SOUND SOURCE LOCALIZATION FOR IMPROVING HEARING AID STUDIES USING MOBILE PLATFORMS

by

Abdullah Küçük

APPROVED BY SUPERVISORY COMMITTEE:

Issa M.S. Panahi, Chair

Carlos Busso

Aria Nosratinia

Mehrdad Nourani

To my daughter, son, wife, family and every person who had faith in me.

SOUND SOURCE LOCALIZATION FOR IMPROVING HEARING AID STUDIES USING MOBILE PLATFORMS

by

ABDULLAH KÜÇÜK, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2021

ACKNOWLEDGMENTS

Firstly, I would like to say a special thank you to my supervisor, Dr. Issa Panahi, without whom I would not have been able to complete this research. His consistent support, guidance, and overall insights in this field have made this an inspiring experience for me. His support extends beyond just our academic collaboration. He has been a brilliant mentor in academics as well as in life. I would like to express my gratitude to Dr. Carlos Busso, Dr. Aria Nosratinia, and Dr. Mehrdad Nourani for serving as committee members for my PhD dissertation defense.

Furthermore, my sincere thanks also goes to my lab members, Mr. Gautam Bhat, Mr. Anton Kovalyov, Mr. Kashyap Patel, Mr. Nikhil Shankar, and Mr. Serkan Tokgöz, for their support and help. We have had very productive discussions regarding research. I would like to thank them for their precious friendships and sharing the difficulties of research with me.

Last but not least, I am thankful to my wife, Beyzanur, and my kids, Omer and Vera, my parents, Rahman and Saadet, and my siblings, Ismail, Yunus Emre, and Esra. They have been my inspiration and motivation throughout my research. Thank you for your support and motivation.

This dissertation was supported by the National Institute of the Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health (NIH) under Award 1R01DC01-5430-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [name of university or educational entity]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_`

standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

November 2021

SOUND SOURCE LOCALIZATION FOR IMPROVING HEARING AID STUDIES USING MOBILE PLATFORMS

Abdullah Küçük, PhD
The University of Texas at Dallas, 2021

Supervising Professor: Issa M.S. Panahi, Chair

Microphone array is one of the powerful techniques that enables to apply effective signal processing algorithms to systems. One of the critical application areas of microphone array is sound source localization (SSL), which refers to identify the speaker of interest using a microphone array. SSL can be used as a preprocessing technique to boost up the entire system efficiency. Recent studies show that smartphones can be an efficient assistive device for hearing aid devices because of smartphones' powerful hardware and software components. Also, Deep Learning (DL) has shown a considerable performance increase in audio signal processing. DL based SSL using the direction of arrival estimation (DOA) methods for two and eight microphone array structures and the distance estimation methods using a single microphone are proposed in this work. The performance of the proposed methods are evaluated in several realistic noisy conditions, reverberations using real-recorded data. Another contribution of this work is to present real-time implementations of the DL based methods on edges devices, i.e., smartphones, tablets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Far-field automatic speech recognition (ASR) is one of the popular areas in research and consumer applications. Far-field audio and speech processing have been developing with the demand of voice-controlled systems in automotive, home automation, and personal assistant devices. Along with these developments, microphone array gains high popularity in signal processing in recent years. Various structures of microphone array such as circular, rectangular, linear haven been used in home automation, personal assistant devices. One of the important application areas of microphone array is speech source localization (SSL). SSL can be used as a preprocessing step for various methods such as beamforming [2], speech enhancement [3], and speech/speaker recognition [4]. One of the remarkable application areas of SSL is hearing aid devices (HAD). It is reported that the people who have severe hearing impairment have difficulty identifying speaker direction [5]. Therefore, 'visual indication' would be very useful for people with hearing impairment. Moreover, SSL enables to enhance the signal to noise ratio (SNR). On the other hand, the wide usage of smartphones makes them a good candidate for the 'visual indication' platform. Furthermore, it has been shown that smartphones are capable of audio signal processing [6, 7, 8, 9, 10] since they have powerful hardware and software components.

The popularity of machine learning (ML)/deep learning (DL) has been drastically increased with the developments in different kind of processing units, i.e., Central Processing Unit (CPU), Graphical Processing Unit (GPU), Tensor Processing Unit (TPU). DL offers effective solutions to complex and nonlinear problems in computer vision [11], speech recognition [12], speech enhancement [13]. Implementing DL approaches to SSL would be a powerful solution, especially when the number of microphones is few and under high noise conditions (low SNRs).

Real-time implementation of DL based methods is quite challenging because of two reasons. The first reason is that the performance of DL methods depends on the dataset. Hence, the dataset should involve various conditions in order to perform well in real-time, which is mostly unseen conditions to the DL model. The second reason is that inference using the DL model can be computationally expensive. Therefore, the resources of the edge devices might not be sufficient for real-time inference. These conditions should be considered to have a robust real-time DL based SSL method. In this work, we aim to develop an efficient DL based SSL method using the direction of arrival estimation (DOA) and speaker distance estimation, which can run on a smartphone/tablet in real-time. We also offer a method for distance estimation between a loudspeaker and microphone in this work.

1.2 Speech Source Localization using Direction of arrival

Speech source localization (SSL) using the direction of arrival (DOA) identifies speaker in interest using a fixed microphone array exploiting spatial clues. As mentioned in the previous section, DOA is a powerful method that enables increasing the signal to noise ratio (SNR), dereverberation, suppression of background noise, and speech enhancement with high perceptual quality [3, 14, 15, 16]. Noise types, SNR, reverberation, number and geometry of microphones directly affect the performance of DOA angle estimation. The dissertation aims to offer deep learning (DL) based DOA angle estimation method for single speech sources that can work in real-time using two or eight circular microphone array.

Since real-time implementation is one of our considerations, the proposed DOA angle estimation method should perform satisfactorily under different noise types and SNRs with low computational complexity. Different DL methods for DOA angle estimation are investigated to have satisfactory performance and low computational complexity. To define a method as real-time, a short frame of speech data (typically 20-40 ms) is required. Processing a short frame of speech data gains tracking ability to the method.

1.3 Speech Source Localization using Distance Estimation

The distance estimation of the speech source is another aspect of SSL. The distance estimation using a high number of microphones, i.e., more than four microphones, is well investigated in the literature. However, speech source distance estimation via fewer microphones has been drawn less attention by the researchers. We have proposed two different distance estimation methods using a single microphone for two different purposes in this work. The first method aims to estimate the distance between a loudspeaker and a microphone to establish an acoustic network using multiple smartphones. The second method targets to estimate the distance between a human talker and a microphone using a single microphone. Single microphone talker distance estimation can be utilized for a distributed ambient telephony system [17], smart home [18] by selecting the nearest microphone, human-robot interaction system [19], intelligent hearing aids [20] by modifying gain based on distance. As the source type and application areas are different for the first and second methods, we have developed two different methods. The real-time implementations of these two methods are also one of our consideration in this work.

1.4 Deep Learning approach for SSL

Deep learning (DL) has gained popularity in various areas such as computer vision [11], speech/speaker recognition[12], speech enhancement[13], audio signal processing [21] since DL offers practical solutions for complicated problems. DL could be an effective solution with reasonable computational complexity for DOA azimuth estimation and speaker distance estimation (SDE), especially for fewer microphones, i.e., two/three microphones for DOA and a single microphone for SDE. These shortcomings lead us to investigate DL methods for DOA angle estimations and SDE. Three steps are followed to have satisfactory performance and low computational complexity. The first step is data acquisition for application-based,i.e.,

single microphone for SDE, two microphones based, or eight circular microphone array for DOA azimuth estimation. The second step is DL model training using various network types, architectures, and hyperparameter tuning in order to have satisfactory performance, even low SNRs and low computational complexity. The last step is a real-time implementation of the DL based DOA angle estimation methods and SDE on edge devices, i.e., smartphones and raspberry pi with eight microphone board.

1.5 Real-time Implementation

One of the dissertation targets is to obtain a real-time implementation of DL based DOA angle estimation and SDE methods on edge devices such as smartphones and raspberry pi with circular eight microphone board. There are some constraints for real-time implementation, such as sampling rate, frame length, power of the processor, battery life, memory size. After obtaining the best model for DOA angle estimation and SDE, Tensorflow Mobile, and Tensorflow Lite [22] are utilized for real-time implementation. An Android-based smartphone is used for realizing the real-time implementation of the two microphones based DOA method as iPhone smartphones do not allow accessing two microphone inputs simultaneously. On the other hand, any smartphones/tablets can be used for SDE and distance estimation method between a loudspeaker and microphone since they are single microphone based. Raspberry pi with circular eight microphones is utilized for real-time implementation of eight microphones based DOA azimuth estimation method.

1.6 Outline of Dissertation

The outline of the dissertation is as follows:

- DL based DOA angle estimation methods for two microphones and their real-time implementation are proposed in Chapter 2.

- Convolutional neural network based DOA angle estimation method for circular eight microphones is defined in Chapter 3.
- Chapter 4 describes a noise-robust method for estimating the distance between a loudspeaker and a single microphone.
- Chapter 5 presents a single microphone based speaker distance estimation method.
- The work is concluded in Chapter 6.

CHAPTER 2

DEEP LEARNING BASED DIRECTION OF ARRIVAL ESTIMATION METHODS USING TWO MICROPHONES

Authors – Abdullah Küçük, Issa Panahi

The Department of Electrical and Computer Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

2.1 Abstract

Deep learning (DL) techniques are gaining popularity due to performance boost in many applications. In this work we propose a DL-based (deep neural network (DNN), convolutional neural network (CNN), and convolutional recurrent neural network (CRNN)), methods for finding the direction of arrival (DOA) of speech source for hearing study improvement and hearing aid applications using popular smartphone with no external components as a cost-effective stand-alone platform. We consider the DOA estimation as a classification problem and use the magnitude/phase of speech signal or real/imaginary part of short time Fourier transform of speech signal as a feature set for any DL method training stage and obtaining appropriate model. The model is trained and derived using real speech and real noisy speech data recorded on smartphone in different noisy environments under low signal to noise ratios (SNRs). The DL-based DOA method with the pre-trained model is implemented and run on Android smartphone in real time. The performance of proposed method is evaluated objectively and subjectively in the both training and unseen environments. The test results are presented showing the superior performance of proposed method over conventional methods.

2.2 Introduction

Direction of arrival (DOA) estimation for speech source localization (SSL) is an important area in signal processing. DOA estimation is a key preprocessing step for a wide variety of audio signal processing algorithms. Examples of audio signal processing algorithms that require a prior DOA estimation step include beamforming [2, 14], speech/talker recognition [4, 23], speech enhancement [3, 24] for hearing aid [25, 26]. DOA for hearing aid applications is the focus of this work. In this work we offer a "visual indicator" for hearing impaired (HI) people as a means to improve their spatial awareness. Since HI people, especially elderly HI

people, have difficulty in identifying the direction of the speaker [5] in a group conversation, therefore the offered 'visual indicator' can be useful to HI people in social events.

We propose using mobile devices, such as smartphones and tablets, as a visual indicator of speech source direction for HI people. Apart from a touchscreen display, mobile devices feature a powerful processor, memory, and at least two built-in microphones needing no additional external component, making them suitable for real-time audio signal processing applications such as DOA estimation[6, 27], speech enhancement[8, 28], audio compression[29], and adaptive feedback cancellation[10]. Additionally, mobile devices are widely available to most people, hence making them a cost-effective solution.

In this work, we propose a DL based SSL methods which are trained in noisy and reverberant environments using real data recorded on a smartphone. Our first contribution is to propose a new feature set for the DL models for SSL as used in this paper. Instead of explicit hand-crafted features from the microphone data (such as those in GCC-PHAT [30] or SRP-PHAT [31]), we use the STFT of the speech signal per microphone. Our second contribution is the efficient real-time implementation of DL-based SSL on Android smartphone using its built-in two microphones for hearing aid applications. Performance of the proposed DL-based SSL algorithm along with the implementation details on Android-based smartphones are analyzed in this work. The proposed method is noise-robust and can work with only two smartphones' microphones in different kinds of unseen noisy environments with low latency and very little computation footprint. The real-time video demo of the Android application in a noisy environment is available in the project website [32]. To reduce computation complexity, we use the real and imaginary part of the raw speech STFT instead of hand-crafted features. Raw STFT features are easier to compute and the CNN or CRNN might learn better features than hand-crafted features. Speech presence is detected using a real-time simple voice activity detector (VAD) developed in [27]. The details about VAD will be given in the following sections. Incoming data frames from two microphones

are processed by the VAD. If VAD decides that frame is speech then inferring process is performed using our pre-trained DL models to estimate the DOA. If the incoming frame is detected as noise, then the previously estimated DOA estimate will be used.

2.3 Conventional DOA Estimation Methods

SSL using DOA has been investigated, and various methods have been developed over the years. Conventional DOA methods can be classified into three classes:

- Methods that decompose the autocorrelation matrix into signal and noise subspace like multiple signal classification (MUSIC) [33]. MUSIC methods can achieve high performance with a large number of microphones.

- Methods that exploit the time difference of signal arrivals like TDOA methods [34]. TDOA methods estimate inter-time difference (ITD) between two microphones and convert ITDs to an angle using apriori information such as distance between microphones, sampling frequency, and sound speed in the medium. One of the popular TDOA methods is the generalized cross-correlation (GCC) with pre/post filtering [6]. TDOA methods perform well for high SNRs and have low computational complexity. However, the performance of TDOA methods is not satisfactory for low SNRs.

- Methods that compute steered response power for estimating DOA like SRP-PHAT [35]. SRP-PHAT computes the power of incoming signals using phase transform and maximizes it using grid search. Although the performance of SRP-PHAT is high with a large number of microphones, it requires high computational power.

2.4 Deep Learning based DOA algorithms

Many researchers have investigated various DL approaches on SSL or DOA for the last five years. We can categorize these approaches into two classes. While the first-class utilizes DL

as a preprocessing step for the DOA, the second group approach uses DL as a replacement of the DOA algorithm.

The works in [31, 36, 37] are examples of the first class. The authors of [31] propose a deep neural network based (DNN) based time-frequency masking in order to filter out non-speech components from the signal. The feature set, the generalized cross-correlation - phase transform (GCC-PHAT), is extracted from the filtered signal. They also show that the proposed method in [31] outperforms using direct GCC PHAT as a feature set for time difference of arrival estimation. One of the recent work [36] proposes DNN architecture for phase enhancement for SSL. The authors of [36] enhance inter-channel phase differences (IPDs) and estimate DOA for each frequency. The DOA is estimated through k-means clustering. The method in [36] has superior performance than conventional methods in [38, 39].

The work in [37] focuses on eliminating regions dominated by noise and reverberation from the signal like [31] . The authors offer a convolutional neural network (CNN) based time-frequency masking method to remove the corrupted region. The filtered signal is fed one of the conventional methods, SRP-PHAT.

The methods in [30, 40, 41, 42, 43, 44] are examples of the second class of DL based DOA methods. The authors of [40] define DOA as a classification problem and utilize a multi-layer perceptron (MLP) as their DL architecture. The architecture consists of an input layer, a hidden layer with a sigmoid activation function, and an output layer with a softmax activation function. The feature set of the method in [40] is GCC-PHAT coefficients. The work in [40] benchmarks the proposed method with state of the art least-square DOA method for eight microphone case.

Interesting work is proposed in [30]. The authors of [30] use synthesized white noise rather than speech signals for their training. Using synthesized white noise has two main advantages. Firstly, the authors of [30] do not need any speech database. Secondly, the

labeling of training data is simple since they do not need to use a voice activity detector (VAD) to remove the silence part of the speech. The phase of the short-time Fourier transform (STFT) is utilized as a feature set. The CNN architecture used as a classifier for DOA estimation. CNN involves three convolutional layers with sixty-four kernels, two fully connected layers with five hundred twelve hidden layers, and an output layer with a softmax activation function. The proposed method in [30] outperforms the SRP-PHAT method for four and eight microphones case.

The work in [41] proposes two different sound localization approaches using the raw waveform of the binaural signal. The first method is an entirely data-driven method using a convolutional neural network (CNN), and the second approach utilizes a gammatone filter for feature extraction. The CNN based method has better the method with the gammatone filter [41]. The CNN includes four convolutional layers, three max-pooling layers, two fully connected layers, an output layer.

The application of CNN to the minimum variance distortionless response (MVDR) scheme is explained in [42]. The authors of [42] extend hybrid beamforming with CNN. The method in [42] uses steered response power (SRP), MVDR, and CNN for acoustic source localization. CNN is utilized in two variants, while the first type uses CNN as a classifier; the second one utilizes as a regressor. The CNN classifier has mostly superior performance than the regressor CNN for uniformly linear eight microphone array[42].

The authors of [45] propose a probabilistic neural network (PNN) for DOA estimation. They aim to solve indoor DOA estimation problem under high reverberation and low signal to noise ratio (SNR). The generalized cross-correlation (GCC) coefficients are used as feature set for PNN. The authors of [45] report 4.6° root mean square error (RMSE) for a high reverberant environment, 600 ms, with low SNR,-10 dB using six microphones.

The hybrid DL method is used for DOA estimation in [44]. The authors combine CNN with long short term memory (LSTM) in [44]. The proposed architecture in [44] involves

three convolutional layers with sixteen filters, one LSTM layer with three hundred nodes, a fully connected layer with one thousand twenty-four nodes, and an output layer. The GCC-PHAT coefficients are utilized as a feature set for the proposed architecture. The authors report that the method in [44] outperforms the method in [30].

2.5 Shortcomings of previous works in DL based DOA estimation

The methods are mentioned in section 2.4 have promising performance in different environments with/out the presence of background noise. However, more investigations are required because of the following reasons:

- Most of the works in section 2.4 train and validate the methods using simulated data. Real-recorded data is required for showing the performance of a proposed method in real-life situations.

- The works in section 2.4 use at least four microphones for evaluating the proposed methods. Only [41] proposed methods for binaural signal (two-channel signal). However, the binaural signal is generated by the head related transfer function (HRTF) to take into account of effect of head (especially ear) shape. Hence, a DL based two-channel (without HRTF) DOA angle estimation method should be investigated.

- Some of the works in section 2.4 are assessed their proposed methods under noisy conditions. However, the authors in section 2.4 mostly used additive white noise rather than more realistic noise types like babble, machinery noise. More research is needed for obtaining the effects of more realistic noise types on DOA angle estimation.

- To our best knowledge, none of the works mentioned in section 2.4 proposes a real-time application of the proposed method on an edge device. Real-time implementation is an indicator of two points: (i) the computational cost of the proposed method is sufficient

for real-time implementation on an edge device (ii) the proposed model is generalized for any unseen environment.

The reasons given above lead us to work on DL based DOA angle methods and real-time implementation on edge devices. Two and eight microphone based methods are offered. The detail regarding two and eight microphones DL based DOA angle estimation method is explained in detail in chapter 3 and 4, respectively.

2.6 Problem Statement

The target is real-time implementation of the proposed method on Android platforms using its built-in two microphones. In a realistic scenario, there is a limited number of human speakers. Therefore, the DOA angle estimation problem formulated as a classification problem instead of a regression problem. The DOA angle resolution is defined as 20° within the range of 0° to 180°. Then data is recorded and simulated based on 20° resolution. The real recorded data has critical importance for real-time implementation. Since the microphones of smartphones are designed for different purposes (such as for voice call or noise cancellation), they have different characteristics. The real recorded data will help to train the DL models in order to implicitly compensate the different characteristics of the microphones. This is one of the remarkable advantages of using the DL method for DOA angle estimation.

2.7 Proposed Methods

We propose DNN, CNN, and CRNN based direction of arrival estimation for two microphones. We give details about the proposed methods in this section.

2.7.1 Deep Neural Network based DOA Angle Estimation

Input Feature Set

The input feature set has critical significance for a DL application. The feature set should include the required information for the applied area. The book [3] states that phase and signal energy are used for SSL for humans. Therefore, we have decided to utilize magnitude and phase as a feature set for the proposed DOA angle estimation method. A 20 ms incoming frame is multiplied with the Hanning window. Then, we calculate FFT of the windowed signal.

$$X_{ch}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{2.1}$$

$X_{ch}(m,\omega)$ is a complex short time Fourier transform (STFT). $x[n]$ is the input signal and $w[n-m]$ is a window to get small frames of the signal. The sequence $f_m = x[n]w[n-m]$ is a short-time section of the speech signal $x[n]$ at time m. The subscript ch denotes the channel number. Next, we find the magnitude and the phase.

$$X_{ch}(m,\omega) = \sqrt{Re(X_{ch}(m,\omega))^2 + Im(X_{ch}(m,\omega))^2} \tag{2.2}$$

$$\angle X_{ch}(m,\omega) = \tan^{-1}\Big(\frac{Im(X_{ch}(m,\omega))}{Re(X_{ch}(m,\omega))}\Big) \tag{2.3}$$

$X_{ch}(m,\omega)$ and $\angle X_{ch}(m,\omega)$ represents magnitude and phase of channel of ch, respectively. $Re(X_{ch}(m,\omega))$ and $Im(X_{ch}(m,\omega))$ denotes real and imaginary part of the STFT for channel ch. The feature set should look like below.

Feature set

$$\Big[Mag.\,of\,1^{st}\,channel\,|\,Phase\,of\,1^{st}\,channel\,|\,Mag.\,of\,2^{nd}\,channel\,|\,Phase\,of\,2^{nd}\,channel\Big] \tag{2.4}$$

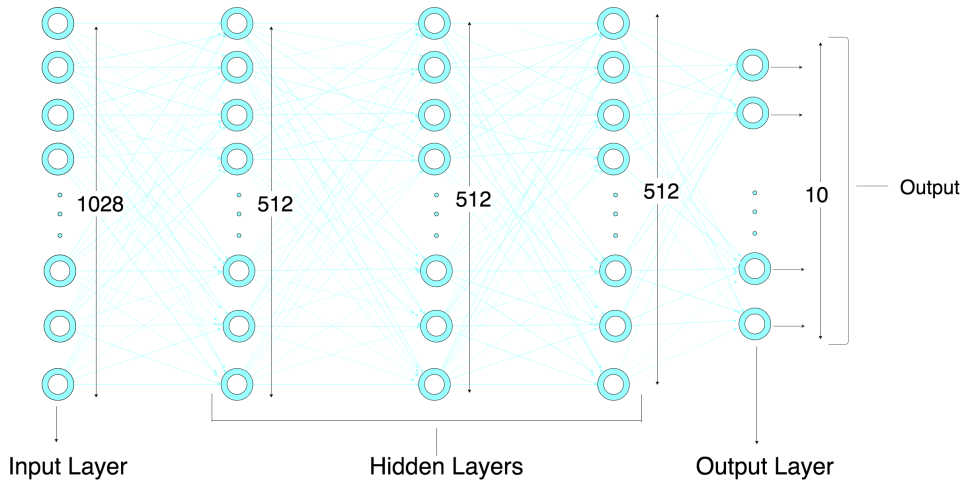


Figure 2.1: Block Diagram of the proposed DNN architecture of DOA angle estimation for two microphones

The feature set's dimension is $1 \times (4 \times (\frac{N_{FFT}}{2} + 1))$. N_{FFT} is equal to number of fast Fourier transform (FFT). Since FFT of a signal is symmetric in the frequency domain, we only use half of it.

Model Architecture

Figure 2.1 shows the proposed DNN architecture for two microphones DOA angle estimation. Various network architectures are investigated in order to come up with the proposed one. The DNN consists of an input layer, three hidden layers, and an output layer. The input layer has 1028 nodes as the NFFT considered in the proposed method was 512. Each hidden layer has 512 nodes with a sigmoid activation function. Since we have 10 classes (20° resolution within the range of 0° to 180°), the output layer has 10 nodes with a softmax activation function. We also use the dropout rate as 0.5 between the hidden layers to avoid overfitting.

2.7.2 Convolutional Neural Network based DOA Angle Estimation

Input Feature Set

Determining input feature-set is crucial for supervised learning. Input feature set should have enough DOA information to be learned by the CNN. In [30], the phase of the signal is considered as an input feature set. [3] states that signal energy is also used for speech source localization for humans. Therefore, we have decided to use real and imaginary part of short time Fourier transform (STFT) of input speech frames which is defined by

$$X_{ch}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{2.5}$$

$X_{ch}(m,\omega)$ is a complex STFT. $x[n]$ is the input signal and $w[n-m]$ is a window to get small frames of the signal. The sequence $f_m = x[n]w[n-m]$ is a short-time section of the speech signal $x[n]$ at time m. The subscript ch denotes the channel number. The real and imaginary part of $X(m,\omega)$ is used as an input feature set for the proposed method. The following matrixes show input feature sets.

$$\begin{matrix} \textbf{Feature set 1} \\ \begin{bmatrix} Imag.\,part\,of\,X_1(m,\omega) \\ Imag.\,part\,of\,X_2(m,\omega) \end{bmatrix} \end{matrix} \quad \begin{matrix} \textbf{Feature set 2} \\ \begin{bmatrix} Real\,part\,of\,X_1(m,\omega) \\ Real\,part\,of\,X_2(m,\omega) \end{bmatrix} \end{matrix} \tag{2.6}$$

The dimension of each feature set is represented by $N_R \times N_C$. We have a total of two input channels hence N_R equals 2. N_C equals to $N_{FFT}/2 + 1$. N_{FFT} is the number of the fast Fourier transform (FFT) points. Since FFT of a signal mirrors in the frequency domain, we get half of the frequency response of data.

Model Architecture

Figure 2.2 shows CNN topology of the proposed method. There are five main layers in the topology viz. input layer, convolutional layer, pooling layer, fully connected (FC) layer,

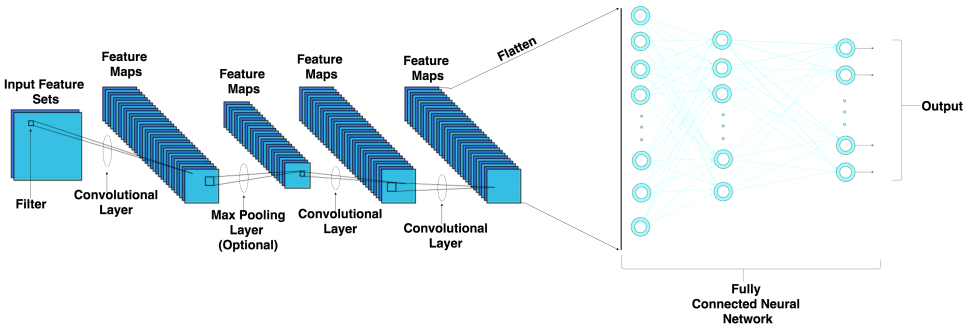


Figure 2.2: Block Diagram of CNN based SSL/DOA angle estimation architecture. The CNN consists 3 components. Input image is first convolved with kernels. The final convolutional layer obtained by stride of 2 is flattened and fed into fully connected layer. Finally, output is fed to the softmax layer.

output layer. The input layer consists of input feature sets defined in the previous section. We have two sets of matrices that consist of the real and imaginary part of STFT of the speech signal. Inputs are processed by convolutional layer. A set of kernels, i.e. filters, is convolved with small parts of input matrices in the convolutional layer. These kernels enable us to find out local information on the spectrum for SSL. After the application of filters, feature maps are generated by each convolutional layer. Pooling layer helps us to reduce the feature map resolution for decreasing computational complexity. In this work, we have used max pooling which takes maximum values in the 2×2 matrix. After max pooling layer, feature maps are flattened and fed into fully connected (FC) layer. FC layer performs classification using activation function. For this study, Rectified Linear Unit (ReLU) [46] is used as an activation function in FC layers. Softmax activation function (gives a probability of each class) is utilized in the output layer. The highest probability is selected as the output class which can be shown as:

$$\hat{\theta}_i = \underset{x}{\operatorname{argmax}}\{p(\theta_c \mid \phi_i)\} \tag{2.7}$$

The angle $\hat{\theta}_i$ denotes estimated DOA angle, $p(\theta_c \mid \phi_i)$ is the probability of c-th class when given the i-th time frame as ϕ_i. The CNN architecture used in this work includes three

convolutional layers. Each convolutional layer has 64 filters with 2×2 strides. There is one optional max pooling layer after each convolutional layer. First FC layer has 512 nodes with ReLU activation function. The second one has 256 nodes with ReLU. We have 10 output classes with Softmax activation layer in the output layer.

2.7.3 Convolutional Recurrent Neural Network based DOA Angle Estimation

Input Feature Set

The input feature set for convolutional recurrent neural network (CRNN) based direction of arrival (DOA) estimation is same with convolutional neural network (CNN) based DOA. Hence, the details can be found in section 2.7.2.

Model Architecture

The proposed architecture is shown in Figure 2.3. We have chosen this architecture after many trials. The model has an input layer, three convolutional layers, two max-pooling layers, one recurrent neural network (RNN) layer, one fully connected layer (FCL), and an output layer in order. There are 64 filters, each with a dimension of 2×2 in every convolutional layer. These filters (kernels) extract the required information for DOA estimation. The max-pooling layer is utilized in order to decrease computational complexity. In our model, the max-pooling layer takes max value in the 2×2 matrix. Then the max-pooling output is flattened and fed to the RNN layer. Since speech is continuous, RNN leads the model to learn context information using previous information. Rectified linear unit [46] is utilized as activation function in FCL. SoftMax is used in the output layer, which has ten nodes since we have ten classes, i.e. within 0° to 180°. We also use the dropout rate as 0.5 between RNN and FCL layers to avoid overfitting.

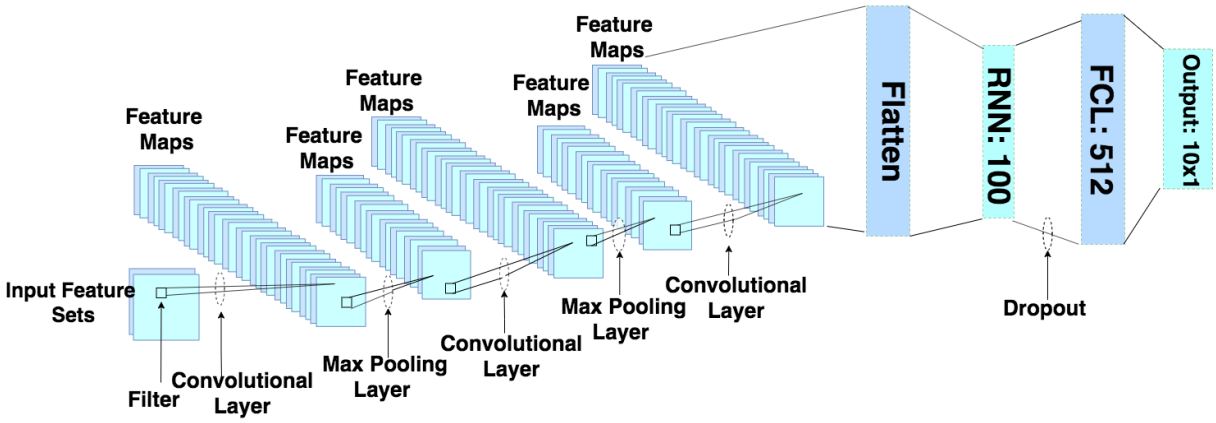


Figure 2.3: Block Diagram of the proposed CRNN architecture for DOA angle estimation

2.8 Voice Activity Detector

In real life, we are exposed to different kind of noises. The presence of background noise leads to performance degradation for DOA estimation. We would like to see the performance of the proposed method under noisy condition. Since we train the model with only speech, the proposed method requires classifier which labels speech segment (includes clean or noisy speech) and noise (or silence) segment. This classifier is called voice activity detector (VAD). If the incoming frame is detected as speech, the frame is fed to pre-trained model making a new estimation, else, the DOA angle estimate from the previous frame is retained:

$$\hat{\hat{\theta}}_i = \begin{cases} \hat{\theta}_{i-1},\ if\ VAD = 0\ (Noise) \\ \hat{\theta}_i,\ if\ VAD = 1\ (Speech) \end{cases} \tag{2.8}$$

Where angle $\hat{\hat{\theta}}_i$ denotes the corrected DOA estimate after VAD for i^{th} frame. Thus, the VAD prevents model to estimate the DOA angle with noise-only frames. In this work, to reduce the real-time processing burden and to reduce CNN size, we utilize a simple single-feature based VAD. That is, 'Spectral Flux (SF)' is used as a feature for VAD [27]. The definition of SF is given by:

$$SF(k,i) = \frac{1}{N} \sum_{k=-\infty}^{\infty} \left(\left| X_i[k] \right| - \left| X_{i-1}[k] \right| \right)^2 \tag{2.9}$$

for k^{th} frequency bin and i^{th} frame. $k = 1, 2, \ldots, N$. $|X|$ denotes the magnitude spectrum of X. A simple thresholding technique is given by:

$$VAD(i) = \begin{cases} 0\ (Noise),\ if\ SF(k,i) < \xi \\ 1\ (Speech),\ if\ SF(k,i) \geq \xi \end{cases} \tag{2.10}$$

where ξ is the calibration threshold.

We have used SF based VAD because it is easy to implement, and it has satisfactory robustness under stationary conditions [27]. In order to make our app robust under non-stationary noise, two parameters are defined for the VAD. The first parameter is decision buffer, D, which makes a VAD decision when D contiguous frames are detected as speech. The second parameter is called threshold, T, which determines initially how many frames is assumed as noise.

Following are the steps in the proposed DOA angle estimation method for efficient real-time implementation:

- Pre-filtering using SF-based VAD to label the incoming frame as noise or speech;
- If the incoming frame is detected as speech, STFT of the incoming frame is fed to DL method;
- Inference is done using the pre-trained DL model.

2.9 Experimental Setup and Evaluation

We present the experimental setup and evaluation for the proposed DL based SSL/DOA methods in this section. The performance metrics and experimental setup for simulated and smartphone recorded data are also explained in this section.

2.9.1 Dataset

Simulated Dataset

Five independent rooms with dimensions as shown in Table 2.1 were simulated using image-source model (ISM) method. A microphone array of 2 mics was placed at the center of each room with 13 cm as the distance between them. We choose 13 cm distance between the microphones as it emulates the two microphones on the smartphone we used for real-time recording. We have considered 10 different DOA angles between 0° to 180 °, with 20° as resolution for recording the data. Simulated recording was generated by convolving the source speech signal with room impulse responses (RIRs). RIRs were generated using ISM method according to different DOA angles. The distance of source varies in each simulated room and is given as per Table 2.1. The noisy data to be played by source was prepared by contaminating clean speech files from HINT[47] and TIMIT[48] databases with collected day-to-day noises at different signal to noise ratios (SNR). We have considered machinery (MRI noise, M), traffic (T), and babble (B) noises with 0, 5 and 10 dB SNR.

Table 2.1: Simulated Data summary

Simulated Data Details	
Room Size	*Room 1 (7,4,2.5)m, Room 2 (6.5, 5.5,3)m, Room 3 (5, 4.5,3) m , Room 4 (7, 5.5,3.5)m, Room 5 (6.2, 5,3)*
Array positions in room	*Center of all rooms*
Source-array distance	*1 m for Room 1, 1.3 m for Room 2, 0.92 m for Room 3, 2 m for Room 4 and 1.05 m for Room 5*
RT_{60}	*400, 350, 300, 400, 300 milliseconds (ms) for Room 1, 2, 3, 4 and 5 respectively*
Sampling Frequency	*16 kHz*

Smartphone Recorded Dataset

Since our goal is to implement the proposed method on the smartphone for people's hearing improvement, we need real smartphone recorded data for training. Training the model on real recorded data makes it more robust to real-life noise and reverberation. Hence, real data is collected using Pixel 1 smartphone to get the model better trained for our applications. The data is recorded in three different rooms. In all rooms and around the smartphone on the table, we place five loudspeakers apart from each other so that the resolution is 20°. Since the DOA angle range is between [0° to 180°] we rotate the smartphone by 90° for this setup to capture another five loudspeaker signals. This way we have a total of ten loudspeakers per setup in each room (shown in Figure 2.4). We have 2 different setups for room 1. The distance between smartphone and loudspeakers for setup 1 and 2 in room 1 are 0.6 m and 2.4 m, respectively. The distance between smartphone and loudspeakers for room 2, 3, and 4 are 1.3 , 0.92 and 1.05 meters, respectively. The dimensions for room 1, 2, 3, and 4 are

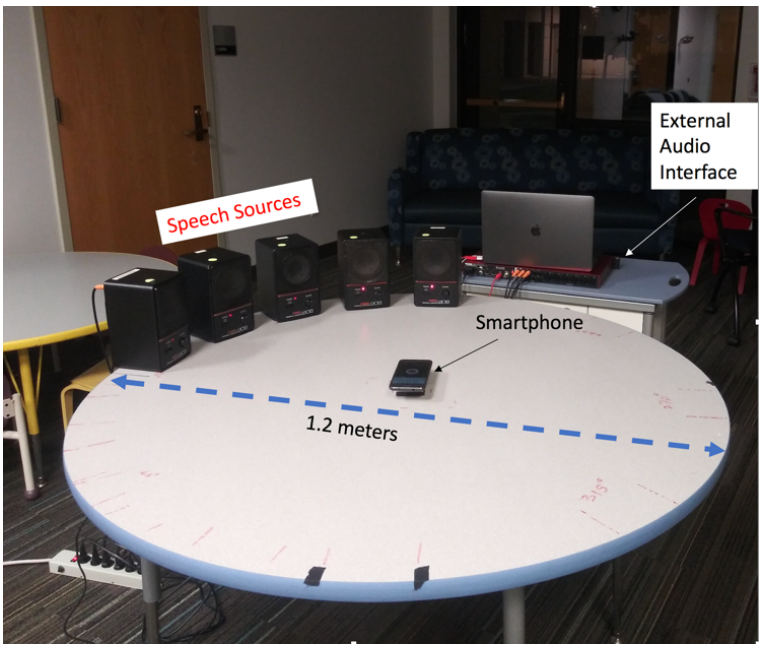


Figure 2.4: One of the setup of data collection used to train the proposed DL-based SSL/DOA estimation method

Table 2.2: Data collection summary

Data Collection Details	
Room Size	*Room 1 (7,4,2.5)m, Room 2 (6.5, 5.5,3)m, Room 3 (5, 4.5,3) m , Room 4 (7, 5.5,3.5)m*
Array positions in room	*2 different location in Room 1*
Source-array distance	*0.6 and 2.4 m for Room 1, 1.3 m for Room 2, 0.92 m for Room 3, and 1.05 m for Room 4*
RT_{60}	*400, 350, 300, 400 ms for Room 1, 2, 3, and 4 respectively*
Sampling. Frequency	*48 kHz*

$7m \times 4m \times 2.5m$, and $6.5m \times 5.5m \times 3m$, and $5m \times 4.5m \times 3m$, and $7m \times 5.5m \times 3.5m$ respectively. Room 4 is only used for testing the real-time Android application as an unseen environment. Reverberation times vary between 300-400ms between the Rooms 1, 2, 3 and 4. Table 2.2 shows the configuration of data recording. Figure 2.4 displays one setup of the data collection in Room 1. For recording, we have used clean speech files from HINT, TIMIT, and LibriVox [49]. The speech files from all three databases are mixed together randomly which would make the data more diverse and realistic. Female and male speakers/speeches are chosen from the speech databases almost equally. 35 minutes recording is done by smartphone for two setups in room 1. And, smartphone recorded 15 minutes for room 2 and 3. Noise files played from a loudspeaker and recorded separately in room 2. The loudspeaker, plays noise files, is located 3.5 meters away from the smartphone (not shown in Figure 2.4). Pixel 1 smartphone is used for separately recording noise files. Noise files were chosen from the DCASE 2017 challenge database [50] which includes real recordings. Vehicle ('Traffic') noise and multi-talker babble ('Babble') noises are selected from the DCASE 2017 database. In the dataset, we also used actual noise in a Magnetic Resonance Imaging (MRI) room with a 3-Tesla imaging system [51] to incorporate a machinery noise ('Machinery') containing strong periodic component. To generate noisy mixtures, clean speech and noise files are mixed at different SNRs levels.

2.9.2 Experimental Setup

The proposed DOA angle estimation methods are evaluated using simulated and real recorded data that explained in previous section. We have generated feature set in MATLAB and transferred to the Python. The magnitude and phase of the speech signal form our feature set for DNN, and the images, which fed to CNN and CRNN, are formed using real and imaginary parts of STFT. The input feature sets for recorded and simulated data are extracted from 20 ms frames, at the sampling frequency of 48, 16 kHz, respectively. Hanning window is applied to each frame and FFT is calculated. Based on the voice activity detector (VAD), each frame is labeled as a silence or speech frame. More information regarding VAD can be found in section 2.8. Then, the magnitude and phase of the speech frames are calculated and reshaped as in (2.4) for DNN, the real and imaginary part of STFT is reshaped like in equation (2.6) for CNN and CRNN. The number of FFT points is 1024, 512 for real recorded and simulated data, respectively. $[number\,of frames \times 1029]$ is the dimension of the feature set with including DOA label (angle information) for DNN training. The dimension of the input feature becomes $[number\,of frames \times 2053]$ with including the label for CNN and CRNN training. We use TensorFlow [22] for training model since it has C/C++ API for Android platforms. The trained model then is used for evaluating the proposed methods. The experiments are conducted using clean speech signal (no presence of background noise) and three different background noise types; multi-talker babble, machinery, traffic at three different SNRs (0, 5, 10 dB).

2.9.3 Performance Metrics

Accuracy (ACC) and root mean square error (RMSE) are utilized for quantifying the performance of the DOA angle estimation.

$$Accuracy(ACC\%) = \frac{N_c}{N_F} \times 100 \tag{2.11}$$

where N_c is the total number of correct DOA estimation and N_F denotes the total number of frames per test case.

$$RMSE = \sqrt{\frac{1}{N_F}\sum_{i=1}^{N_F}\left(\theta_i - \hat{\theta}_i\right)^2} \tag{2.12}$$

where N_F is the total number of frames per test case. θ_i is the true DOA angle and $\hat{\theta}_i$ is the estimated DOA angle for the i^{th} frame.

2.10 Results and Discussions

We first present results for our experiments using simulated data. Figure 2.5 shows a comparison of the proposed methods against other deep learning based DOA methods using simulated data. The first proposed method is deep neural network (DNN) based DOA angle estimation, denoted by Proposed DNN. Our second DOA developed method (The Proposed CNN-Speech Phase) uses phase information of speech as input for CNN, whose topology is the same as ours convolutional neural network (CNN) based DOA angle estimation method (The Proposed CNN). The last proposed method is convolutional recurrent neural (CRNN) network based DOA angle estimation method which is represented by Proposed CRNN, see Figure 2.5. The method from [40] is MLP based method which uses GCC Phat information as an input and is denoted as GCC MLP. The method from [30] utilizes synthesized white noise phase information in training. It uses the model for estimation on speech files and CNN-Noise Phase is used in [30]. The reason for including our second proposed method is to make a comparison fair with the method in [30] which uses white noise in the training part. C-LSTM denotes the method in [44]. These methods are compared when there is not presence of background noise, where 90% of data is utilized for training and the rest is for testing. As it is seen from Figure 2.5, C-LSTM and Proposed CNN and CRNN have accuracy performance more than 90%. The proposed CNN Speech Phase has almost 90% accurracy.

The proposed DNN method has nearly 85% accuracy performance using two microphone. However, the rest of the algorithms, GCC MLP and CNN Noise Phase, performs worst for two microphone case even though there is no presence of background noise. Since this is two microphone DOA estimation, the method in [40] and [30] might require more training data for better performance. The last observation from Figure 2.5 is method in [40] that performs better compared to the method in [30] even though [40] uses MLP with input sets of GCC-PHAT.

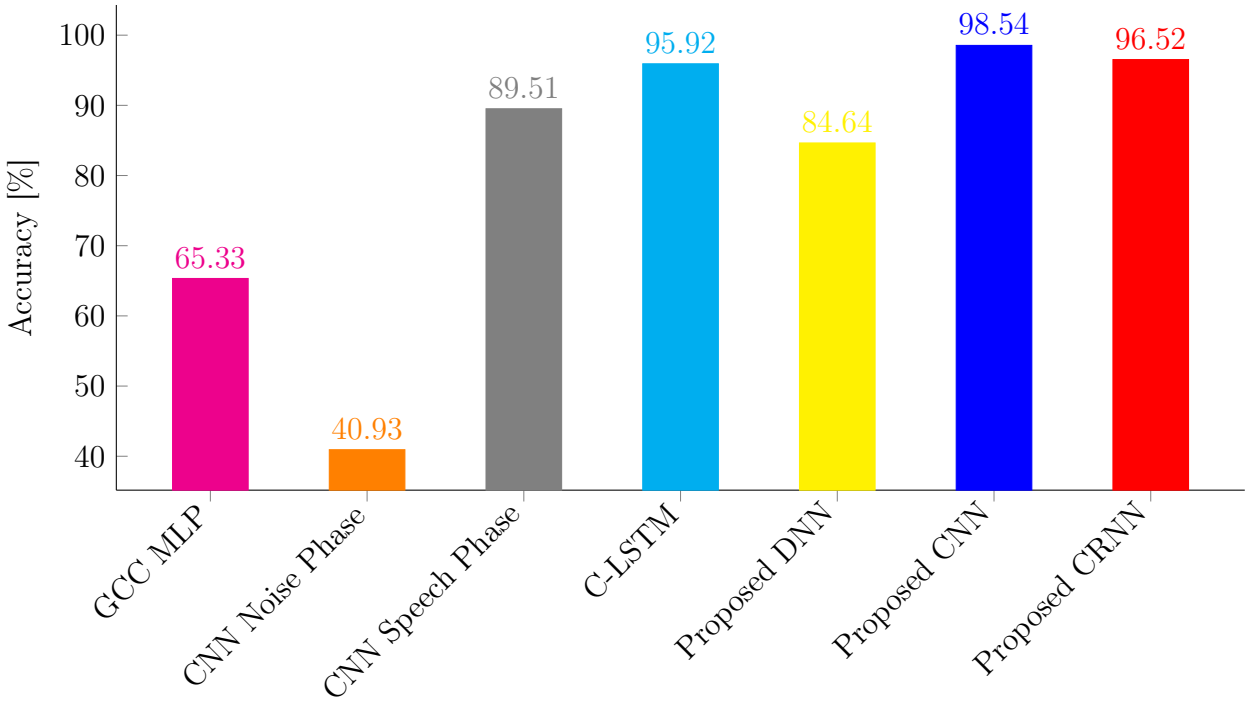


Figure 2.5: Accuracy comparison results for the silent environment using simulated data

Figure 2.6 shows the comparison of the proposed method CNN and CRNN versus method in [30] and in [44] for an unseen environment when there is not presence of background noise with smartphone recorded data. The training data for the two methods of Figure 2.6 is generated using data from Room 1 and 2.We have two different setups for Room 1 as it was

mentioned in section 2.9.1. Room 3 data is used for generating test data. There are two cases for comparison: While case 1 uses one frame information for forming image to feed CNN or CRNN, case 2 uses ten consecutive frames. The accuracy of the proposed CNN method for case 1 and 2 is 73% and 81%, respectively- a relative improvement of about 8%. While the proposed CNN based DOA method performs best for case 1, the proposed CRNN based has highest accuracy for case 2. However, the performance of the method in [30] for case 1 and 2 is about 27% and 36%, respectively when using smartphone recorded data. A reason for such huge difference between accuracies shown in Figure 2.6 could be that [30] trains model using synthesized white noise while we have used speech for training in our method and in implementing the method of [30] for the comparison analysis. Another reason could be that [30] uses two identical microphones for data generation while in our comparison analysis between the two methods we have used the two built-in microphones of the smartphone which could have mismatch characteristics, e.g. different gains. On the other hand, C-LSTM [44] accuracy performance for Case 1 and 2 is 10.01% and 77.58%, respectively. The method in [44] is not capable of classifying for Case 1 because the C-LSTM model always estimates only one class. Since we have ten classes, the performance of C-LSTM is around 10% for Case 1. The multi-frame approach enables the C-LSTM method to perform better and increases its accuracy of DOA angle estimation to 77.58%. These results prove that the proposed CNN and CRNN methods has highest accuracy performance and are good candidates for real-time implementation

Figure 2.7 displays the CRNN-based DOA angle estimation accuracy of the proposed method under different noisy conditions. The results are collected for babble, traffic, and machinery noises at 0 dB, 5 dB, and 10 dB. As we mentioned at the beginning of this section, the test data is entirely different from the data collected and used for training. We have trained a model for each noise type and SNR level. The results are collected using the multi-frame approach, 10 frames for each DOA angle estimation. As the first observation

from Figure 2.7, the accuracy increases with an increase in SNR. The second observation is that the performance accuracy under babble noise is the lowest among the use of the other two noise types. Since babble noise contains multiple speeches, performance degradation is expected. The next observation is that the accuracy difference between all noise types is highest at 0 dB SNR. Low performance at 0 dB for all noise types can be explained with the low performance of the single feature based VAD. The performance could be increased using superior VAD. The performance differences decrease as the SNR increases. Moreover, the performance under machinery noise is higher than traffic noise at 0 dB. However, the accuracy performance under traffic noise is highest for 5 dB and 10 dB SNRs. After 10 dB SNR, the accuracy results become closer to clean speech condition (when there is no presence of background noise).

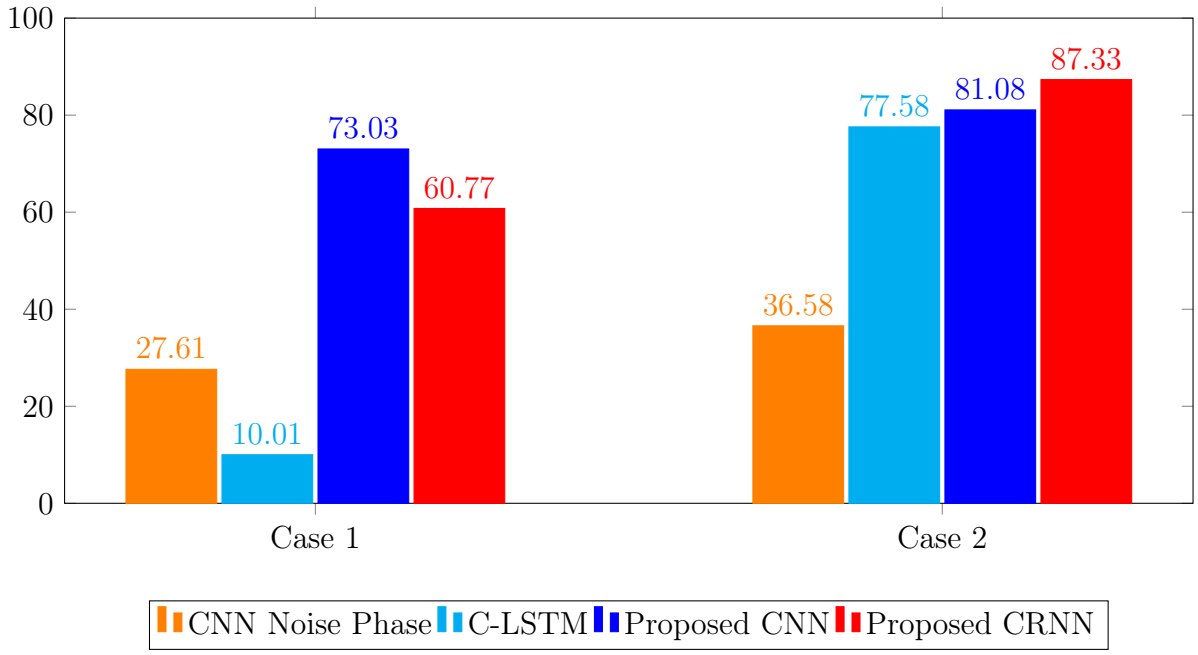


Figure 2.6: Comparison of accuracy results for Unseen Silent Environment

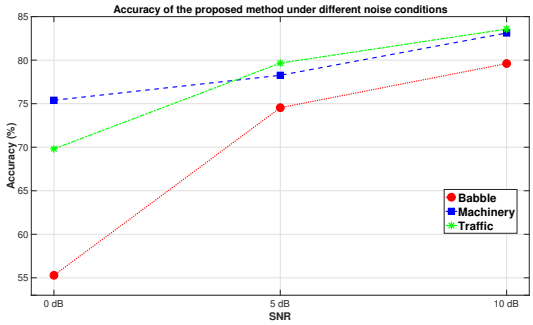


Figure 2.7: Accuracy performance of the proposed CRNN-based DOA angle estimation method under different noise types (babble, machinery, traffic) at 0dB, 5dB, and 10dB SNRs.

2.11 Real-time Implementation on Android based Smartphone

This section presents the real-time implementation of the proposed CNN and CRNN-based DOA angle estimation algorithm on the Android-based smartphone. We use Android operating system (OS) that allows us to access the two built-in microphones of the phone/tablet.

2.11.1 Offline Training

The model for CNN and CRNN based DOA angle estimation is trained offline using smartphone recorded data. Then, the pre-trained model is implemented on Android-based platforms. The data for training is generated on MATLAB using real recorded data. Frame size is considered as 20ms and the sampling frequency is 48 kHz. Each frame is multiplied by the Hanning window and STFT is calculated. Then, the real and imaginary part of STFT is reshaped like in (5.2). Since frame length is 20ms, the number of FFT point is selected as 1024 at 48 kHz. The dimension of the input feature becomes $[numberofframes \times 2053]$ with including label for training. TensorFlow [22] is utilized for training model. The benefit of using TensorFlow is it has C/C++ API for Android platforms. Half of the data is used for training and the other half is used for testing for model training. This makes the model robust to new incoming data and prevents overfitting since we use half of the data for testing.

2.11.2 Implementation

Figure 2.8 shows the block diagram of real-time processing modules of the proposed DOA estimation methods on Android platforms. We define real-time as estimating the talker direction in 250-275 ms when the frame size is 20 ms and 'Wait Buffer' capacity is 10 frames. The stereo input/output framework for audio signal processing [52] is employed for real-time implementation. The smartphone captures 20ms frames at 48 kHz. The captured frames are stored in 'Input Buffer'. Then each frame is multiplied by the Hanning window and FFT is calculated. Next, features are extracted for the VAD. A simple efficient spectral flux based VAD is employed in this work. Details regarding VAD can be found in section 2.8. If an incoming frame is labeled as a speech frame by the VAD, it is forwarded to 'Wait Buffer'. When the size of 'Wait Buffer' reaches 10 frames, the stored buffer feeds to the pre-trained CRNN model. The model gives 1×10 probability results since we have ten classes. We find the max of it and update the graphical user interface (GUI), shown in 2.9. All these computations are done on the smartphone which offers a cost-effective DL based solution for DOA angle estimation.

As it is seen in Figure 2.9, we have four main buttons. A 'Start' and 'Stop' button are used for starting and stopping the app. 'Setting' button is for updating the VAD parameters which enable the application in different background noise types and SNR levels. The user-friendly GUI, Figure 2.9, shows the direction of the speech source numerically and with graphical blue markers. The GUI also has a 'Result Saver' button for calculating the real-time performance of the proposed CNN/CRNN based DOA estimation method in terms of accuracy and RMSE. The video demonstration of the proposed CRNN based DOA estimation can be seen on our SSPRL research lab website [32]. It should be noted that an ambiguity exists, as shown in Figure 2.9 display, due to the source symmetry w.r.t the two microphones used. The last main button is 'Result Saver' . This button is used for saving estimated DOA angles. When we touch the button, the popup screen asks the direction of the talker. We

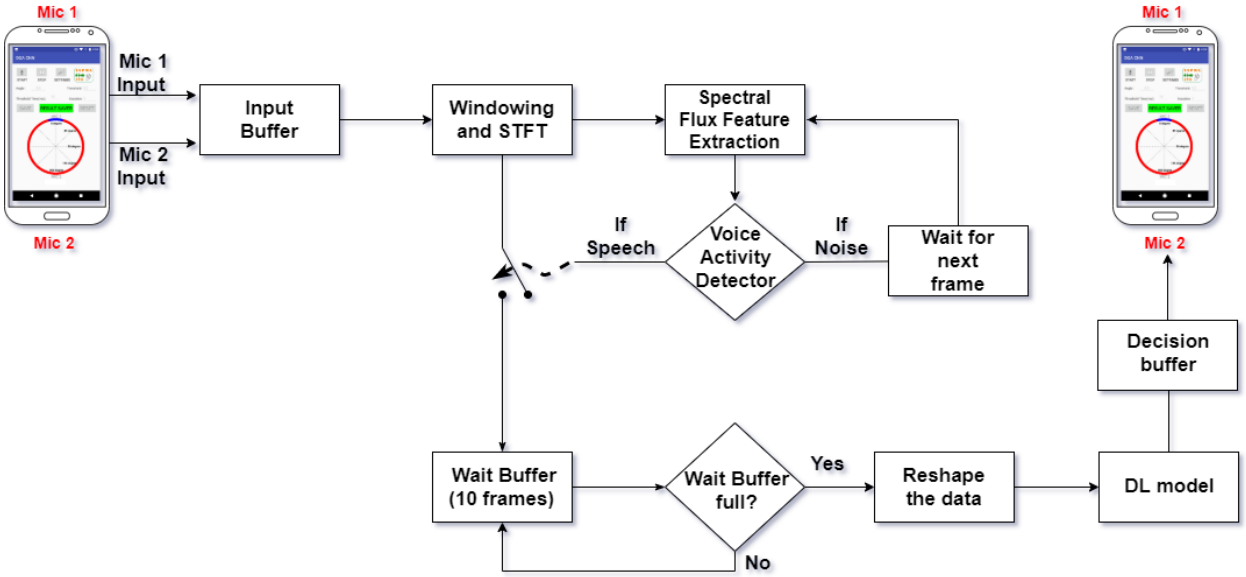


Figure 2.8: Block diagram of real-time processing modules for the proposed DOA angle estimation methods

calculate the accuracy and RMSE of real-time app based on the direction of the talker for 100 estimations. Another good feature of the app is 'Voice Assist'. The talker location is also shown with 'Voice' for every 10 estimations. It says 'The talker at θ degree', θ indicates the estimated DOA angle. We've named this feature as 'Voice Assist'.

We have a manual solution for ambiguity problem because of two microphones. The proposed app shows only one blue marker when the talker is at 0° and 180° (see Figure 2.9 for angles). Two blue markers (because of ambiguity problem) are displayed on GUI when the speaker is at a different angle than 0° and 180°. The solution would be rotating the smartphone clockwise. If rotating smartphone clockwise decreases the estimated angle, the talker is the right-hand side. Otherwise, the talker is left-hand side. This manual solution would resolve the ambiguity problem for two microphone based DOA angle estimation method. The video demonstration of the proposed CNN based DOA estimation can be seen at http://www.utdallas.edu/ssprl/hearing-aid-project/ [32].

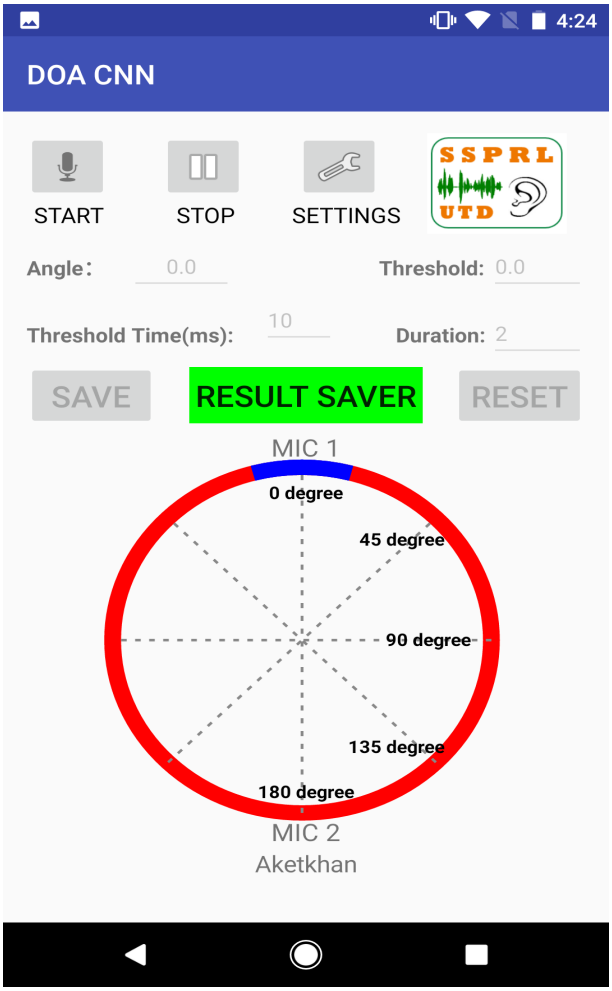


Figure 2.9: GUI display of the developed Android app for DOA angle estimate on Android Pixel 1

2.11.3 Implementation Highlights

Four parameters are critical for the real-time implementation; training size, wait buffer, reshaping and model size. First one is the training size for training of the model. Since we have a limited dataset size, when the training size is large, like 80% to 90%, the model is overfitted to the collected data. Hence the trained model is not able to work well with new unseen data in real-time operation. To resolve this issue, we have defined training data size as 50% for training the CNN model. For the second one, we use a 'Wait Frame Buffer' to get more accurate estimations. Thus, the DOA angle estimations are done based on ten

frames (each frame is 20ms) for real-time implementation of our algorithm. The third one is reshaping part of the model. Hence, the model is fed by data whose dimension is 10×2052 (for inference, the label isn't required, hence the dimension is not 10×2053) for real-time implementation. Finally, the optimization operations are applied to the model to decrease model size. 32-bit floating-point representation is used to store the model weights. However, we quantize the weight values to 8-bits per parameter using Bazel [53] which causes very little loss of accuracy in the result.

2.11.4 Performance Evaluation in Real-time Implementation

CNN based DOA angle estimation application

The performance of the proposed DOA app is tested using Pixel 1 and Pixel 3 Android smartphones. A hundred different estimations are stored with ground truth for each degree to evaluate the real-time operation of the proposed application. The phones are placed in the center of the table and the room dimension is $6.5 \times 5.5 \times 3$ meters. The measured reverberation time is 350 milliseconds. There was a fan noise in the background when collecting the results. The accuracy results are used for evaluating performance results.We define a modified accuracy (ACCmod) measure which accepts DOA estimations correct if it is within $\leq$20 of the actual angle. Modified accuracy results for Pixel 1 and 3 can be seen in Table 2.3. It is worth noting that our CNN model is trained using collected data with Pixel 1 but inferred on Pixel 3. We only see a 5% absolute degradation is accuracy. This means that our CNN model is rather robust to smartphone.

Table 2.3: Accuracy (error $\leq 20^{\circ}$) for real-time using CNN model trained for google pixel 1, but inferenced on pixel 3

Android Pixel 1	*89%*
Android Pixel 3	*83%*

CRNN based DOA angle estimation application

The performance of the proposed algorithm/app is tested in a room whose dimension is $7 \times 5.5 \times 3$ using an Android-based Pixel smartphone. Two built-in microphones of the smartphone are used for the app. The distance between the two built-in microphones is 13 cm. The location of the smartphone microphones is shown in Figure 2.8. The ten frame-based approach (Case 2) of the proposed method is implemented on the Pixel smartphone. Results of hundred DOA angle estimations are stored together with the ground truth values for four angles, namely$\{0°, 40°, 140°, 160°\}$, in order to evaluate the real-time performance of the proposed app. The phone is placed in the center of a table whose dimension is 2.43×1.21 meters in the room. A human talker was speaking 1.05 m away from the smartphone at four different angles ($\{0°, 40°, 140°, 160°\}$) with respect to the smartphone. There was no presence of background noise during performance evaluation in real-time. The accuracy and RMSE results are the criteria used for evaluating the performance of our real-time application. Table 2.4 shows the accuracy and RMSE results of the proposed real-time method/app run on pixel smartphone.

Average accuracy of 94.24% and more was obtained for the proposed method at the four different DOA angles. According to Table 2.4, the proposed app has lower RMSE for low degrees compared to high degrees. The reason for the difference in RMSE could be the difference in the characteristics of the two built-in microphones of the smartphone. The average RMSE result is 14.75°, which is less than our recorded data resolution, 20°. Table 2.4 shows that the proposed app will be a good candidate for presenting the direction of the speaker/speech source visually to the HI user.

The last measurement is CPU and Memory usage of the proposed Android app is displayed in Figure 2.10. This snapshot in Figure 2.10 was taken on Pixel 1 when the 'Wait Buffer' size was 10. As it is seen from the figure, CPU and Memory consumption increases and decreases parallelly. The reason for this is 'Wait Buffer' and VAD. When 'Wait Buffer'

Table 2.4: Accuracy and RMSE results of the real-time application

Angles	***Accuracy*** *(%)*	***RMSE(°)***
0°	95	3
40°	96	7
140°	94	20
160°	92	29
Average	94.24	14.75

size reaches ten frames, the image is formed and fed into the CNN. Then, the pre-trained model inferences as shown (in yellow arrows) in the figure. During this process, CPU and Memory usage reaches 40-44% and around 850MB, respectively. One reason for high usage is that the calculations are done using 10 frames. Another reason is TensorFlowAPI on Android utilizing Java which increases the CPU consumption. Although the consumptions are little more than expected, this situation is temporary. After the estimations were done, CPU consumption decreases to below 8% and memory usage reduces to less than 30 MB. The CNN based DOA app can run continuously on limited battery for over one hour without crashing of the app or any memory problems. Figure 2.10 clearly shows that the proposed app perfectly suits for real-time applications.

2.12 Conclusion

This chapter presents deep learning based DOA angle estimation methods and its real-time implementation on Android-based smartphone for hearing improvement. The system pipeline and architectures of the models were optimized for getting high accuracy and decreasing the computational complexity. The models were trained using real data recorded by the smartphone which enabled the model to implicitly compensate for gain mismatch of the two built-in microphones of the smartphone in the training phase. The proposed methods were compared with the recent deep learning based DOA methods and their superior performance were shown. The proposed app accuracy and error performance were also

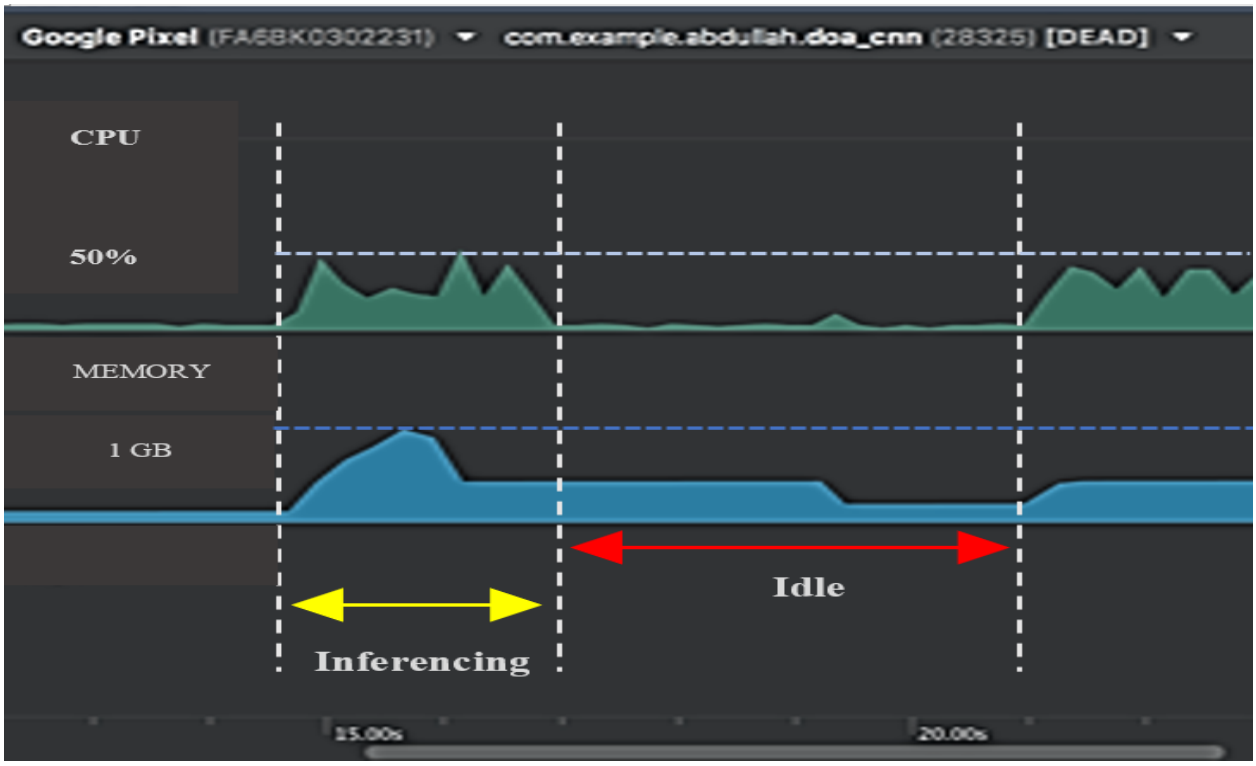


Figure 2.10: CPU and Memory consumption of Proposed Method on Pixel 1

evaluated. CPU and memory consumption of the app running on the smartphone were evaluated. As per the results of our experiments, the proposed smartphone-based DOA app suits the real-life scenarios for hearing aid applications very well.

CHAPTER 3

CONVOLUTIONAL NEURAL NETWORK BASED DIRECTION OF ARRIVAL ESTIMATION METHOD USING EIGHT MICROPHONES

Authors – Abdullah Küçük, Yiya Hao, Anshuman Ganguly, and Issa Panahi

The Department of Electrical and Computer Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

Corresponding author: Abdullah Küçük

3.1 Abstract

In this chapter, we present a real-time convolutional neural network (CNN)-based Speech source localization (SSL) algorithm that is robust to realistic background acoustic conditions (noise and reverberation) using eight microphone. We have implemented and tested the proposed method on a resource-constrained hardware prototype (Raspberry Pi) for real-time operation. We have used the combination of the imaginary-real coefficients of the short-time Fourier transform (STFT) and Spectral Flux (SF) with delay-and-sum (DAS) beamforming as the input feature. We have trained the CNN model using noisy speech recordings collected from different rooms and inference on an unseen room. We perform real-time inferencing of our CNN model on the prototyped platform with low latency (21 milliseconds (ms) per frame with a frame length of 30 ms) and high accuracy (i.e. 89.68% under Babble noise condition at 5dB SNR). We provide quantitative comparison with six other previously published SSL algorithms under several realistic noisy conditions, and show significant improvements by incorporating the Spectral Flux (SF) with beamforming as an additional feature to learn temporal variation in speech spectra. Lastly, we provide a detailed explanation of real-time implementation and on-device performance (including peak power consumption metrics) that sets this work apart from previously published works. This work has several notable implications for improving the audio processing algorithms for portable battery-operated Smart loudspeakers and hearing improvement (HI) devices.

3.2 Introduction

Speech source localization (SSL) estimation generates the important direction information that can be used to improve the performance of many audio/speech signal processing methods such as microphone array beamforming [2, 3, 14], speech enhancement [3, 54], speech/speaker recognition [4, 23], and hearing improvement (HI) devices such as Roger Select [55] and

Roger Table Mic [56]. Many commercial products are available to the public which use some types of microphone arrays and some forms of SSL methods aimed at specific applications. Considering all these, however, the robustness, accuracy, and cost-effectiveness of the SSL-based methods remain a challenging issue, especially in noisy environments at low signal to noise ratios (SNRs).

In this work, a novel eight-microphone uniform circular array (UCA) based SSL estimator using convolutional neural networks (CNN) is proposed. Previous CNN based methods such as [7] show that using imaginary-real coefficients as the feature map can work in several realistic environments but still suffer from the background noise especially when SNR is low. As the augmentation of [7], another feature, spectral flux, is included in the feature map. Additionally, a delay-and-sum (DAS) beamformer [57] is added to enhance the SNR before computing spectral flux. Thus, the feature map contains both of the imaginary-real coefficients of the short-time Fourier transform (STFT) and the spectral flux with beamforming which can essentially improve the performance of the proposed estimator. Several microphone array can solve the 180° ambiguity issue such as V-shape, circular (UCA), and spherical arrays. In this work, the UCA of eight microphones is selected for the proposed method. Such structure has been used in many commercial products such as smart loudspeakers [58, 59]. Fig. 3.1. shows the block diagram of the proposed SSL platform. Noisy speech data is received through the UCA microphones, then the imaginary and real coefficients are calculated by the STFT. Meanwhile, the STFT outputs are sent to a DAS beamforming module (which converts the signals into eight beams), then the spectral flux is generated from the signals of eight beams. The imaginary-real coefficients and the spectral flux are combined and reshaped into the feature map, then fed to the proposed SSL/DOA estimator. Once the direction of the speech source $\hat{\theta}$ is estimated by the algorithm, it will be displayed by turning on the proper LED pointing out the speech source direction. There are 35 LEDs positioned circularly on top of the development board covering the entire 360°

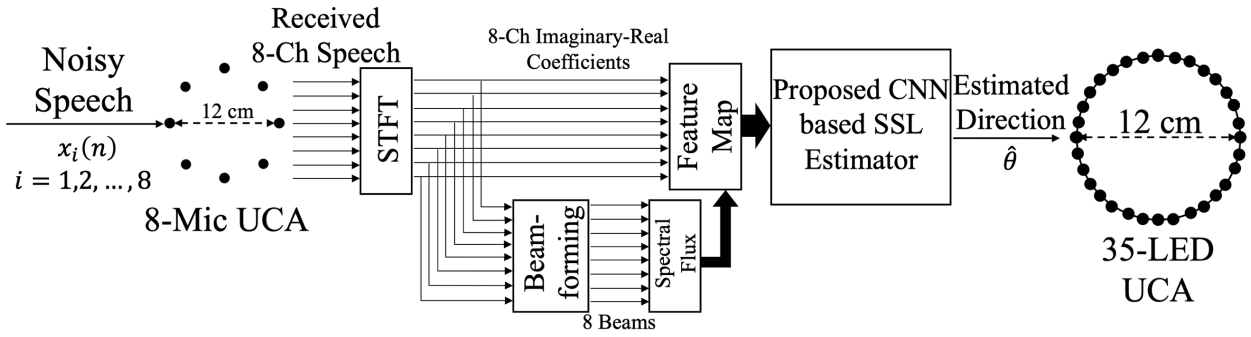


Figure 3.1: The block diagram of the proposed real-time platform using eight uniform circular array (UCA) of microphones.

azimuth in the horizontal plane. The proposed method has been implemented to run in real-time on the prototyped platform which formed with a Raspberry Pi and an internet-of-things (IoT) development board with UCA microphones. The proposed method has shown excellent performance and accuracy offline or in real-time under realistic noisy environments. The real-time testing was completed in a separate room which is different as the room for the data collection room.

The major contribution of this work is the augmentation of the imaginary-real coefficients with spectral flux plus beamforming. The utilization of spectral flux as one of the features can incorporate temporal dependency between successive signal frames, and feed into the CNN model training. The beamforming technique essentially improves the performance of the spectral flux based method. A pre-processing stage by beamformer enhances the SNR of the input signal for spectral flux. Typically, CNN models treat each feature vector to be independent of adjacent frames, hence including spectral flux can yield better models that are more aware of voiced-activity-detection (VAD) type activities. Although some models such as a recurrent neural network (RNN) model can essentially learn the above temporal representations, they are usually more memory intensive and have higher latency than their CNN counterparts. Another contribution of this work is the prototyped platform with beamforming which converts the proposed method from an offline trained model into

a realtime SSL estimator. The end-products similar like [55] and [56] can be built based on the prototype. The proposed method, therefore, offers both scientific significance and practical importance. In this work, we use the term "eight-microphone" or "eight-channel" to specify the number of SSL sensors/microphones used. In the figures, we also use the term "MIC" or "CH" denoting the microphone. The term "Beam" denotes the output signal after beamforming.

3.3 Feature Representation for Training

The feature representation needs to contain enough information for the estimation purpose. In our proposed method, the imaginary-real coefficients from the STFT and the spectral flux after beamforming are combined as the feature set. The speech information is included in the imaginary-real coefficients of the current frame (i.e. the voiced segments of the speech such as vowels have harmonic characteristics). The spectral flux contains information of the magnitudes for the current frame and the previous frame which provides the model with the short-term memory.

3.3.1 Imaginary and Real Coefficients

For the proposed CNN method, the N-point STFT is applied to every data frame of the time-domain signal, shown as

$$X_k^i(m) = \sigma_k^i(m) + \tau_k^i(m) \tag{3.1}$$

where, $X_k^i(m)$, stands for the output of N-point STFT of $x_k^i(m)$ (from i^{th} microphone for k^{th} frame). $\sigma_k^i(m)$ denotes the real part of the $X_k^i(m)$, and $\tau_k^i(m)$ denotes the imaginary part of $X_k^i(m)$. m denotes the frequency bin. In the proposed method, the real parts $\sigma_k^i(m)$ and the imaginary parts $\tau_k^i(m)$ as one of the features feed the CNN models for training, and forming the following vectors,

$$\tau_k^i = [\tau_k^i(1)\ \tau_k^i(2)\ ...\ \tau_k^i(\frac{N}{2}+1)]^T \tag{3.2}$$

$$\sigma_k^i = [\sigma_k^i(1)\ \sigma_k^i(2)\ ...\ \sigma_k^i(\frac{N}{2}+1)]^T \tag{3.3}$$

Using (3.2) and (3.3), the feature $\Phi_k^i(m)$ can be represented by the following matrices,

$$\Phi_k^l(m) = [\tau_k^1\ \tau_k^2\ ...\ \tau_k^8]^T,\ l = 1 \tag{3.4}$$

$$\Phi_k^l(m) = [\sigma_k^1\ \sigma_k^2\ ...\ \sigma_k^8]^T,\ l = 2 \tag{3.5}$$

where, l is the number of feature channel. Hence, $\Phi_k^1(m)$ represents the imaginary coefficients feature set, and $\Phi_k^2(m)$ stands for the real coefficients feature set.

3.3.2 Spectral Flux

The imaginary-real feature can cover the frequency domain information of the speech. However, it only covers k^{th} signal frame information excluding any relations between adjacent frames. This disadvantage can be resolved by adding spectral flux into the feature set for proposed CNN model which offers the short-time memory. In conventional signal processing SSL methods, the performance of using spectral flux has already been utilized and shown by scholars such as [27]. It is interesting to note that spectral flux works so well without any phase information. The reason could be that spectral flux contains the signal power information between successive frames which helps SSL by the human hearing system. Another reason might be that instead of the absolute values of the captured samples, spectral flux only contains the relative values (the STFT magnitude difference between successive frames) which are more robust for the disunity issue of microphone array introduced by hardware.

In the proposed method, the signals from eight microphones are converted to the frequency domain by STFT, then processed by the beamforming module. Then they are converted to eight beams. That is,

$$BF_k^q(m) = \frac{1}{L}\sum_{i=0}^{L-1} W^{q,i}(m) X_k^i(m) \tag{3.6}$$

$$\Upsilon_k^q(m) = A_k^q(m)\, e^{j\theta_{BF_k^q(m)}} \tag{3.7}$$

where, $BF_k^q(m)$ denotes the beamformer output at q^{th} beam for k^{th} frame. L stands for the total number of the microphones which equals to eight in this work. $W^{q,i}(m)$ denotes the finite impulse response (FIR) filter weights in frequency domain at i^{th} microphone for q^{th} beam. In (3.7), $A_k^q(m)$ is the magnitude of $BF_k^q(m)$, and $\theta_{BF_k^q(m)}$ stands for the phase of the $BF_k^q(m)$. Hence, the spectral flux coefficients for two successive frames can be calculated as follows,

$$M_k^q(m) = |A_k^q(m)| - |A_{k-1}^q(m)| \tag{3.8}$$

$$S_k^q = [S_k^q(1)\ S_k^q(2)\ ...\ S_k^q(\frac{N}{2}+1)]^T \tag{3.9}$$

where, $M_k^q(m)$ is the magnitude differences between two adjacent frames. Then the spectral flux-based feature constructed as

$$\Phi_k^l(m) = [S_k^1\ S_k^2\ ...\ S_k^8]^T,\ l = 3 \tag{3.10}$$

As (3.10) shows, spectral flux as the third feature channel has been inserted into feature map Φ_k^l. The details of the training input formats are discussed in Section 3.4.

3.4 Convolutional Neural Network Model

In this section, the CNN model of the proposed method is presented. The architecture of the proposed CNN model contains one input layer, three convolution layers, one pooling layer, two fully-connected layers, and one output layer. The size of each feature map is $M \times K$, where $M = 8$ since there are eight microphones/beams, and $K = \frac{N}{2} + 1 \times H$ where N is the

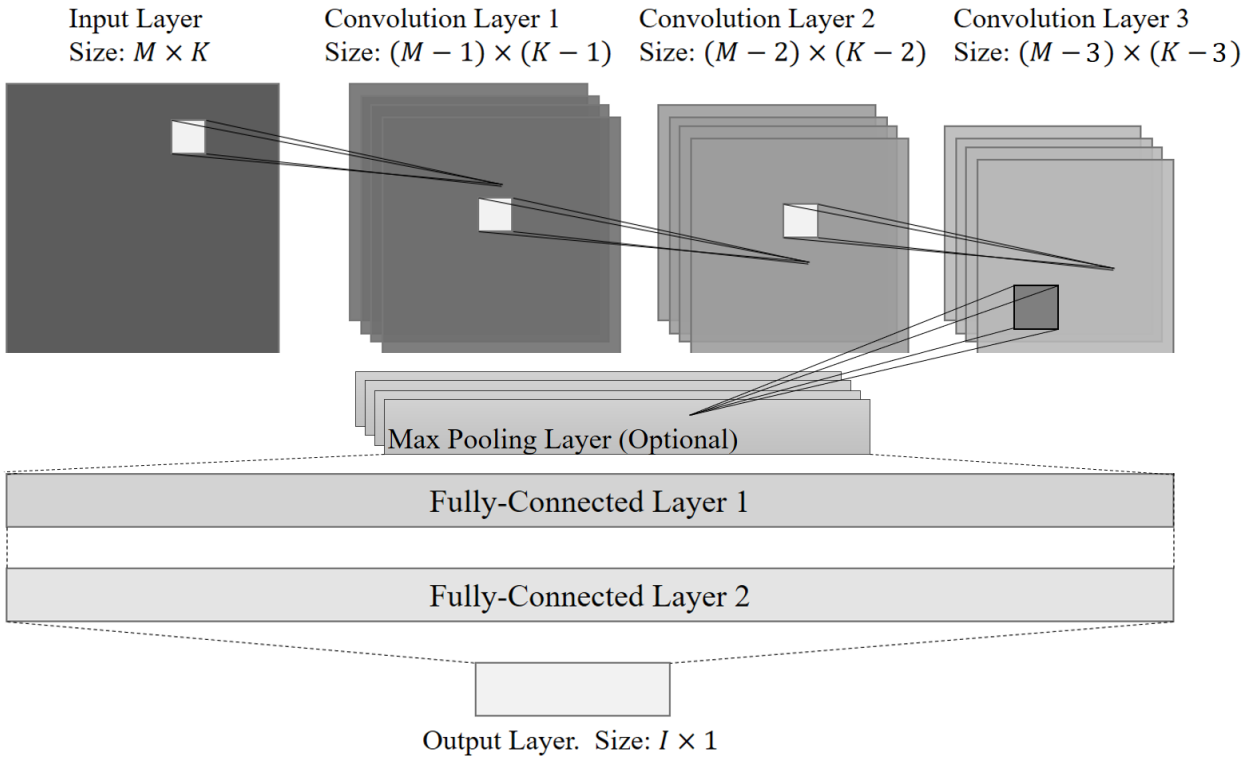


Figure 3.2: The CNN model of the proposed method. The size of the input layer is 8×771. The size of the output layer is 8×1

number of the STFT point. In proposed work, $H = 3$ which stands for the Φ_k^1, Φ_k^2, and Φ_k^3. The CNN model is shown as Figure 3.2

3.4.1 Data Labelling

In order to train the CNN model, the realistic speech signals have been captured and used to create datasets for training and testing purposes.

The recorded data data was labeled and reshaped into the feature set Φ_k^l. The frame size equals to 30 milliseconds (ms) at 16kHz sampling frequency, resulting in 480 samples for each frame. Therefore, the STFT size is set to $N = 512$ points. After STFT, there are $\frac{N}{2} + 1 \times 3 = 771$ coefficients, and the total size of the input feature is 8×771. The imaginary real coefficients and spectral flux (Φ_k^1, , Φ_k^2, $and\Phi_k^3$) of eight microphones/beams are put into the eight different rows. Each row contains the three features of one microphone or beam

pointing at one direction (e.g. the first row contains three features at the direction of 0°). The dataset

$$\widetilde{\Phi} = \begin{bmatrix} \tau_k^1 & \sigma_k^1 & S_k^1 \\ \tau_k^2 & \sigma_k^2 & S_k^2 \\ \vdots & \vdots & \vdots \\ \tau_k^8 & \sigma_k^8 & S_k^8 \end{bmatrix} \tag{3.11}$$

A ground truth θ_k (for k^{th} frame) is put at the end of the vector representing the actual direction.

3.4.2 Convolutional Neural Network (CNN) model

Once the pre-processing including labeling and reshaping has been completed, the input feature maps are fed into the CNN model for training. A set of filters of size 2×2 in the convolution layer is applied to learn the correlations among all the feature coefficients. Each filter convolves with the first 2×2 samples of the input feature map then shifts one step towards the right-hand side to do the next convolution. Each convolutional layer contains 64 filters. After three convolution layers, a pooling layer is followed to downsample the data. The size of the fully-connected layer equals to $(M-3) \times (K-3) = 5 \times 768 = 3840$ Then the modeled coefficients are sent to the first fully-connected layer. The rectified linear units (ReLU) activation function [46] is used inside the fully-connected layers. After two fully-connected layers, the coefficients will be mapped to the output layer with the size of $I \times 1$ which treats the whole system as a classification problem. In this case, we set the $I = 8$ which means the resolution is 45°. This resolution is used since it can cover typical situations encountered by a user with people around, such as in a business meeting, group conversations, and dining in a restaurant.

The softmax function is applied to generate the probability for each coefficient θ_k inside the output layer. The categorical crossentropy is used as the lost function. The final SSL - the DOA estimated azimuth angle is then given by,

$$\hat{\theta}_k = \underset{\theta_k}{\operatorname{argmax}}\{p(\theta_k \mid \Phi_k^l)\} \tag{3.12}$$

where $p(\theta_k \mid \Phi_k^l)$ denotes the conditional probability of θ_k using $\Phi_k^1,\ \Phi_k^2,\ and\ \Phi_k^3$. $\hat{\theta}_k$ is the final estimated direction (the DOA angle estimate) at k^{th} frame. In the experiment setup, the feature sets contain 90 minutes clean speech for each direction with 45° resolution. The CNN models shuffle the feature sets and apply 90 percent of the data to train the model, and 10 percent of the data to validation.

After the the whole training is completed, a frozen model is generated as the proposed CNN based SSL estimator. The proposed method has been implemented on the prototyped platform in real-time. Therefore, both of the offline validation/testing results and the real-time performance of the proposed method have been measured. The proposed model is built, trained and implemented based on Tensorflow (version 2.0) [22].

3.5 Data Collection

The performance of a learning model using a simulating dataset is unconvincing, especially in the realistic scenarios. Therefore, a data collection scheme is presented to obtain a realistic dataset for model training.

3.5.1 Data Collection Scheme - The Setup

Figure 3.3 shows the setup of the data collection in room A. Multiple loudspeakers are placed at the edge of a circular table. The clean speech signals are played via the loudspeakers while another loudspeaker locating under the table can play the noise creating diffused background

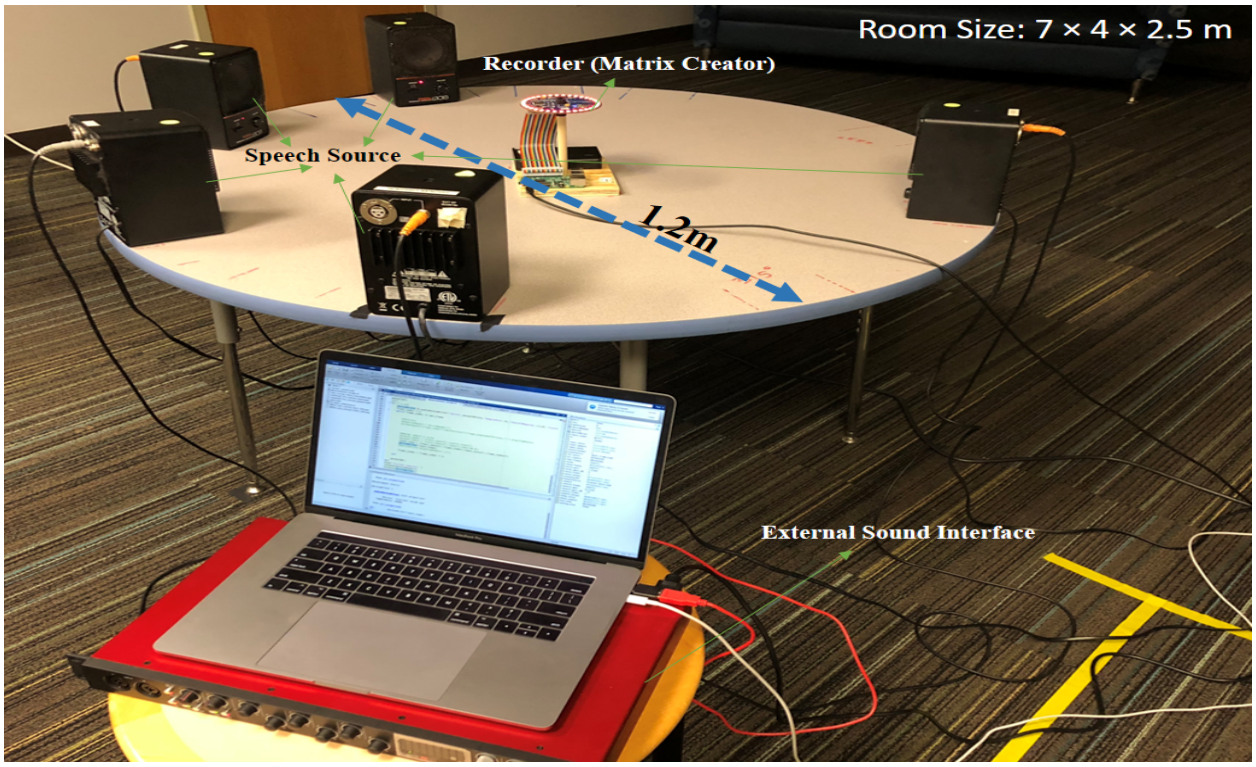


Figure 3.3: The Setup of Data Collection (Room A).

noise. All loudspeakers are connected to an external audio interface which is controlled by a script running on a MacBook via a USB3.0 cable. The prototyped platform as the recording device with eight MEMS microphones sits in the center of the circular table. The training speech is made based on HINT database [47]. The total length of the training speech is 90-minutes long for each direction/loudspeaker. The data collection was completed in room A, B, and C. The real-time testing was completed in room D. The setup information is presented in Table 3.1. Details of the prototyped platform is presented in section 3.7.

3.5.2 Collection Procedures

Sound level calibration is required before the collecting session. A sound pressure level (SPL) meter is used to calibrate the output levels of all loudspeakers to 65 dB SPL. The level of the noise loudspeaker is set at different SNRs for conducting the experiments. After the

Table 3.1: Collection Setup

	Quantity	Details
Room A	1	$7 \times 4 \times 2.5$ *meters*
Room B	1	$4 \times 4 \times 3$ *meters*
Room C	1	$8 \times 4 \times 3$ *meters*
Room D	1	$6 \times 4 \times 3$ *meters*
Speech Loudspeaker	8	Fostex 6301B
Noise Loudspeaker	1	Bose SoundLink Mini II
Circular Table	1	1.2 meters diameter
Audio Interface	1	Focusrite Scarlett 18i20
Recording Device	1	Matrix Creator (8 MEMS MIC)
Clean Speech	90 min	HINT Database
Noise	2 Types	Babble and Machinery

sound level calibration, the speech signal from the first loudspeaker starts to play while the noise loudspeaker is playing at the same time. The first loudspeaker plays the speech for 90 minutes, then the second loudspeaker starts to play from another location/direction. Using the same manner, the rest of the loudspeakers play speech signals from different directions one after another. Once the data collection session is done, the recorded audio data will be dissected into different pieces (one single piece stands for one loudspeaker direction). Then the azimuth directions are labeled to corresponding speech pieces as discussed above. The collected dataset is currently available for public use in [32].

3.6 Measured Results and Discussion

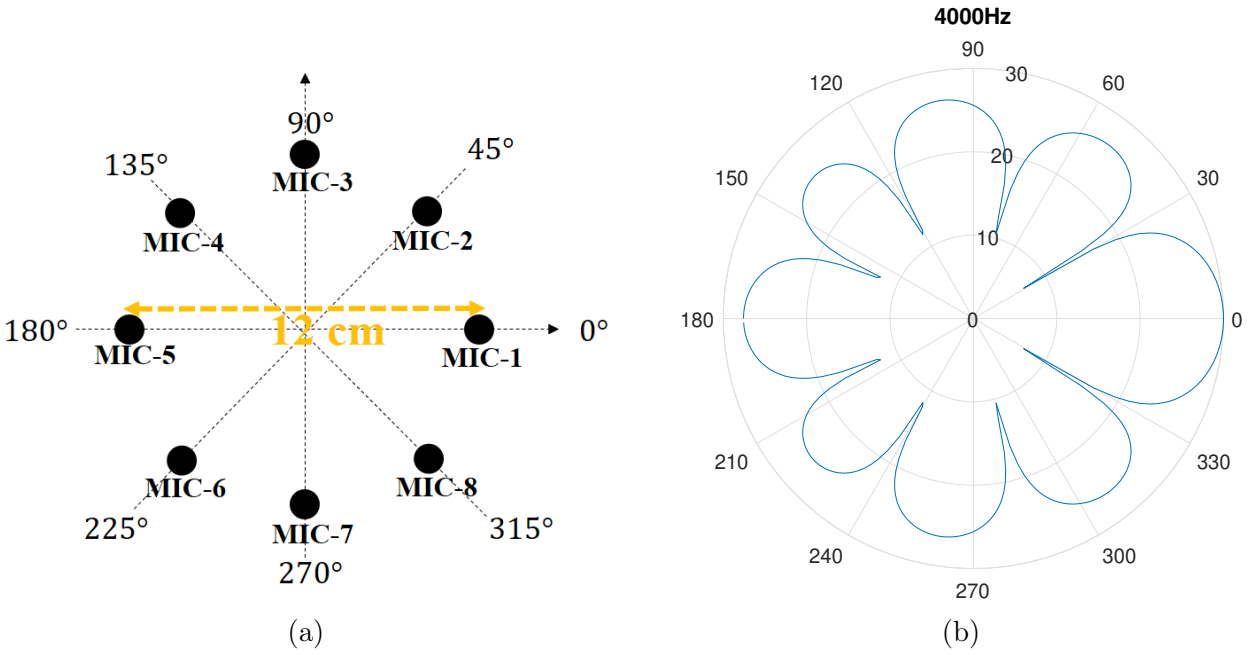
In this section, we present several offline test results to show the performance of the proposed method (denotes as 8CHImagReal-SF-BF) compared with other published methods to the cases we considered. The comparisons are trained/tested with the same dataset as the proposed method. The comparisons include a conventional signal processing SSL estimator based on the generalized cross-correlation (GCC) [60] (denotes as 8CH-GCC), an MLP neural network based eight-microphone SSL estimator using GCC-Phat as the feature set

[40] (denotes as 8CH-GCCPhat-MLP), and a CNN-based SSL estimator using the phase of the white noise as the feature set [30] (denotes as 8CH-Phase-WN). One more comparison has also been added using the same CNN model as 8CH-Phase-WN but using the phase of the speech instead of the white noise (denotes as 8CH-Phase). Another two comparisons use the same CNN model as the proposed method. One of them uses the feature of the imaginary-real coefficients (same as the published work in [7]) (denotes as 8CH-ImagReal). In order to measure the improvement by beamforming, another method using the imaginary-real coefficients and spectral flux without beamforming is included as well (denotes as 8CH-ImagReal-SF). The experiments include the offline testing and real-time testing. The offline testing is based on the collected data in room A, B, and C. The 10 percent of the collected data is used for testing (90 percent of the collected data is used for training). The real-time experiments were completed in room D with the prototyped platform. The dimension of the rooms is shown in Table 3.1. The offline measured results are presented in this section, and the real-time test results is presented in section 3.7.

Fig.3.4a shows the UCA geometric positions. The MIC- 1 is located at 0° . DAS beamforming has been used to enhance the SNR for spectral flux feature (DAS modifies the phase information so that phase-related features such as imaginary-real is unsuitable). DAS beamforming has low computation complexity compared to other beamformers such as MVDR [61] which ensures real-time implementation. The directivity pattern of the first beam towards 0° at 4kHz of the beamformer is shown in Fig. 3.4b. Eight linear-phase fractional-delay filters convolve with their corresponding microphone signals to generate the first beam. All eight beams point to their own directions from 0° to 315° and have 45° between every two adjacent beams.

3.6.1 The Performance of the Proposed Method Under Quiet Condition

In this section, the measured results under quiet and noisy conditions are presented. 90-minutes long collected speech dataset for eight directions are used for training and testing.



(a) (b)

Figure 3.4: (a) The geometric positions of the eight-microphone UCA, and (b) the directivity pattern of the first beam towards 0° at 4000Hz.

90 percent of the collected data is used for training, and the rest is used for the testing. The accuracy is quantified based on the root mean square error. The accuracy (ACC) measure is defined by,

$$ACC = \frac{N_C}{N_F} \tag{3.13}$$

where, N_F is the total number of the frames per test case and N_C is the total number of the frames with the correct direction estimation. N_C can be denoted as,

$$N_C = \sum_{k=1}^{N_F} c_k, c_k = \begin{cases} 0, \ \theta_k \neq \hat{\theta}_k \\ 1, \ \theta_k = \hat{\theta}_k \end{cases} \tag{3.14}$$

where, c_k represents the estimated correction of k^{th} frame. θ_k is the actual direction and $\hat{\theta}_k$ is the estimated direction for the k^{th} frame. ACC can present the performance of the estimator partly, because the result is correct only if the estimated direction is same as the

actual direction. However, if the estimated direction is only one class different from the actual direction, the ACC result will still show the estimation is failed even it just one class different. To quantify the performance additionally, the ACC_w is introduced. It is defined as,

$$ACC_w = \frac{\widetilde{N_C}}{N_F} \tag{3.15}$$

where, $\widetilde{N_C}$ denotes the number of the correction frame with a wide angle.

$$\widetilde{N_C} = \sum_{k=1}^{N_F} \widetilde{c_k}, \widetilde{c_k} = \begin{cases} 0, \ |\theta_k - \hat{\theta}_k| > 45^\circ \\ 1, \ |\theta_k - \hat{\theta}_k| \geq 45^\circ \end{cases} \tag{3.16}$$

In the quiet environment, the ACC of the proposed method, 8CH-ImagReal-SF-BF, is measured and compared with other methods including 8CH-GCCPhatMLP, 8CHPhase-WN, 8CH-Phase, 8CH-GCC, 8CH-ImagReal, and 8CH-ImagReal-SF (Fig. 3.5). The performance of 8CHGCC 8CHGCC, as a conventional signal-processing based estimator, is worse than most of the other neural network based estimators except 8CH-Phase-WN. The proposed method reaches the best performance with 93% ACC among all estimators. The proposed method is better than 8CH-ImagReal which shows the improvement of the combination features (imaginaryreal coefficients plus spectral flux) comparing to using imaginary-real coefficients alone. Meanwhile the proposed method is also better than 8CH-ImagReal-SF. This is the improvement by beamforming which boosts the SNR of the input for spectral flux. The ACC_w results in Fig. 3.6 prove it again by presenting the accuracy with a wider angle. 8CH-ImagReal reaches 95% ACC_w but still lower than the proposed method which reaches the best results again at 97% ACC_w. According to both of the ACC and ACC_w results, the proposed method is better than the 8CH-ImagReal. This fact proves that for the feature set, the combination of the imaginary-real and the spectral flux with beamforming performs better than using imaginary-real alone.

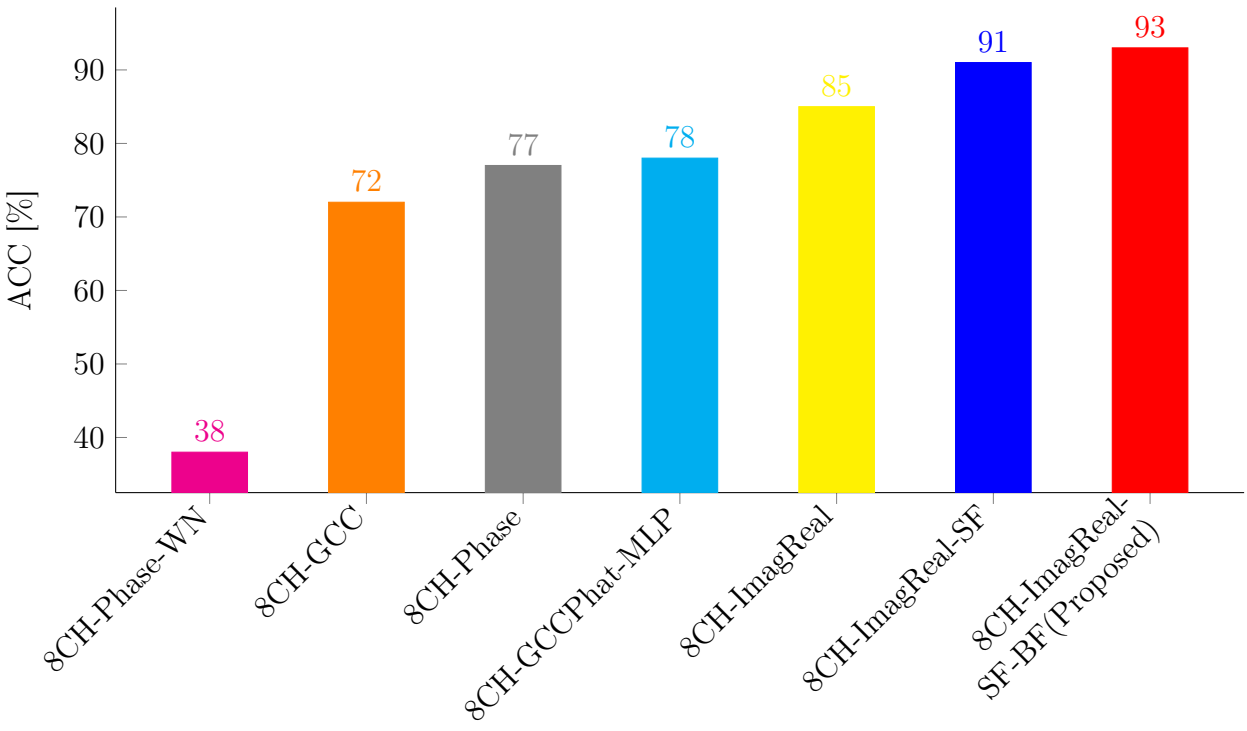


Figure 3.5: ACC comparison results for the silent environment

3.6.2 The Performance of the Proposed Method Under Noisy Condition

All the results above are only based on the clean speech signals. In order to test and evaluate the performance of the proposed SSL method, noisy speech data are collected as follows. Speech is played by eight loudspeakers one-by-one circularly placed on a table at 0° to 315° angles with 45° resolution. Meanwhile, noise is played by a loudspeaker placed under the table simulating diffused noise. The setup is presented in Section 3.7. Fig.3.7 and 3.8 show the offline ACC three different SNR levels. To compare the proposed method to other estimators, another three CNN-based estimators are measured. 8CH-GCC is included as a conventional signal processing based estimator. 8CH-GCCPhat-MLP and 8CHImagReal

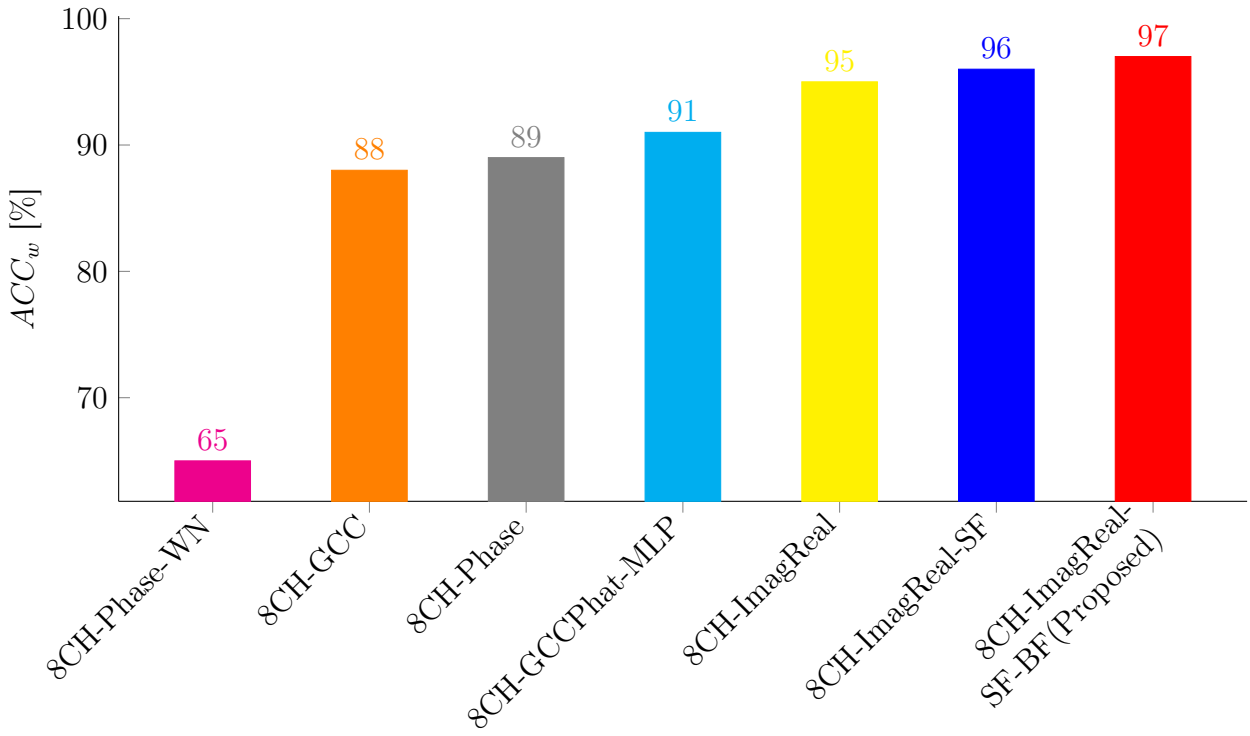


Figure 3.6: ACC_w comparison results for the silent environment

are also included because they are the best two published estimators besides the proposed method in the previous measurement. The offline ACC results show that the proposed method is robust to background noise even in low SNRs under babble noise (as one of the toughest noisy situations – a non-stationary noise). The ACC of the proposed method at 0dB SNR under machinery noise is above 85%, and even reaches 92% when the SNR is enhanced to 5dB.

Fig. 3.9 and 3.10 show the ACC_w results of the proposed method and comparisons. Under machinery noise, the proposed method gets 95% ACC_w at 5dB SNR, and still gets 81% ACC_w at -5dB SNR. Under babble noise, the ACC_w of the proposed method is slightly lower than the ACC_w under machinery noise, but still more robust to background noise than other comparisons.

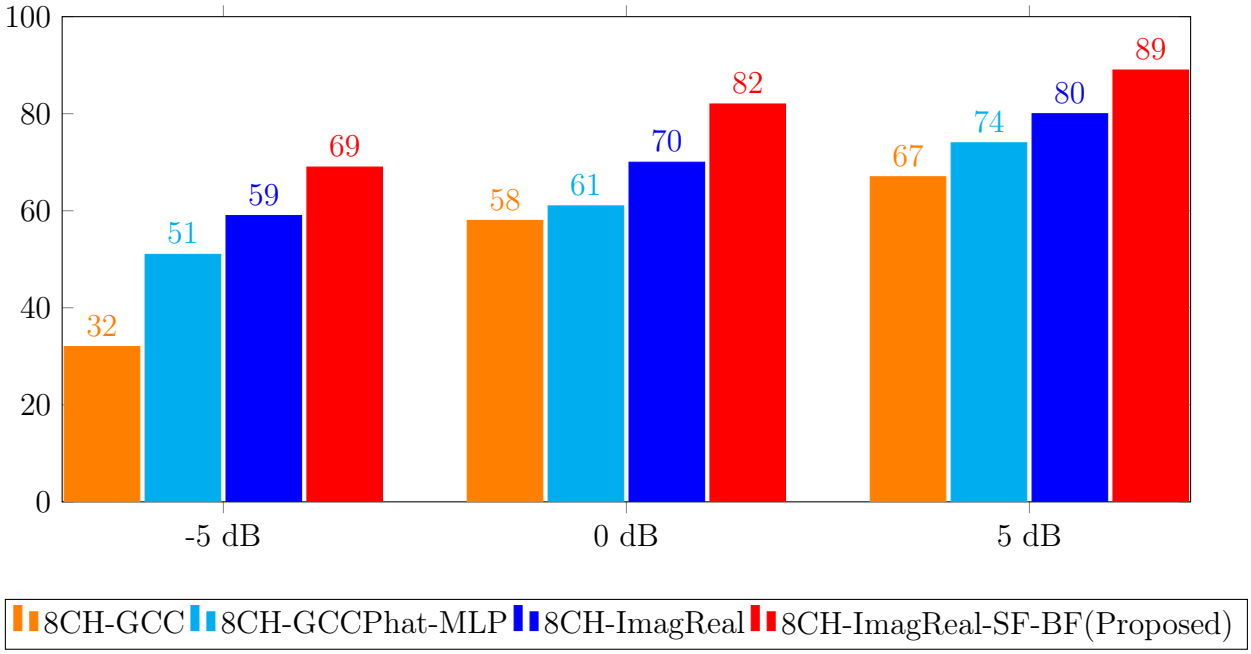


Figure 3.7: The offline ACC results (%) under babble conditions.

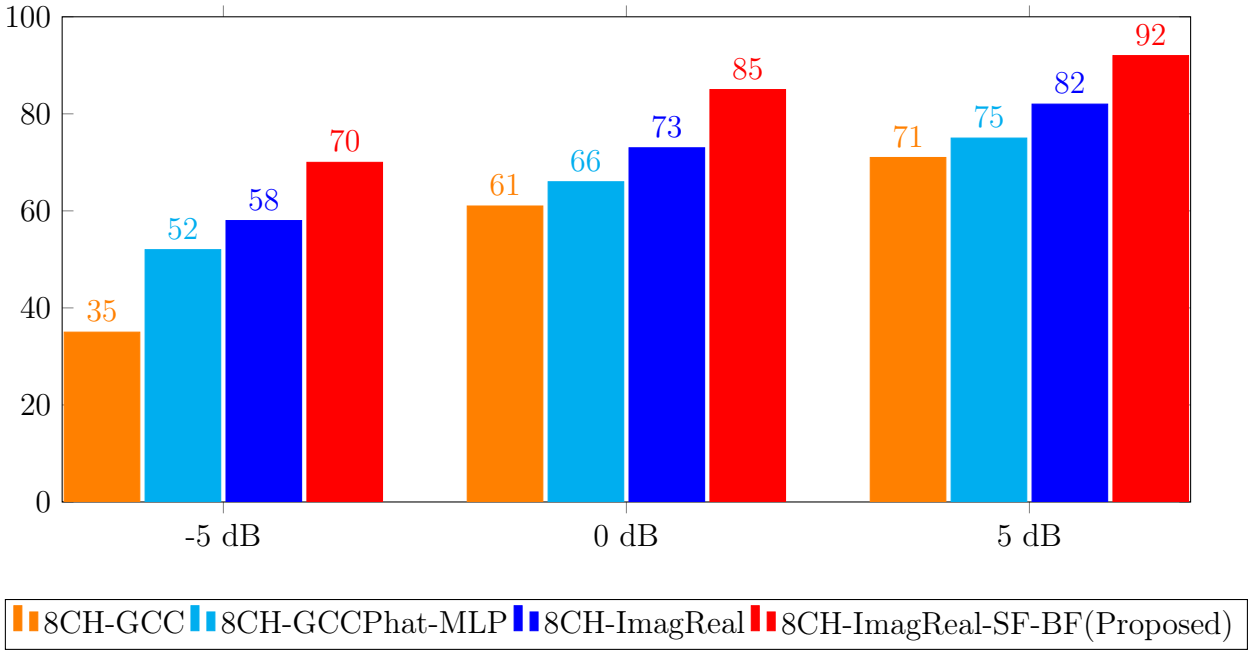


Figure 3.8: The offline ACC results (%) under machinery conditions.

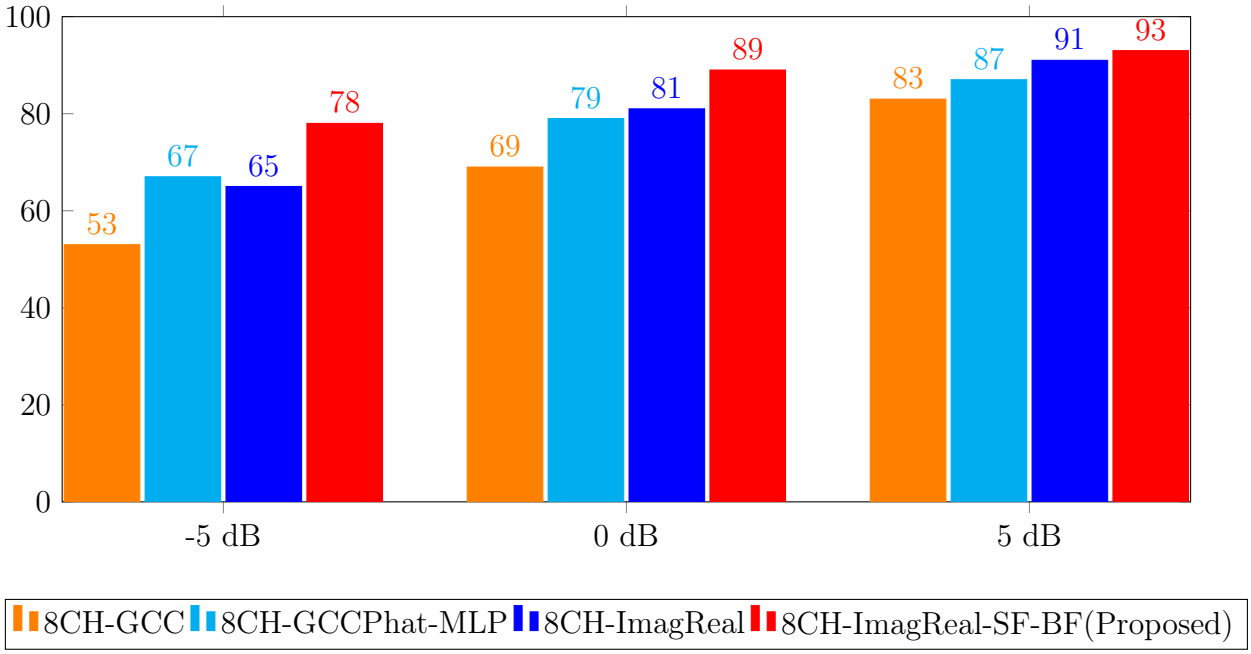


Figure 3.9: The offline ACC_w results (%) under babble conditions.

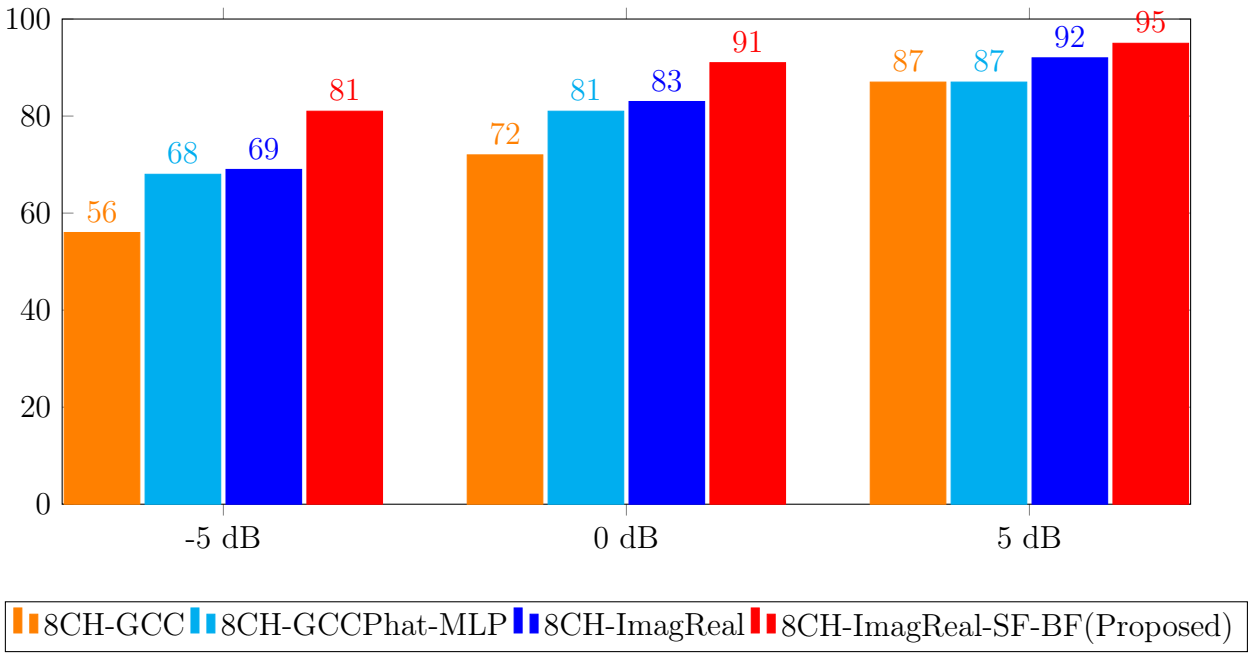


Figure 3.10: The offline ACC_w results (%) under machinery conditions.

3.7 Real-Rime Implementation and Real-Time Measured Results

Offline results can partially prove and show the performance of the methods. However, it is always necessary to implement the method in real-time, capture the realistic data, and test it on the fly. The proposed method and several other comparisons have been implemented in real-time. The algorithms are written in C/C++ and Python-based on frame based data. A single-board computer - the Raspberry Pi 3 (RP3) [62], and an IoT development board - matrix creator (MC) [63] have been used as the real-time implementation platform. Such platform has been used as the recording device as well in the proposed data collection sessions. Fig. 3.11. shows the hardware platform for real-time implementation. The RP3 and a mobile power bank are sitting on the bottom. The MC with the microphone array is lifted sixteen centimeters high in order to reduce the sound reverberation and reflection effects from the table.

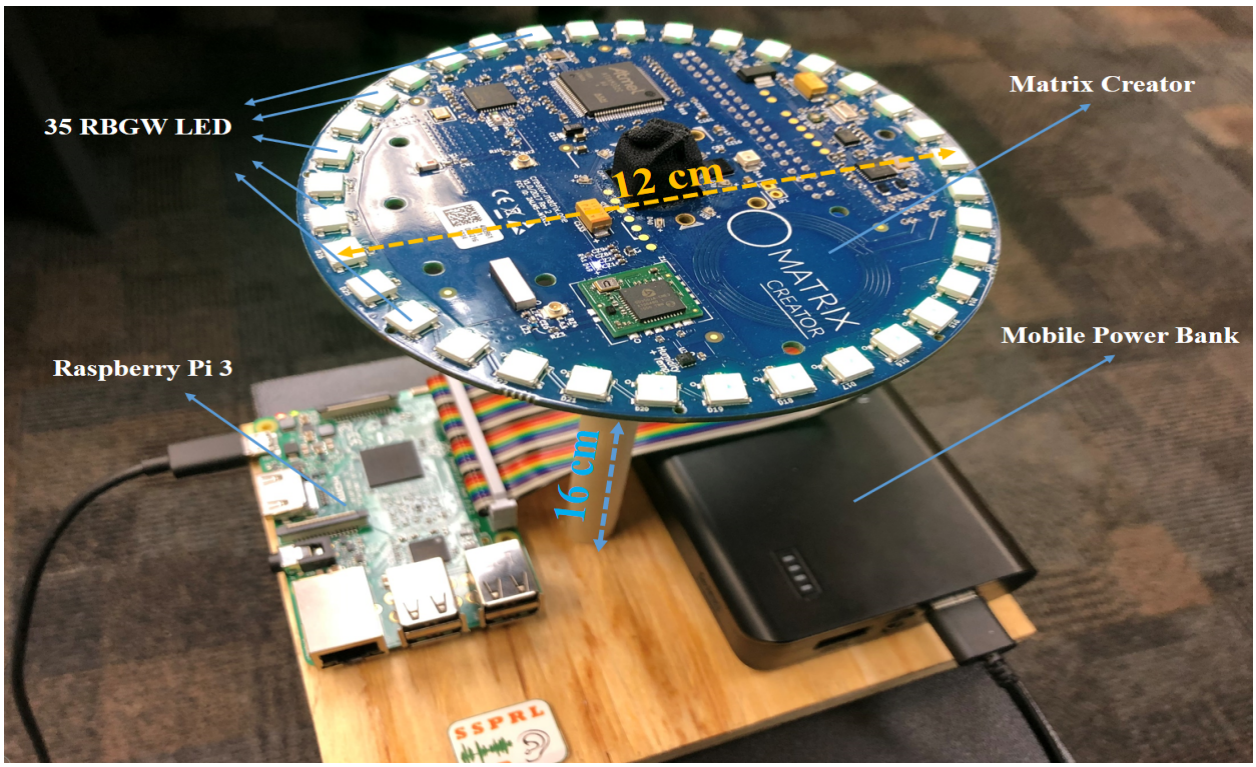


Figure 3.11: The entire hardware connection and setup.

3.7.1 Hardware Platform

As we discussed above, two hardware modules have been used as our hardware platform for real-time implementation. The first one is a single-computer RP3, and another one is an IoT development board of MC which is an extendable board for RP3 via the 40 pins general-purpose input/output (GPIO) connection. In MC, eight-microphone UCA (omnidirectional MEMS microphones) is located at the edge of a small round board on the backside. 35 RBGW-LED lights are also located at the edge of the board as a ring covering 360° on the front side, see Fig.3.11. Both microphones and lights are controlled by a Spartan 6 FPGA board. The details of the hardware of the prototyped platform is shown in Fig.3.12.

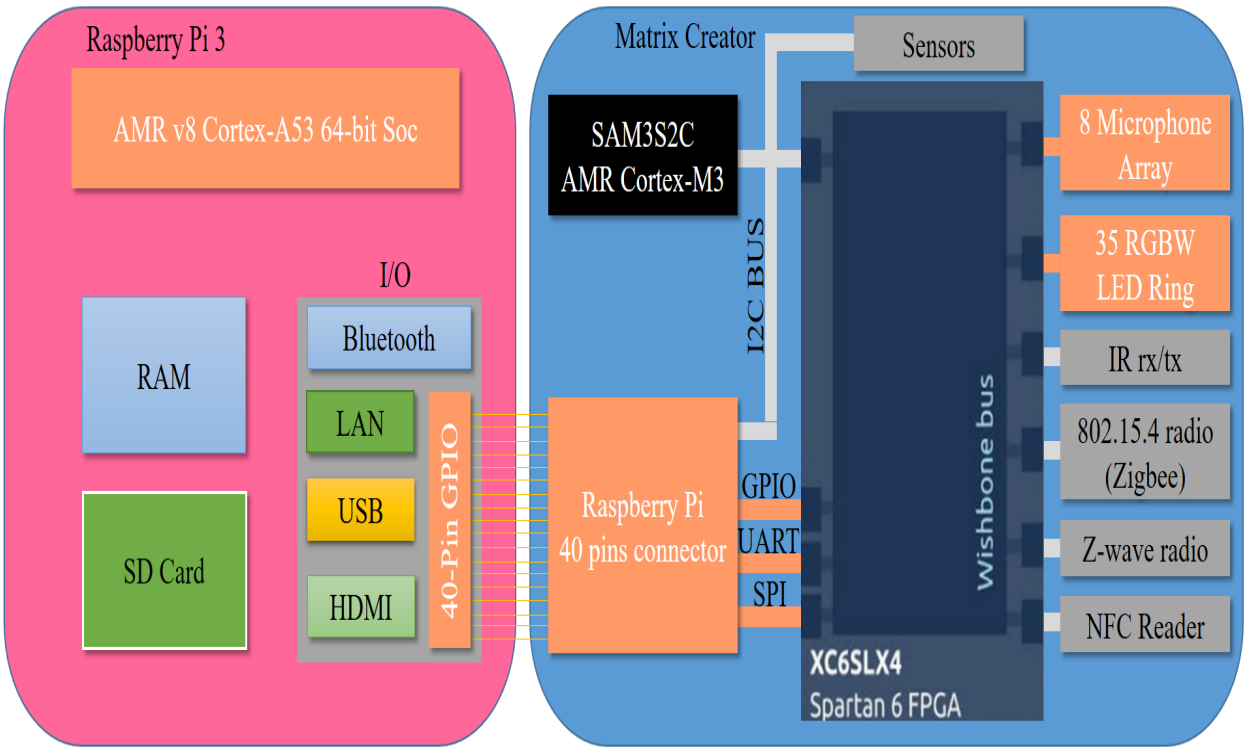


Figure 3.12: The details of the hardware of the prototyped platform.

3.7.2 Frame-based Algorithm in C/C++ and Python

In order to implement the proposed method in real-time, the pre-trained model is frozen. The proposed CNN model is put into the RP3 running in Python on a Linux operating system (OS) using Tensorflow. The computations need to be reduced so that the RP3 is sufficient to handle the real-time processing. The block diagram of the real-time implementation is presented in Fig.3.13. The speech signals are captured via the eight-microphone array from the MC board. The microphones on MC are all digital MEMS, which means the output signals have already been converted to digital data from analog. Spartan 6 FPGA gathers and buffers the signal data, then it directly sends them into the RP3 via a serial port protocol - the serial peripheral interface (SPI). In the RP3, an executable file takes control to receive the speech signal data from SPI. The executable file is written in C++ and embedded C and then compiled by GNU [64] compiler collection. In the executable file, the received speech signals are pre-processed to generate the feature maps. Then the feature maps are sent to the pre-trained frozen model, and the model will estimate and predict the direction (DOA angle) based on the input feature maps. Once the estimated direction angle $\hat{\theta}_k$ is produced, the executable file will then light up the corresponding LED in the MC surface (via SPI) to display the estimated direction of the speech source. Furthermore, in order to evaluate the real-time performance, the estimated direction was sent to the server as well via SPI. The server controls the loudspeakers playing, meanwhile calculating the real-time estimation results.

3.7.3 Real-time Performance of the Proposed Method Under Noisy Conditions

The real-time performance of the proposed method was tested via the prototyped platform. The comparisons including 8CH-GCC, 8CH-GCCPhat-MLP and 8CHImagReal are implemented on the same platform as well. The experiments were completed in room which is different as to the data collection rooms. The experiments were under babble and machinery

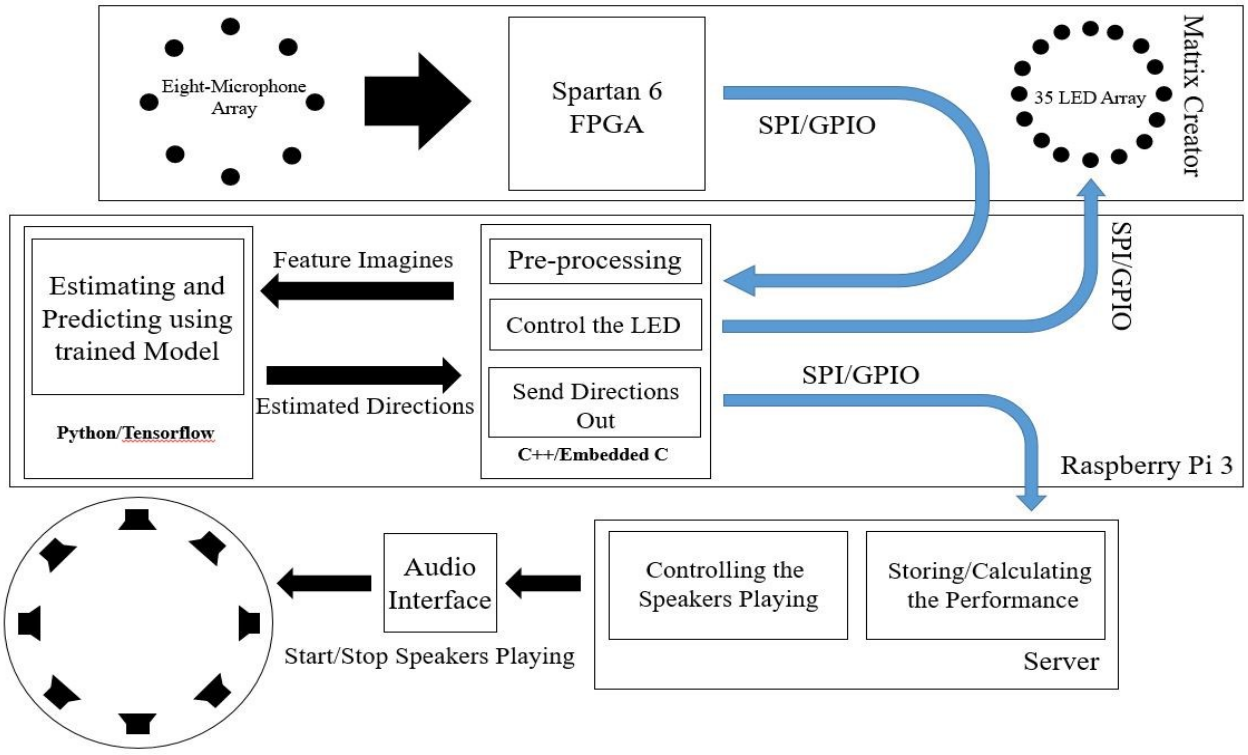


Figure 3.13: The block diagram of the real-time implementation.

noise with 90-minute speech played from the loudspeakers. The ACC and ACC_w results are presented in Fig. 3.14-3.17. In our experiments, both of the ACC and ACC_w results of 8CH-GCCPhat-MLP are decreased extremely comparing to the offline test. The reasons include (i) the real-time processing may introduce interference and calculation delay to jeopardize the performance, (ii) the model of the 8CH-GCCPhat-MLP is overfitted to the training data. Although the real-time performance of all estimators is degraded (compared to offline performance), the proposed estimator still reaches the best results with ACC and ACC_w real-time measured results show that (i) the proposed method is not overfitted to the training data, (ii) the proposed method is more robust to background noise over the comparisons. The proposed method can be furtherly built as a final/commercial HI product by including other processing modules such as a VAD detector, an auditory processing module, or a speech enhancement module.

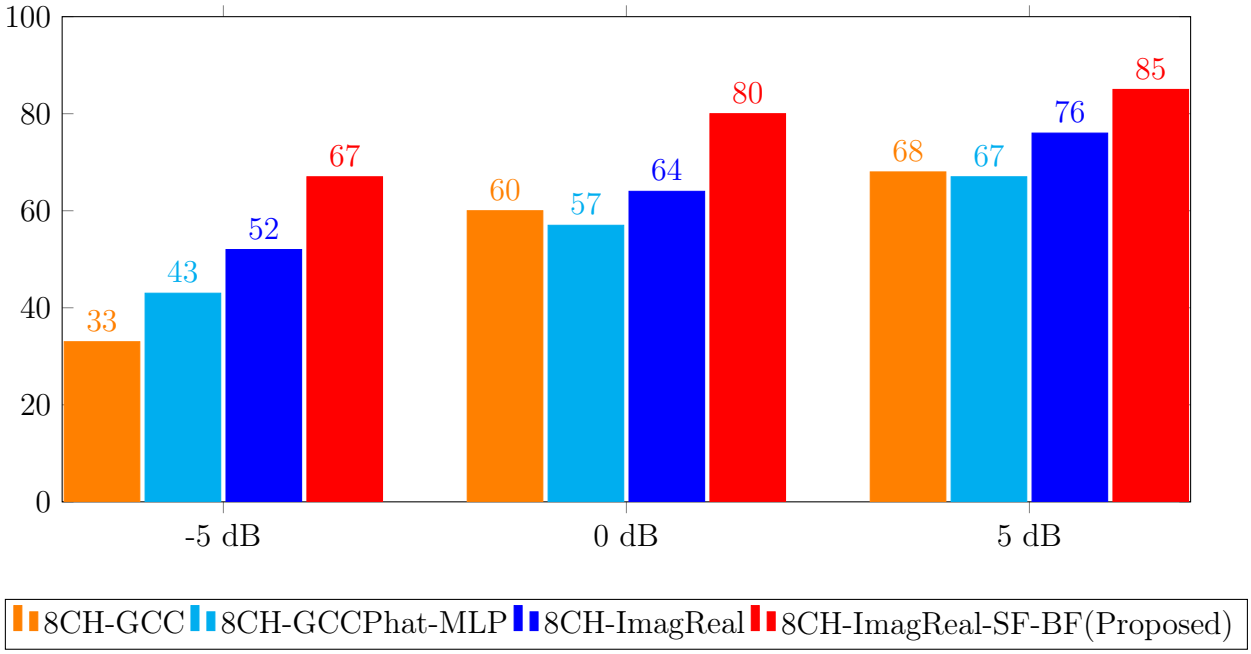


Figure 3.14: The real-time ACC results (%) under babble conditions.

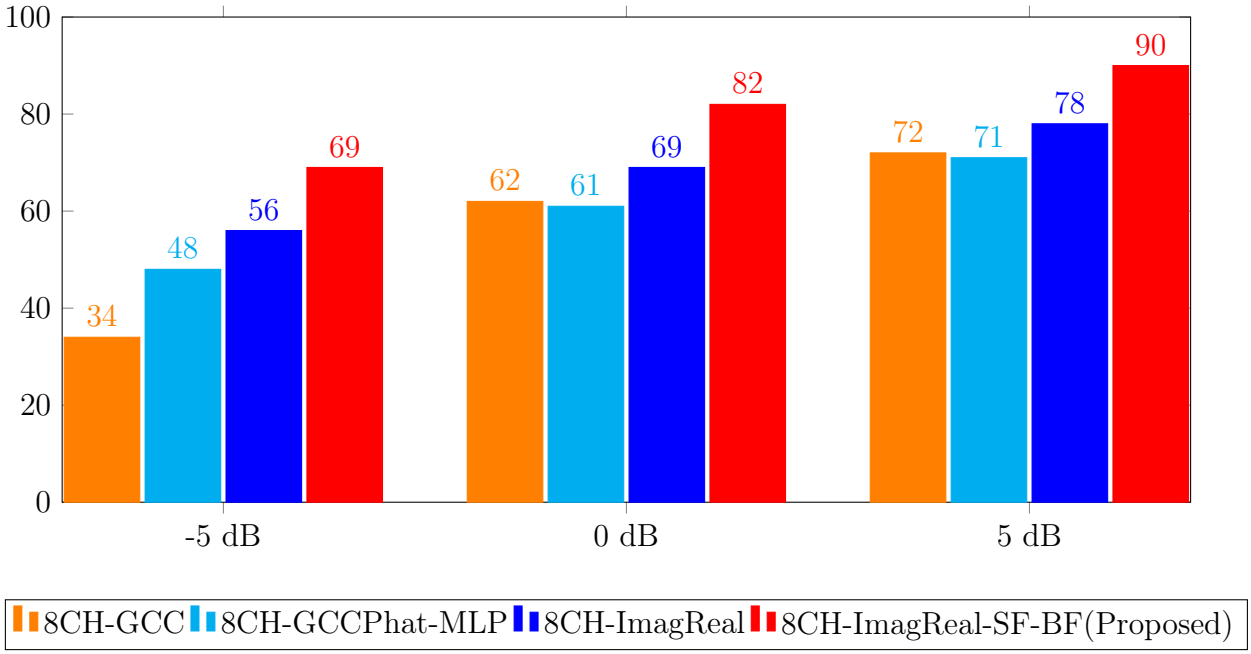


Figure 3.15: The real-time ACC results (%) under machinery conditions.

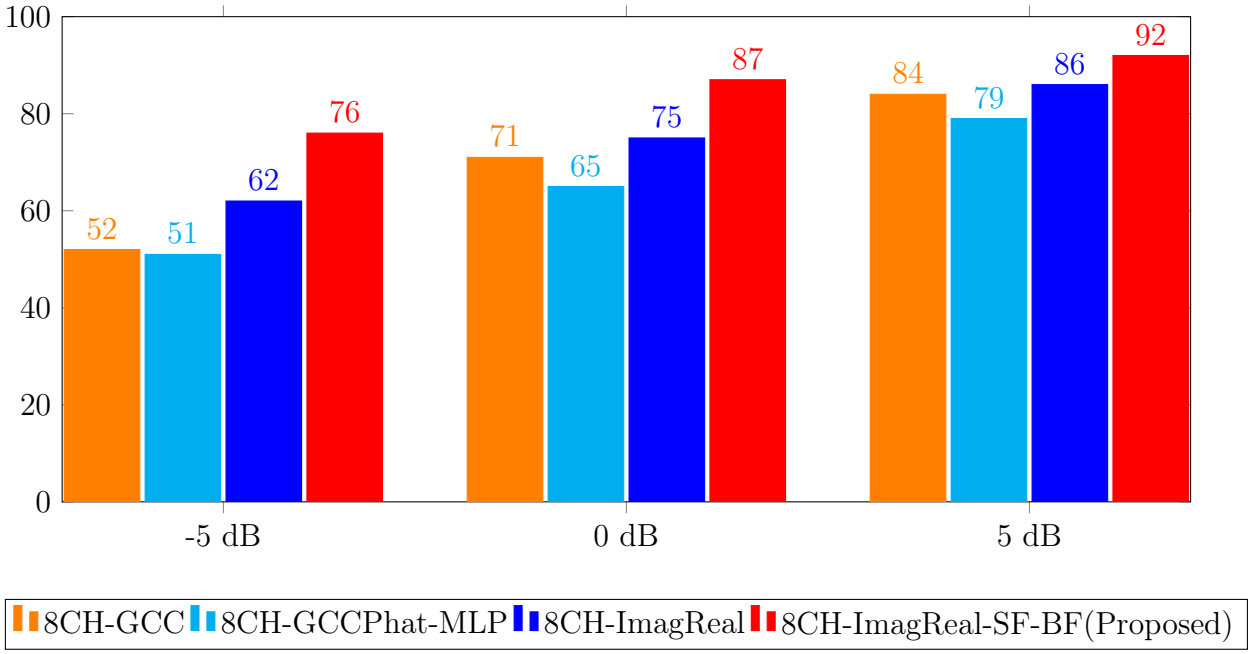


Figure 3.16: The real-time ACC_w results (%) under babble conditions.

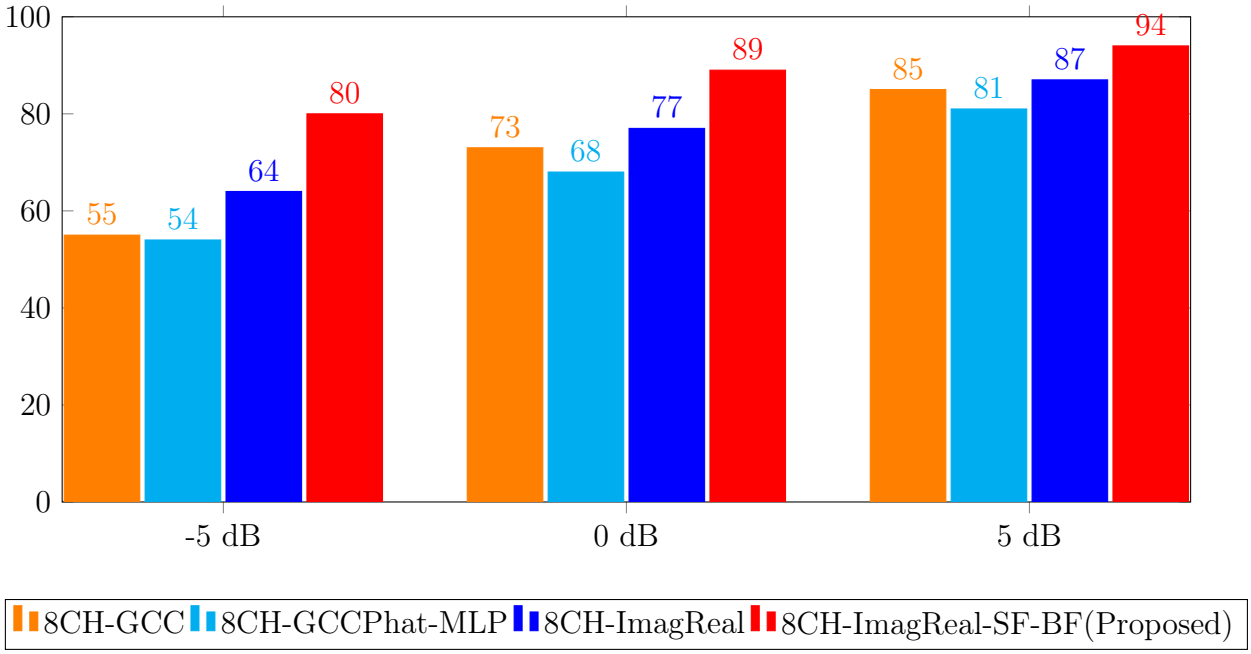


Figure 3.17: The real-time ACC_w results (%) under machinery conditions.

3.7.4 The Power Consumption of the Prototype Platform

To develop a robust SSL estimator in real-time, the power consumption is therefore important to consider. In our hardware setup, the capacity of the power bank sitting on the bottom (Fig.3.11) is 20k milliamps per hour. Our power consumption measurement has been completed with the fully charged power bank, the results are presented in Fig.3.18, where Y-axis shows the watts consumption per hour, and X-axis shows the methods. In Fig. 3.18, "IDLE" stands for the power consumption of the prototype operating system running without any extra processing or calculation. The total power consumption of the platform for the proposed method, including all processing stages, is only 1.15 watts per hour, slightly larger than the power consumption of 8CHGCCPhat- MLP. Since our setup is

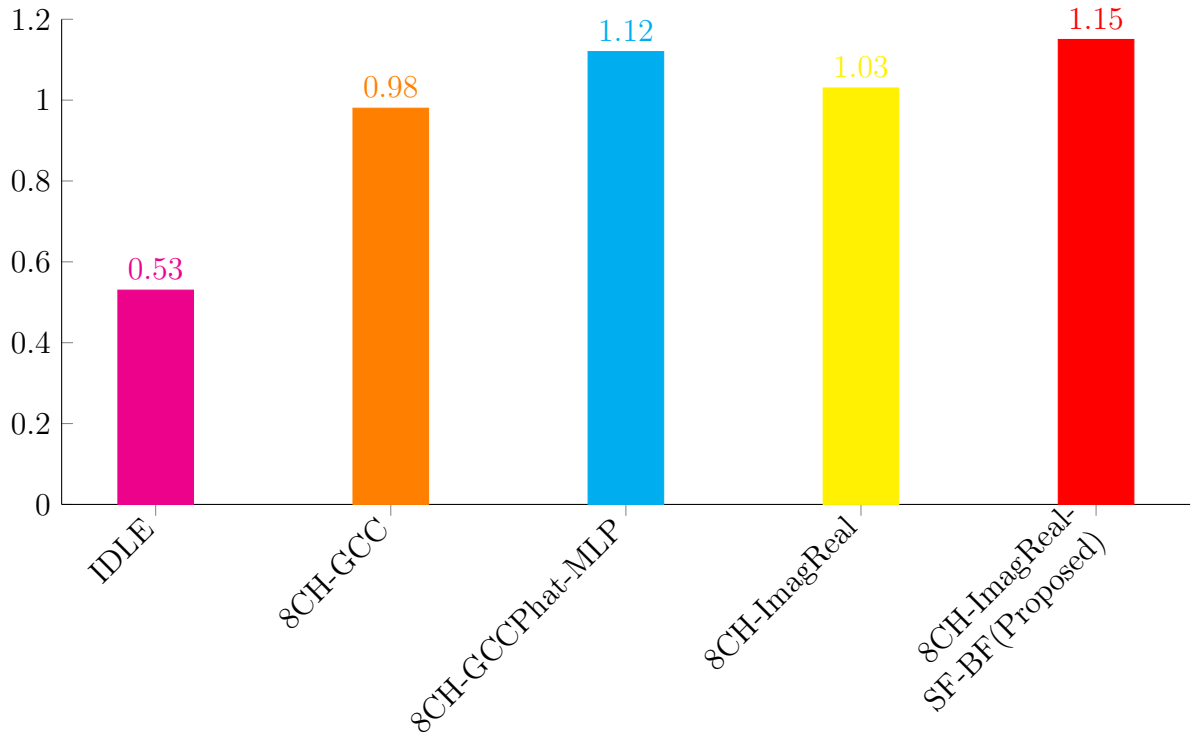


Figure 3.18: Power consumption of the prototype (watt hours).

only a prototype unit using the development boards, the power consumption shown here is much more than what is needed for the implementation of the proposed method. This is so since many other unnecessary modules unrelated to the proposed method are also running on the boards. The end-product, as a dedicated hearing improvement unit, will only need to keep and run the modules required for the implementation of the proposed method, hence the power consumption will be very small. Additionally, the size of the end-product will be much smaller and compact compared to the prototype platform.

3.8 Conclusion

In this chapter, we proposed a CNN-based SSL estimator using an eight-microphone UCA. Imaginary-real coefficients and spectral flux are used as feature set for the CNN model. Beamforming is used as well to enhance the SNR when computing the spectral flux. The offline and real-time results show that the proposed SSL method, as an augmentation method for imaginary-real coefficients CNN based DOA method, is scalable and robust under different types of noise and performs better than other neural network based estimators. A prototype platform for implementing the proposed method in real-time was also developed using a resource-constrained device, Raspberry Pi, plus an IoT development board. The prototype platform not only shows the robustness but also presents and establishes a real-time platform. The end-products including HI devices can be built based on the platform with a VAD (to "freeze the estimation" when no speech detected). Such products help to improve the hearing capability of people with hearing loss by identifying the direction and location of the speakers in noisy environments and where there maybe several people such as in a group meeting or a social gathering.

CHAPTER 4

NOISE ROBUST SINGLE MICROPHONE-BASED SOUND SOURCE DISTANCE ESTIMATION METHOD

Authors – Abdullah Küçük, Issa Panahi

The Department of Electrical and Computer Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

Corresponding author: Abdullah Küçük

4.1 Abstract

Supervised learning based methods for source localization are being adapted to different acoustic conditions and various microphone arrays. This chapter presents a single microphone based speaker distance estimation method. The method exploits a convolutional neural network (CNN) to classify the distance between speaker and microphone in predefined distance classes, from 75 cm to 3 meters (m) with 75 cm resolution. We propose two models in this paper: the first model classifies three predefined distances; 75 cm, 150 cm, and 225 cm; the second model is for four classes; from 75 cm to 3 m with 75 cm increments. We use a novel input feature set for speaker distance estimation – using real and imaginary parts of the short-time Fourier transform (STFT) for CNN-based distance estimation. Furthermore, we utilize Image Source Model (ISM) generated RIRs for training the model and publicly available speech files to test our model. The proposed method is also tested with two different background noises, namely babble (non-stationary) and machinery (stationary), at various signal-to-noise (SNR) levels of 0 dB, 10 dB, and 20 dB. The experiments show the generalization ability of the proposed method for unseen speakers and unseen environments. Furthermore, through experimental evaluation with simulated and also with measured acoustic impulse responses, the ability of the proposed distance estimation approach to adapt to unseen acoustic conditions and its robustness to unseen SNR is demonstrated. Another unique aspect of this work is a real-time implementation of the proposed method on an Android-based smartphone using its built-in microphone, needing no external hardware or component.

4.2 Introduction

Sound source localization (SSL) has a wide range of applications in signal processing. The research community mostly focuses on estimating the azimuth and/or elevation of the sound

source which is also called the direction of arrival estimation [34, 65]. Sound source distance estimation can be complimentary for SSL and viable for many applications, such as establishing wireless acoustic networks, automatic microphone array positioning and calibration, and speaker source positioning. One of the exciting and novel ideas is to establish a wireless acoustic network using multiple smartphones for hearing aid studies to further improve audio signal processing, e.g., beamforming, SSL, speech enhancement (SE).

Mobile devices, such as smartphones and tablets, have become an essential component of our daily lives and have a powerful processor, battery, memory, touch screen and display panel, speaker, and at least two microphones needing no external or additional components. It is also reported that smartphones are capable of different audio signal processing methods in real-time [6, 9, 10, 29, 52, 66]. Hence, we could establish our microphone array using multiple smartphones. However, the microphones' distance has a critical importance for audio signal processing algorithms using a microphone array. The proposed method can enable us to estimate the distance between two smartphones using only a single microphone. Several approaches have been considered to estimate the distance between microphone and signal source [67, 68, 69, 70, 71]. The authors of [67] exploit the statistical features and binaural cues for feeding to Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) in order to estimate the speech source. The method in [68] offers two different distance estimation methods based on binaural speech signal direct-to-reverberant energy ratio (DRR). DRR is very popular method for the distance estimation method for binaural signals. The works in [67, 68] propose source distance estimation algorithms using binaural speech signal. The distance estimation of speech source is not our goal here. Instead, we aim to estimate the distance of sound source, which can play a single tone sinusoidal wave, chirp signal or white noise, from a microphone. The other study in [69] exploits the time difference of arrival (TDOA) of sound for determining the known interspacing of microphones that are located along a fixed axis of a linear array. The method in [69] reports simulation

test results utilizing a linear array with five microphones. [70] utilizes three microphones for estimating the source direction and distance. Method of [70] exploits the geometry of the microphone array and the use of TDOA technique. The distance estimation is tested in [70] for four different distances, 2, 4, 6, 8 meters. The method in [71] uses a setup similar to that of our method. In [71], a microphone is mounted on a loudspeaker, and a real-time max likelihood method is used to estimate the distance between two loudspeakers playing back a stereo music signal. All the aforementioned methods have used microphone arrays for the distance estimation. Unlike these methods, we present and analyze here a distance estimation method using a single microphone.

This paper proposes a noise-robust single microphone-based distance estimation between a microphone and loudspeaker. The method requires a loudspeaker to play a single tone sinusoidal signal and a microphone to receive that sinusoidal signal. The proposed algorithm then processes the signal and estimates the distance between the microphone and the sound source. Based on our experimentation, as well as the listener's comfort level, the frequency for a single tone sinusoidal signal is determined to be 330Hz which provides an optimal accuracy for the distance estimation by proposed algorithm. Moreover, frequency-based prefiltering techniques are proposed which increases the range of estimated distance and enhances the robustness of proposed method in the presence of background noise and reverberations. The performance of proposed method is evaluated with and without the presence of background noise and reverberations. It is also evaluated using the data convolved with the measured room impulse responses (RIRs). We used RIRs recorded for real rooms [72] which are available to public. The signal model and proposed method are presented in Section 4.3. The experimental setup is given in Section 4.4. Section 4.5 discusses the performance analysis and test results of proposed method. Conclusion is in 4.6.

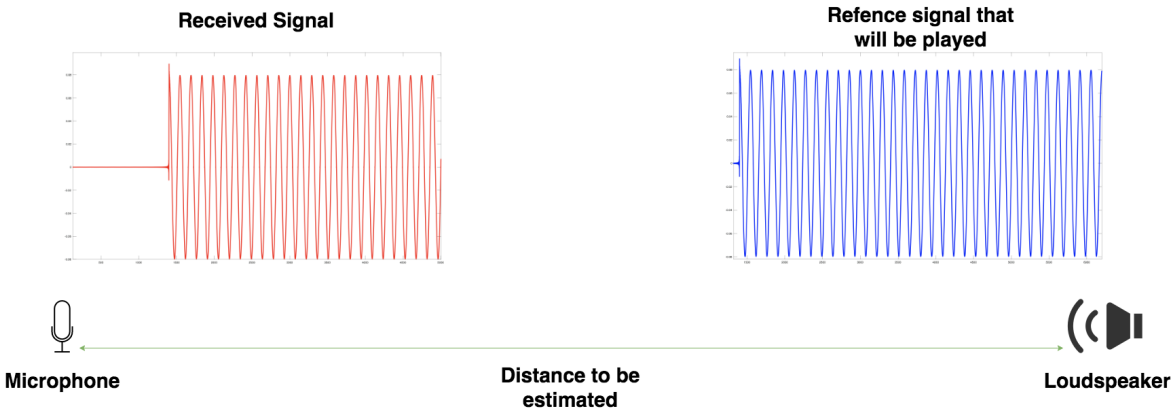


Figure 4.1: The scenario for the proposed distance estimation method

4.3 Proposed Method

The method is proposed for estimating the distance between a sound source (like loudspeaker) and a microphone, as shown in Figure 4.1. The loudspeaker plays a single tone sinusoidal signal, also called a reference signal in this paper, and the distance is estimated based on the captured sinusoidal signal by the microphone. As shown in Figure 4.1, the received signal has a time delay. The proposed method aims to calculate the time delay using cross-correlation between the reference and received signals. We also propose a frequency weighting function to increase efficiency of the proposed method. The proposed algorithm assumes that the loudspeaker and microphone are synchronized. In other words, the loudspeaker starts playing the reference signal and the microphone begins capturing the signal at the same time.

4.3.1 Signal Model and Proposed Method

We use a single tone sinusoidal signal as reference signal for our method.

$$x(n) = \sin(\omega_0 n) \tag{4.1}$$

and $\omega_0 = 2\pi f_0$ where $f_0 = F_0/F_s$. F_0 is frequency in Hz (330 Hz in our setup) and F_s refers to sampling rate. The received signal would be as following:

$$y(n) = \alpha\, x(n-\eta) + v(n) \tag{4.2}$$

where η is the time delay to be estimated. α is attenuation factor and $v(n)$ is noise and assumed additive and uncorrelated with $x(n)$. To determine the η, generalized cross correlation function between $x(n)$ and $y(n)$:

$$r_{xy}(m) = x(m) * y(-m) \tag{4.3}$$

m is index of time. (4.3) can also be written as following:

$$r_{xy}(m) = \alpha\, r_{xx}(m-\eta) + r_{xv}(m) \tag{4.4}$$

where $r_{xx}(m)$ is autocorrelation of $x(n)$ and $r_{xv}(m)$ is cross correlation between $x(n)$ and $v(n)$. Since $x(n)$ and $v(n)$ are uncorrelated, $r_{xv}(m)$ disappears. For ideal conditions (when there is not presence of signal distortion effect), $r_{xx}(m-\eta)$ ensures sharp peak for $m = \eta$ in (4.4) since the autocorrelation gets its maximum value at $lag = 0, r_{xx}(0)$ and α is constant. However, under non-ideal conditions, the effect of noise and reverberation causes spurious spikes [34]. The prefilter aims to ensure the sharper peak of $r_{xx}(m-\eta)$ than spurious peaks. Many different prefilter techniques are proposed for time difference of arrival (TDOA) estimation using generalized cross-correlation (GCC) like PHAT, Eckart, Roth, SCOT [34]. All these prefiltering methods assume two signals are exposed to noise and reverberation. However, the assumption for the proposed method is different from GCC. Only the received signal $y(n)$ is exposed to noise and reverberation for our case. For distance estimation, we propose two prefiltering techniques which normalize cross power spectral density function, $R_{XY}(\omega)$.

$$R_{XY}(\omega) = X(\omega)Y^*(\omega) \tag{4.5}$$

$X(\omega)$ and $Y(\omega)$ is the complex conjugate of reference and received signal, respectively. * denotes complex conjugate. The modified cross power spectra with the proposed prefilters are shown as following:

$$\ddot{R}_{XY}(\omega) = \frac{X(\omega)Y^*(\omega)}{|Y(\omega)|^\beta} \tag{4.6a}$$

$$\ddot{R}_{XY}(\omega) = \frac{X(\omega)Y^*(\omega)}{\sqrt{X(\omega)X^*(\omega)Y(\omega)Y^*(\omega)}^\beta} \tag{4.6b}$$

where the β represents power normalizing factor. (4.6a) and (4.6b) enable decreasing the impact of magnitude difference and focusing the impact of phase difference. Now, the modified cross correlation will be:

$$\ddot{r}_{xy}(m) = \int_{-\infty}^{\infty} \ddot{R}_{XY}(\omega)e^{j\omega m}d\omega \tag{4.7}$$

(4.7) is inverse Fourier transform and shows relationship between $\ddot{r}_{xy}$ and $\ddot{R}_{XY}$. The time delay can be calculated using (4.7):

$$\hat{\eta} = \underset{m}{\operatorname{argmax}}\, \ddot{r}_{xy}(m) \tag{4.8}$$

where $\hat{\eta}$ is the estimated time delay. Now, we can calculate the distance with the knowledge of the speed of sound.

$$\hat{d} = \frac{\hat{\eta}}{F_s}c \tag{4.9}$$

where $\hat{d}$ is the estimated distance. c is the speed of the sound. At $20°C$ $(68°F)$, the speed of sound in air is about 343 meters per second and sampling frequency equals 48 kHz for this work.

The period of the sinusoidal signal also has critical importance for the distance estimation algorithm. We need a large period of the signals so that the phase shift to be estimated falls within the fundamental period of the signal. This can be achieved either by a small frequency

of the single tone sine or a high sampling rate. The max sampling rate could be 48 kHz which is widely used in real systems. Also, our simulations show that the best frequency of the single tone sine for the proposed distance estimation algorithm is 330 Hz. The period for the single tone sinusoidal signal at 330 Hz and 48 kHz sampling frequency is 0.0333 seconds(s).

4.4 Experimental Setup

4.4.1 Performance Metric

Mean absolute error (MAE) is utilized for quantifying the performance of the proposed distance estimation algorithm.

$$MAE(m) = \frac{1}{N_S} \sum_{i=1}^{N_S} \left| d_i - \hat{d}_i \right| \tag{4.10}$$

where N_S is the number of signal frames evaluated and m represents the distance. d_i is the correct distance between a microphone and a loudspeaker. $\hat{d}_i$ refers to the estimated distance. We also use MAE_total which is the mean of $MAE(m)$ over distances.

4.4.2 Dataset and Experimental Setup

Clean (no background noise present) and noisy sinusoidal signals are used for evaluating the proposed method. The room impulse functions are generated via the Image Source Model (ISM) [73]. The room size is $21 \times 21 \times 5$ m, and a single microphone is placed at the center of the room. The loudspeaker is located at fifty six different distances. The distance varies 10 cm to 500 cm in 10 cm step size, and the step size becomes 50 cm between 500 cm to 800 cm. We have also generated noisy sinusoidal signals by adding real smartphone recorded noise to synthesized sinusoidal signals. Each sine signal is received for 2 seconds, and we use the entire 2 seconds data to estimate the distance between the microphone and loudspeaker. Each sinusoidal at a specific distance is mixed with ten different noise segments per noise

type and signal to noise ratio (SNR) levels. We have considered three noise types, namely; multi-talker babble (B), machinery (M), traffic (T). Babble and traffic noise files are chosen from DCASE 2017 database[50]. Actual noise in the Magnetic Resonance Imaging room with a 3-Tesla imaging system is used as machinery noise. More information regarding real noise recording can be found in [7]. We have also considered three SNRs, namely 0, 5,10 dB.

We have also used publicly available real-recorded Aachen Impulse Response (AIR) dataset [72] other than ISM generated RIRs in order to show robustness of the method in real environment conditions. The AIR dataset includes binaural recordings but we have used only one channel for our purpose. Five different rooms and eleven different distances from 50 cm to 868 cm are utilized for the experiment. The RIRs are convolved with 2 seconds synthesized sine wave at 330 Hz for the experiment.

4.5 Results and Discussions

The proposed method is evaluated using ISM simulated data within the range of 10 cm to 8 m. Since we were not able to find a method in the literature that estimates the distance between loudspeaker and microphone using a single microphone, we used the proposed method without prefiltering as a baseline, denoted Baseline in this paper, for our experiments. Table 5.7 displays the proposed method's MAE_total performance under different noise types and SNR levels and when there is no presence of background noise. The sine wave at 330 Hz mixed with ten different each noise type at three different SNRs, i.e., 0 dB, 5 dB, and 10 dB for Table 5.7. $Proposed\_\beta Mag$ and $Proposed\_\beta Scott$ represent the equation (4.6a) and (4.6b) respectively. We have tuned the β values to get the best performance. The β is 1.2 and 0.9 for $Proposed\_\beta Mag$ and $Proposed\_\beta Scott$, respectively. Table 5.7 displays presents that both prefiltering technique enable to decrease the MAE drastically.

Table 4.1: The proposed method total MAE performance under different noise types and SNR levels. Bolds represents the best performance. The MAE values are in cm.

	Clean	**Babble**			**Machinery**			**Traffic**		
	Clean	***0 dB***	***5 dB***	***10 dB***	***0 dB***	***5 dB***	***10 dB***	***0 dB***	***5 dB***	***10 dB***
Baseline	135.10	120.07	120.86	133.84	97.48	92.26	93.56	113.12	118.8	102.7
Proposed_βMag	**0.1038**	**0.1052**	**0.1048**	0.1044	0.1121	0.1094	**0.1044**	**0.1044**	**0.1039**	**0.1038**
Proposed_βScott	**0.1038**	**0.1052**	**0.1048**	**0.1043**	**0.1077**	**0.1073**	0.1045	**0.1044**	**0.1039**	**0.1038**

In table 5.7, while the baseline has around 120 cm MAE performance, both prefiltering methods have around 0.1 cm MAE performance for all the cases. Table 5.7 proves that the proposed method with prefiltering is robust to various noise types and SNR levels.

The second experiment is conducted to compare the prefiltering methods. The performances of both prefiltering methods are very close to each other in Table 4.1. Therefore, we have experimented with the proposed methods in the harder condition, i.e., -5 dB Machinery noise. The same procedure as in the first experiment is followed for the second experiment. Figure 4.2 displays the MAE for each distances for both proposed prefiltering methods. The left (red) figure shows the performance of the proposed method in equation (4.6a), and the right (blue) figure is for equation (4.6b). We have obtained undesired peaks the in the distance estimations of *Proposed* β*_ Mag*. Hence, the figure shows that the proposed method in equation (4.6b) has superior performance for low SNR with having maximum 0.401 cm MAE.

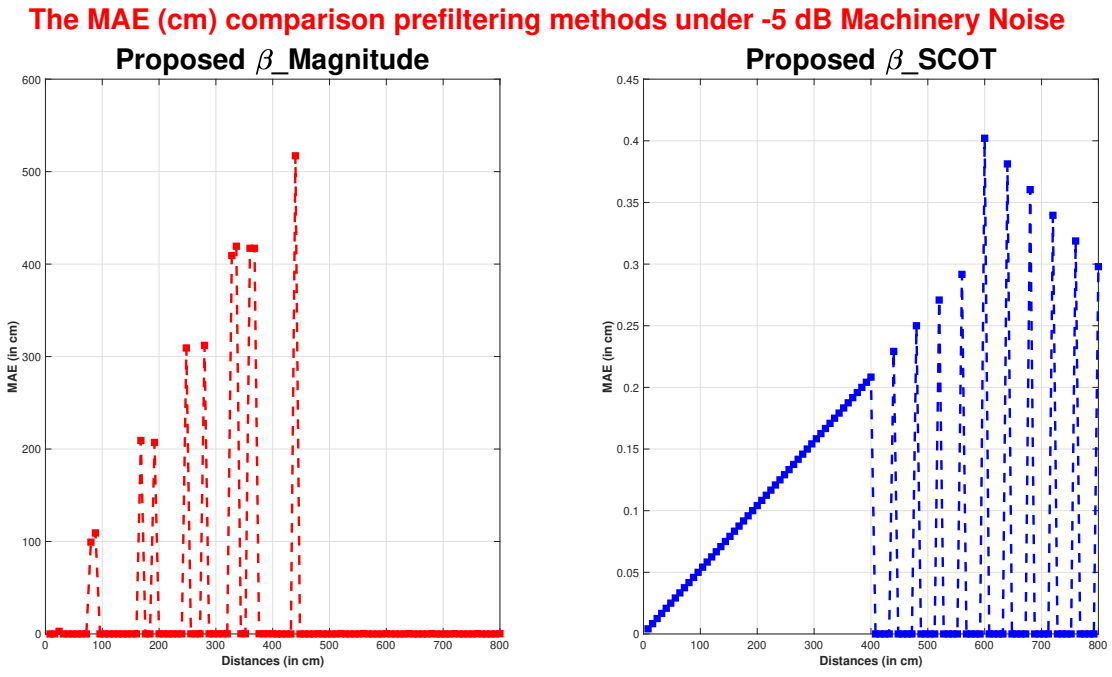


Figure 4.2: The comparison of the proposed prefilterings under -5 dB machinery noise

Table 4.2: The performance of the proposed method in real environment

Room	***RT_{60} in s***	***Distance in cm***	***Estimation in cm***	***Difference in cm***	***Error (%)***
Studio Booth	0.08	*50*	50.29	0.29	0.58
Stairway	0.20	*100*	99.87	0.13	0.13
Meeting	0.21	*190*	197.37	7.37	3.88
Aula Carolina	0.69	*300*	306	6	2
Lecture	0.72	*400*	397.37	2.63	0.66
Lecture	0.79	*556*	550.37	5.63	1.02
Lecture	0.80	*710*	701.5	8.5	1.20
Lecture	0.81	*868*	856.37	7.63	0.88

Table 4.2 shows the performance of the proposed method via real measured RIRs. The dataset used for Table 4.2 was explained in section 4.4.2. The rooms, reverberation time (RT_{60}), distance, estimation of distance,the absolute difference ($|correct\ distance - distance\ estimation|$), and the percentage error (4.11) are given in Table 4.2. The difference is low when the reverberation time is small, but it increases with increase of reverberation time. The maximum difference between correct and estimated distance is 8.5 cm. There could be an explanation for the high difference; the AIR dataset [72] was not designed for distance estimation purposes. The given distance is between the center of two microphone array and loudspeaker, and the distance between two microphones is 17 cm. However, we have used the right channel for our experiment. Hence, the given distance is not the exact measurement of the distance between the right microphone and loudspeaker. Moreover, the data collection setups in the AIR dataset are not fixed for all rooms. Considering this situation, the error of $\frac{17}{2} = \mp 8.5$ cm could be acceptable for the AIR database. Additionally, the maximum error percentage is 3.88; hence, we can say that the proposed method performs satisfactorily well in real environments with low and high reverberation time.

$$Error(\%) = \frac{|correct\ distance - distance\ estimation|}{correct\ distance} \times 100 \tag{4.11}$$

4.6 Conclusion

This chapter presents a single microphone based noise robust sound source distance estimation algorithm. The method estimates the distance between a loudspeaker which plays a single tone sinusoidal signal at 330 Hz and a microphone. The algorithm assumes that the loudspeaker and microphone are synchronized. We also offer two new prefilters in frequency domain which enables the method to increase its estimation range and reduce the estimation error. The performance of the proposed algorithm is evaluated in the presence of background noise and reverberation. Also, publicly available real measured RIRs are used for assessing the performance of the method in real environment conditions. All the conducted experiments show that the proposed algorithm would be a viable solution for single microphone based sound source distance estimation.

CHAPTER 5

SINGLE MICROPHONE SPEAKER DISTANCE ESTIMATION USING CONVOLUTIONAL NEURAL NETWORK

Authors – Abdullah Küçük, Issa Panahi

The Department of Electrical and Computer Engineering, EC33

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

Corresponding author: Abdullah Küçük

5.1 Abstract

This chapter presents a single microphone based speaker distance estimation method. The method exploits convolutional neural network (CNN) to classify the distance between speaker and microphone in predefined distance classes, from 75 cm to 3 meters (m) with 75 cm resolution. We propose two models in this paper: the first model classifies three predefined distances; 75 cm, 150 cm, and 225 cm, the second model is for four classes; from 75 cm to 3 m with 75 cm increments. We use a novel input feature set – using real and imaginary parts of the short-time Fourier transform (STFT) for CNN-based distance estimation. We use Image Source Model (ISM) generated RIRs for training the model and publicly available speech files to test our model. The proposed method is also tested with two different background noises, namely babble and machinery, at a various signal-to-noise (SNR) levels of 0 dB, 10 dB, and 20 dB. The experiments show the generalization ability of the proposed method for unseen speakers and unseen environments. Through experimental evaluation with simulated and also with measured acoustic impulse responses, the ability of the proposed distance estimation approach to adapt to unseen acoustic conditions and its robustness to unseen SNR is demonstrated. Another unique aspect of this work is a real-time implementation of the proposed method on an Android-based smartphone using its built-in microphone, needing no external hardware or component.

5.2 Introduction

Speech source localization (SSL) refers to identify the location of the target speech source in terms of azimuth, elevation, distance, or coordinate with respect to a fixed microphone array. SSL has vital importance on audio signal processing, such as beamforming [2, 3, 14, 74], speech enhancement (SE) [3, 24], speaker/speech recognition [4, 23], hearing aid [6, 26, 27, 75]. Sound source distance estimation (SSDE) determines the speaker's location

in terms of distance. SSDE can be complementary for well-known direction of arrival (DOA) methods to improve the effectiveness of SSL (which is SSL in terms of azimuth and/or elevation) [76, 77, 78]. SSDE can be utilized for a distributed ambient telephony system [17], smarthome [18] by selecting the nearest microphone, human-robot interaction system [19], intelligent hearing aids [20] by modifying gain based on distance.

Even though SSL is a hot topic in audio signal processing, researchers have mostly investigated methods for estimating azimuth and elevation of speech source. SSDE has drawn relatively less interest than azimuth and elevation estimation of speech source by the scientific by research community. Developments in machine learning and deep learning (DL) has led researchers to study SSDE algorithms to obtain SSDE's potential benefit for many systems such as distributed ambient telephony system, smarthome, human-robot interaction system, and intelligent hearing aids.

SSDE algorithms can be divided into two categories based on the sound source type. The first type could be loudspeakers. Since we can play specific signals such as single tone sinusoidal signal, white noise, and chirp signal via loudspeaker. Having a prior knowledge of played signals enables us to use many techniques and satisfactory distance estimation accuracy [79]. The second source type could be human talker, so the sound type is speech. Speech is a non-stationary signal and depends on the talker's age, vocal tract, excitation, gender, emotion and so on. Dependency on many variables makes SSDE quite complicated. In this work, we focus on speaker distance estimation using a single microphone.

Estimating the absolute distance of the speaker is more challenging in a closed area because of reverberation. This situation leads the researchers to develop methods with prior information on the room properties [80, 81]. The authors of [80] use previously measured room impulse responses for the estimation. The method in [81] utilizes specific room characteristics such as the wall's surface area and absorption coefficients for estimating the distance. However, the detailed knowledge of a room characteristics is mostly not available in

practice. Another distance estimation approach is learning-based methods [82, 83, 84, 85, 1], which do not require prior knowledge of the room information. The first learning-based method for distance estimation of a sound source using a pair of microphones is offered in [82]. The method in [82] requires a few training data points to fit the model. The power ratio of the direct and reflected signal is utilized in [82]. The method in [82] does not require prior information but needs a few training data. The second learning-based method is for distance estimation of the binaural speech signal [83]. The method in [83] utilizes statistical features extracted from the speech signal and their binaural cues. The standard deviation of the difference of magnitude spectra of the left and right binaural signal is defined and exploited as a feature in [83]. Gaussian Mixture Model (GMM) and Support Vector Machine (SVM) are the classifiers in [83]. The other recent method in [84] proposes a solution for the problem of distance estimation using binaural hearing aid microphones in reverberant rooms. The authors of [84] define two approaches to estimate the Direct-to-Reverberant Energy Ratio (DRR) of binaural signals. The first method for estimating DRR utilizes interaural magnitude square coherence, whereas the second one is based on stochastic maximum likelihood beamforming. The authors of [84] also offer two calibration procedures to estimate the distance. While the first calibration method utilizes the reverberant room's critical distance, the second procedure uses the listener's own voice. The methods in [83, 84] are for binaural signals, especially for hearing aid studies. On the other hand, it is shown that smartphones can be an assistive device for hearing aid studies [7, 8, 9, 10, 29]. Therefore, single microphone based distance estimation can be viable for hearing aid applications. Single microphone based speaker distance estimation methods may be scarce in the literature. We could find only two works that have been published regarding speaker distance estimation using single microphone. The first single microphone based algorithm is proposed in [85]. The method in [85] extracts a couple of statistical and source-specific features from the speech signal and uses a pattern recognition method, GMM,

as a distance estimator. The performance of the proposed method is evaluated when the speaker is at five different locations for known/unknown cases [85]. For the latter method, the average performance accuracy is 72.4% and 66.8% for known and unknown cases, respectively. Another recent method in [1] exploits the power of DL methods for talker distance estimation using a single microphone. The author of [1] formulates the distance estimation problem as an image classification problem. The method in [1] utilizes a hybrid network involving a convolutional neural network (CNN) and recurrent neural network (RNN) as a classifier and a log-scaled mel spectrogram as a feature set. The model attempts to classify three predefined distances; 1m, 2m, and 3m. The author of [1] also states that further study is required to improve the model's generalization ability. A method for single microphone based speaker distance estimation which has satisfactory performance in a various realistic environments is required to be investigated.

A convolutional neural network (CNN) based speech distance estimation using a single microphone is proposed in this work. We use the real part and imaginary part of the short-time Fourier transform rather than hand-crafted features, such as DRR, as a feature set for CNN. The proposed CNN is trained with recorded speech signal convolved with the Image Source Model (ISM) generated room impulse responses (RIRs). The proposed method has satisfactory accuracy performance for unseen speakers and environments. We have also shown the performance of the proposed method using the dataset, which is publicly available RIRs and speech files. Additionally, we have investigated and improved the proposed method's performance under stationary and non-stationary background noises. Moreover, we have implemented the proposed method on the Android-based smartphone. The real-time implementation proves the generalization of the proposed method for the unseen environment and speaker. The implementation also shows the effectiveness of the method in terms of computational complexity. Simultaneously, real-time implementation operation enables us to track the speaker's distance with respect to the microphone. The implementation details of the real-time application are also shared in this chapter.

5.3 Problem Statement

In this work, we focus on using a single microphone, built-in and available on almost all smartphones, to estimate a speech source's distance under low signal to ratios (SNRs). The smartphone alone is used in our approach with no external microphone(s) and hardware of any kind. While the Android operating system (OS) allows using multiple microphones simultaneously, iOS OS permits using only one microphone for audio applications. To address both OSs, we study on single microphone based SSDE.

SSDE using a single microphone is one of the most complicated problems in audio signal processing since speech is nonstationary and has many dependents such as vocal tract, excitation, age, gender, etc. Also, one channel input does not carry enough information to locate speaker distance precisely. These reasons lead us to formulate the SSDE problem as a classification problem with reasonable resolution rather than a regression problem. In this work, we offer two solutions for the different application requirements; one is for up to 3 meters (m) with 75 cm resolution, the second one is for up to 2.25 m with 75 cm resolution. The first model should be chosen for the lower range and relatively high performance; the second model is for the higher range.

5.4 Proposed Method

The proposed block diagram for a single microphone based speaker distance estimation algorithm is presented in Figure 5.1. The proposed method requires a 2.01 s speech signal per estimation; therefore, the block diagram involves a buffer to store. The proposed CNN model is trained with only speech frames; hence, it requires a voice activity detector (VAD) to label the speech frames. We also filter out the content higher than 8 kHz of speech in order to decrease the computational complexity. The details regarding the proposed method can be found in the following subsections.

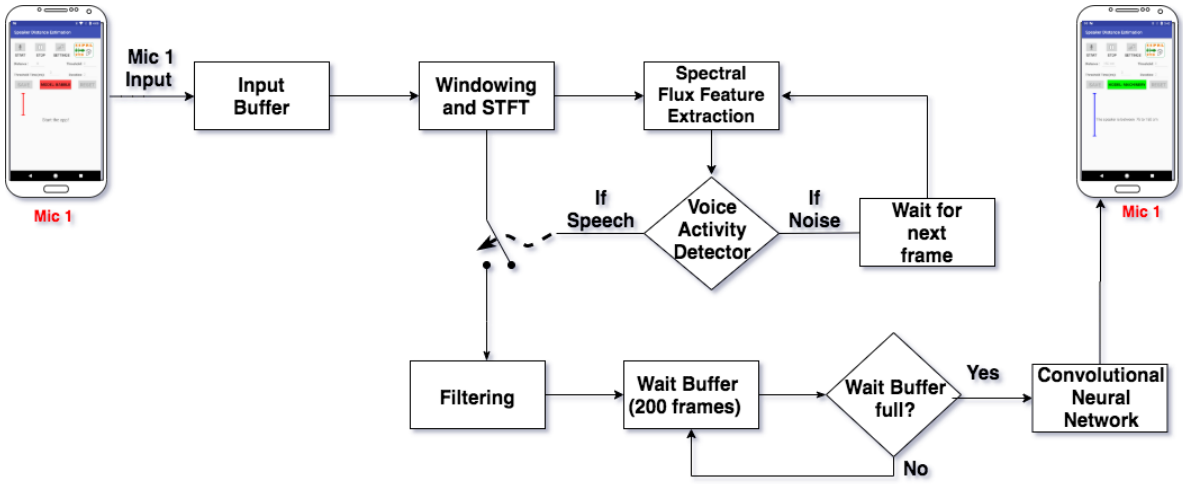


Figure 5.1: Block diagram of smartphone-based real-time processing modules in the proposed CNN-based Speaker Distance estimation application

5.4.1 Model Architecture

The proposed architecture for three class classification is shown in Figure 5.2. The starting point for the architecture is the general architectural principles of the Visual Geometry Group (VGG, a group of researchers at Oxford who developed this architecture) models for image classification [86]. We have optimized the proposed architecture for the speaker distance estimation problem after many trials. The model has an input layer, six convolutional layers, three max-pooling layers, two fully-connected layers (FCL), and an output layer in order. The proposed architecture consists of stacked two convolution layers with 3×3 kernels followed by max-pooling layers. Together, these layers are called block in this paper. The proposed method has three blocks. $32, 64, 128$ are the number of filters in convolutional layers for first, second, and third blocks, respectively. These filters (kernels) enable the network to extract the local information for the distance estimation. The max-pooling layer is for reducing computational complexity. In the proposed model, the max-pooling layer takes the max value in the 2×2 matrix. Next, the output of the max-pooling layer is flattened and fed to the FCL. Each FCL has 64 nodes with rectified linear unit [46]. SoftMax is utilized in the output layer, which has three nodes since we have three classes. We also use the dropout

rate as 0.3 between blocks and FCL layers to avoid overfitting. Layer 1 and 2 regularization was also applied to convolutional layers and FCL for a possible overfitting issue. The CNN architecture used for four classes is pretty much is the same with Figure 5.2. We utilize 0.5 as dropout rate and four nodes with Softmax activation function in output layer for the architecture for four classes case. There are $2,356,612$ trainable parameters for both the proposed CNN models.

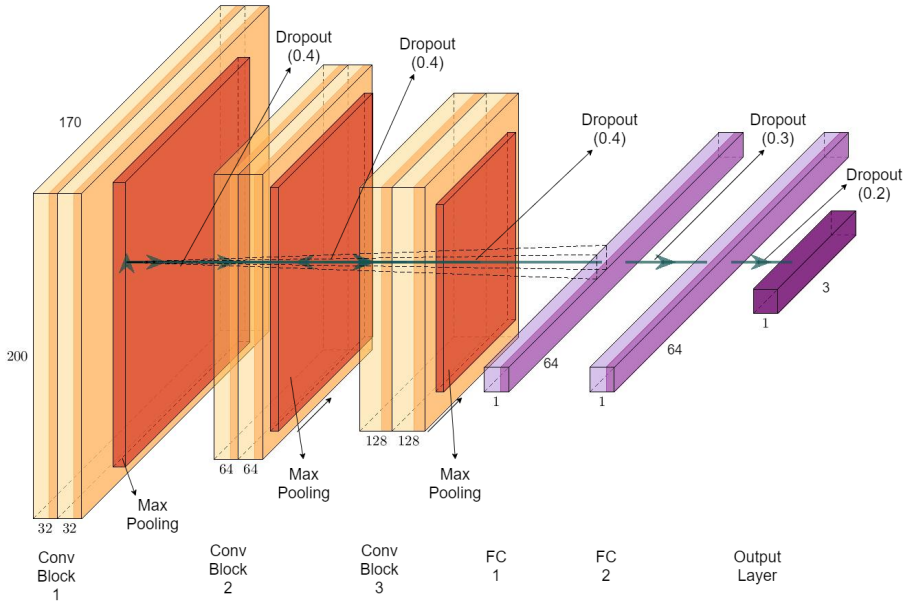


Figure 5.2: Block diagram of CNN based distance estimation architecture.

5.4.2 Feature Extraction

The input feature set has a direct effect on DL algorithm performance. Therefore, the input feature selection is one of the critical points for the DL task. We have used real and imaginary parts of short-time Fourier transform (STFT) of speech as an input feature set for DOA estimation in our previous work, and we obtained promising results [7]. We use the same feature set for speaker distance estimation since it is one of the rawest features obtained

from speech and enables filters in convolutional layers extract more useful information for the distance estimation. The STFT is formulated by:

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{5.1}$$

$X(m,\omega)$ is a complex STFT. $x[n]$ is the input signal and $w[n-m]$ is a window to obtain small frames of the speech signal. The sequence $f_m = x[n]w[n-m]$ is a short-time part of the speech signal $x[n]$ at time m. The real and imaginary part of $X(m,\omega)$ is used as an input feature set for the proposed method. The following matrixes show input feature sets.

$$\begin{array}{cc} \textbf{Feature channel 1} & \textbf{Feature channel 2} \\ 200 \updownarrow \begin{bmatrix} Imag.\,part\,of\,X(m,\omega) \\ \vdots \end{bmatrix} & \begin{bmatrix} Real\,part\,of\,X(m,\omega) \\ \vdots \end{bmatrix} \end{array} \tag{5.2}$$

The dimension of each feature set is represented by $N_F \times N_C$. N_F depends on the number of frames which are used for an estimation. N_{FFT} is the number of the fast Fourier transform (FFT) points. Since most speech contents sit up to 8 kHz in the spectrum, we get up to 8 kHz frequency response of data. N_C equals to $8000/(F_s/N_{FFT})$ where F_s is sampling frequency. For 48 kHz sampling rate and 20 ms frame length, N_C is 170, and N_F is 200 for this work.

5.4.3 Voice Activity Detector

In our daily lives, we are exposed to different kinds of noises. Since the task is speaker distance estimation, a classifier is required for discriminating noise and speech frames. This classifier is called voice activity detector (VAD). The proposed model is trained via only speech frames. VAD, hence, labels speech segment (includes clean or noisy speech), which is required by the proposed method. If the incoming frame is labeled as the speech by the VAD, the frame is stored for the pretrained model making a new estimation; else, the distance estimation from the previous estimation is retained:

$$\hat{\hat{d}}_i = \begin{cases} \hat{d}_{i-1},\ if\ VAD = 0\ (Noise) \\ \hat{d}_i,\ if\ VAD = 1\ (Speech) \end{cases} \tag{5.3}$$

Where distance $\hat{\hat{d}}_i$ represents the corrected the distance estimate after VAD for i^{th} frame. Thus, the VAD prevents the model from estimating the distance with noise-only frames. In this work, we utilize a simple single-feature-based VAD, which shows satisfactory performance in our previous works [7, 27], to reduce the real-time processing burden. That is, 'Spectral Flux (SF)' is used as a feature for VAD [27]. SF is defined by:

$$SF(k,i) = \frac{1}{N} \sum_{k=-\infty}^{\infty} \left(\left|X_i[k]\right| - \left|X_{i-1}[k]\right| \right)^2 \tag{5.4}$$

for k^{th} frequency bin and i^{th} frame. $k = 1, 2, \ldots, N$. $|X|$ denotes the magnitude spectrum of X. A simple thresholding technique is given by:

$$VAD(i) = \begin{cases} 0\ (Noise),\ if\ SF(k,i) < \xi \\ 1\ (Speech),\ if\ SF(k,i) \geq \xi \end{cases} \tag{5.5}$$

where ξ is the calibration threshold.

Two parameters enable VAD robust under nonstationary noise. The first parameter is decision buffer, D, which makes a VAD decision when D contiguous frames are detected as speech. The second parameter is called a threshold, T, which initially determines how many frames are assumed as noise.

5.5 Real Time Implementation on Android Based Smartphone (Pixel)

This section presents the real-time implementation of the proposed speaker distance estimation application on the Android Pixel smartphone.

5.5.1 Offline Training

The model is trained and will then be put on and implemented by the Android-based smartphone. The dataset generation process is given in section 5.6.2. The speech files with babble and machinery noise background noises are used for real-time; hence, we have two pretrained model in the app, users can choose one of them based the environment they are in. After the data generation process, TensorFlow [22] is utilized to implement the proposed architecture and train the model. The reason for using TensorFlow is that it has a C/C++ API, which can be used on Android platforms.

5.5.2 Real-time Implementation

The detailed block diagram of real-time processing modules in the proposed method is given in Figure 5.1. The Android-based smartphone captures a 20 ms signal frame at 48 kHz sampling rate. Input buffer stores up to 50% overlapped 20 frames of captured signals. Next, we pull out a frame and apply Hanning window. The short-time Fourier transform (STFT) is utilized for converting to the frequency domain. Spectral Flux (SF) feature is extracted via equation (5.4) in order to label the incoming frame as speech or noise. Threshold and Duration parameter for VAD (section 5.4.3 for details) are user-definable to give users the flexibility to adapt the app in different environmental conditions. After VAD labels two hundred speech frames (this makes 2.01 s of time using 50% overlapped 20 ms data), we apply a filter to take out information up to 8 kHz. After filtering, the data is forwarded to the Wait buffer. When the Wait buffer reaches two hundred frames, the data, whose size is 200×170, feed the pretrained model. The proposed model gives an output of 1×3 array; each value represents each class's probability. Next, we find the maximum of the output of the CNN model. Finally, we use the similar procedure in equation (5.10) to update the Graphical User Interface (GUI), which is presented in Figure 5.3.

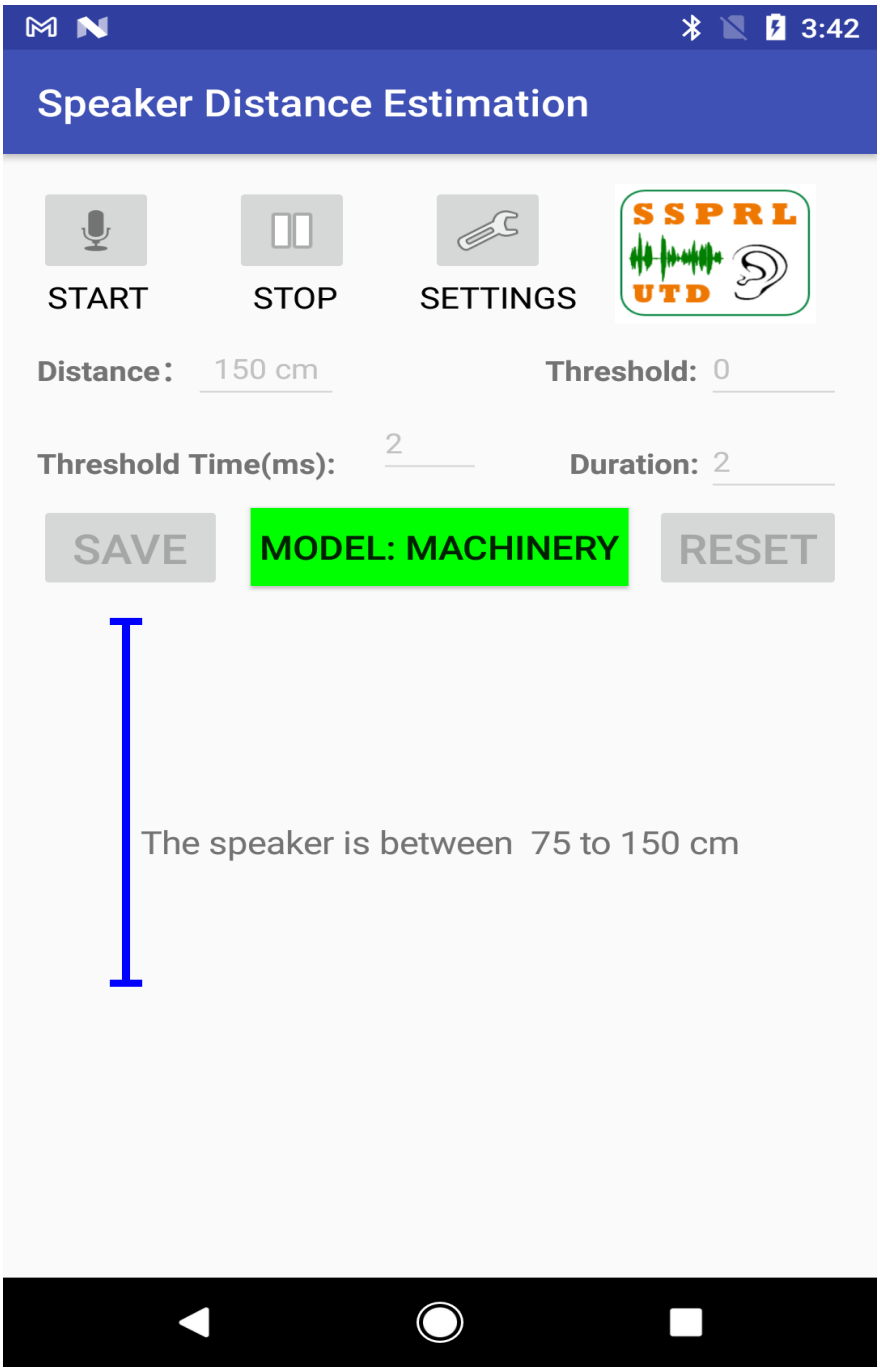


Figure 5.3: GUI display of the developed android app for Speaker distance estimate on Android Pixel 1.

As shown in Figure 5.3, the proposed app has four main buttons to handle all operations in the app. The application begins and halts using the 'Start' and 'Stop' buttons, respectively.

The 'Setting' button is designed for user-definable VAD. Users can change the Threshold and Duration parameters based on the environment that they are in. The estimated distance by the proposed method is displayed by text and graphical. As in Figure 5.3, the blue bar is next to the text. Different colors for the bar are used for different distances, i.e., red is for the distance is up to 75 cm, blue is when the estimated distance between 75 and 150 cm, and green is for the distance is more than 150 cm. The fourth and last main button is the pretrained model selection button. This button is for selecting the model trained with either babble or machinery background noise. The app users can prefer the model based on which environment they are in. For example, if they are in a restaurant or cafe, the model trained with babble noise is convenient. The model selection also can be made automatically by inserting noise classification block before the CNN inferencing block in the pipeline in Figure 5.1.

5.5.3 Implementation Highlights

We have applied three critical parameters, which we determined in our previous work [7], to run inference on an edge device, i.e., Android Pixel 1 for this work. Our proposed model can be run on any Android-based smartphone, which has at least one microphone. The parameters, as mentioned earlier, are wait buffer, reshaping, and the model size. The speaker distance estimation problem depends on many variables such as gender, age, voice tract, as we mentioned before. Depending on many parameters makes the speaker distance estimation problem sophisticated. Hence, single-frame based estimations do not work well as we presented in section 5.7.1. This situation leads us to define a buffer, namely Wait Buffer, in order to estimate using multi-frame, i.e., two hundred frames for our case. The next important parameter for real-time implementation is reshaping. The pretrained model requires the data with a shape of $2 \times 200 \times 170$, which is three dimensional. However, handling three-dimensional reshaping in an Android environment could be problematic. Because of

this situation, we have added the Reshape layer into our model. Now, the model expects a data shape of 200×340. Last but not least, there is a size limit for an application in the Android platform. Since we embed the pretrained model into the app, the model size should be as less as possible. To do that, we have decreased the number representation from 32-bit to 8-bit using the quantization technique. Decreasing number representation also reduces computational complexity. The price of this operation is only very little loss of accuracy.

5.6 Experimental Setup

We present the experimental setup and evaluation for the proposed single microphone and CNN based speaker distance estimation (SDE) method in this section. The performance metrics and experimental setup for simulated and publicly available data are also explained in this section.

5.6.1 Performance Metrics

There are four key classification metrics, namely Accuracy, Recall, Precision, F1 score. These metrics are calculated using confusion matrix. The confusion matrix displays not only the performance of the model, but also which classes are being predicted correctly and incorrectly, and what type of errors are being made [87]. Table 5.1 illustrates the confusion matrix.

Table 5.1: The confusion matrix

		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Accuracy measures how well the model predicts correctly. However, only accuracy is not enough for evaluating the performance of the model. We use recall (sensivity) and

precision in order to obtain the effects of false negative and false positive predictions on the performance. F1 score stands for the harmonic mean of the model's precision and recall. The formulas are shown as (5.6)-(5.9).

$$Accuracy\ (\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{5.6}$$

$$Recall\ = \frac{TP}{TP + FN} \tag{5.7}$$

$$Precision\ = \frac{TP}{TP + FP} \tag{5.8}$$

$$F1\ score\ = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{5.9}$$

5.6.2 The Distance Dataset

We have generated the distance dataset by convolving publicly available speech files with the generated and measured room impulse responses (RIRs). The details about the datasets are given in the following subsections.

The Speech and Noise Files

To obtain a generalized supervised deep learning model, the dataset should involve a diversified amount of data. The distance dataset should have two distinct properties: the first is a wide range of speakers since the distance estimation depends on the age, gender, vocal tract of the speaker. The second distinct property is environmental conditions, which will be explained in the next section. In order to have a wide range of speakers in our dataset, we have chosen speech files from three different datasets; TIMIT [48], AN4 [88], and Microsoft deep noise suppression challenge [89]. [88] is designed for speaker identification purposes. It has 74 different speakers in it, and the female and male talker number is almost equal.

TIMIT is one of the popular datasets in audio signal processing, and we have selected ten speech files from it. [89] is recently published for noise suppression task. We have used this dataset because it includes recorded speech files in different conditions with a different speaker. Sixteen speech files are selected from [89]. We have a total of a hundred speech files, and the number of female and male talker are almost the same. We have not included children's speech files in our dataset. The total length of speech files is 492640 seconds(s). Last but not least, the silence portion of the speech signal is using VAD, explained in section 5.4.3, removed for experimental evaluation, since the nonspeech contents lead the model performance degradation.

Noise files are smartphone recorded noise files. Noise files were selected from the DCASE 2017 challenge database [50], which includes real recordings. Vehicle ('Traffic') noise and multi-talker babble ('Babble') noises are selected from the DCASE 2017 database. In the dataset, we also used actual noise in a Magnetic Resonance Imaging (MRI) room with a 3-Tesla imaging system [51] to incorporate a machinery noise ('Machinery') containing a strong periodic component. To generate noisy mixtures, clean speech and noise files are mixed at different SNRs levels.

The Simulated RIRs

As it is mentioned in the previous section, the distance dataset should be designed with different environmental conditions. In order to do that, we have generated a hundred rooms with various reverberation times using the Image Source Model (ISM). The room dimension randomly changes from $6 \times 6 \times 3$ to $10 \times 10 \times 5$. The reverberation time (RT_{60}) also changes from 0.2 s to 0.7 s based on the room size. In other words, RT_{60} is proportional to the room size. A microphone is located in the center of the room, and a speech source is located in four different positions in the room. The speech source locations are 75 cm, 150 cm, 225 cm, 300 cm away from the microphone. The details regarding ISM generated RIRs can be seen in Table 5.2.

Table 5.2: ISM generated RIRs

The Generated RIRs Details	
Room Size	*Randomly changes from* $6\times6\times3$ *to* $10\times10\times5$
Number of rooms	*100*
Microphone position in room	*Center of the room*
Source-microphone distance	*Four positions in each room: 75, 150, 225, 300 cm w.r.t the microphone*
$\boldsymbol{RT_{60}}$	*Randomly changes from 0.2 s to 0.7 s.*
Sampling Frequency	*48 kHz*

The Measured RIRs

Aachen impulse response (AIR) public dataset [72] is used to show the proposed method's generalization ability. Even though the dataset's main application field is dereverberation, we can use the RIRs for distance estimation. As the dataset [72] includes measurements with different distances. RIRs of five different rooms with various distances from [72] are utilized for test data generation. The details of the RIRs that have been used for our experiment can be seen in Table 5.3.

Table 5.3: Aachen Impulse Response Database

Room	***Source-Microphone Distance (m)***	$\boldsymbol{RT_{60}(s)}$
Studio booth	*0.50*	*0.08*
Stairway	*1.00*	*0.18*
Meeting room	*1.90*	*0.21*
Aula Carolina Aachen	*3.00*	*0.35*
Lecture room	*4.00*	*0.72*

The Dataset Generation

Dataset generation is one of the critical processes for DL based method. While eighty rooms and ninety-five speech files are used for training, twenty rooms and five speakers are used

Pseudo code Part 1: *The Generated RIRs*

No Requirement

Randomly choose R room dimensions between $6 \times 6 \times 3$ and $10 \times 10 \times 5$
Assign randomly selected $RT_{60} \in [0.2, 0.7]$ based on room dimension
Define speech source locations L_S as 75, 150,225, 300 cm away from the microphone
for each r $\in$ R **do**
    for each l $\in L_S$ **do**
        Simulate RIR for one distance and single mic via [73]
        Save RIR
    end for
end for ▷ Number of RIR files = $R \times L_S$

for testing. The proposed method's data generation procedure is presented in the form of two-part pseudocode in Pseudocode Part 1 and 2. Pseudocode Part 1 describes ISM based RIR generation. Firstly, we randomly select a hundred room dimensions between $6 \times 6 \times 3$ and $10 \times 10 \times 5$ in Pseudocode Part 1. Next, reverberation times (RT_{60}) for the rooms are randomly but room size dependent assigned. Then, RIRs are generated based on room size, RT_{60} for four different distances between a microphone and a speech source, namely $75, 150, 225$, and 300 cm. Finally, we save the RIRs.

The dataset generation procedure for the convolutional neural network (CNN) is given in Pseudocode Part 2. The RIRs with various acoustic conditions and speech files are required for Pseudocode Part 2. Please recall that the silence portions of the speech files are removed using the VAD, explained in section 5.4.3. The proposed method requires two seconds (s) of speech data for an estimation. Short-time Fourier transform (STFT) is utilized for the dataset realization. The frame length is 20 ms, and 50% overlap is used. The window is Hanning window. The STFT of two hundred frames at 20 ms are saved in tfrecord format, is a simple format for storing a sequence of binary records in TensorFlow machine learning framework [22]. The details of the procedure are given above in Pseudocode Part 2.

Pseudo code Part 2: *The Generated Dataset*

Require: N_R RIR files and N_S speech files
Optional: Noise file (n) and SNR value in dB
for each $n_s \in N_s$ **do**
  for each $n_r \in N_R$ **do**
    s = n_s convolve with n_r for each distance
    if noise file and SNR value is provided **do**
      s = s + n at desired SNR
    end if
    for each i $\in N_F$ (Number of frame) **do**
      $x_{frame} = win \times s_{frame}$ ($\triangleright s_{frame}$ is 50 % overlapped 20 ms frame and win is Hamming window)
      X = FFT(x_{frame})
      X_{Imag} = Imag(X) and X_{Real} = Real(X)
      Dataset[row_{idx},:] = [$X_{Real}|X_{Imag}$]
      if row_{idx} equals to two hundred **do**
        Assign label to dataset
        Save dataset in tfrecord format
        row_{idx}=0
      end if
    end for
  end for
end for

5.7 Experimental Evaluation

In this section, different experiments with simulated and measured RIRs as well as noisy data at various SNRs are presented to objectively evaluate the performance of the proposed method . The dataset details and generation process for all the experiments are given in section 5.6. For all the experimental evaluations except the one presented in section 5.7.2, the proposed method attempts to classify three categories (when the distance are 75 cm, 150 cm, and 225 cm). In section 5.7.2, we also consider four case scenario (when the distance are 75 cm, 150 cm, 225 cm and 300 cm). To form an input feature set, the input signals are transformed to the STFT domain using frame length 20 ms with 50% overlap and number of FFT ($N_{FFT} = 1024$) and filtering out high-frequency components, resulting in K = 170. The

sampling rate of the speech signal is $F_s = 48$ kHz, and multi-frame (two hundred frames) based estimations are used for all the experiments. We have four test cases for the accuracy performance evaluation of the proposed method. The cases are as following:

1. *Known Speaker-Known Environment (K.S-K.E):* The test data involves the same speech files and RIR files in training data

2. *Unknown Speaker-Known Environment (U.S-K.E):* Only the same RIR files are used for both training and test data.

3. *Known Speaker-Unknown Environment (K.S-U.E):* Only the same speech files are used for both training and test data.

4. *Unknown Speaker-Unknown Environment (U.S-U.E):* The test data includes different speech and RIR files from training data.

Obviously, the fourth case shows the most generalized model performance among all four cases. The first three cases are significant to display the performance of the model in various cases.

5.7.1 The Effect of number of frames

We have experimented with evaluating the number of the frame for a single estimation effect on performance. To do that, we have used the speech files without background noise. Four different numbers of frames, namely 1, 50, 100, and 200, are utilized in this experiment. The obtained results are shown in Figure 5.4. As it is expected, the number of frames for an estimation increases the performance raises. As a higher number of frames means more information for an estimation. When the number of frames is 1, the method has 53.2% accuracy. The accuracy performance for 50 and 100 frames is 82.8% and 90.2%, respectively. While we have used 200 frames for each estimation, the best performance is obtained with

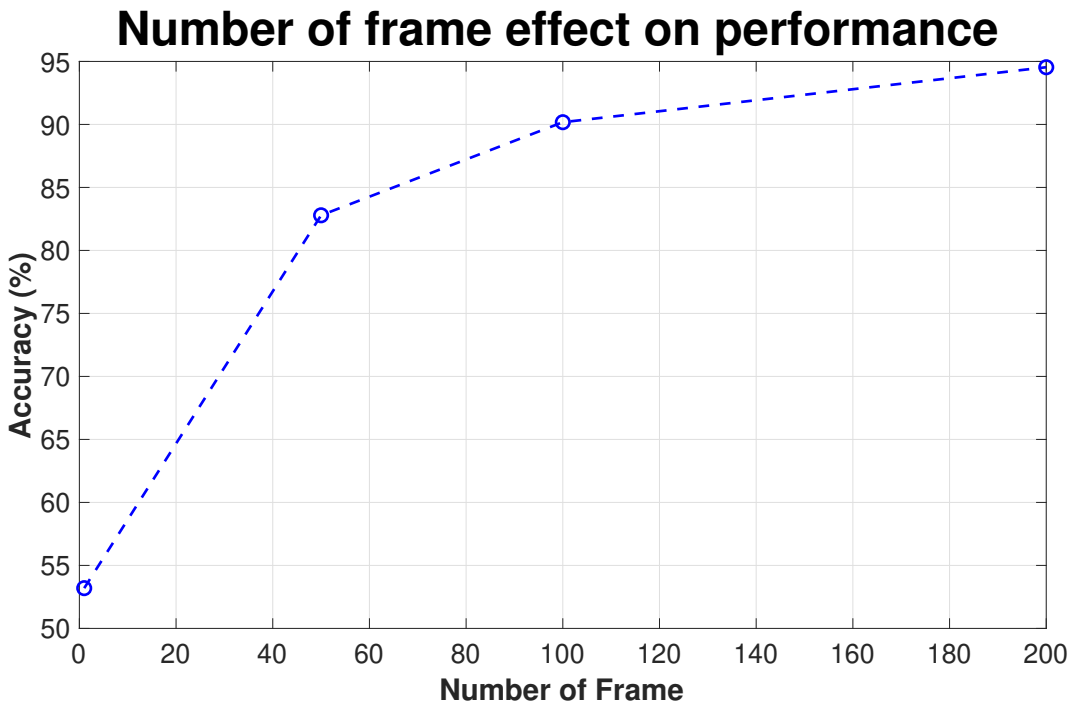


Figure 5.4: The performance of different number frame used for an estimation

94.5%. We have not considered the number of frames more than 200 because of real-time implementation concerns. Two hundred frames and 20 ms each frame make 2.01 s data for each estimation. More than 2.01 seconds for estimation would not be considered real-time since the proposed method cannot track the speaker. Since we have obtained the best accuracy with 200 frames, the results below are obtained using 200 frames for each estimation.

5.7.2 The Effect of number of classes on the performance

We present the performance of the proposed method when there is no presence of background noise. Two different models are trained for 3 and 4 classes. The details regarding the model architectures are given in section 5.4.1. The convolutions of the speech files (section 5.6.2 for details) and the simulated RIRs (see section 5.6.2 for details) are used for the first experiment. The training dataset set involves convolutions of eighty different rooms and

Table 5.4: The performance of the proposed method for 3 and 4 classes when there is no background noise

Cases	**3 Classes**	**4 Classes**
1. Known Speaker-Known Environment	97.78	97.50
2. Unknown Speaker-Known Environment	96.67	90.00
3. Known Speaker-Unknown Environment	96.67	94.17
4. Unknown Speaker-Unknown Environment	94.55	88.64

ninety-five various speakers. Three distinct rooms are convoluted with three various speakers for test data.

The proposed method's accuracy performance for three and four classes is presented in Table 5.4. While the column of 3 classes in Table 5.4 represents the performance of the proposed method estimates only three classes, the column of 4 classes denotes the accuracy of four class-based estimations of the method. Firstly, the three classes' overall accuracy performance is higher than the four classes based estimation as we expect. Since when there is a less number of classes for categorizing, the method makes fewer mistakes. Secondly, the highest accuracy performances for three and four classes are 97.78% and 97.50%, respectively, for the first case, as expected. The data for single microphone based speech source distance estimation is inclined to overfitting. Because of this problem, we used many dropouts, Layer 1 and 2 regularization techniques to overcome the overfitting issue. Moreover, the performance for case 2 is equal to case 3 for three class estimation. Contrarily, four class estimation has a better accuracy rate for K.S-U.E compared to for the U.S-K.E case. Lastly, we can see that the proposed method has a satisfactory result with 94.55% and 88.64% accuracy for unseen speaker and environment case for three and four classes, respectively.

Table 5.5 presents recall, precision, and F1 scores of case 4 for 3 (Table 5.5-a) and 4 (Table 5.5-b) classes. The most mistakes are made when the distance is 225 cm, and the best performance is seen when the distance is 75 cm for three classes. An increasing class number from 3 to 4 leads to more false positives and negatives in third and fourth classes in

Table 5.5: Recall, Precision, and F1 scores of unknown speaker and environment case for 3 and 4 classes

(a) 3 classes

	Recall	Precision	F1 score
75 cm	1.00	1.00	1.00
150 cm	0.927	0.910	0.918
225 cm	0.909	0.926	0.917

(b) 4 classes

	Recall	Precision	F1 score
75 cm	1.00	1.00	1.00
150 cm	0.909	0.925	0.917
225 cm	0.745	0.788	0.766
300 cm	0.891	0.831	0.860

Table 5.5-b. The most erroneous estimations is obtained for class 3 (225 cm) in Table 5.5-b and the recall, precision, and F1 scores less than 0.8. There are also many false negatives and positives for class 4 (300 cm). We obtain the highest correct estimations when the distance is 75 cm for four classes. Based on we obtain the results from Table 5.4 and 5.5, we continue to present results for three predefined classes since the performance of three classes is higher than four classes and the distance up to 225 cm is enough for our application.

5.7.3 Experiments with the Measured RIRs

In this subsection, we conduct an experiment using measured RIRs (see section 5.6.2 for details). This experiment is significant in two ways. Firstly, the experiment displays the performance of the method in a realistic environment. Secondly, the proposed method is tested when the speech source is not exactly 75 cm, 150 cm, or 225 cm away from the microphone. Since we have trained the model when the distance between the speech source and the microphone is exactly 75 cm, 150 cm, and 225 cm, the model is required to test when the distance varies. The distances and RT_{60} information for the experiment can be seen from Table 5.3. The label assignment is conducted by the following formula:

$$Label(i) = \begin{cases} 0,\ if\ 0 < distance \leq 75, \\ 1,\ if\ 75 < distance \leq 150, \\ 2,\ if\ 150 < distance. \end{cases} \tag{5.10}$$

where i is a dummy variable which indexes frame count.

Figure 5.5 shows the accuracy performance of the proposed method with measured RIRs. We have used a pretrained model with simulated data to test the proposed model's generalization ability in a realistic environment. In other words, measured RIRs have not been used for the training stage. Therefore, we have two cases for this experiment; (i) Known Speaker-Unknown Environment (Case 3) (ii) Unknown Speaker-Unknown Environment (Case 4). The proposed method achieves 88.66% and 85.83% accuracy for Case 3 and 4, respectively. There is only almost a 3% accuracy gap between the performance of the proposed method for Case 3 and 4.

The recall, precision, and F1 scores for the case of the unknown speaker and environment for the results with measured RIRs are presented in Table 5.6. The most erroneous estimations are obtained when the distance is 100 cm in the stairway. The number of false negatives is more than false positives for Class-1. Even though the F1 score for Class 1 is 0.739, a training model with measured RIRs can improve this. The proposed method performs very

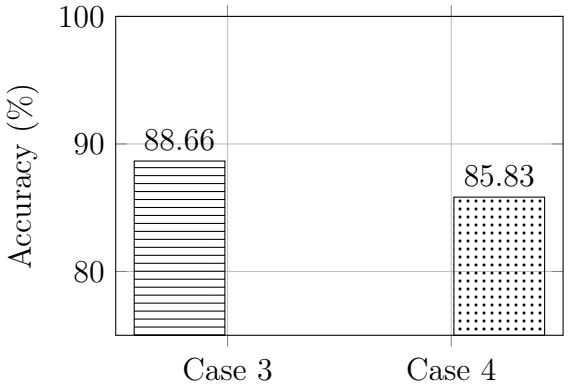


Figure 5.5: The performance of the proposed method with measured RIRs

Table 5.6: Recall, Precision, and F1 scores of unknown speaker and environment case for the results with measured RIRs

	3 classes		
	Recall	**Precision**	**F1 score**
Class-0	0.930	0.949	0.939
Class-1	0.720	0.758	0.739
Class-2	0.855	0.788	0.820

well in the studio booth, meeting room, Aula Carolina Aachen, and lecture room despite low and high reverberation times. Even though the performance of the proposed method in the stairway is low than expected, the overall accuracy performance is 85.83%.

5.7.4 The comparison of the proposed method with the state of art method

The measured RIR dataset [72] is also used for performance comparison of the proposed method to the method in [1]. Figure 5.6 displays the accuracy (%) comparison in two different rooms for Case 4. We have followed the same procedure in [1] for the comparison. The results for the compared method are obtained from their paper [1]. The proposed method has superior performance for both rooms with 86.3% and 83.7% accuracy. In [1],

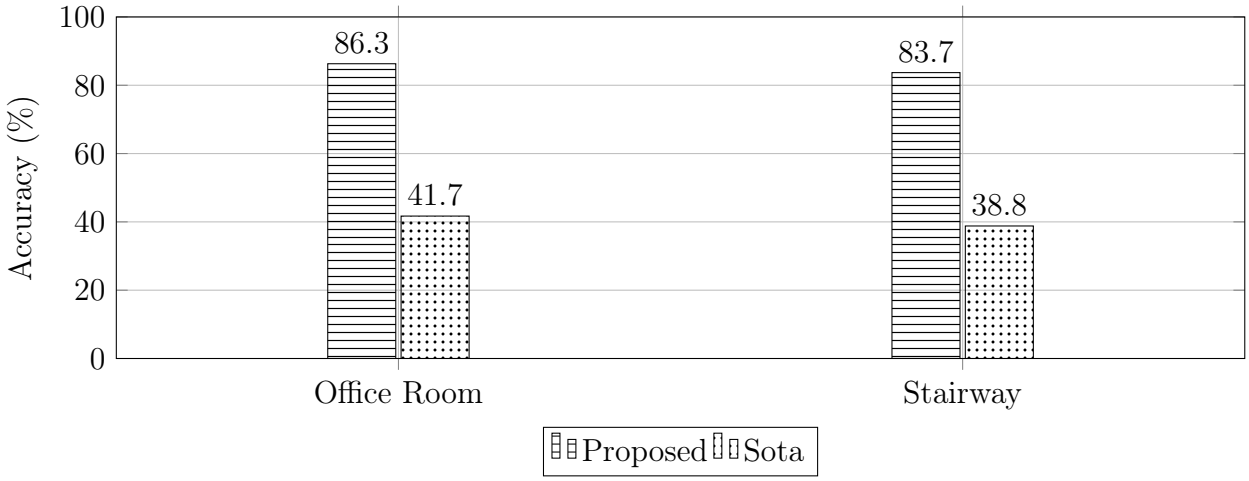


Figure 5.6: The accuracy comparison of the proposed method to the single channel state of art method [1]

the authors also state that their method requires improvement to increase the generalization ability of their algorithm. On the other hand, our proposed method has satisfactory results for unseen environments and speakers. There could be two reasons for better performance: we have used more diversified data, and the model architecture and feature set are better for a single microphone based distance estimation.

5.7.5 Experiments with background noise

The proposed method's performance under different noise types and SNR levels can be obtained in Table 5.7. We have obtained the accuracy performance results under babble and machinery noise at 0 dB, 10 dB, and 20 dB. The results, which are given under the 'Clean' column, are also collected when the background noise does not exist. We have trained a model with two different background noises and compared the performance of these models in Table 5.7. The first model is trained with the speech files with babble and machinery background noises at three SNR levels, i.e., 0 dB, 10 dB, and 20 dB. The performance result for the first model is given in the 'Model 1' column. The following model is trained with speech files without the presence of background noise, displayed 'Model 2' columns in the table 5.7. The training data size for both models is equal. The trained model is tested with different noise conditions and without background noise.

Table 5.7: The performance of the proposed method under different noise types and SNR levels

	Babble								**Machinery**							
	Clean		**0 dB**		**10 dB**		**20 dB**		**Clean**		**0 dB**		**10 dB**		**20 dB**	
	Model 1	*Model 2*	*Model 1*	*Model 2*	*Model 1*	*Model 2*	*Model 1*	*Model 2*	*Model 1*	*Model 2*	*Model 1*	*Model 2*	*Model 1*	*Model 2*	*Model 1*	*Model 2*
K.S.-K.E.	93.33	**97.78**	**81.11**	57.78	85.56	**86.67**	87.78	**95.56**	96.67	**97.78**	**89.65**	33.33	96.67	**98.33**	94.58	**95.56**
U.S-K.E.	95.56	**96.67**	**87.78**	41.67	**88.76**	80.00	**89.78**	86.67	93.27	**96.67**	**90.91**	33.33	93.26	**96.67**	91.67	**93.33**
K.S.-U.E.	94.55	**96.67**	**80.03**	46.67	83.33	**86.67**	81.11	**93.33**	94.45	**96.67**	**92.13**	33.33	94.48	**96.67**	**92.11**	91.11
U.S-U.E.	93.33	**94.55**	**79.15**	39.39	**84.56**	75.76	88.16	**92.73**	93.33	**94.55**	**82.54**	33.33	**88.34**	70.3	89.76	**90.91**

Each row in the Table 5.7 represents one case, which is explained at the beginning of this section. As expected, Model 2 performs better than Model 1 when there is no presence of background noise since Model 2 is trained with only speech signals without background noise. On the other hand, Model 1 accuracy performance is satisfactory with 93.33% for unseen talker and room for Clean case in both Babble and Machinery. When we compare models for the high noisy case, i.e. 0 dB, Model 1's performance is much better than Model 2's for both babble and machinery noises. While Model 1 has 79.15% and 82.54% accuracy for babble and machinery noises(U.S-U.E), respectively, Model 2's accuracy rate is 39.39% and 33.33% for babble and machinery noises(U.S-U.E), respectively. As raise in SNR, the performance of both models increases under both background noise types. The accuracy rates (U.S-U.E) for both noise types at 20 dB is close enough to the performance of the Clean case for both models.Overall performance of Model 1 under machinery noise is better than Model 1's performance under babble noise since the babble noise is more challenging than machinery noise case, and it involves speech contents, which leads to more erroneous estimations. Table 5.7 proves that a training model with and without background noise, like Model 1, makes the model more robust for background noises.

We have also conducted an experiment to assess the performance of Model 1a-1b and Model 2 under unseen SNR levels for both babble and machinery noise. 3 dB and 12 dB SNRs are considered for the experiment. The obtained results are shown in Figure 5.7. The accuracy rate difference between Model 1a-1b and Model 2 is high for the low SNR for both background noise. Model 1a has 32.23% superior accuracy than Model 2 for 3 dB babble noise. The difference becomes 41.67% under 3 dB machinery noise. The performance of both Model 1a-1b and 2 are almost equal for 12 dB for both noise types. While Model 1a performs 88.43% accuracy, Model 2 has 86.67% accuracy for babble noise at 12 dB. The performance of Model 1b and 2 for machinery 12 dB is equal. This experiment is also another proof that training the proposed method with and without background noise enables the model robust for unseen conditions.

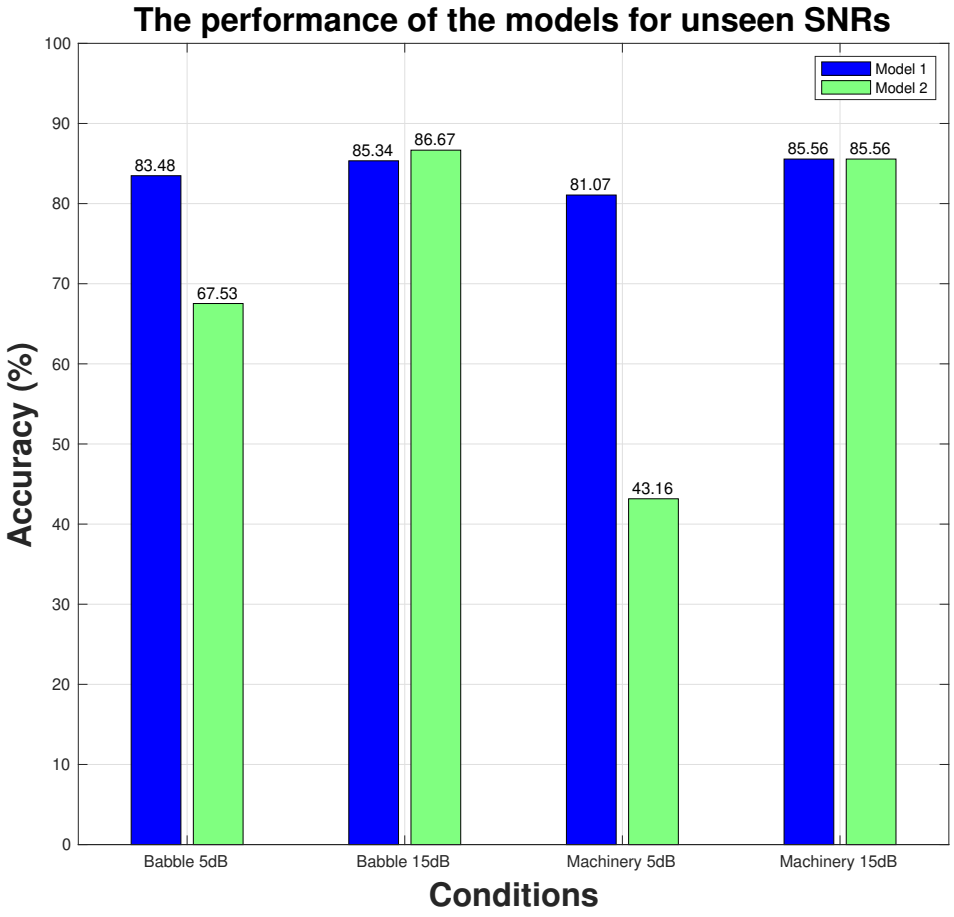


Figure 5.7: The performance of the models for unseen SNRs. Blue bar graph (left bar for each noise and SNR case) represents the performance of Model 1. Model 2's performance is denoted by the green bar graph (right bar for each noise and SNR case).

5.7.6 The CPU and Memory Consumption

The last experiment measures CPU and Memory consumption of the proposed Android app is displayed in Figure 5.8. The details of the real-time implementation of the proposed method are given in section 5.5. When Wait Buffer reaches 200 frames, the model inferences the speaker distance. The model made three estimations in the inferencing stage in Figure 5.8. The maximum CPU usage is 27%, and the usage is not constant. Additionally, memory consumption is around 190 MB during the inferencing process. When we consider that

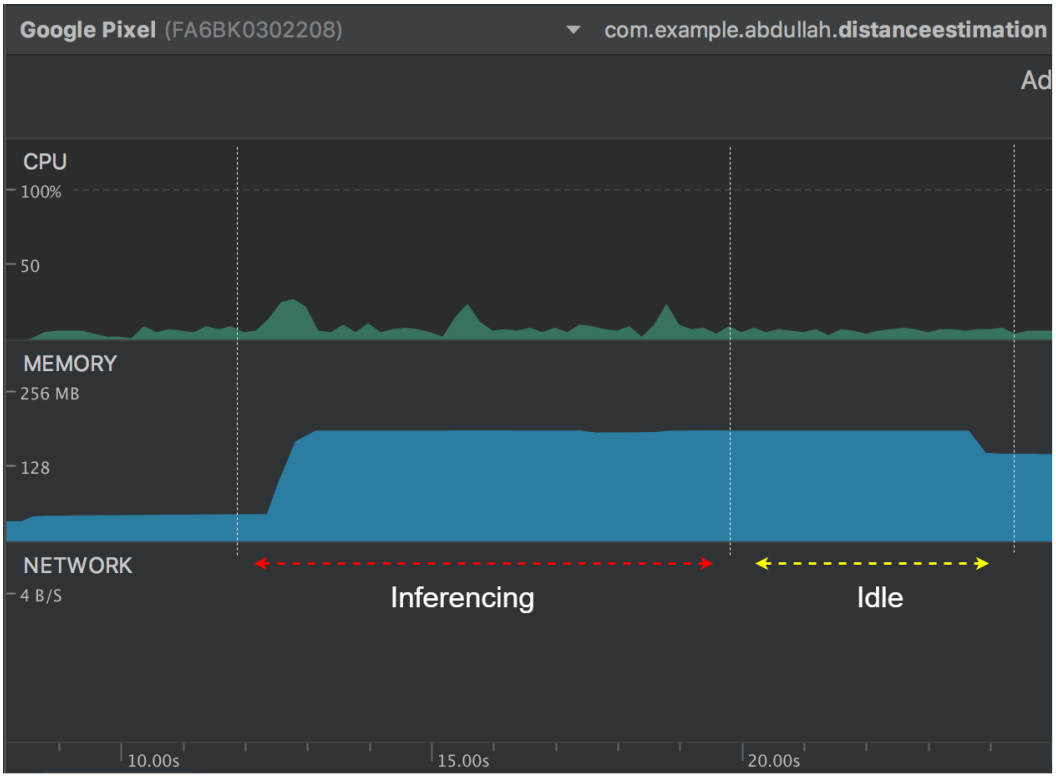


Figure 5.8: Screenshot of CPU and memory consumption of the proposed method on pixel 1.

most recent smartphones have at least 2GB memory, the proposed application consumes less than 10% memory. Moreover, the usage of CPU and Memory decreases in the idle stage. This experiment shows that the proposed distance estimation application perfectly suits the real-time implementation on the smartphone.

5.8 Conclusion

A convolutional neural network (CNN) based speech source distance estimation using single microphone algorithm and its real-time implementation are presented in this chapter. We optimize the proposed system pipeline and architecture in order to get high accuracy with low computational complexity. The proposed method has satisfactory accuracy rate for unseen speaker and environment. The proposed model is trained with background noisy

data to make the model robust in noisy environment. The proposed method is also tested for publicly available room impulse responses (RIRs) and performs satisfactorily well. The real-time implementation details also presented in this work. As per the results of our experiments, the proposed smartphone-based app suits real-life scenarios.

CHAPTER 6

CONCLUSION

In this dissertation, speech source localization (SSL) for hearing aid studies and its real-time implementation on mobile devices are studied. In Chapter 2, three different deep learning based SSL using the direction of arrival (DOA) angle estimation methods for two microphones are proposed. Details regarding deep learning architectures, feature sets, and hyperparameters are given in Chapter 2. Convolutional neural network (CNN) and convolutional recurrent neural network (CRNN) based DOA angle estimation methods perform best among other state-of-the-art DOA angle estimation methods. The real-time implementation of CNN and CRNN based DOA angle estimation methods are explained in Chapter 2.

Chapter 3 presents CNN based SSL method for eight microphones and its real-time implementation on resource-constrained hardware. The new feature set, which includes imaginary and real part of the short-time Fourier transform of the speech and spectral flux, is defined for eight microphones in Chapter 3. The proposed method also compared with the state of art DOA angle estimation methods for eight microphones. The real-time implementation of the proposed method on a raspberry pi with Matrix Creator is given in Chapter 3.

A noise-robust sound source distance estimation algorithm is described in Chapter 4. The algorithm estimates the distance between a loudspeaker which plays a single tone sinusoidal signal at 330 Hz, and a microphone. A weighting function in the frequency domain is offered for the sound source distance estimation method in order to increase the efficiency of the proposed method. The developed method's performance is evaluated in the presence of background noise and reverberation in Chapter 4.

Chapter 5 presents a convolutional neural network based single microphone speaker distance estimation method. The proposed method aims to estimate the talker distance in three/four predefined distances, which starts from 75 cm up to 300 cm with a 75 cm

increase. The data generation, training, and testing processes for the proposed method are given in Chapter 5. The proposed method performance is evaluated with Image source modeling (ISM) generated room impulse responses (RIRs) and measured RIRs. The real-time implementation details are also presented in Chapter 5.

REFERENCES

[1] M. Yiwere and E. J. Rhee, "Sound source distance estimation using deep learning: An image classification approach," *Sensors*, vol. 20, no. 1, pp. 172, 2020.

[2] E. Lindemann, "Two microphone nonlinear frequency domain beamformer for hearing aid noise reduction," in *Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Accoustics*. IEEE, 1995, pp. 24–27.

[3] I.J. Tashev, *Sound capture and processing: practical approaches*, John Wiley & Sons, 2009.

[4] I. Mccowan, J. Pelecanos, and S. Sridharan, "Robust speaker recognition using microphone arrays," 2001.

[5] W. Noble, D. Byrne, and B. Lepage, "Effects on sound localization of configuration and type of hearing impairment," *The Journal of the Acoustical Society of America*, vol. 95, pp. 992–1005, 1994.

[6] A. Ganguly, A. Küçük, and I. Panahi, "Real-time smartphone application for improving spatial awareness of hearing assistive devices," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2018, pp. 433–436.

[7] A. Küçük, A. Ganguly, Y. Hao, and I. M. S. Panahi, "Real-time convolutional neural network-based speech source localization on smartphone," *IEEE Access*, vol. 7, pp. 169969–169978, 2019.

[8] N. Shankar, A. Küçük, C. K. A. Reddy, G. S. Bhat, and I. M. S. Panahi, "Influence of mvdr beamformer on a speech enhancement based smartphone application for hearing aids," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2018, pp. 417–420.

[9] G. S. Bhat, N. Shankar, C. K. A. Reddy, and I. M. S. Panahi, "A real-time convolutional neural network based speech enhancement for hearing impaired listeners using smartphone," *IEEE Access*, vol. 7, pp. 78421–78433, 2019.

[10] P. Mishra, A. Ganguly, A. Küçük, and I. M. S. Panahi, "Unsupervised noise-aware adaptive feedback cancellation for hearing aid devices under noisy speech framework," in *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Dec 2017, pp. 1–5.

[11] A. Krizhevsky, I. Sutskever, and G. E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[13] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 2017, pp. 1–5.

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

[15] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

[16] M. S Brandstein and H. F Silverman, "A practical methodology for speech source localization with microphone arrays," *Computer speech & language*, vol. 11, no. 2, pp. 91–126, 1997.

[17] A. Härmä, "Ambient telephony: Scenarios and research challenges," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[18] C. Lu, C. Wu, and L. Fu, "A reciprocal and extensible architecture for multiple-target tracking in a smart home," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 1, pp. 120–129, 2011.

[19] I. Meza, C. Rascon, G. Fuentes, and L. A Pineda, "On indexicality, direction of arrival of sound sources, and human-robot interaction," *Journal of robotics*, vol. 2016, 2016.

[20] T. Zhang, F. Mustiere, and C. Micheyl, "Intelligent hearing aids: The next revolution," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 72–76.

[21] A. Sehgal and N. Kehtarnavaz, "A convolutional neural network smartphone app for real-time voice activity detection," *IEEE Access*, vol. 6, pp. 9017–9026, 2018.

[22] Google, "TensorFlow (2021)," `https://www.tensorflow.org/`.

[23] M. L. Seltzer, "Microphone array processing for robust speech recognition," *CMU, Pittsburgh PA, PhD Thesis*, 2003.

[24] M. S. Brandstein and S. M. Griebel, "Nonlinear, model-based microphone array speech enhancement," in *Acoustic signal processing for telecommunication*, pp. 261–279. Springer, 2000.

[25] I. Panahi, N. Kehtarnavaz, and L. Thibodeau, "Smartphone-based noise adaptive speech enhancement for hearing aid applications," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2016, pp. 85–88.

[26] B. Widrow and F. Luo, "Microphone arrays for hearing aids: An overview," *Speech Communication*, vol. 39, no. 1-2, pp. 139–146, 2003.

[27] A. Ganguly, A. Kucuk, and I. Panahi, "Real-time smartphone implementation of noise-robust speech source localization algorithm for hearing aid users," in *Proceedings of Meetings on Acoustics 173EAA*. ASA, 2017, vol. 30, p. 055002.

[28] C. Karadagur Ananda Reddy, N. Shankar, G. Shreedhar Bhat, R. Charan, and I. Panahi, "An individualized super-gaussian single microphone speech enhancement for hearing aid users with smartphone as an assistive device," *IEEE Signal Processing Letters*, vol. 24, pp. 1601–1605, Nov 2017.

[29] Y. Hao, M. C. R. Charan, G. S. Bhat, and I. M. S. Panahi, "Robust real-time sound pressure level stabilizer for multi-channel hearing aids compression for dynamically changing acoustic environment," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 1952–1955.

[30] S. Chakrabarty and E. A. P. Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2017, pp. 136–140.

[31] P. Pertilä and M. Parviainen, "Time difference of arrival estimation of speech signals using deep neural networks with integrated time-frequency masking," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 436–440.

[32] "Smartphone-based open research platform for hearing improvement studies," `http://www.utdallas.edu/ssprl/hearing-aid-project/`.

[33] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[34] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[35] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1997, vol. 1, pp. 375–378 vol.1.

[36] J. Pak and J. W. Shin, "Sound localization based on phase difference enhancement using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1335–1345, 2019.

[37] P. Pertilä and E. Cakir, "Robust direction estimation with convolutional neural networks based steered response power," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 6125–6129.

[38] N. T. N. Tho, S. Zhao, and D. L. Jones, "Robust doa estimation of multiple speech sources," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2287–2291.

[39] N. Ma, T. May, and G. J. Brown, "Exploiting deep neural networks and head movements for robust binaural localization of multiple sources in reverberant environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2444–2453, 2017.

[40] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 2814–2818.

[41] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end binaural sound localisation from the raw waveform," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 451–455.

[42] D. Salvati, C. Drioli, and G. L. Foresti, "Exploiting cnns for improving acoustic source localization in noisy and reverberant conditions," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 103–116, 2018.

[43] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 405–409.

[44] Q. Li, X. Zhang, and H. Li, "Online direction of arrival estimation based on deep learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 2616–2620.

[45] Y. Sun, J. Chen, C. Yuen, and S. Rahardja, "Indoor sound source localization with probabilistic neural network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6403–6413, 2018.

[46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[47] M. Nilsson, S. D. Soli, and J. A Sullivan, "Development of the hearing in noise test for the measurement of speech reception thresholds in quiet and in noise," *The Journal of the Acoustical Society of America*, vol. 95, pp. 1085–1099, 1994.

[48] J. S. Garofolo and et al, "TIMIT acoustic-phonetic continuous speech corpus LDC93S1.," Philadelphia: Linguistic Data Consortium, 1993.

[49] "Acoustical liberation of books in the public domain," `https://librivox.org/`.

[50] A. Mesaros, T. Heittola, and T. Virtanen, "TUT Acoustic Scenes 2017, Development Dataset," `https://zenodo.org/record/400515/`.

[51] A. A. Milani, G. Kannan, I. M. S. Panahi, and R. Briggs, "A multichannel speech enhancement method for functional mri systems using a distributed microphone array," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sep. 2009, pp. 6946–6949.

[52] A. Küçük, Y. Hao, A. Ganguly, and I. M. Panahi, "Stereo I/O framework for audio signal processing on android platforms," *The Journal of the Acoustical Society of America*, vol. 143, pp. 1955–1956, 2018.

[53] "Bazel (2021)," `https://bazel.build/`.

[54] Y. Rao, Y. Hao, I. M. S. Panahi, and N. Kehtarnavaz, "Smartphone-based real-time speech enhancement for improving hearing aids speech perception," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 5885–5888.

[55] Phonak, "Phonak Hearing Aids, Roger Select," `https://www.phonak.com/us/en/hearing-aids/accessories/rogerselect.html`.

[56] Phonak, "Phonak Hearing Aids, Roger Table Mic.," `https://www.phonak.com/us/en/hearing-aids/accessories/roger-tablemic.html`.

[57] M Klemm, IJ Craddock, JA Leendertz, A Preece, and R Benjamin, "Improved delay-and-sum beamforming algorithm for breast cancer detection," *International Journal of Antennas and Propagation*, vol. 2008, 2008.

[58] Amazon Alexa, "Alexa Voice Service [Online]:Available," `https://developer.amazon.com/docs/alexa-voiceservice/audio-hardware-configurations.html`.

[59] Google, "Google Home," `https://store.google.com/gb/product/google_home`.

[60] Y. Hao, *Smartphone Based Multi-Channel Dynamic-Range Compression for Hearing Aid Research and Noise-Robust Speech Source Localization Using Microphone Arrays*, Ph.D. thesis, 2019.

[61] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the mvdr filter," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 290–300, 2001.

[62] Raspberry Pi, "Raspberry Pi Model B [Online] Available:," `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`.

[63] Matrix One, "Matrix Creator [Online] Available:," `https://www.matrix.one/products/creator`.

[64] RM Stallman, "Using and porting the GNU compiler collection," Free Software Foundation, 1999 Jul.

[65] J. Gu and P. Wei, "Joint svd of two cross-correlation matrices to achieve automatic pairing in 2-d angle estimation problems," *IEEE Antennas and Wireless Propagation Letters*, vol. 6, pp. 553–556, 2007.

[66] A. Küçük and I. M. S. Panahi, "Convolutional recurrent neural network based direction of arrival estimation method using two microphones for hearing studies," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2020, pp. 1–6.

[67] E. Georganti, T. May, S. van de Par, and J. Mourjopoulos, "Sound source distance estimation in rooms based on statistical properties of binaural signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1727–1741, 2013.

[68] M. Zohourian and R. Martin, "Binaural direct-to-reverberant energy ratio and speaker distance estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 92–104, 2020.

[69] P. D. Jager, M. Trinkle, and A. Hashemi-Sakhtsari, "Automatic microphone array position calibration using an acoustic sounding source," in *2009 4th IEEE Conference on Industrial Electronics and Applications*, May 2009, pp. 2110–2113.

[70] W. Jiang, Z. Cai, M. Luo, and Z. L. Yu, "A simple microphone array for source direction and distance estimation," in *2011 6th IEEE Conference on Industrial Electronics and Applications*, June 2011, pp. 1002–1005.

[71] J. K. Nielsen, N. D. Gaubitch, R. Heusdens, J. Martinez, T. L. Jensen, and S. H. Jensen, "Real-time loudspeaker distance estimation with stereo audio," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Aug 2015, pp. 250–254.

[72] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–5.

[73] E. A. Lehmann and A. M. Johansson, "Diffuse reverberation model for efficient image-source simulation of room impulse responses," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1429–1439, Aug 2010.

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

[75] D. Byrne and W. Noble, "Optimizing sound localization with hearing aids," *Trends in Amplification*, vol. 3, no. 2, pp. 51–73, 1998.

[76] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, and A. Mouchtaris, "Localizing multiple audio sources in a wireless acoustic sensor network," *Signal Processing*, vol. 107, pp. 54–67, 2015.

[77] H. Wang and P. Chu, "Voice source localization for automatic camera pointing system in videoconferencing," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, vol. 1, pp. 187–190 vol.1.

[78] M. Farmani, M. S. Pedersen, Z. Tan, and J. Jensen, "Informed tdoa-based direction of arrival estimation for hearing aid applications," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 953–957.

[79] J. K. Nielsen, N. D. Gaubitch, R. Heusdens, J. Martinez, T. L. Jensen, and S. H. Jensen, "Real-time loudspeaker distance estimation with stereo audio," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, Aug 2015, pp. 250–254.

[80] E. Larsen, C. D. Schmitz, C. R. Lansing, W. D. O'Brien, B. C. Wheeler, and A. S. Feng, "Acoustic scene analysis using estimated impulse responses," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, 2003, vol. 1, pp. 725–729 Vol.1.

[81] Y. Hioka, K. Niwa, S. Sakauchi, K. Furuya, and Y. Haneda, "Estimating direct-to-reverberant energy ratio using d/r spatial correlation matrix model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2374–2384, 2011.

[82] A. Brendel and W. Kellermann, "Learning-based acoustic source-microphone distance estimation using the coherent-to-diffuse power ratio," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 61–65.

[83] E. Georganti, T. May, S. van de Par, and J. Mourjopoulos, "Sound source distance estimation in rooms based on statistical properties of binaural signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1727–1741, 2013.

[84] M. Zohourian and R. Martin, "Binaural direct-to-reverberant energy ratio and speaker distance estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 92–104, 2020.

[85] E. Georganti, T. May, S. van de Par, A. Harma, and J. Mourjopoulos, "Speaker distance detection using a single microphone," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1949–1961, 2011.

[86] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[87] KDnuggets, "KDnuggets Classification Metrics (2020)," `https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html//`.

[88] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, USA, 1992.

[89] C. KA Reddy, H. Dubey, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "Icassp 2021 deep noise suppression challenge," *arXiv preprint arXiv:2009.06122*, 2020.

BIOGRAPHICAL SKETCH

Abdullah Küçük is currently pursuing his PhD at the Statistical Signal Processing Research Laboratory (SSPRL) at The University of Texas at Dallas, Richardson, TX. He received his Bachelor of Engineering degree in Electronics Engineering from Kadir Has University, Istanbul, Turkey, in 2011 and his MS in Electrical Engineering from The University of Texas at Dallas in 2018. His current research interests include real-time speech source localization, microphone array processing, voice activity detector, deep learning for audio signal processing. He currently has published work in two journals and seven conference papers on topics such as speech source localization, deep learning based direction of arrival estimation, speech enhancement, and acoustic feedback cancellation. He also submitted a journal and conference paper for publication. He previously worked as an Applied Scientist Intern at Amazon Lab 126 in Summer 2019. He has very good understanding of digital and statistical signal processing concepts. He is also well versed in developing machine learning based algorithms for audio and speech applications.

CURRICULUM VITAE

Abdullah Küçük

July, 2020

Contact Information:

Department of Electrical Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: abdullah.kucuk@utdallas.edu

Educational History:

BE, Electronics Engineering (Honors), Kadir Has University, Turkey, 2011
MS, Electrical Engineering, The University of Texas at Dallas, TX, 2018
PhD, Electrical Engineering, The University of Texas at Dallas, TX, 2021

Industrial Experience:

Applied Scientist Intern, Amazon Lab 126, Boston, MA, May 2019 – August 2019
Software Test Engineer, Huawei Technologies, Istanbul, Turkey, November 2013 – March 2015
Software Engineer (Cofounder)r, 3 Aces Software Company, Istanbul, Turkey, November 2012 – October 2013

Research Experience:

PhD Research Assistant, Statistical Signal Processing Research Lab (SSPRL) at UTD October 2016 – Present:
Recently working on deep learning based Direction of Arrival (DOA) estimation. Worked on implementation of DOA for two microphones. Developed and improved Android application which estimates DOA using two mics of the smartphone. Worked on Audio Compression algorithm to improve quality and intelligibility of speech for Hearing Aids. Coded in MATLAB and C. Worked on implementation Speech Enhancement algorithms to the Android phones.

Publications:

- Abdullah Küçük, A. Ganguly, Y. Hao and I. M. S. Panahi, "Real-Time Convolutional Neural Network-Based Speech Source Localization on Smartphone," in IEEE Access, vol. 7, pp. 169969-169978, 2019.

- Abdullah Küçük, and I. M. S. Panahi, "Direction of arrival estimation using deep neural network for hearing aid applications using smartphone." Proceedings of Meetings on Acoustics 178ASA. Vol. 39. No. 1. Acoustical Society of America, 2019.

- Abdullah Küçük, and and I. M. S. Panahi, "Convolutional Recurrent Neural Network Based Direction of Arrival Estimation Method Using Two Microphones for Hearing Studies," 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), Espoo, Finland, 2020, pp. 1-6, doi: 10.1109/ MLSP49062.2020.9231693

- Abdullah Küçük, and and I. M. S. Panahi, "Noise Robust Single Microphone-based Distance Estimation Method and Its Real-Time Implementation on the Smartphone for Hearing Aid Studies", International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2022, (Submitted)

- Abdullah Küçük, and and I. M. S. Panahi, "Convolutional Neural Network based Single Microphone Speaker Distance Estimation Method for Hearing Studies", in IEEE/ACM Transactions on Audio, Speech, and Language Processing, (Submitted)

- Y. Hao, Abdullah Küçük, A. Ganguly and I. M. S. Panahi, "Spectral Flux-based Convolutional Neural Network Architecture for Speech Source Localization and its Real-time Implementation," in IEEE Access, doi: 10.1109/ACCESS.2020.3033533.

- Abdullah Küçük, A. Ganguly, I. Panahi, "Improved pre-filtering stages for GCC-based direction of arrival estimation using smartphone", Proceedings of Meetings on Acoustics, Acoustic Society of America (ASA 2018), Minneapolis, MN, May 2018.

- Abdullah Küçük, Y. Hao, A. Ganguly, I. Panahi, "Stereo I/O Framework for Audio Signal Processing on Android Platforms", Proceedings of Meetings on Acoustics, Acoustic Society of America (ASA 2018), Minneapolis, MN, May 2018, 143(3), pp.1955-1956.

- A. Ganguly, Abdullah Küçük, I. Panahi, "Real-time Smartphone application for improving spatial awareness of Hearing Assistive Devices", 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 433-436. IEEE, 2018.

- N. Shankar, Abdullah Küçük, C. Reddy, G. Bhat, I. Panahi, "Influence of MVDR beamformer on a Speech Enhancement based Smartphone application for Hearing Aids", Engineering in Medicine and Biology Society (EMBC), 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 417-420. IEEE, 2018.

- A. Ganguly, Abdullah Küçük, I. Panahi, "Real-time Smartphone implementation of noise-robust Speech source localization algorithm for hearing aid users",Proceedings of Meetings on Acoustics 173EAA, vol. 30, no. 1, p. 055002. Acoustical Society of America, 2017.

- P. Mishra, A. Ganguly, Abdullah Küçük, I. Panahi, "Unsupervised Noise-aware Adaptive Feedback Cancellation for Hearing Aid Devices under Noisy speech framework" - In 2017 IEEE signal processing in medicine and biology symposium (SPMB) (pp. 1-5).

Honors:

Best Individual Contributor 2014 in Huawei R&D Center, Nov 2014
Graduating 3rd ranking in Electronics Engineering, Jun 2011
Full Scholarship from Kadir Has University, Sep 2006