed CNN-RNN model in this thesis and other state-of-the-art deep learning models presented in the literature including CNN, RNN, AlexNet, and GoogleNet on the same dataset. This table shows that the proposed CNN-RNN model is able to surpass other state-of-the-art models.

Table 2-1 Results of comparing the classifier's accuracy when using original images or original and generated images.

	Classification Accuracy (%)
Original	93.3
Original + Generated	98.0

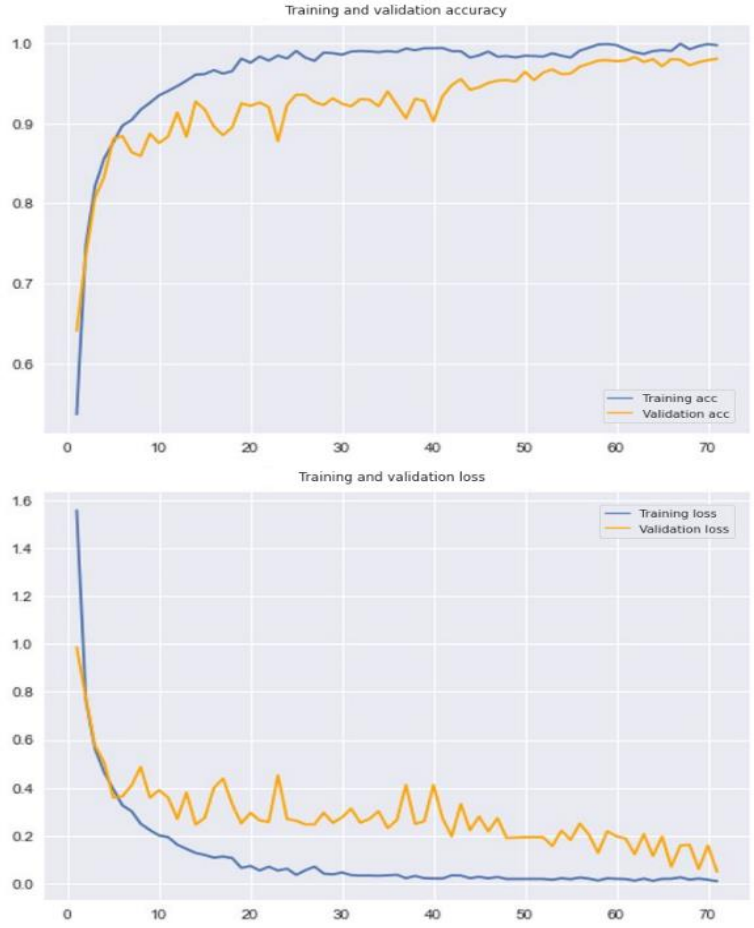


Figure 2-18 overall accuracy and loss

Table 2-2 Previous state-of-the-art ESC models vs Proposed model

Framework	Classification Accuracy (%)	Ref.
PiczakCNN	73.7	[18]
AlexNet	92	[92]
GoogleNet	93	[92]
RNN	82.09	[93]
MC-Net + LMC	95	[94]
WCCGAN	94	[95]
The proposed model	98	

LMC: Log-Mel spectrogram + Spectral Contrast
WCCGAN: Weighted Cycle-Consistent Generative Adversarial Network

2.5. Conclusion

In this chapter, a unified CNN-RNN structure is proposed to address the environmental sound classification task. Several deep learning techniques including transfer learning and data augmentation are used to improve the model performance. For data augmentation, a generative model using DCGAN is used to address the lack of data problem for environmental sound classification. This data augmentation method can produce spectrograms with similar structures to the training set. Applying a CNN-RNN algorithm on a mix of the real dataset and generated images show that the DCGAN method has the ability to improve the performance of the environmental sound classification task. Experimental results on UrbanSound8K datasets demonstrate that the proposed approach achieves superior performance to the state-of-the-art methods.

Chapter 3. Sound Filtration

3.1. Introduction

People are living in noisy environments nowadays. In a noisy environment, it is not easy to focus, have a conversation or relax. Environmental noise, also known as noise pollution or sound pollution, is the propagation of noise with ranging impacts on human activity. Most of the environmental noises are considered harmful to a degree. The source of environmental noise is mainly caused by human activities, machines, transport, and so on. The environmental noises can be categorized into different groups including continuity (continuous and discrete), periodicity (periodic and aperiodic), probability (deterministic and random), and stationarity (stationary and non-stationary) [96].

In the latest category, the stationary category, the main distinction between stationary and non-stationary sound is that the features of a stationary sound, such as frequency and spectral content, do not vary over time, whereas the features of a non-stationary sound change with time. Examples of stationary sounds are engine, white noise, and air conditioning while speech, traffic noises, and washing machine noises are examples of non-stationary sounds. Most of the sounds in the environment are non-stationary. Among non-stationary sounds, some of them are more complex and slightly different from others, such as speech, dog barking, and sirens. These sounds are highly dynamic since they have multiple frequency components in a short time interval (less than 30 ms), which makes them difficult to identify, understand, and analyze by computers.

Several techniques are introduced to attenuate stationary and non-stationary noises in the real world. The main goal of these techniques is to take the audio signal from a microphone, which contains a mix of noise sound and desired sound, clean the mixed sound to get rid of the noise and send the cleaned sound to a speaker. The attenuating process traditionally is addressed using statistical signal processing tools [62]–[64]. However, they are not able to reduce non-stationary and dynamic noises due to their complex nature, and thus, the clean signal is partly corrupted due to the existence of the non-stationary sounds [97]. The state-of-the-art works in noise

attenuation and noise suppression are employed the unique characteristics and capabilities of the machine learning and deep learning techniques to address the problem with conventional signal processing approaches [67], [72], [98] . Figure 3-1 shows the overall approach using deep learning-based techniques for noise reduction. In this figure, the noisy signal which contains a mixture of desired sound and noise is considered as the input sound. Then, features of the mixed sound are used to prepare the input data to be fed to the deep neural network. The deep neural network results in a clean signal, which only contains the desired sound.

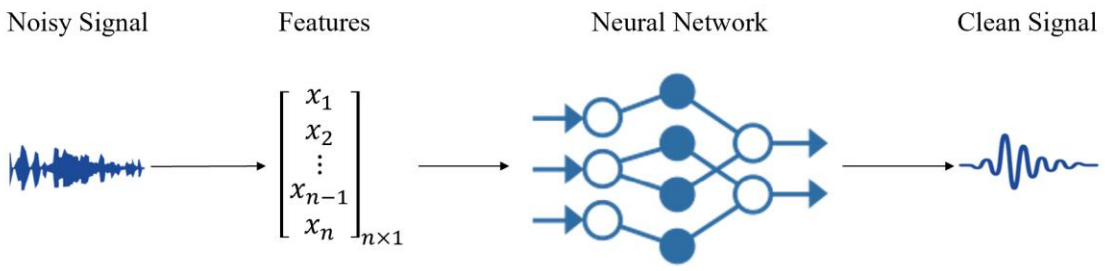


Figure 3-1 A schematic of deep learning-based techniques for noise reduction

Although deep learning methods have shown impressive performance to remove stationary and non-stationary noises [71], [72], [97], compared to the traditional methods, there is still room to improve their performance and the quality of the resulted clean sound. Moreover, in terms of non-stationary sounds, due to their complexity and dynamic nature, there are few publications and efforts that successfully investigate the problem of noise attenuation with non-stationary sounds. In this chapter of the thesis, the effort is to well address this problem and investigate the intelligence approach considering the limitation in terms of quality and intelligibility of clean signal and real-time processing. The successfully applied strategies are evaluated to show the performance and the quality of the proposed techniques to address the limitations and improve the accuracy and the performance of the noise suppression strategies compared to previous works.

In this chapter, a deep-learning-based architecture is proposed to address the environmental noise attenuation task. The proposed method employs signal processing and deep learning techniques to achieve superior performance in both noise reduction levels and speech quality levels. The main goal of this section is to address the existing

challenges in the area of noise reduction and speech enhancement which are mentioned as follows.

- Noise reduction in complex noisy situations: The existing techniques for speech enhancement need to be improved in order to be effective on non-stationary noises with a complex and highly dynamic structure such as dog barking, siren, car horn, and a baby crying.

- Speech quality: Existing techniques for speech enhancement in noisy environments corrupt the speech signal and decrease its intelligibility. In this thesis, signal processing tools are used along with powerful neural network models to address this problem.

In the following, the proposed method, experimental settings, and obtained results are presented and discussed.

3.2. Experimental Methodology

In this section, the procedure of the speech enhancement algorithm based on the DNN network is explained. Figure 3-2 shows the overall block diagram for the entire system, including the pre-processing, the model structure, and the post-processing. According to Figure 3-2, a raw noise signal is added to a clean speech to construct the input to the algorithm. In this thesis, mixed-signal is considered as the combination of a noise signal and a clean speech. Then, the input signal passed through the pre-processing box, proposed DNN architecture, and post-processing box in order to produce a clean signal as the output. The pre-processing, the DNN architecture, and the post-processing steps are described in detail in the following sections.

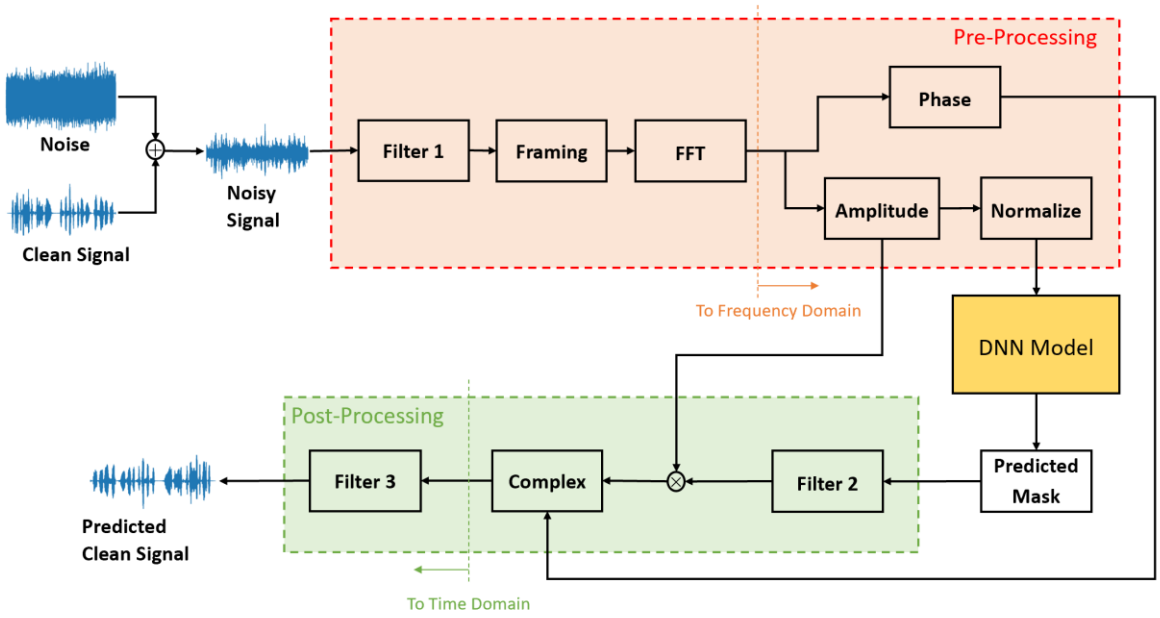


Figure 3-2 The proposed speech denoising architecture

3.2.1. Pre-processing

Before the explanation of pre-processing box, the generated mixed signal is introduced. Figure 3-3 shows the mixed-signal generation process. The mixed-signal in the input of Figure 3-2 is generated using a combination of a clean speech and a noise signal which are assumed to be uncorrelated and can be formulated as,

$$X(k) = S(k) + \mathrm{G} * N(k) \tag{8}$$

where $X(k)$ is the mixed-signal, $S(k)$ is the clean speech signal, $N(k)$ is the noise signal, G is the noise gain, and k is the discrete-time index. The noise gain creates the various distributions of noise and clean signals in the mixture, resulting in various signal-to-noise ratios (SNRs). These variants are considered to simulate a noisy real-world environment. The aim is to extract the signal $S(k)$ from the mixed-signal $X(k)$.

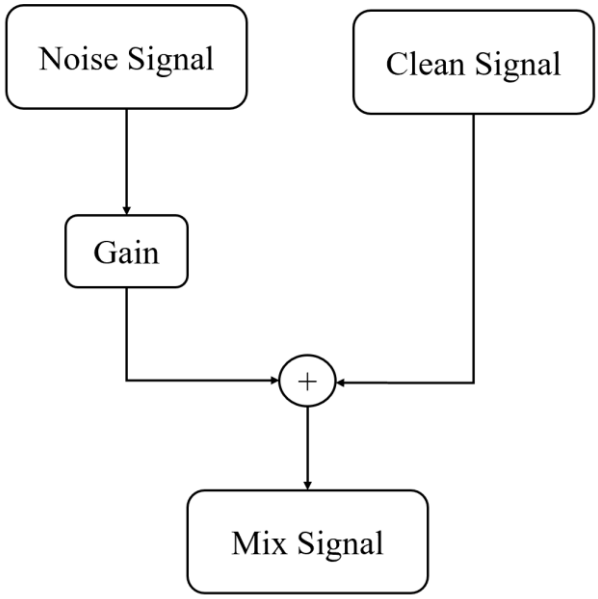


Figure 3-3 The mixing process of noise and clean signal

In Figure 3-2, the pre-processing part is highlighted by a dashed red box filled with light red. The first step of pre-processing part is Filter 1. This is a bandpass filter that is considered a passive attenuation gate for improving the quality or intelligibility of the mixed signal. This filter allows frequency range between 40 Hz and 40 kHz to pass and cuts off other frequencies outside this range. Suppressing the range of noises is considered to be useful since those do not exist in the human voice range [99].

Human ears can tolerate a latency between 20-40 ms [100] and the system needs to be configured to have the latency in this range to work smoothly in real-time. Therefore, the next step is framing, where the filtered mix signal, $\hat{X}(k)$, is divided into short-length frames. A-frame size is chosen such that the human auditory system cannot understand the associated delay in processing, decision making, and aversive sound suppression for each sound segment or frame. Then a Hanning window with 50% overlap is used to keep the information at the edge of the frames.

In the signal processing context, the well-known Fast Fourier Transform (FFT) is commonly used to transfer a signal from the time domain into the frequency domain. The FFT provides amplitude and phase representation of a signal in the frequency domain. The amplitude is encoded as the magnitude of the complex number while the

phase is encoded as the angle. The following equations represent the relationship between the real and imaginary parts and with the amplitude and phase.

$$Real\ Part = Magnitude\ \times\ Cos(Phase) \tag{9}$$

$$Imaginary\ Part = Magnitude\ \times\ Sin(Phase) \tag{10}$$

Figure 3-4 shows the relationship between amplitude and phase, with the imaginary and real parts of the complex number. Moreover, the concept of Inverse FFT (IFFT) is used in order to construct a time-domain signal from the imaginary and real part of the frequency domain representation of a signal. Therefore, given the magnitude and the phase of a signal in the frequency domain, the time domain signal can be calculated using the IFFT.

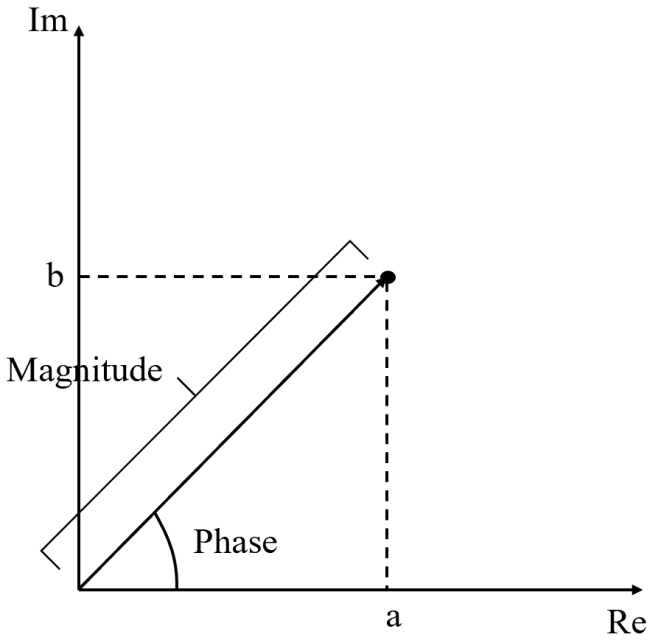


Figure 3-4 The relationship between amplitude and phase of a signal in terms of a = real (Re) and b = imaginary (Im) parts.

In this thesis, the Short Time Fourier Transform (STFT) which is a form of FFT with smaller time frames is used to convert the framed signal into the frequency domain to obtain audio features. STFT is the most widely used technique which is able to represent features of both stationary and non-stationary sounds [101]. Figure 3-5 shows

the STFT of three noises including dog barking, siren, and engine. As shown in this figure, the sounds that we are dealing with have various range of frequencies.

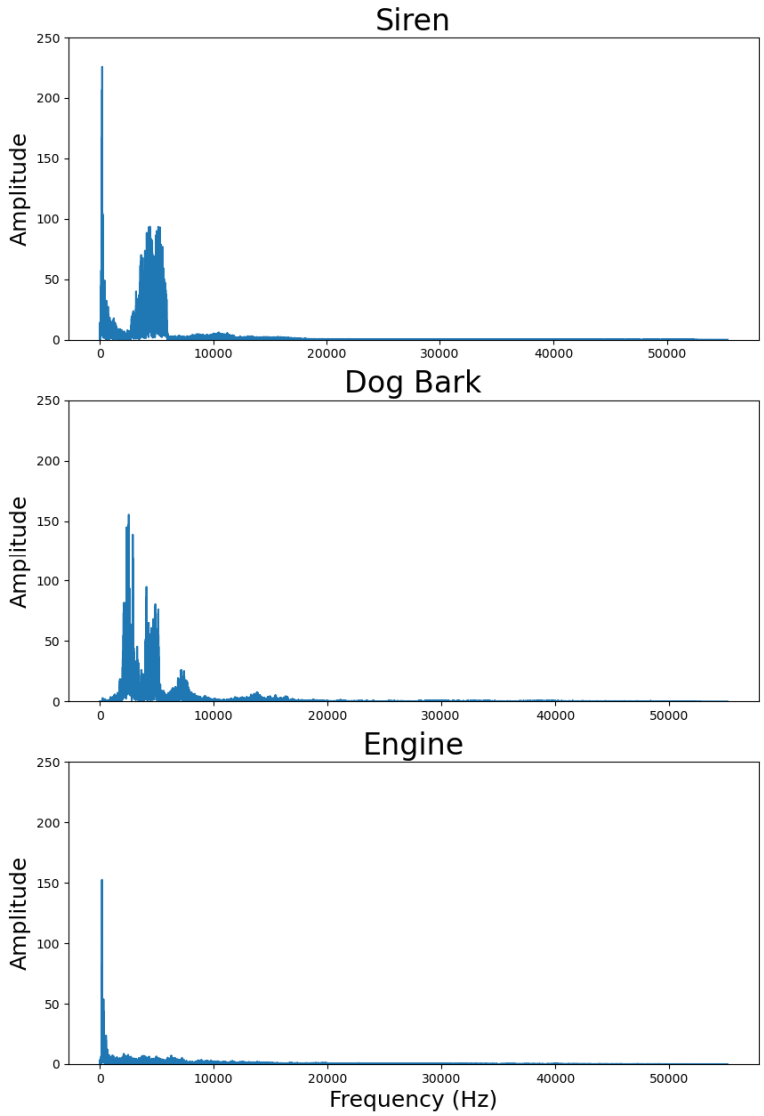


Figure 3-5 the FFT of three noises with different structures

The STFT produces a complex matrix spectrogram that is linearly scaled and factored into a real-valued phase term and a complex-valued amplitude term [102].

$$\hat{X}_{FFT}(n,m) = \sum_{m=0}^{N-1} x(k+nN)w(k)e^{-j2\pi mk/N} \tag{11}$$

where n, m, N, and w represent the frame index, the discrete-frequency index, the frame length, and the analysis window function, respectively. The polar form of STFT of the mixed signal is represented by,

$$\hat{X}_{FFT}(n,m) = \left|\hat{X}_{FFT}(n,m)\right| e^{j\angle \hat{X}_{FFT}(n,m)} = F(n,m) + jA(n,m) \quad (12)$$

where $F(n,m)$ are the amplitude and $A(n,m)$ is the phase.

To prepare the features of mixed-signal and the target clean speech as the input and the output of the DNN model, the logarithm of the amplitudes $A(n,m)$ is calculated. $A(n,m)$ represents the Log Power Spectrum (LPS) features of the mixed-signal and can be formulated by,

$$LPS = \log\left(\frac{2}{N} A(n,m)\right) \quad (13)$$

where N is the length of the signal frame. The LPS vector is normalized between 0 and 1 to make the training process faster and they are fed to the DNN model as the input.

3.2.2. DNN Model

The DNN is introduced by an orange box in Figure 3-2. The DNN's input is the normalized and flattened LPS vector, which is the output of the pre-processing box. The DNN model is constructed such that its output is a predicted mask vector. The predicted mask is measured to extract the speech components in a time domain. In order to construct the output for the DNN architecture and the supervised noise attenuation task, the mask vectors are computed by dividing the squared absolute values of clean speech signal magnitudes by the squared sum of mixed-signal and clean signal magnitudes, using the following equation,

$$M(k) = \left(\frac{S^2(n,m)}{S^2(n,m) + X^2(n,m)}\right)^{0.5} \quad (14)$$

where the $S^2(n,m)$ and $X^2(n,m)$ are the energy of speech and mixed signals at the n^{th} frame and m^{th} frequency bin, respectively [65].

The proposed DNN speech enhancement model architecture is illustrated in Figure 3-6. The model is a fully connected neural network including seven dense layers and two GRU layers. The first two layers are GRUs to extract information from sequential data and to preserve a memory from previous layers, as an inherent capability of the GRUs. Each GRU is then followed by a batch normalization to speed up the training process. In the proposed architecture, the last GRU layer is then followed by three double branches dense layers. The two branches include a branch consisting of a dense layer with an activation function, to extract features related to the nonlinear part of the feature map, and another branch consisting of a dense layer without an activation function, to obtain features related to the linear part of the feature map. To illustrate, the linear branch maintains information from the previous layer to the next layer. Therefore, using a linear and nonlinear activated branched avoids missing information, which is an inherent effect of using nonlinear activation functions.

As is shown in Figure 3-6, in the right direction, the inputs, which are the GRUs' output features, are passed through a dense layer with the Leaky ReLU as a nonlinear activation function followed by batch normalization. In the same manner, at the left direction, the inputs are passed through a dense layer without an activation function followed by a batch normalization. The linear and nonlinear parts are then added together to create the final feature map in each step. This strategy is repeated three times over the rest of the DNN model to prepare the final feature map. Finally, in order to prepare the final output and the predicted masks, a dense layer is used to map the final features to the same dimension as the constructed mask vectors.

The Mean Square Error (MSE) loss function, which is well-known for regression problems [103], is used to compute the squared error between the predictions and the real mask vectors. MSE is defined as,

$$MSE(n) = E[(\hat{S}(n) - S(n))^2] \tag{15}$$

where $S(n)$ and $\hat{S}(n)$ denote the real mask vector and its prediction in the n-th frame. Also, ADAM optimizer is used to solve the optimization algorithm and to update DNN architecture's weights in GRUs and dense layers.

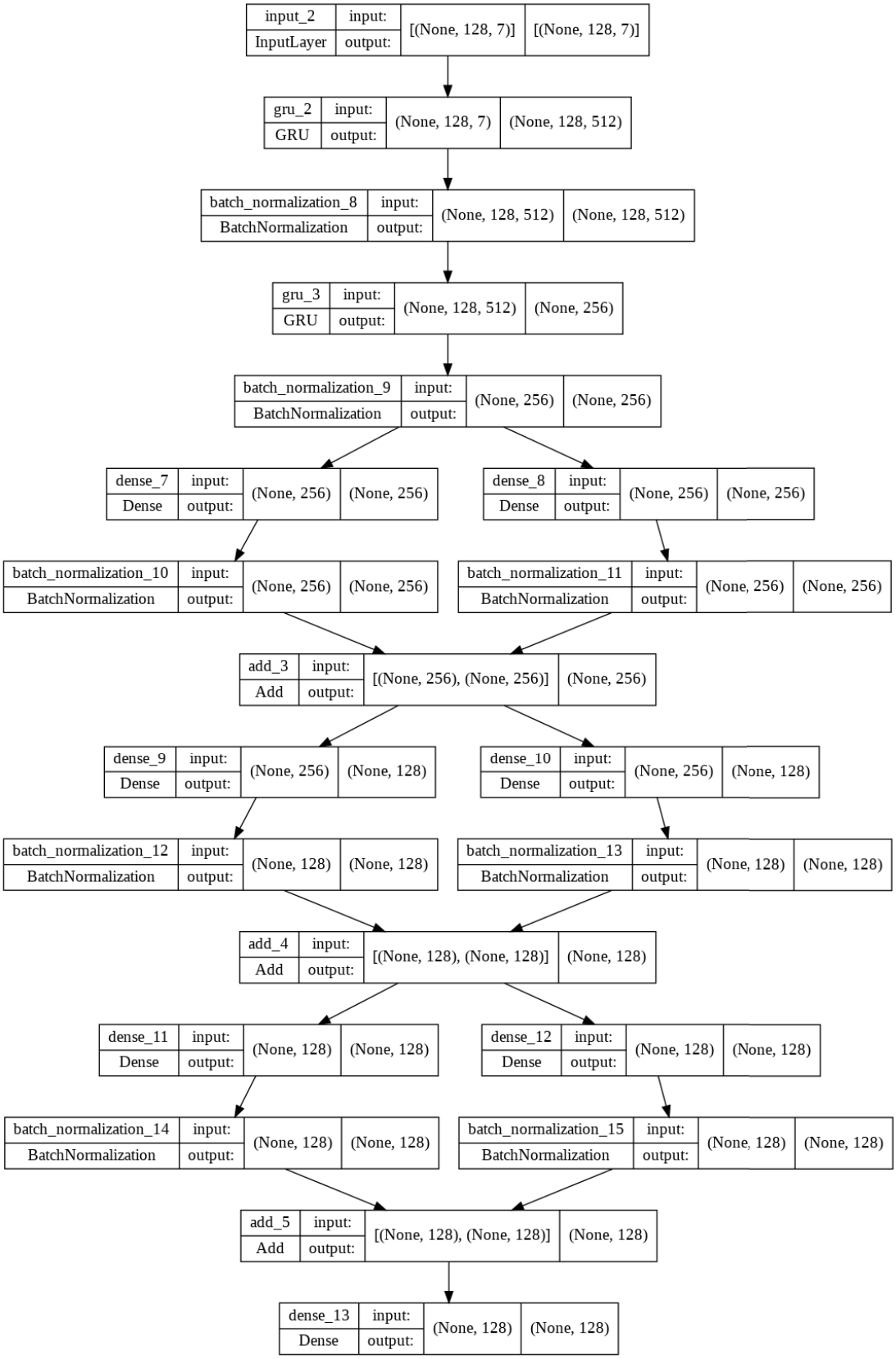


Figure 3-6 The DNN model structure for filtration

3.2.3. Post-processing

In Figure 3-2, the post-processing part is highlighted by a dashed green box. Post-processing steps include the predicted mask modification, reconstruction, and tuning of the clean speech signal in the time domain. Instead of using the raw output of the DNN model, the mask vector is passed through filter 2, a Gaussian smooth function that is used for denoising the predicted mask vectors [104]. The level of smoothing is controlled by the standard deviation of the Gaussian number. The smoothed masked then is multiplied by the amplitude of the mixed signal to create the clean speech signal amplitudes, computed by,

$$A_C = M \times A_M \tag{16}$$

where A_C, M and A_M denote the predicted amplitude of clean signal, mask, and amplitude of mix signal.

Up to this point, the amplitude of the clean speech signal is computed in the frequency domain. The phase of the clean speech and the mixed speech are considered the same and therefore, the phase of the clean speech can be extracted from the mixed-signal and stored in order to construct the clean speech. To transform the clean signal to the time domain, the IFFT is used using the mixed-signal phase and computed amplitude from the post-processing step. During the IFFT process, the speech signal is reconstructed by vectorizing the complex part of the computed real and the imaginary values of the signal. This step is completed in the complex box in Figure 3-2. At the end, the predicted clean speech is passed through filter 3, which is a bandpass filter for a final tune to improve the quality of the predicted signal.

3.2.4. Data Preparation for Noise Filtration Task

The performance of the proposed architecture is examined using the Librispeech [105] and ESC-50 [106] datasets. The LibriSpeech dataset is a collection of approximately 1,000 hours of human speech and ESC-50 has 50 classes with 40 samples in each class. 200 utterances from various English speakers (female and male) are randomly selected from the first dataset as the clean speech for training and testing purposes. For the noise dataset, the ESC-50 dataset, which includes stationary noises

(e.g. engine), non-stationary noises (e.g. vacuum cleaner), and highly dynamic noises (e.g. siren sounds) is employed. The reason for choosing ESC-50 as the noise signal dataset for the noise attenuation task is that this dataset includes most of the challenging sound for autistic individuals. The noise samples are selected such that they include a clear noise over the entire audio sample. Clean speech and noise signals are consisting of a vector representing the values over time and they are down-sampled to 8kHz to adjust the dataset size for training and also speed up the real-time processing practice. The selected noise and clean speech signals are added together with three different SNRs. This way, the architecture can be trained and learned from different noisy environments in terms of the power and intensity of the noise signal. The mixed signals are then framed and used as the input to the DNN model. In this thesis, 20% of the created dataset is selected for validation. The model is trained for 100 epochs with an MSE function as a loss function using a batch size of 128. To reduce overfitting and improve generalization error, the dataset is shuffled and then fed to the model. Moreover, 20 unseen mixed signals are specified as the test set to evaluate the performance and assess the capability of the trained model.

3.3. Simulation Results

In this section, the simulation results of the proposed DNN architecture are presented. First, the visual representation of the predicted clean speech and the predicted masks are illustrated in order to visually see the performance and the effectiveness of the proposed model. Afterward, a comprehensive study is performed to evaluate the performance and the accuracy of the proposed DNN in this thesis and to compare it with other state-of-the-art methods available in the literature.

To visually observe the experimental results, the spectrograms of the clean speech signal, the noise signal, the mixture of the speech signal with the noise at SNR of 0 dB, and the enhanced speech predicted by the proposed method in this thesis, are presented in Figure 3-7. According to Figure 3-7, it is observed that the proposed method effectively suppresses the background noise, and the predicted speech signal is close to the clean version. In highly dynamic noise structures such as dog barking and siren, eliminating the noise signal without any corruption to the speech signal is complicated and difficult. Figure 3-7 shows that the proposed method in this thesis is

capable of suppressing the highly dynamic noises and the resulted filtered speech is close to the clean speech spectrogram.

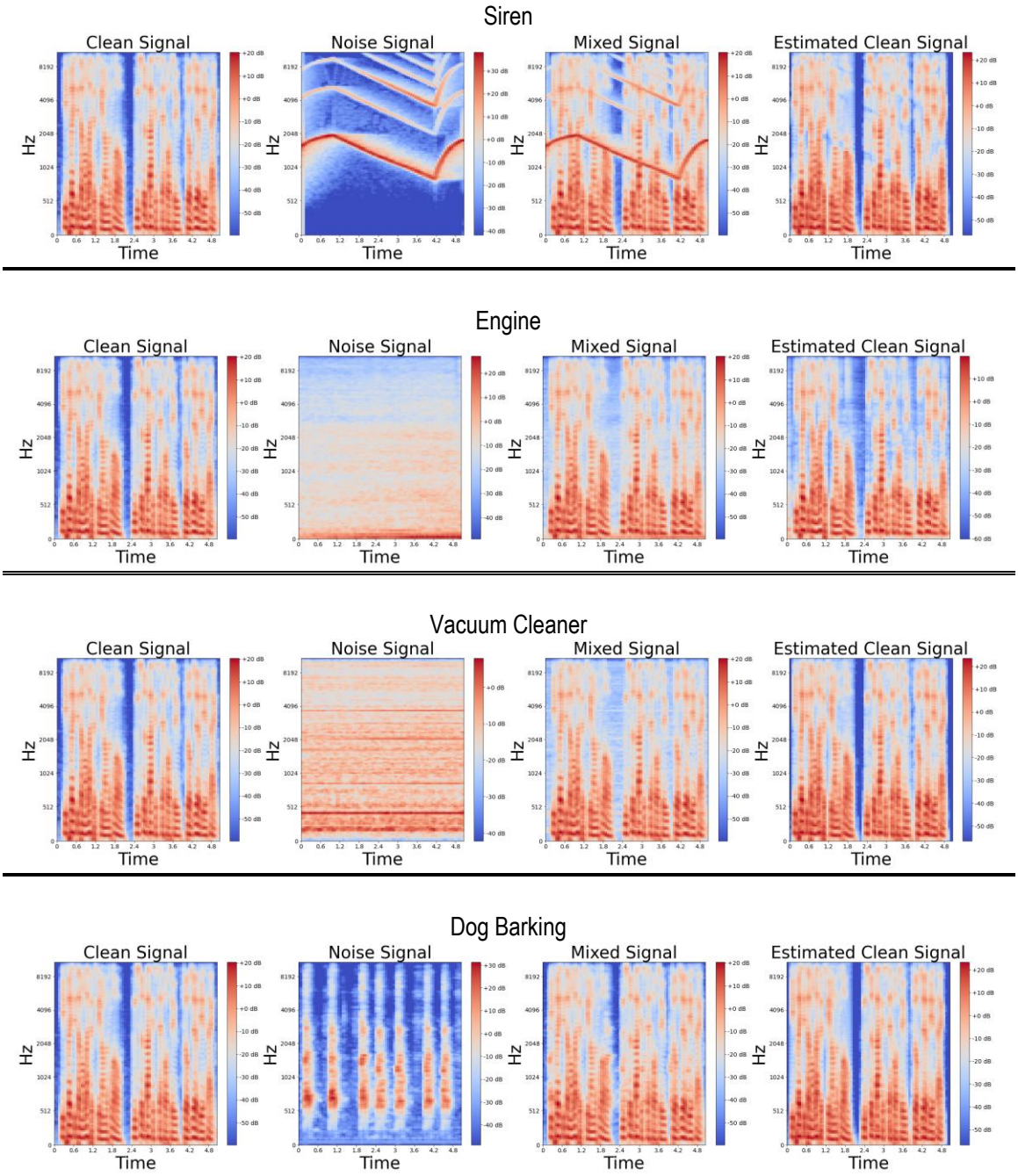


Figure 3-7 Spectrogram performance of the proposed method for different noise structure types

To further discuss the effectiveness and capability of the proposed method, the estimated ratio mask vectors, which is the output of the DNN model, are compared with the computed ratio mask in equation 14, which are considered the real ratio mask vectors. Figure 3-8 shows four randomly selected samples of the estimated ratio mask vectors compared to the real ones computed directly from the clean reference and the mixed signals. The simulation results show that the predicted and real masks are very close to each other, which indicates the accuracy and performance of the proposed DNN model. According to Figure 3-8, the DNN model can predict the fluctuations in the ratio masks to generate a clean speech at the end of the process.

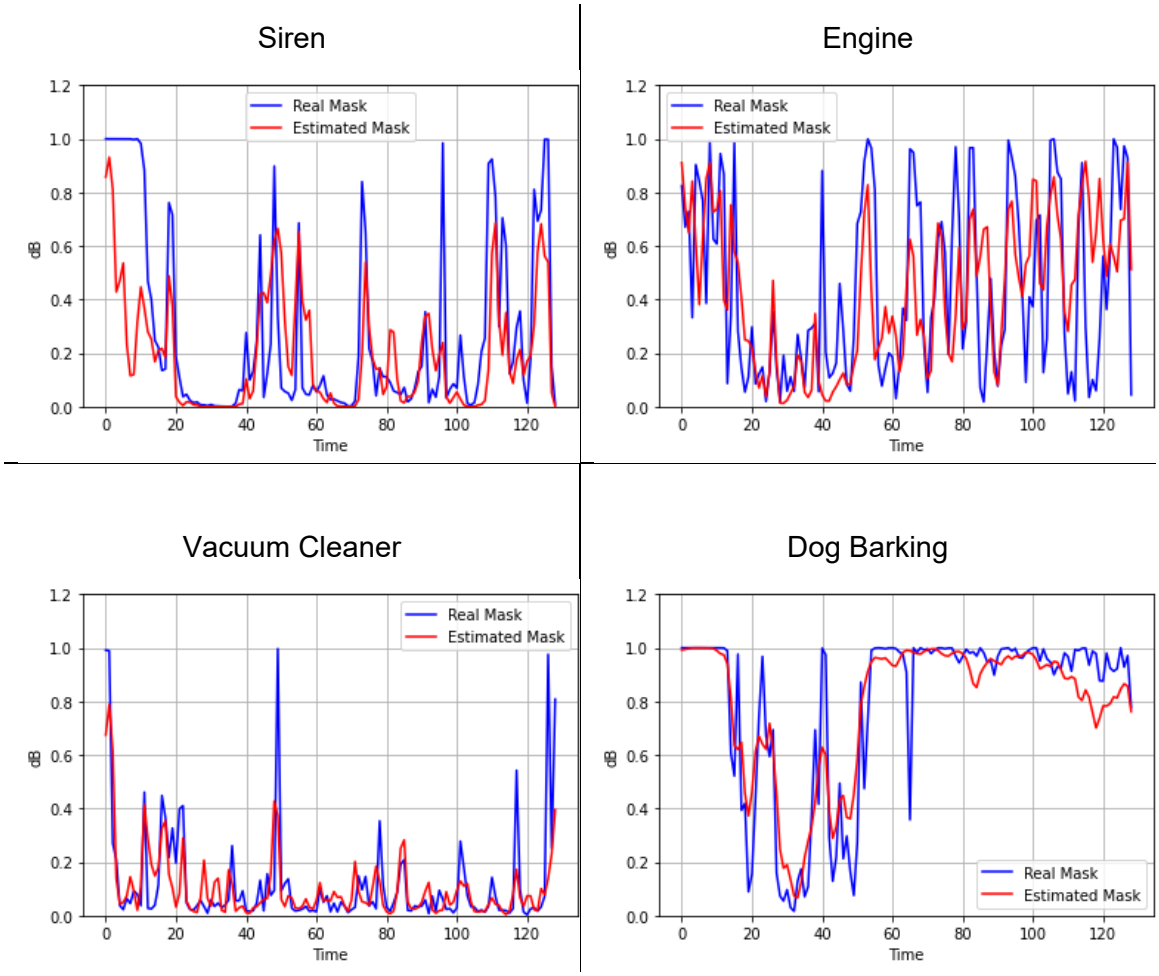


Figure 3-8 A comparison between the estimated masks by the proposed method and the real masks

To evaluate the performance of the proposed algorithm, four standard and most commonly used speech quality measurements are used.

- Perceptual Evaluation Speech Quality (PESQ): this metric is used to evaluate speech quality, calculated by comparing the predicted and clean speech signals. PESQ is calculated as a linear combination of average disturbance values and average asymmetrical disturbance values between a reference signal and an estimated signal. Although it assesses the noise speech quality, some features such as loudness, loss, delay, and echo are not expressed in the PESQ score [107]. In the PESQ test, the representative values are from -0.5 to 4.5, demonstrating the minimum and maximum speech quality.

- Short-Time Objective Intelligibility (STOI): reflects the improvement in speech intelligibility and has a strong correlation to subjective listening test scores [108]. STOI is based on an intermediate intelligibility measure for the short-time time-frequency domain and uses a simple Digital Fourier Transform (DFT)-based time-frequency decomposition [108]. STOI score ranges from 0 to 1 where a better speech intelligibility receives a higher STOI score.

- Segmental Signal to Noise Ratio (Seg SNR): evaluates the overall speech quality and the performance of noise reduction. Seg SNR computes the segmental signal-to-noise ratio (SNR) in dB by comparing a noisy signal with a clean reference signal. The Seg SNR for a speech signal is formulated as follows,

$$Seg\ SNR = mean(10log_{10}^{(\frac{\sum_{n=1}^{N} x^2(n)}{\sum_{n=1}^{N}(x(n)-\hat{x}(n))^2})}) \quad (17)$$

where N is the length of the signal frame, x(n) is the clean speech signal and $\hat{x}(n)$ is the enhanced or predicted speech signal [109].

- The Log-Likelihood Ratio (LLR): represents the quality level of speech signals by measuring the difference between the linear prediction of the clean reference signal and the degraded signal [110]. LLR score ranges from 0 to 2 where higher LLR scores mean better speech qualities.

Table 3-1 illustrates the performance of the proposed DNN architecture for the noise attenuation task. The noise signals are selected such that they include various noise types including, highly dynamic noises (siren), non-stationary noises (vacuum cleaner), and stationary noises (engine). Dog barking is also considered the fourth noise

sample to show the algorithm's performance in a difficult situation where the noise has structural similarity to the speech. This resemblance is notified from the spectrograms representing the clean speech and the dog bark in Figure 3-7. The clean speech and noise signals are mixed with four different SNRs, including, -3 dB, 0 dB, 3 dB, and 10 dB, to comprise various noisy conditions. Table 3-1 presents the experimental results of the average performance based on PESQ, STOI, Seg SNR, and LLR scores on the selected test samples. For all these measurements, higher scores indicate better performances. According to Table 3-1, the simulation results show that the proposed method provides high speech intelligibility with significant speech quality improvements. The results indicate that the enhanced speech quality is the highest at 10dB and is degraded as it gets to -3dB. This is due to the noise level amplification, which illustrates the high power and intensity of the noise. Moreover, the results demonstrate that the proposed framework has significantly enhanced speech quality even at higher noise levels of -3dB and 0dB.

To compare the simulation results with other existing methods, the Wiener filtering method [111] and an LSTM network [112] are simulated and the results are provided in the form of bar charts in Figure 3-9. According to this figure, the proposed method in this thesis has enhanced speech quality and intelligibility compared to those two methods in [111] and [112] which represents the effectiveness and reliability of the proposed method in suppressing different noises. In the stationary and non-stationary noises such as engine and vacuum cleaner, the simulation results show that the LSTM networks perform close to our method. Figure 3-9 shows that the Wiener filtering method and LSTM network do not have satisfactory performances in eliminating the complex noisy structures such as dog barking and siren. On the other hand, the proposed method in this thesis achieved a higher level of noise suppression for those complex conditions. According to Seg SNR and STOI scores, which are representing the noise reduction level and speech quality, the method here significantly improves the denoised speech signal and accomplishes a higher level of noise reduction compared to the other two methods in [111] and [112].

Table 3-1 The performance measurements of the proposed method for different noise structure types

	SNRs	Test Results			
		PESQ	STOI	Seg SNR	LLR
Siren	-3 dB	1.6162	0.516	1.2519	1.2749
	0 dB	2.3564	0.6504	3.1065	1.5719
	3 dB	2.8372	0.668	3.7530	1.6983
	10 dB	3.2567	0.698	3.9237	1.8601
Engine	-3 dB	1.5196	0.6026	2.5174	1.3725
	0 dB	2.4095	0.7788	4.2853	1.6011
	3 dB	2.6525	0.7848	4.7296	1.6791
	10 dB	3.1156	0.8163	4.8516	1.8164
Vacuum Cleaner	-3 dB	1.4998	0.5576	1.9869	1.4671
	0 dB	2.5080	0.6804	3.2365	1.5697
	3 dB	2.7352	0.6941	4.0709	1.5738
	10 dB	3.3297	0.7116	4.8583	1.5764
Dog Barking	-3 dB	1.8148	0.5644	1.4928	1.3941
	0 dB	2.4782	0.6844	3.8941	1.4813
	3 dB	2.7153	0.6891	4.1668	1.5095
	10 dB	3.1299	0.7164	4.8946	1.5818

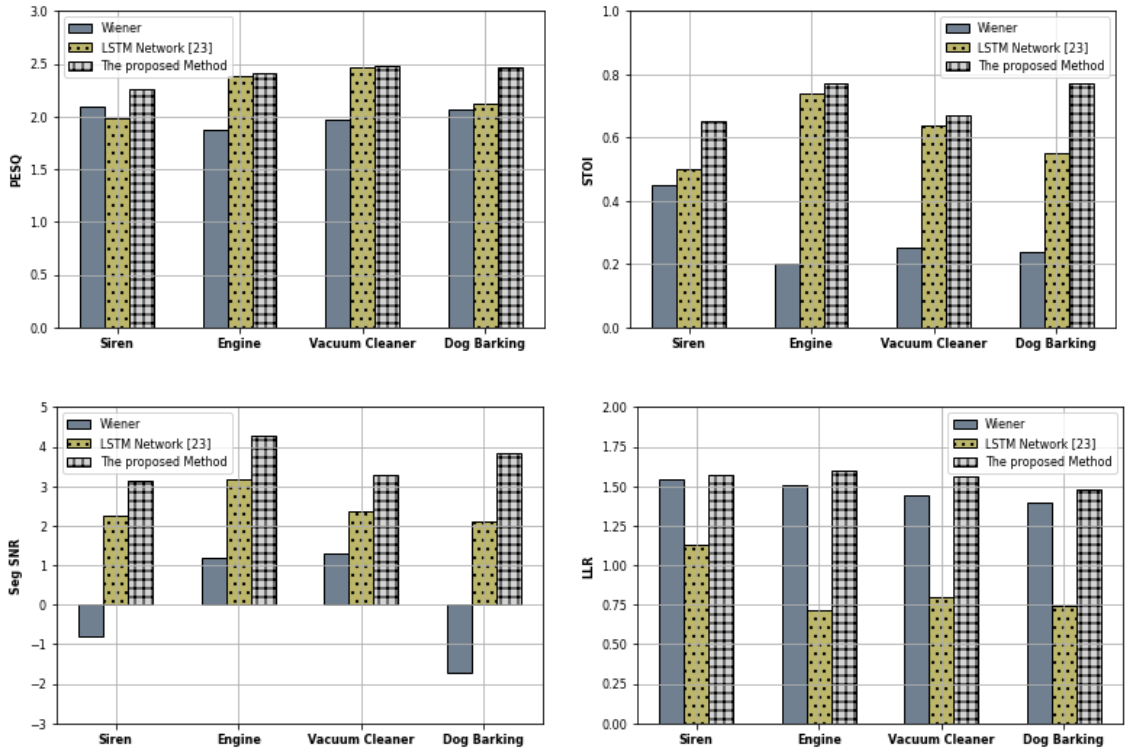


Figure 3-9 Performance comparison of the proposed algorithms with other algorithms in terms of PESQ, STOI, Seg SNR and LLR scores.

3.4. Conclusion

This chapter proposes an innovative solution to address the speech enhancement problem using a DNN-based learning framework for real-time applications. The proposed framework receives a noisy speech signal as an input and extracts its features to feed to a DNN model. The proposed DNN model is designed and trained to generate a ratio mask to eliminate the background noise from the noisy speech signal. Signal processing techniques are utilized to increase speech quality and integrability before and after the DNN model. The proposed techniques and DNN model are applied to the Librispeech dataset for the clean speech sources and ESC-50 datasets for the noise sources. The simulation results indicate that the proposed method can remove the background noise from the speech signal, even in the presence of complex noisy conditions (e.g., unsteady real-world environments) and difficult noise types (non-stationary and highly dynamic noises). Four speech quality measurements including PESQ, STOI, Seg SNR, and LLR are employed to evaluate the performance of the proposed method in terms of the noise reduction level, speech quality, and intelligibility. Moreover, the spectrograms and the predicted masks are provided to demonstrate the significant performance of the proposed technique under different circumstances from low to high SNRs.

Chapter 4. Integration of Sound Classification and Filtration

4.1. Introduction

The model inference is considered an important step in the production of machine learning and deep learning models. The model inference is generally referred to the process of using a trained model to infer a result from previously unseen data. At first glance, the training and model inference may seem to be similar, as in both cases data is fed through the model. However, in training, a model will process the data in order to update the learnable parameters of the model, reduce the cost function and evaluate the desired outputs to reach an acceptable level of accuracy and loss. In contrast, the model inference process is used to evaluate the ability and the performance of the model against a new set of data. Figure 4-1 shows the overall schematic of an inference structure. In this figure, in the training stage, the training dataset including input and labeled output data pairs is used to build a trained model which meets the modeling criteria such as accuracy and loss. After the model has been successfully trained, in the inference stage, the trained model is utilized to make a prediction based on new unseen data to evaluate the model in real applications.

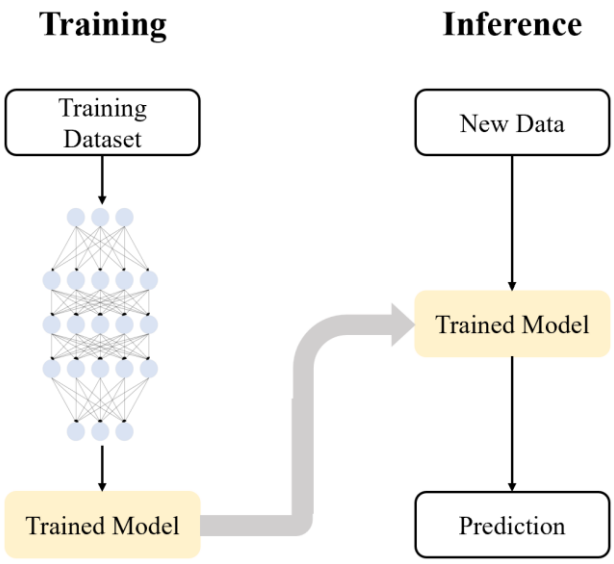


Figure 4-1 The overall schematic of an inference structure

For AI models, there are two types of inferences or model deployments including batch inference and real-time inference.

- The batch inference is an asynchronous procedure that makes predictions based on a set of data for end-users or commercial applications such as text translation, accounting systems, and email tagging. These predictions are not required to be in real-time.

- Real-time inference also known as on-demand predictions requires the model to make predictions immediately. This inference is useful for analyzing data from streaming and interactive applications such as robotic systems, speech recognition, and self-driving cars.

Since human ears are highly sensitive to delay (less than 40ms latency), a real-time inference is required for this application to identify and filter aversive sounds instantly. Therefore, the system needs to work super-fast in its computational and predicational processes. For real-time inference, there are several important metrics to optimize such as latency, throughput, and cost. Latency is the delay between a user's action and the response of the application to the user's action. Throughput is a measure of how many units of information a system can process in a given amount of time. Cost also includes the cost of providing powerful processors, memory, and network. In general, machine learning and deep learning applications aim to minimize latency and cost while maximizing throughput.

Deep learning-based algorithms have achieved remarkable performance in various applications from computer vision to natural language processing. However, deep learning models require excessive computational power due to their number of layers, neurons, and layers' structures. This makes the inference process complex for the deep learning model to be deployed in real-time and also increases the cost to provide such powerful processors.

As it is mentioned early, in this project, the system needs to respond in less than 40 ms, which includes recording data, executing inference, data pre-processing, model prediction, data post-processing, and returning the results to the system or application. Today's mobile devices and CPUs cannot support this computation power in terms of latency and energy consumption. One approach to address the latency and power

requirement for this purpose is the use of accelerators such as Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs). GPUs tend to be optimized for parallel throughput and are often used in training infrastructure. While TPUs are both useful in training and have advantages for large complex models and large batch sizes, especially during inference. There are two ways to use an accelerator, i.e. cloud computing and AI edge computing.

- Cloud computing is the delivery of computing services including servers, storage, databases, networking, and software over the Internet (“the cloud”) to offer faster innovation. Cloud-based system utilizes GPUs that work well in highly parallel applications in a data center.

- AI edge computing refers to the deployment of AI applications on the devices - 'on the edge'. It’s called “AI edge” because the AI processing computation is done near the user at the edge of the network, where the data is located, rather than cloud computing facility or private data center.

In the edge devices such as mobile phones, the average GPU memory size is much smaller than the average GPU size in a data center such as Amazon Web Services (AWS) and Google Cloud. On the other hand, access to a limited GPU available on the device comes at a price and also can lead to a battery draining quickly. Therefore, deploying a deep learning model to a data center and then exposing it through an Application Programming Interface (API) can be a reasonable choice for inference.

In this chapter, the integration of an aversive sound classifier and filter which are described in chapters 2 and 3, is presented. In the following, the framing strategy is explained to show how new samples are organized to make the system works in real-time. Afterward, the integration procedure of the classifier and filter is explained. Then, the processing time and required specifications are presented. In the following, the application representation including the graphical user interface is shown. Afterward, several pilot sessions are conducted with autistic adults to examine the noise attenuation system introduced in this thesis in real-life. Finally, the conclusion is discussed.

4.2. Data Framing

Framing refers to the process of capturing data in short time intervals for further data processing, prediction, and inference. Operating in real-time requires a framing strategy such that the system could perform information processing in real-time using time-series data. In our application, the frequency representation of non-stationary and dynamic sounds such as Fast Fourier Transform (FFT) is not effective for a large-size signal since the spectral features in non-stationary signals change quickly over time. Therefore, data framing is required to split data for the use of FFT for feature extraction.

In this thesis, frame size is chosen very small such that the human cannot perceive the delay associated with processing, decision making, and aversive sound suppression for each sound segment or frame. This very small frame size is not capable of representing information and express a sound signature. both concerns including the delay and representing information can be addressed using a cumulative framing strategy. Figure 4-2 represents the cumulative framing strategy used in this thesis.

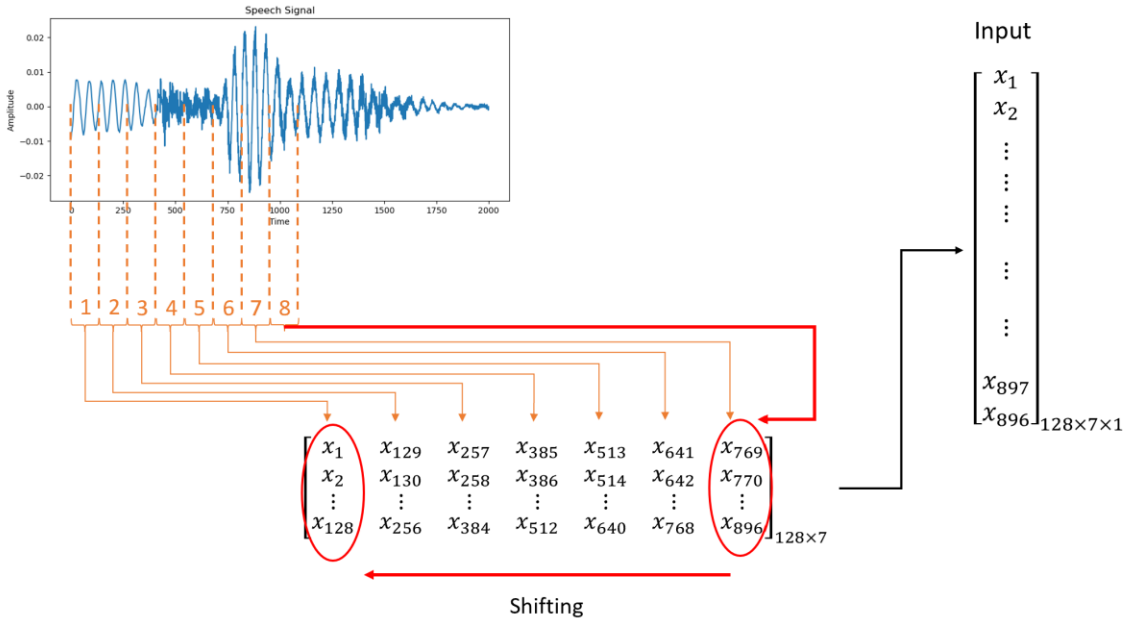


Figure 4-2 The cumulative framing strategy used in this thesis

According to Figure 4-2, a 16 ms frame size is used. Each framed signal consists of 128 data samples with an 8 kHz sampling rate. When a frame is recorded by

a microphone, the data preparation process is applied to the signal frame, in order to extract the required features. Since the extracted features are poor to represent the information contained in the frame, seven recorded frames are considered to be cumulated to represent meaningful information. Those frames are recorded by microphone, and each placed in a column of an input matrix. The input matrix size for the proposed deep neural network for the noise suppression task is 128×7. The created feature matrix is flattened to a vector and is fed to the neural network model. Afterward, a new frame of 128 data samples is recorded by microphone and after the preparation process required for feature extraction, is placed in the last column on the feature matrix. Accordingly, the first column on the feature matrix is removed. The first 6 columns of the new matrix are the same as the last 6 columns of the previous matrix.

For the classification task, the input matrix size is 128×173 which is required to create the three-dimensional feature map. To create this feature map, the signal frame of 128×1 is transferred to a mel-spectrogram with 128 mels which creates the first column of 128×1 in the feature map matrix. 173 frames are added together to create the final first dimension of the feature map. The percussive and harmonic mel spectrograms are also formed to create the final three-dimensional feature as the input to the CNN-RNN model. After each recording, the first column will be removed, and all the columns are shifted one space to the left. The new input vector will be fed to the model and this process is repeated.

In order to efficiently employ the memory usage and the storage space for the system, and accelerate the data processing phase, the recorded data is stored once in the memory and the data preparation for classification and filtration performs in parallel. Also, the shared feature characteristics in classification and filtration are calculated and stored once to be used in both models. This framing cumulative strategy provides the system with an accelerated speed compared to sequential computation and comes to the system with the ability to work in real-time with an unnoticeable delay. Additionally, keeping the data samples from the previous steps, provides the model with meaningful features and information, compared to providing the system only with the new data.

4.3. Model structure and task modification for individuals with ASD

The goal of this thesis is to design a framework as an intervention technique for individuals with ASD to attenuate selected aversive sounds. The classifier introduced in Chapter 2 was originally trained to classify the environmental aversive sounds into 10 different classes including drilling, dog barking, siren, air conditioner, car horn, children playing, engine idling, gunshot, jackhammer, and street music. The classifier was trained on UrbanSound8K. This way, the introduced CNN-RNN model has a unique capability compared to other state-of-the-art methods and machine learning structures in the literature in terms of effectiveness, accuracy, and performance. In this chapter, three aversive sounds including siren, dog barking, and drilling are selected for the inference and integration of the classification and filtration task. These are the three top aversive sounds among autistic individuals [8]. Therefore, the classifier output is modified to return the probability of the existence of these three aversive sounds and the other outputs are considered non-aversive. Moreover, over one hour of recorded environmental non-aversive sounds is added to the training and evaluation datasets in order to enrich the UrbanSound8K dataset. Therefore, the output of the classifier contains four classes including siren, dog barking, drilling, and non-aversive.

The filtration introduced in Chapter 3 is used to suppress the aversive sound in the mixed-signal received from the environment. The proposed deep neural network model has been trained on the ESC-50 dataset which contains 50 classes of aversive sounds, with 40 samples in each class. Also, 200 utterances from various English speakers (female and male) have been randomly selected from the LibriSpeech dataset as the clean speech to be added to the ESC-50 data set to create the mixed-signal for training and testing purposes.

Based on comprehensive research and study by [8], individuals with ASD want to attenuate selective aversive sounds. For example, an individual may be required to attenuate siren and dog barking even though the other individual may find the dog barking a neutral sound and requires filtering siren and drilling. Therefore, a single filter that works in every situation may not be effective. In general, sound sensitivity in individuals with ASD is a subjective problem. As a result, in order to address the problem of aversive sound attenuation for individuals with ASD, the general application filter

introduced in Chapter 3 is divided into three different sub-filters, which are able to attenuate the aversive sounds including siren, dog barking, and drilling, separately. Three sub-filters are developed using the deep neural network architecture introduced in Chapter 3 using transfer learning and fine-tuned by three different datasets which contain a mixed signal that is constructed using siren, dog barking, and drilling. Therefore, the modified classifier is capable of predicting the probability of each of the three introduced aversive sounds in the environmental sound, and the resulting probability is used as an indicator to activate different filters.

4.4. Integration of classification and filtration

In this section, the modified filtration and classification algorithms are integrated to build the desired framework, which is capable of classifying and filtering the selected aversive sounds, simultaneously. Figure 4-3 shows the overall block diagram of the entire system, integrating the classification and filtration tasks. The first step in this process is to receive the environmental sound through the microphone. The microphone records a small sound frame (16 ms) which consists of 128 data samples using an 8 kHz sampling rate. The cumulative framing strategy is then used to prepare the appropriate data structure and correct data size which is used in the classification and filtration blocks. Afterward, the structured data is fed to the classification and filtration simultaneously. However, the filtration is out of the loop until an aversive sound is identified by the classifier. The input to the classifier is used to create the appropriate feature map for the classification task. The classifier, which contains the CNN-RNN network, determines whether the environmental sound contains siren, dog barking, drilling, or it is a non-aversive sound. If the classifier specifies the environmental sound as a non-aversive sound, the input sound from the microphone is directly sent to the speaker. Otherwise, the classifier tags the environmental sound as one of the aversive sound classes, and returns the probability of each aversive sound. In this scenario, the classifier output is used to activate the filtration stage and choose the appropriate trained deep neural network model, based on the identified class, in order to predict mask vectors for clean signal creation. Based on Figure 4-3, the selected deep neural network model uses the extracted features, from the cumulative framing strategy block, as the input to the model to predict the mask vectors. Mask vectors that contain the amplitudes of the clean signal are then post-processed and using the phase of the original

environment signal, and the IFFT technique, to form the clean signal. The clean signal is then passed through the speaker for the user and the system also notifies the user about the presence of an aversive sound with its class name. This notification occurs through the graphical user interface described in the next section.

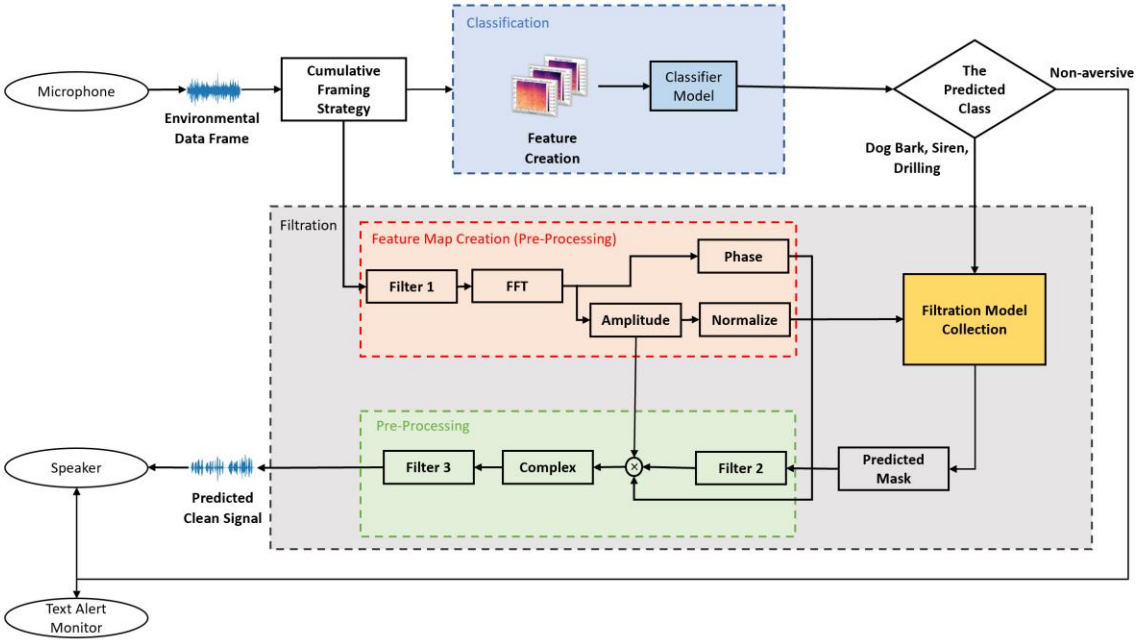


Figure 4-3 The integration of Classifier and Filter

4.5. Graphical User Interface (GUI) Design

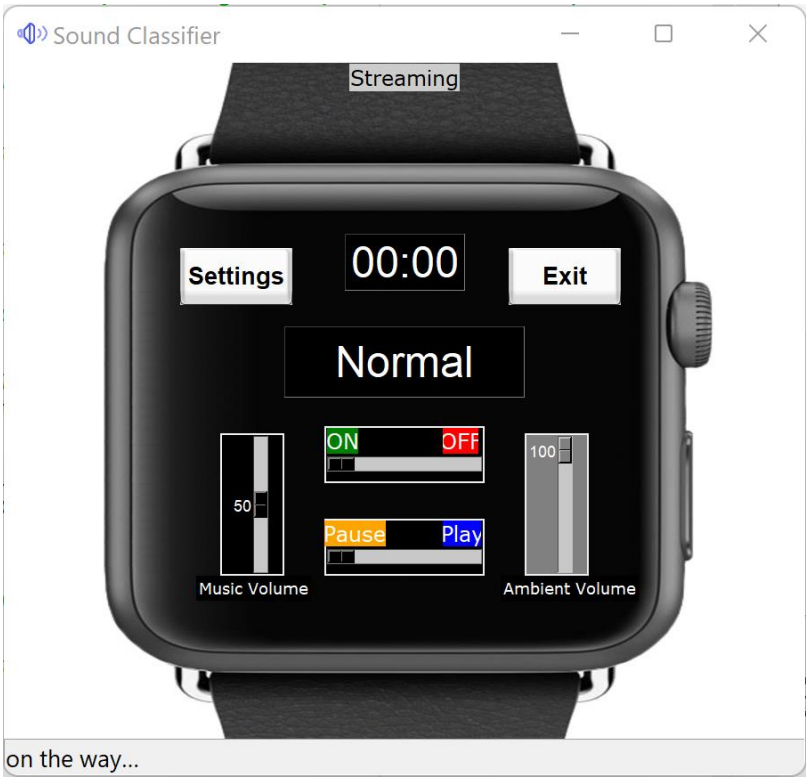
In this section, a graphical user interface (GUI) is designed using the Python Tkinter library to make communication between users and the software. The interface is responsible for all communications, settings, customizations, and user-defined operations. The GUI can be compatible with every operating system including Windows, MacOS, IOS, and Android. The background of the GUI is designed like a smartwatch to show that this software can be implemented into a smartwatch. Figure 4-4 shows the GUI's first and second pages. In Figure 4-4 top view, the status bar “Normal” indicates the system mode. This status bar alerts the user about the presence of the aversive sound and active operation by the system such as suppressing or attenuating the sound. For example, the operation modes can be, “Normal”, meaning that there is no aversive sound identified by the system, and “Aversive”, for when an aversive sound being identified by the system. In the Normal mode, the filtration algorithm is not activated, so

the speakers play intact environmental sounds. Users are able to turn the volume of upcoming environmental sounds on the speakers up and down using the ambient volume slider.

An activation slider is configured to manually turn on and off the filtration algorithm. This slider is configured for situations where the system misidentifies a sound as an aversive sound and provides the user with the ability to inactivate the filtration. “ON” mode is shown in green color and “OFF” mode is shown in red color. As another optional feature of this software, the GUI provides playing music. In case the system fails to identify an aversive sound in the environment, the user can play music, white noise, or any preferred sounds to mask the aversive sound. Users can also control the volume of the music using the music volume slider.

The Settings button is configured for users’ customizations such as choosing their preferred music and selecting the aversive sounds (see below). The exit buttons will close the entire program. The software also presents the time in hours and minutes using the clock status bar.

A



B

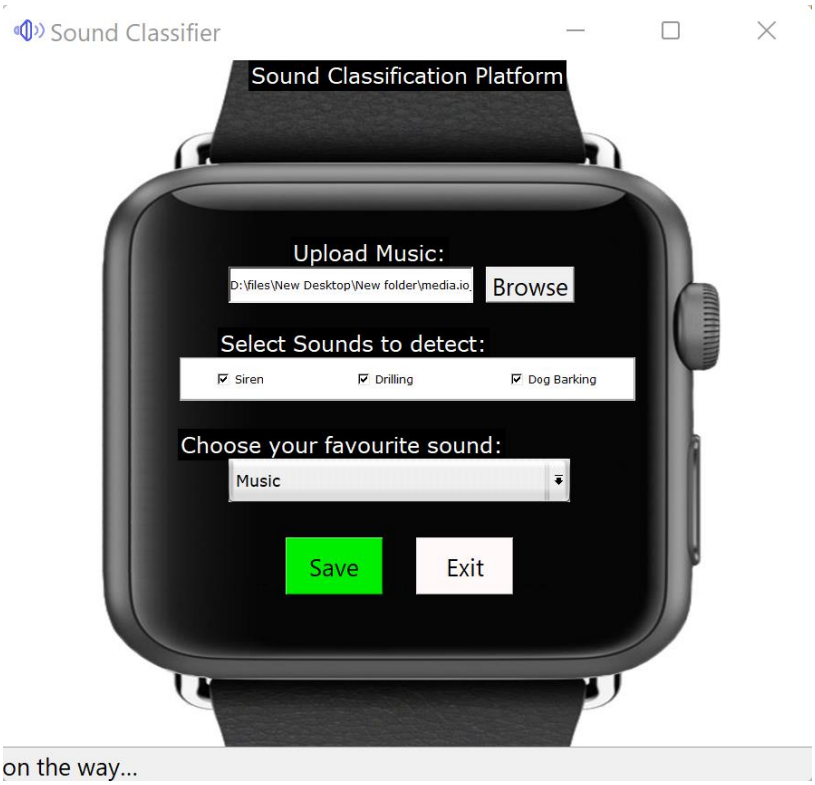


Figure 4-4 A)The GUI first page B) The GUI second page

Figure 4-4 B shows the second page of the GUI. This page is activated by pressing the setting button on the first page. On this page, the users are able to choose and upload their preferred music and sound to be stored in the memory, select the aversive sounds to be suppressed, and also choose their preferred music or some pre-uploaded neutral sounds such as white noise, rain, wind, ocean, and waves to be played. Users can upload their favorite music by using the browse button on the right. The save button in green color will save all the changes by the user and update the system. The exit buttons on this page will close the second page and turn back to the first page. Figure 4-5 shows the options menu in which the users can define what kind of sounds they are willing to hear facing an aversive sound.

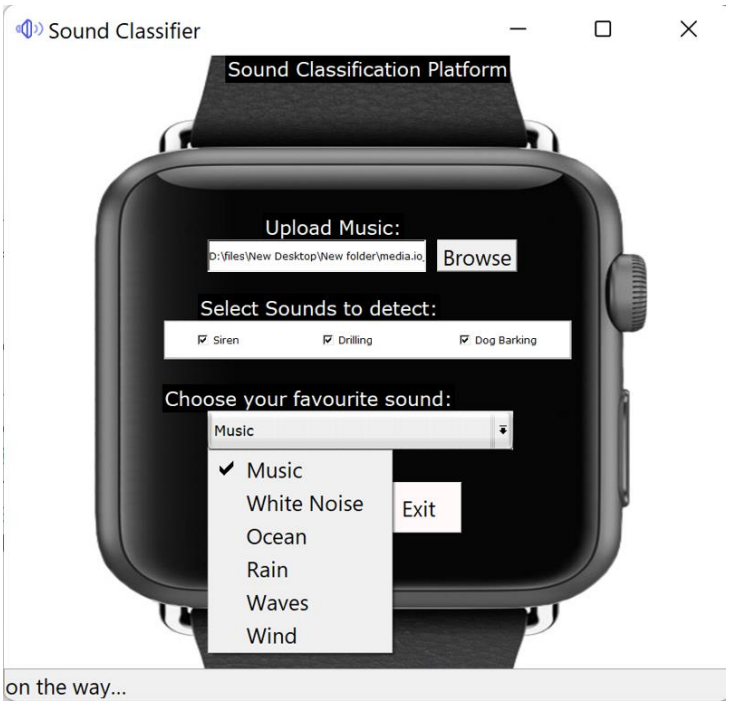


Figure 4-5 The favorite sounds option menu

4.6. Experimental Setup and Simulation Results

The proposed intervention in this thesis was designed to capture a very small chunk of environmental sound (16 ms) in order to operate in real-time. As mentioned, human ears are sensitive to delay and tolerate a latency of less than 40 ms. The processing time to execute the pre-processing and post-processing steps including feature map creation and clean signal reconstruction is 1-2 ms using 4 CPUs with 16 GB RAM. The required processing time to record sample data is 16 ms, while the overall time required for sample data recording and processing steps is around 17 to 18 ms. Even though this number is less than the acceptable delay for human ears, the processing time of the predictions in deep neural network algorithms is still needed to add to the required time for recording the data samples and processing steps. Due to the high performance and the accuracy of the proposed deep learning architects in this thesis, their several complex layers, neurons, and millions of parameters to generate such an outcome, are computationally intense. Considering the two deep neural network models for classification and filtration in the process, they add a huge delay of around 400 ms to the system which is unacceptable. Real-time deep learning-based processing is expensive to achieve considering the associated delay in the system unless the

platform contains an accelerator like GPU and TPU, which could speed up the processing and reduce the computational time.

As mentioned earlier, cloud-based deployment and AI edge deployment are used to address the deployment of deep learning models. In this thesis, the cloud-based approach is used due to its advantages over the AI edge deployment. The cloud computing advantages are high-speed computation, scalability, simple deployment, painless try and error, trouble-free debugging, unlimited storage capacity, backup and restore data, data loss prevention, and efficiency and cost reduction. Even though the AI edge deployment also has many advantages in the current system application, considering the resources and implementation time on AI edge, the cloud computing is considered more efficient.

Nowadays, there are several deep learning cloud services available for training, inference, and deployment assignments. The most commonly used service providers are AWS, Google Cloud, and Microsoft Azure. In comparison, AWS is now running for almost 7 years and as a result, they have more capital, more infrastructure, and better and more scalable services available for users compared to other web services. In this thesis, an Amazon EC2 G3 service is used which is the latest generation of Amazon EC2 GPU graphics instances that deliver a powerful combination of CPU, host memory, and GPU capacity. G3 instances provide access to NVIDIA Tesla M60 GPUs, each with up to 2,048 parallel processing cores, and 8 GB of GPU memory. Considering high-speed internet access, the system introduced in this thesis takes around 65-75 ms to perform all the processing steps, data recording, and inference using the deep neural network models which are less than the previous processing time of 400 ms. As it is clear from the simulation results, the use of AWS EC2 with 32 CPUs and 8 GB GPU RAM reduces the processing time by the order of 6 compared to an operating system using 4 CPUs with 16 GB RAM. Even though the processing time is still out of the range of human ears' latency tolerance, the system is successfully deployed on a cloud service with small noticeable latency. While not ideal, this latency was small enough that we proceeded with initial testing sessions with individuals with ASD in order to examine the performance of the system. This improvement in reducing latency is a huge step toward the deployment of the system to operate in real-time for real applications. The processing time is also affected by the high-speed internet availability and it is expected

that AI edge deployment solve the latency problem entirely due to its proven performance in other commercial applications.

4.7. Testing Sessions Experiment and Results

In order to examine the noise attenuation system introduced in this thesis in real-life, testing sessions were scheduled with adults on the autism spectrum who self-reported a history of negative reactions to sound. For this purpose, a comprehensive test scenario was designed, and the documentation of a use case was prepared. The testing scenarios document describes an action the user may undertake with the system. It also describes a situation when the user may find themselves in while using that system. In other words, the scenario document is an instruction for the testing sessions, the participant specifications, individual requirements, measurable and un-measurable metrics, result normalization, and system performance assessment. The aim of testing sessions was to find out information on participants' comfort levels while listening to various sounds with and without interventions. According to the document, different scenarios were considered to be tested on the participant. To illustrate, a summary of the steps during each session is provided. The pre-session interviews and preparation steps are ignored in the summary below.

- Step 1: Play the siren and dog barking in order to get to the comfortable listening level of the participant. The participant wears headphones connected to the laptop. The test taker plays the sounds each in 5s through zoom. Zoom is running on iPad with speakers on. The test takers increase the volume from 0 upwards until the participant raises hand to indicate the loudest comfortable listening level.

- Step 2: Play each aversive sound and get comfort ratings incrementally. Participants will rate their comfort with the aversive sounds in 10 increasing increments in volume from 0 to the max comfortable listening level found in step 1.

- Step 3: Using the aversive sound filtration tool, to find the loudest comfortable listening level for music. Participants will adjust the volume and find the loudest volume that is still comfortable for them for the music sound, starting from zero.

The music is played continuously through the platform on the laptop, through the earphones worn by the participant. Participants can manually adjust the volume slider to increase/decrease the volume. After they find their comfort level, they will pause the music and go to the next step. The system records the settings automatically.

- Step 4: Using the aversive sound filtration tool, to find the loudest comfortable listening level for white noise. Participants will adjust the volume and find the loudest volume that is still comfortable for them for the white noise sound, starting from zero. The white noise is played continuously through the platform on the laptop, through the earphones worn by the participant. Participants can manually adjust the volume slider to increase/decrease the volume. After they find their comfort level, they will pause the white noise and go to the next step. The system records the settings automatically.

- Step 5: Test the effect of masking the siren with music. The test taker plays the aversive sound over Zoom on the iPad at volume specified in Step 1 and then participants press the play button for music simultaneously at the level specified in Step 3. Then the participant pauses the music and provide a comfort rating.

- Step 6: Test the effect of masking the siren with white noise. The test taker plays the aversive sound over Zoom on the iPad at volume specified in Step 1 and then participants press the play button for white noise simultaneously at the level specified in Step 4. Then the participant pauses the music and provide a comfort rating.

- Step 7: Test the effect of masking the siren using the filter without speech. The test taker plays aversive sound through Zoom on the iPad; the participant then presses the filtration button, which filters out the aversive sound from the signal being delivered to the ears. The participant then provides a comfort rating.

- Step 8: Test the effect of masking the siren using the filter with speech. The test taker plays aversive sound + speech through Zoom on the iPad; the participant then presses the filtration button, which filters out the aversive sound from the signal being delivered to the ears. The participant then provides a comfort rating.

- Step 9: Repeat steps for the dog barking sound
- Step 10: Qualitative feedback

In this experiment, four different sounds were included: two neutral sounds (birds and rain) [113], and two aversive sounds (siren and dog barking). The neutral sounds were included between the aversive sounds in step 1 in order to make the participant comfortable during the testing session.

For this pilot test, eight participants with ASD were recruited to evaluate the performance of the system. Inclusion criteria for the participant included: a history of sensitivity to sounds, participant's normal hearing (thresholds ≤ 20 dB HL between 250 to 8000 Hz), and a verbal IQ over 85 for participating in a testing session. The demographical information of the participants is presented in Table 4-1.

Table 4-1 The demographical information of the participants

Participant number	1	2	3	4	5	6	7	8
Age (years)	29	25	50	19	19	21	46	31
Primary Language	English	English	English	English	English	English	Spanish	English
Highest Education	Bachelor Degree	Bachelor Degree	Half a Bachelor Degree	College Certificate	Elementary School	Elementary School	Bachelor Degree	Bachelor Degree
Living Situation	With roommate	-	Alone	With parents	With roommate	With parents	With spouse/ partner	Alone

Due to the covid-19 pandemic, the testing sessions are held remotely at first (n=6). With the ease of Covid-19 restrictions, as of the preparation of this thesis in the summer of 2022, two in-person sessions were also run based on the participant's preference. Remote sessions were conducted using a web conferencing application and each participant received the testing materials including a windows laptop, a headphone, and a tablet. The tablet was used as an external camera so the experimenter could talk to the participants while the testing session was conducted. The sessions were run by two experimenters. One experimenter provided instructions to participants and was responsible for communicating with participants. The second experimenter (myself) was responsible for session setup, software operation, solving any technical issues during the test, and performing the test.

At each step, participants were asked to rate their comfort level based on a 4-point Likert rating scale, illustrated in Table 4-2 (very comfortable = 1; very uncomfortable = 4).

Table 4-2 Comfort rate based on a 4-point Likert rating scale

	Comfort Rate
Very Comfortable	1
Somewhat Comfortable	2
Somewhat Uncomfortable	3
Very Uncomfortable	4

There was substantial variability in comfort ratings between participants. For example, one participant found the volume 20% of the maximum volume the highest level of uncomfortability and therefore, the comfort level would be 4 for this experiment. Another participant found the volume 100% of the maximum volume somewhat comfortable or very comfortable.

Table 4-3 shows the average comfort rating for different scenarios for the eight individuals with ASD participating in testing sessions. According to this table, the average comfort rate for siren and dog barking, at the maximum level defined by the participant, without any intervention technique is about 3.7. It means that the average participants are very uncomfortable while the siren and dog barking are playing through the microphone. When music and white noise were played to mask the siren and dog barking, average comfort rating changed to around 3, showing that on average participants' comfort improved somewhat but that they were still somewhat uncomfortable. The use of filtration technique (no speech present) changed comfort ratings to an average 2.0, indicating that the average participants became somewhat comfortable when the filtration technique was applied (when there was no speech in the sound playing through the microphone). The last scenario (filtration with speech present) was to use the filter while there was speech in the original signal. According to Table 4-3, the average comfort rating was 3.2. Even though this comfort rate is better than the original comfort level for unfiltered aversive sounds ((3.7), it is not satisfactory to be very comfortable for the participant. A comprehensive study based on qualitative feedback has been performed in order to figure out the reasons. While the participants are comfortable with the level of aversive sound filtration, the high level of rating is due to the unpleasant speech. A couple of participants find the speech synthetic while others

find the speech subject boring and uninteresting. There were cases that report speech intelligibility as the reason for a high rating. In the next rounds of testing sessions, more participants need to be recruited and more scenarios need to be defined to diverse the comfort rates and to further study the performance of the system in real-life.

Table 4-3 Average comfort rate for different scenarios

	Average Comfort Rate
Comfort rate with siren	3.7
Comfort rate with dog barking	3.6
Comfort rate with siren and music masking	2.7
Comfort rate with siren and white-noise masking	2.7
Comfort rate with siren and noise suppression tool without speech	2.1
Comfort rate with siren and noise suppression tool with speech	3.3
Comfort rate with dog bark and music masking	3.5
Comfort rate with dog bark and white-noise masking	2.7
Comfort rate with dog bark and noise suppression tool without speech	2
Comfort rate with dog bark and filtration tool with speech	3.1

4.8. Conclusion

In this chapter, the classification and filtration algorithm designed in chapters 2 and 3 are integrated to present the integrated system structure. The input environmental sound is presented using a cumulative framing strategy to save memory and computation costs. Then, the modified version of the classification and filtration task specified for individuals with ASD is explained. Afterward, the integrated system is introduced and different scenarios which may happen in the operation are explained. The designed GUI that interacts between the backend software and the user has been explained. The deployment strategy is then introduced and the simulation results along with the testing session results are described. According to the simulation results and testing sessions, the proposed integrated system is able to work in real-time using the cloud computation technique and is able to identify the aversive sound in the environment and filter the predicted aversive sound through the filtration process.

Moreover, most of the experiment performed in this thesis shows that the participants who are suffered from ASD find the system introduced here as a successful strategy to filter the aversive sound and increase their comfort level while using the presented software.

Chapter 5. Conclusion

5.1. Introduction

In this thesis, a new intervention technique is proposed to ameliorate auditory sensitivity in children with ASD. The proposed technique employs a deep learning-based algorithm to identify the specified aversive sounds in the environment and a noise filtering system to eliminate the intensity of the identified sounds. The specified aversive sounds are among the sounds that are commonly reported among the ASD population. The proposed deep learning-based technique receives a mixed signal consisting of the aversive sound such as siren and non-aversive sounds such as speech as an input and extracts its features to feed to the identification model for classification. In chapter 2, the detailed description of the proposed classification algorithm is provided. The proposed classifier is a unified CNN-RNN framework for environmental sound classification which benefits from the advantages of CNN and RNN. The input of the algorithm is a three-dimensional feature map including mel-spectrogram and decomposing audio time-series data into harmonic and percussive components, and the output is the predicted class. Furthermore, several deep learning techniques including transfer learning and data augmentation are used to improve the model performance. For data augmentation, a generative model using DCGAN is used to address the lack of data problem for environmental sound classification. This data augmentation method can produce spectrograms with similar structures to the original dataset. The performance of the proposed classification algorithm is examined using the UrbanSound8K dataset and achieved an overall accuracy of 98% to detect a target sound in the environment and is able to surpass other state-of-the-art algorithms.

When an aversive sound is detected by the classifier, the filtering system will be activated to remove the aversive part of the signal. In chapter 3, the detailed description of the proposed filtration algorithm is provided. The proposed algorithm is a DNN-based framework which employs deep learning and signal processing techniques to suppress the specified aversive sound from a mixed signal and pass the non-aversive sounds. The algorithm receives a mixed signal as an input and extracts its features to feed to a DNN model. The proposed DNN model is designed and trained to generate a ratio mask in order to create the clean signal. Signal processing techniques are utilized to increase

the output sound quality and integrability before and after the DNN model. The proposed techniques and DNN model are trained on the Librispeech dataset as the clean speech sources and the ESC-50 dataset as the aversive sound sources. The simulation results indicate that the proposed method can remove the aversive sound from the mixed signal, even in the presence of complex noisy conditions (variant SNRs) and difficult noise types (non-stationary and highly dynamic noises).

In Chapter 4, the proposed classification and filtration algorithms are integrated to present the overall intervention technique. This system is able to identify an aversive sound in the environment, alert the user of the presence of the aversive sound and filter the aversive sound while passing non-aversive sounds. The entire system is formed in the shape of a GUI to communicate with users and give them the ability to customize and control the system. The designed software is using a framing strategy which able the system to work in real-time. The software is deployed on the AWS data center using a GPU and is compatible with every operating system. To evaluate the performance of the aversive sound attenuation software on auditory sensitivity in real-life, several testing sessions are conducted in collaboration with individuals with ASD to assess the functionality of the proposed software in the real setting and obtain valuable feedback for further improvement. Eight participants with ASD and sound sensitivity history are recruited to examine the performance of the system. The details of testing session steps are provided in Chapter 4. According to the session results, the proposed intervention technique is able to ameliorate comfort level in the presence of aversive sounds.

5.2. Future Works

A list of possible expansions to the current work is as follows:

1. In Chapter 2 and 3, the classification and filtration models can be trained on more data samples to diversify the dataset and improve the model performance in real field. The dataset can be captured from the real environment or in an acoustic laboratory.

2. In chapter 4, the existing software can be implemented into a standalone device on the edge using an artificial intelligence (AI) chip to avoid network latency and increase data security. Some of the AI chips utilize GPU, CPU, memory, power

management and high-speed interfaces to enable faster training and deployment.

3. In Chapter 4, a long-term field testing can be conducted with children with ASD in their daily lives to understand the impact of the device to improve the quality of life for the children and their families. Improving the quality of life is achieved by removing some of the barriers to increase social interactions and participation and improve mental health.

References

[1] B. Haesen, B. Boets, and J. Wagemans, “A review of behavioural and electrophysiological studies on auditory processing and speech perception in autism spectrum disorders,” *Research in Autism Spectrum Disorders*, vol. 5, no. 2. Elsevier, pp. 701–714, Apr. 01, 2011, doi: 10.1016/j.rasd.2010.11.006.

[2] “Autism Prevalence Higher in CDC’s ADDM Network | CDC Online Newsroom | CDC.” https://www.cdc.gov/media/releases/2021/p1202-autism.html (accessed Mar. 14, 2022).

[3] A. Ben-Sasson, S. A. Cermak, G. I. Orsmond, H. Tager-Flusberg, M. B. Kadlec, and A. S. Carter, “Sensory clusters of toddlers with autism spectrum disorders: differences in affective symptoms,” *J. Child Psychol. Psychiatry.*, vol. 49, no. 8, pp. 817–825, Aug. 2008, doi: 10.1111/J.1469-7610.2008.01899.X.

[4] E. J. Marco, L. B. N. Hinkley, S. S. Hill, and S. S. Nagarajan, “Sensory Processing in Autism: A Review of Neurophysiologic Findings,” *Pediatr. Res.*, vol. 69, no. 5 Pt 2, p. 48R, May 2011, doi: 10.1203/PDR.0B013E3182130C54.

[5] G. Dawson and R. Watling, “Interventions to facilitate auditory, visual, and motor integration in autism: a review of the evidence,” *J. Autism Dev. Disord.*, vol. 30, no. 5, pp. 415–421, 2000, doi: 10.1023/A:1005547422749.

[6] J. Kern, M. Trivedi, C. Garver, … B. G.-, and undefined 2006, “The pattern of sensory processing abnormalities in autism,” *journals.sagepub.com*, vol. 10, no. 5, pp. 480–494, Sep. 2006, doi: 10.1177/1362361306066564.

[7] M. W. M. Kuiper, E. W. M. Verhoeven, and H. M. Geurts, “Stop Making Noise! Auditory Sensitivity in Adults with an Autism Spectrum Disorder Diagnosis: Physiological Habituation and Subjective Detection Thresholds,” *J. Autism Dev. Disord.*, vol. 49, no. 5, pp. 2116–2128, May 2019, doi: 10.1007/s10803-019-03890-9.

[8] N. E. Scheerer, T. Q. Boucher, B. Bahmei, G. Iarocci, S. Arzanpour, and E. Birmingham, “Family Experiences of Decreased Sound Tolerance in ASD,” *J. Autism Dev. Disord.*, 2021, doi: 10.1007/S10803-021-05282-4.

[9] Z. J. Williams, J. L. He, C. J. Cascio, and T. G. Woynaroski, “A review of decreased sound tolerance in autism: Definitions, phenomenology, and potential mechanisms,” *Neurosci. Biobehav. Rev.*, vol. 121, pp. 1–17, Feb. 2021, doi: 10.1016/J.NEUBIOREV.2020.11.030.

[10] S. Khalfa *et al.*, “Increased perception of loudness in autism,” *Hear. Res.*, vol. 198, no. 1–2, pp. 87–92, Dec. 2004, doi: 10.1016/J.HEARES.2004.07.006.

[11] M. M. Phillips, D. P., & Carr, “Disturbances of loudness perception,” *J. Am. Acad.*

Audiol., vol. 9, no. 5, pp. 371–379, 1998.

[12] J. J. Brout *et al.*, “Investigating Misophonia: A Review of the Empirical Literature, Clinical Implications, and a Research Agenda,” *Front. Neurosci.*, vol. 0, no. FEB, p. 36, Feb. 2018, doi: 10.3389/FNINS.2018.00036.

[13] J. Claiborn, J. M. Claiborn, T. H. Dozier, S. L. Hart, and J. Lee, “SELF-IDENTIFIED MISOPHONIA PHENOMENOLOGY, IMPACT, AND CLINICAL CORRELATES,” *Psychol. Thought*, vol. 13, no. 2, pp. 349–375, Oct. 2020, doi: 10.37708/psyct.v13i2.454.

[14] P. Jastreboff, M. J.-S. in Hearing, and undefined 2014, “Treatments for decreased sound tolerance (hyperacusis and misophonia),” *Citeseer*, Accessed: Jun. 23, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1067.3711&rep=rep1&type=pdf.

[15] H. Weber, K. Pfadenhauer, M. Stöhr, and A. Rösler, “Central hyperacusis with phonophobia in multiple sclerosis,” *Mult. Scler.*, vol. 8, no. 6, pp. 505–509, Dec. 2002, doi: 10.1191/1352458502ms814oa.

[16] J. Law, E. Rubenstein, … A. M.-P. A., and undefined 2016, “Auditory sensitivity issues in children with autism spectrum disorders: Characteristics and burden,” *iancommunity.org*, Accessed: Nov. 15, 2021. [Online]. Available: https://iancommunity.org/sites/default/files/galleries/conference-presentations/Law_PAS_2016.pdf.

[17] E. Boucher, T. Q., Scheerer, N. E., Iarocci, G., Bahmei, B., Arzanpour, S., & Birmingham, “Misophonia, hyperacusis, and the relationship with quality of life in autistic and non-autistic adults.”

[18] K. P.-2015 I. 25th I. W. on and undefined 2015, “Environmental sound classification with convolutional neural networks,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7324337/?casa_token=7jlPqDKatOAAAAAA:ookWSDS23gdvKhT_6k0JUb4b271LnXDf92PPgUttb15kAtOf9H8d9btlg95_8fkU9cC1NVk24xI.

[19] H. B. Sailor, D. M. Agrawal, and H. A. Patil, “Unsupervised filterbank learning using Convolutional Restricted Boltzmann Machine for environmental sound classification,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017, vol. 2017-August, pp. 3107–3111, doi: 10.21437/Interspeech.2017-831.

[20] D. Yu and L. Deng, *Automatic Speech Recognition*. London: Springer London, 2015.

[21] N. Shankar, G. S. Bhat, and I. M. S. Panahi, “Efficient two-microphone speech enhancement using basic recurrent neural network cell for hearing and hearing

aids," *J. Acoust. Soc. Am.*, vol. 148, no. 1, pp. 389–400, Jul. 2020, doi: 10.1121/10.0001600.

[22] H. K. Maganti, D. Gatica-Perez, and I. McCowan, "Speech enhancement and recognition in meetings with an audio-visual sensor array," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 15, no. 8, pp. 2257–2269, Sep. 2007, doi: 10.1109/TASL.2007.906197.

[23] S. Duan, J. Zhang, P. Roe, and M. Towsey, "A survey of tagging techniques for music, speech and environmental sound," *Artif. Intell. Rev.*, vol. 42, no. 4, pp. 637–661, Dec. 2014, doi: 10.1007/s10462-012-9362-y.

[24] E. Alexandre, L. Cuadra, … M. R.-I. T. on, and undefined 2007, "Feature selection for sound classification in hearing aids through restricted search driven by genetic algorithms," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4317557/?casa_token=veVuJSfdLzEAAAAA:GAxDr0zhvT97eqmA3kCU0bm4K_gSP0WeVjvt2BiMYtwBew_PsOnf5zmNc6ToVwfTL1Ko-EGgHbw.

[25] D. Mital, G. L.-R. and A. Systems, and undefined 1989, "A voice-activated robot with artificial intelligence," *Elsevier*, Accessed: Jun. 23, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/092188908990033X.

[26] K. Łopatka, P. Zwan, and A. Czyzewski, "Dangerous sound event recognition using support vector machine classifiers," *Adv. Intell. Soft Comput.*, vol. 80, pp. 49–57, 2010, doi: 10.1007/978-3-642-14989-4_5.

[27] D. Barchiesi, D. Giannoulis, … D. S.-I. S., and undefined 2015, "Acoustic scene classification: Classifying environments from the sounds they produce," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7078982/?casa_token=HdGKDdUX1Z4AAAAA:9arCAXTbDchg7og9wlGRUlQhgmvvlW6yYhZbGkdT9D4F5kTCro6Ty7FwBX1if2rbvhAfR_ng758.

[28] F. González-Hernández, L. S.-F.-A. Acoustics, and undefined 2017, "Marine mammal sound classification based on a parallel recognition model and octave analysis," *Elsevier*, Accessed: Jun. 23, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003682X16305254.

[29] L. Ballan, A. Bazzica, … M. B.-… on M. and, and undefined 2009, "Deep networks for audio event classification in soccer videos," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5202537/?casa_token=njtjCErHyQwAAAAA:JOoxPtFuD1LhvO1VpSFMFTWHCfkJkk2LPYmx8kP_eHR3nkamo2RiLs_CexQLzNKnMOEwgPylmPU.

[30] M. Vacher, J.-F. Serignat, and S. Chaillol, "Sound Classification in a Smart Room Environment: an Approach using GMM and HMM Methods," 2007. Accessed:

Jun. 23, 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00957418.

[31] B. Kostek, P. Szczuko, P. Zwan, and P. Dalka, “Processing of musical data employing rough sets and artificial neural networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3400 LNCS, pp. 112–133, 2005, doi: 10.1007/11427834_5.

[32] E. Pyshkin and A. Kuznetsov, “Searching for music: from melodies in mind to the resources on the web,” 2010. Accessed: Jun. 23, 2021. [Online]. Available: http://code.google.com/p/musicip-libofa.

[33] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, “Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Aug. 2015, vol. 2015-August, pp. 151–155, doi: 10.1109/ICASSP.2015.7177950.

[34] J. Salamon, J. B.-2015 I. I. C. on, and undefined 2015, “Unsupervised feature learning for urban sound classification,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7177954/?casa_token=3_4ovfK8hBAAAAAA:yEJuw8aJsr4ZmGjYPLAyaCU5PtrYgElXwxISLNsoJQM1QUn-DU-e3p9vX5RCQ8e1E1rBN4Kt7x8.

[35] J. Geiger, K. H.-2015 23rd E. S. Processing, and undefined 2015, “Improving event detection for audio surveillance using gabor filterbank features,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7362476/?casa_token=vCQ3z1UBB4oAAAAA:PFoamBSdpzJxfXEjK0KzEgZktXjv9ha8SS4ywoByReNItgKzFLSyLTJD_5IvjCIiL_b7E446vto.

[36] S. Sameh and Z. Lachiri, “Multiclass support vector machines for environmental sounds classification in visual domain based on log-Gabor filters,” *Int. J. Speech Technol.*, vol. 16, no. 2, pp. 203–213, Jun. 2013, doi: 10.1007/s10772-012-9174-0.

[37] F. Su, L. Yang, T. Lu, and G. Wang, “Environmental sound classification for scene recognition using local discriminant bases and HMM,” in *MM’11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops*, 2011, pp. 1389–1392, doi: 10.1145/2072298.2072022.

[38] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, “Context-dependent sound event detection,” *Eurasip J. Audio, Speech, Music Process.*, vol. 2013, no. 1, 2013, doi: 10.1186/1687-4722-2013-1.

[39] L. Lu, H. Jiang, and H. Zhang, “A robust audio classification and segmentation method,” in *Proceedings of the ninth ACM international conference on Multimedia - MULTIMEDIA ’01*, 2001, p. 203, doi: 10.1145/500141.500173.

[40] V. Peltonen, J. Tuomi, A. K.-… S. Processing, and undefined 2002, "Computational auditory scene recognition," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5745009/?casa_token=bOtjAnnBqzMAAAAA:x1HUaK5GdEppbKqqUhAe-40GO1CFBh_un5RbCq1cyrQIBITJrwx-nl7ACDsBdkgSstay60-PMlQ.

[41] B. K.-P. of the IEEE and undefined 1986, "Spectral analysis and discrimination by zero-crossings," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1457931/?casa_token=FpyxTnikTDQAAAAA:lOA5iKh1usYSDj6w4EydO6VzVWvGO49zmNyBMr5DxvDi68iXQm25Lqa5X8Dmw6eOarBkhK7t6sE.

[42] J. Markel and A. Gray, *Linear prediction of speech*. 2013.

[43] T. Zhang, C. K.-I. T. on speech and audio, and undefined 2001, "Audio content analysis for online audiovisual data segmentation and classification," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/917689/?casa_token=pNKYcrT2in0AAAAA:a60XhdyRsWpXnyxpnFo4cKdQESENa9jAFJKRxwcAf50TE8IBYXmB_2LJWWuPmkXOlzO6zKmugDY.

[44] G. Tzanetakis, P. C.-I. T. on speech and, and undefined 2002, "Musical genre classification of audio signals," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1021072/?casa_token=IbQJn1O5aFUAAAAA:FNU5YBY6E6QS9yv583pcN1a3260SwiLYOJ_qC18Tzvg2ToLU1d1rl4HrRKTMRAkl0cgExpvvmXc.

[45] S. Chu, S. Narayanan, C. K.-I. T. on Audio, and undefined 2009, "Environmental sound recognition with time–frequency audio features," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5109766/?casa_token=44P1Rm-XuDMAAAAA:1x0QJoOFELJ4w0ta_u-b2MCbYrT-ajiwC5-i5_7suBxDCNloxWZiH21AfNh_qiDn5n42K-oJMrs.

[46] H. Lu, H. Zhang, and A. Nayak, "A Deep Neural Network for Audio Classification with a Classifier Attention Mechanism," Jun. 2020, Accessed: Jun. 23, 2021. [Online]. Available: http://arxiv.org/abs/2006.09815.

[47] J. Salamon, J. B.-I. S. P. Letters, and undefined 2017, "Deep convolutional neural networks and data augmentation for environmental sound classification," *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7829341/?casa_token=spe1dJbHwzAAAAAA:OWTZs8w60PHZkTqKCmjcPYRUwz-ETB59nM9yEA0jJTGBYo-84grtlKpNj9EZTV_R--JDBFRJj8Q.

[48] C. Villanueva, J. Vincent, A. Slowinski, and M.-P. Hosseini, "Respiratory Sound Classification Using Long-Short Term Memory." Accessed: Jun. 23, 2021.

[Online]. Available: https://arxiv.org/abs/2008.02900.

[49] M. D. R, A. M. Kavitkar, and V. Soumya, “Sound Recognition Using Recurrent Neural Network,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 6, no. 4, pp. 815–819, 2018.

[50] F. Roche, T. Hueber, S. Limier, and L. Girin, “Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models,” *Proc. Sound Music Comput. Conf.*, pp. 415–422, Jun. 2018, Accessed: Jun. 23, 2021. [Online]. Available: http://arxiv.org/abs/1806.04096.

[51] T. V. and H. H. O. Gencoglu, “Recognition of acoustic events using deep neural networks | IEEE Conference Publication | IEEE Xplore.” https://ieeexplore.ieee.org/abstract/document/6952140?casa_token=TypXI0mEwvwAAAAA:uggA6ERZl0-aNcxrPyUOIy7ilfkZSHYIaqBoyuAzRg9bsHxq43WxXWRH1rW-u-4OiBW93kCiTGE (accessed Jun. 23, 2021).

[52] H. H. and T. V. E. Cakir, T. Heittola, “Polyphonic sound event detection using multi label deep neural networks,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7280624/?casa_token=qcsc1OQYjnkAAAAA:_6dAMjEltRP-RtM1gJE-4tN_rGLQo8u7PpYRGR2ilOapCJ1sFTO-vZjzlz3KHcLSUPErOXd2gNY.

[53] E. Cakır, G. Parascandolo, T. H.-… on Audio, undefined Speech, and undefined 2017, “Convolutional recurrent neural networks for polyphonic sound event detection,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7933050/?casa_token=X3IMPWMTtrsAAAAA:EranRFJYXYIPJJkRszqQfJ747N8Zsa2attiXYNiBN1ygkzMKsn0le5S9Ez1Z3DKTOibv7Lkb8X0.

[54] A. Graves, A. Mohamed, G. H.-2013 I. international, and undefined 2013, “Speech recognition with deep recurrent neural networks,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6638947/?casa_token=lHGs3SD9QLwAAAAA:JEkFg6oP7UVFJ5AG3rUDWb0Ut5qdi4-Zmd-Ue6hsWEL72yxMTbTfSxHiwNIbwve4He5CennsiqM.

[55] G. Parascandolo, H. Huttunen, and T. Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, May 2016, vol. 2016-May, pp. 6440–6444, doi: 10.1109/ICASSP.2016.7472917.

[56] D. Tang, B. Qin, and T. Liu, “Document Modeling with Gated Recurrent Neural Network for Sentiment Classification,” Association for Computational Linguistics, 2015. Accessed: Jun. 23, 2021. [Online]. Available: http://ir.hit.edu.cn/.

[57] Z. Zuo *et al.*, “Convolutional Recurrent Neural Networks: Learning Spatial Dependencies for Image Representation.” Accessed: Jun. 23, 2021. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W03/html/Zuo_Convolutional_Recurrent_Neural_2015_CVPR_paper.html.

[58] S. Sigtia, E. Benetos, S. D.-I. T. on, and undefined 2016, “An end-to-end neural network for polyphonic piano music transcription,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7416164/?casa_token=GNWdyEPM-dcAAAAA:vZUD-okHEA6ecRaJSQ4sl08q_f2XiAdtHCvvzP49vm1WGKfuvjfBEuJ3hlGTMHJvi5GUz506FtI.

[59] Y. Liu, X. Zhu, Z. Qin, and J. Li, “Emotion Classification with Data Augmentation Using Generative Adversarial Networks ‘Prediction of Sea Level of East Coast of Britain’ View project Biological Image Processing and Analysis View project Emotion Classification with Data Augmentation Using Generative Adversarial Networks,” *Springer*, vol. 10939 LNAI, pp. 349–360, 2018, doi: 10.1007/978-3-319-93040-4_28.

[60] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” Nov. 2016, Accessed: Jun. 23, 2021. [Online]. Available: https://arxiv.org/abs/1511.06434v2.

[61] I. J. Goodfellow *et al.*, “Generative Adversarial Nets.” Accessed: Jun. 23, 2021. [Online]. Available: http://www.github.com/goodfeli/adversarial.

[62] R. Talmon, I. Cohen, and S. Gannot, “Transient Noise Reduction Using Nonlocal Diffusion Filters,” *IEEE Trans. Audio, Speech Lang. Process.*, vol. 19, no. 6, pp. 1584–1599, 2011, doi: 10.1109/TASL.2010.2093651.

[63] R. Talmon, I. Cohen, S. G.-2011 I. International, and undefined 2011, “Clustering and suppression of transient noise in speech signals using diffusion maps,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5947500/?casa_token=_Kf5nJUjwS8AAAAA:lWp_Me42lTGcFWzC9hVvP0wIfbEEvX21SBJRVyYjRNtLiL_2nVI-egObC_bVRGolA_WovBCauNY.

[64] K. Cao and M. Wang, “Transient noise suppression algorithm in speech system ARTICLES YOU MAY BE INTERESTED IN,” *aip.scitation.org*, vol. 1864, p. 20006, Jul. 2017, doi: 10.1063/1.4992823.

[65] R. Ullah, M. S. Islam, Z. Ye, and M. Asif, “Semi-supervised transient noise suppression using OMLSA and SNMF algorithms,” *Appl. Acoust.*, vol. 170, p. 107533, Dec. 2020, doi: 10.1016/j.apacoust.2020.107533.

[66] N. Mohammadiha, P. Smaragdis, and A. Leijon, “Supervised and unsupervised speech enhancement using nonnegative matrix factorization,” *IEEE Trans. Audio,*

Speech Lang. Process., vol. 21, no. 10, pp. 2140–2151, 2013, doi: 10.1109/TASL.2013.2270369.

[67] S. Tamura and A. Waibel, “NOISE REDUCTION USING CONNECTIONIST MODELS.,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1988, pp. 553–556, doi: 10.1109/icassp.1988.196643.

[68] S. D. Kamath and P. C. Loizou, “A MULTI-BAND SPECTRAL SUBTRACTION METHOD FOR ENHANCING SPEECH CORRUPTED BY COLORED NOISE.” Accessed: Jun. 23, 2021. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.4102&rep=rep1&type=pdf.

[69] P. S.-1996 I. I. C. on Acoustics and undefined 1996, “Speech enhancement based on a priori signal to noise estimation,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/543199/?casa_token=fOmqul4OfeMAAAAA:1sgmyVnyzsjB1JabY3HKOSQvx_jo_VJ3wxaufL5G0ZzKbTieEopmxXH_6Nb8RSWnowB_2ReenRY.

[70] J. Lim, A. O.-I. T. on Acoustics, undefined Speech, and undefined 1978, “All-pole modeling of degraded speech,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1163086/?casa_token=dkMatbPwgjMAAAAA:9ppqKkHF2yjVpPm0u6ayGWHGtT0BiA2C8ioH5geBGs22y8m2VupSBW1SR8gzohB7EoxgdAiPbww.

[71] Y. Xu, J. Du, L. Dai, C. L.-I. T. on Audio, and undefined 2014, “A regression approach to speech enhancement based on deep neural networks,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6932438/?casa_token=o95kQhdsCfMAAAAA:UG36qiH0_BRd4E06HzOWJ35nG_P7xS9Yrwh9FGt_U14i8QugzjI3XmphdrKbdOSTMjFIHfibSCQ.

[72] M. Nikzad, A. Nicolson, Y. Gao, J. Zhou, K. K. Paliwal, and F. Shang, “Deep Residual-Dense Lattice Network for Speech Enhancement,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, pp. 8552–8559, Apr. 2020, doi: 10.1609/aaai.v34i05.6377.

[73] M. Strake, B. Defraene, K. Fluyt, W. Tirry, and T. Fingscheidt, “Speech enhancement by LSTM-based noise suppression followed by CNN-based speech restoration,” *EURASIP J. Adv. Signal Process.*, vol. 2020, no. 1, Dec. 2020, doi: 10.1186/s13634-020-00707-1.

[74] Y. Hu *et al.*, “DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2020, vol. 2020-October, pp. 2472–2476, doi: 10.21437/Interspeech.2020-2537.

[75] Y. Tsao, S. Matsuda, X. Lu, and C. Hori, “Speech enhancement based on deep denoising Auto-Encoder Speech Enhancement Based on Deep Denoising Autoencoder,” 2013. Accessed: Jun. 23, 2021. [Online]. Available: https://www.researchgate.net/publication/283600839.

[76] S. Fu, C. Liao, Y. T.-I. S. P. Letters, and undefined 2019, “Learning with learned loss function: Speech enhancement with quality-net to improve perceptual evaluation of speech quality,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8902088/?casa_token=cAlRcuSZCZwAAAAA:jgzfMcYD1sEomMV678hOCy7kllFuTUV8m41AmEgNXExHb9Z8KmXZ_-zybSM85tqWKtmJeEWOuJ0.

[77] Y. Xia, S. Braun, C. Reddy, … H. D.-I. 2020-2020, and undefined 2020, “Weighted speech distortion losses for neural-network-based real-time speech enhancement,” *ieeexplore.ieee.org*, Accessed: Jun. 23, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9054254/?casa_token=8fEvcswgevAAAAAA:x42tZaHD-GGCSws7JSak1zbli-HimNSfE6BMrF4y0NMwVWq0FG4dYEmJSp9r27m3n0tdmIjGcMw.

[78] L. Deng, D. Yu, L. Deng, and D. Yu, “Deep Learning: Methods and Applications,” *Found. Trends R Signal Process.*, vol. 7, pp. 197–387, 2013, doi: 10.1561/2000000039.

[79] I. Tabian, H. Fu, and Z. S. Khodaei, “A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures,” *Sensors 2019, Vol. 19, Page 4933*, vol. 19, no. 22, p. 4933, Nov. 2019, doi: 10.3390/S19224933.

[80] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, “CNN-RNN: A Unified Framework for Multi-label Image Classification,” 2016.

[81] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[82] K. Cho *et al.*, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.” Accessed: Jun. 24, 2021. [Online]. Available: https://arxiv.org/abs/1406.1078.

[83] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” pp. 248–255, Mar. 2010, doi: 10.1109/CVPR.2009.5206848.

[84] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 1–9, Oct. 2015, doi: 10.1109/CVPR.2015.7298594.

[85] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Sep. 2014, Accessed: Jan. 30, 2022. [Online]. Available:

https://arxiv.org/abs/1409.1556v6.

[86] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, Dec. 2015, doi: 10.48550/arxiv.1512.03385.

[87] S. Alvarez-buylla Puente, “Single and Multi-Label Environmental Sound Classification Using Convolutional Neural Networks Master’s thesis in the Programme Sound and Vibration,” 2018. Accessed: Jun. 24, 2021. [Online]. Available: https://odr.chalmers.se/handle/20.500.12380/255604.

[88] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019, doi: 10.1186/S40537-019-0197-0/FIGURES/33.

[89] D. Sarkar, R. Bali, and T. Ghosh, “Hands-on transfer learning with Python : implement advanced deep learning and neural network models using TensorFlow and Keras.”

[90] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, Feb. 2015, doi: 10.48550/arxiv.1502.03167.

[91] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” Dec. 2015, Accessed: Jun. 24, 2021. [Online]. Available: https://arxiv.org/abs/1412.6980v9.

[92] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, “Classifying environmental sounds using image recognition networks,” *Procedia Comput. Sci.*, vol. 112, pp. 2048–2056, Jan. 2017, doi: 10.1016/J.PROCS.2017.08.250.

[93] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” Dec. 2014, Accessed: Jun. 24, 2021. [Online]. Available: http://arxiv.org/abs/1412.3555.

[94] Y. Su, K. Zhang, J. Wang, and K. Madani, “Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion,” *Sensors 2019, Vol. 19, Page 1733*, vol. 19, no. 7, p. 1733, Apr. 2019, doi: 10.3390/S19071733.

[95] M. Esmaeilpour, P. Cardinal, and A. Lameiras Koerich, “Unsupervised feature learning for environmental sound classification using Weighted Cycle-Consistent Generative Adversarial Network,” *Appl. Soft Comput.*, vol. 86, p. 105912, Jan. 2020, doi: 10.1016/J.ASOC.2019.105912.

[96] “Difference Between Stationary and Non-Stationary Signals.” https://askanydifference.com/difference-between-stationary-and-non-stationary-signals/ (accessed Jun. 27, 2022).

[97] N. Pan, J. Chen, and B. H. F. Juang, “Comparative study of deep learning based and traditional single-channel noise-reduction algorithms,” *2019 Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. APSIPA ASC 2019*, pp. 1880–1884, Nov. 2019, doi: 10.1109/APSIPAASC47483.2019.9023278.

[98] L. Xu, Z. Wei, S. F. A. Zaidi, B. Ren, and J. Yang, “Speech enhancement based on nonnegative matrix factorization in constant-Q frequency domain,” *Appl. Acoust.*, vol. 174, p. 107732, Mar. 2021, doi: 10.1016/J.APACOUST.2020.107732.

[99] B. Kirubagari, R. S.-B. I. J. of, and undefined 2012, “A Noval Approach in Speech Enhancement for Reducing Noise Using Bandpass Filter and Spectral Subtraction,” *journal.bonfring.org*, Accessed: Jun. 24, 2021. [Online]. Available: http://www.journal.bonfring.org/abstract.php?id=6&archiveid=109.

[100] M. A. Stone and B. C. J. Moore, “Tolerable hearing aid delays. I. Estimation of limits imposed by the auditory path alone using simulated hearing losses,” *Ear Hear.*, vol. 20, no. 3, pp. 182–192, Jun. 1999, doi: 10.1097/00003446-199906000-00002.

[101] I. Y. Soon and S. N. Koh, “Speech Enhancement Using 2-D Fourier Transform,” *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 717–724, 2003, doi: 10.1109/TSA.2003.816063.

[102] S. Kelkar, … L. G.-I. T. on, and undefined 1983, “An extension of Parseval’s theorem and its use in calculating transient energy in the frequency domain,” *ieeexplore.ieee.org*, Accessed: Jun. 24, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4180434/?casa_token=u5_qN2iFIN8AAAAA:Ii_WskxUv5B2pyaKU5nD-e2zrjcKJ3YCBgGVRJUmPxGGWBK19HiCf4ftDtGQmVHEj5C7oZGl28I.

[103] J. Fürnkranz *et al.*, “Mean Squared Error,” *Encycl. Mach. Learn.*, pp. 653–653, 2011, doi: 10.1007/978-0-387-30164-8_528.

[104] A. Singh and J. Singh, “Comparative Analysis of Gaussian Filter with Wavelet Denoising for Various Noises Present in Images,” *Indian J. Sci. Technol.*, vol. 9, no. 47, 2016, doi: 10.17485/ijst/2016/v9i47/106843.

[105] V. Panayotov, G. Chen, … D. P.-2015 I. international, and undefined 2015, “Librispeech: an asr corpus based on public domain audio books,” *ieeexplore.ieee.org*, Accessed: Jun. 24, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7178964/?casa_token=rz40qCHe1bAAAAAA:azelbtYhDFYyLAcif0miodEmrZguoeZvwzH_6HsJyLvHEQarTwV6mkJ48HPhKZvb_0dqfvnqcYI.

[106] K. J. Piczak, “ESC: Dataset for environmental sound classification,” in *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, Oct. 2015, pp. 1015–1018, doi: 10.1145/2733373.2806390.

[107] M. Gogate, K. Dashtipour, A. Adeel, and A. Hussain, “CochleaNet: A robust language-independent audio-visual model for real-time speech enhancement,” *Inf. Fusion*, vol. 63, pp. 273–285, Nov. 2020, doi: 10.1016/j.inffus.2020.04.001.

[108] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “A short-time objective intelligibility measure for time-frequency weighted noisy speech,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2010, pp. 4214–4217, doi: 10.1109/ICASSP.2010.5495701.

[109] Q. Huang, C. Bao, X. Wang, and Y. Xiang, “Speech enhancement method based on multi-band excitation model,” *Appl. Acoust.*, vol. 163, p. 107236, Jun. 2020, doi: 10.1016/j.apacoust.2020.107236.

[110] Y. Hu and P. C. Loizou, “Evaluation of objective quality measures for speech enhancement,” *IEEE Trans. Audio, Speech Lang. Process.*, vol. 16, no. 1, pp. 229–238, Jan. 2008, doi: 10.1109/TASL.2007.911054.

[111] T. V. Sreenivas and P. Kirnapure, “Codebook constrained wiener filtering for speech enhancement,” *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 383–389, 1996, doi: 10.1109/89.536932.

[112] N. L. Westhausen and B. T. Meyer, “Dual-Signal Transformation LSTM Network for Real-Time Noise Suppression,” *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2020-October, pp. 2477–2481, May 2020, Accessed: Jun. 24, 2021. [Online]. Available: http://arxiv.org/abs/2005.07551.

[113] W. Yang *et al.*, “Affective auditory stimulus database: An expanded version of the International Affective Digitized Sounds (IADS-E),” *Behav. Res. Methods*, vol. 50, no. 4, pp. 1415–1429, Aug. 2018, doi: 10.3758/S13428-018-1027-6/TABLES/8.