SPEECH FEATURES ANALYSIS OF THE JOINT SPEECH SEPARATION AND AUTOMATIC SPEECH RECOGNITION MODEL

TAWSEEF KHAN

**FACULTY OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY**

UNIVERSITY OF MALAYA

KUALA LUMPUR

2021

SPEECH FEATURES ANALYSIS OF THE JOINT SPEECH SEPARATION AND AUTOMATIC SPEECH RECOGNITION MODEL

TAWSEEF KHAN

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SOFTWARE ENGINEERING (SOFTWARE TECHNOLOGY)

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY

UNIVERSITY OF MALAYA

KUALA LUMPUR

2021

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Tawseef Khan
Matric No: WOC170008 / 17043552/1
Name of Degree: Master of Software Engineering (Software Technology)
Title of Dissertation: SPEECH FEATURES ANALYSIS OF THE JOINT SPEECH SEPARATION AND AUTOMATIC SPEECH RECOGNITION MODEL

Field of Study: Speech Recognition

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every right in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

SPEECH FEATURES ANALYSIS OF THE JOINT SPEECH SEPARATION AND AUTOMATIC SPEECH RECOGNITION MODEL

ABSTRACT

Speech recognition of target speakers from a mixture of voiced noises from interfering speakers in a single channel is a complex task. This is because the speech signal pattern of both the target and interfering speakers are similar and can be challenging to distinguish from one another. If the target speaker's speech can be correctly identified, such a system can be used in interviews, courtrooms, transcribing video subtitles, etc. During conversations between multiple speakers, it is common for the voices to overlap. In such cases, it is important to separate the speech of the target speaker based on one single audio signal. To date, ASR models are good at recognizing lexical data in white/background noises though they are unable to perform well with other voiced noises. Recently a joint speech separation and ASR model was proposed that can handle both the task of speech separation and recognition into one component in an end-to-end fashion. Two key factors affecting the accuracy of ASR models are the type of features used to build the model and the signal-to-noise ratio (SNR) of the target signal. This research compares different features to find the optimum features for the joint speech separation and ASR model at different SNR levels. Ten features that were previously used in speech separation of voiced noise have been used to test the accuracy of the model at SNR levels -10, -5, 0, 5, +5 (dB). The experiment evaluates the Word Error Rate (WER) of Speech separation and ASR separately within the joint speech separation and ASR model. Ten features that were used for speech separation in previous studies were evaluated, which are STFT, LOG-POW, LOG-MEL, LOG-MAG, GF, GFCC, MFCC, PNCC, RASTA-PLP (Relative Spectral - Perceptual Predictive), and AMS. At SNR level -10, GF and GFCC was found to have the lowest WER. For SNR levels -5, 0, 5, 10 the lowest WER was achieved by GF, PNCC, STFT, and GF.

Keywords: Speech Separation, Automatic Speech Recognition, Acoustic Model, Signal-to-Noise Ratio, Word Error Rate

ANALISIS CIRI-CIRI PERTUTURAN BAGI MODEL BERSAMA PEMISAHAN UCAPAN DAN PENGECAMAN PERTUTURAN AUTOMATIK

ABSTRAK

Pengecaman pertuturan oleh pengguna sasaran dari campuran beberapa suara yang menyebabkan gangguan dalam satu saluran adalah tugas yang amat kompleks. Ini kerana, corak isyarat pertuturan pengguna sasaran dan isyarat gangguan mempunyai ciri-ciri suara yang hampir serupa dan amat mencabar untuk membezakan satu sama lain. Sekiranya ucapan penutur sasaran dapat dikenal pasti dengan tepat, sistem seperti ASR dapat digunakan dalam wawancara, ruang sidang, menyalin sari kata video, dan lain-lain. Semasa perbualan antara beberapa orang, adalah perkara biasa bagi suara untuk bertindih. Dalam kes sedemikian, adalah penting untuk memisahkan ucapan penutur sasaran berdasarkan satu isyarat audio tunggal. Sehingga kini, sistem ASR mampu mengenali data leksikal dalam gangguan suara putih / latar belakang walaupun tidak dapat berfungsi dengan baik dengan jenis—jenis gangguan suara yang lain. Baru-baru ini, model pemisahan pertuturan (SS) bersama dan model ASR telah dicadangkan bagi membolehkan pemisahan ucapan dan pengecaman tugas menjadi satu komponen secara hujung-ke-hujung. Dua faktor utama yang mempengaruhi ketepatan sistem ASR adalah jenis ciri suara yang digunakan untuk membina model dan nisbah isyarat-ke-bunyi (SNR) dari isyarat sasaran. Penyelidikan ini membandingkan teknik pengekstrakan ciri-ciri yang berbeza untuk mencari ciri-ciri optimum bagi pemisahan pertuturan gabungan dan model ASR pada tahap SNR yang berbeza. Sepuluh jenis ciri-ciri yang sebelum ini pernah digunakan dalam kajian sebelum ini telah digunakan untuk menguji ketepatan model pada tahap SNR -10, -5, 0, 5, +5 (dB). Keupayaan ciri-ciritersebut diukur dengan menilai Kadar Kesalahan Kata (WER) pemisah pertuturan dan ASR secara berasingan, dan secara bersama. Sepuluh ciri-ciri yang digunakan untuk pemisahan pertuturan dalam kajian ini adalah STFT, LOG-POW, LOG-MEL, LOG-MAG, GF, GFCC, MFCC, PNCC, RASTA-PLP, dan AMS. Pada tahap SNR -10, GF dan GFCC didapati mempunyai WER yang terendah. Untuk tahap SNR -5,0,5,10 WER terendah dicapai oleh GF, PNCC, STFT, dan GF.

Keywords: Pemisahan Pertuturan, Pengecaman Pertuturan Automatik, Model Akustik, Nisbah Isyarat-ke-Bising, Kadar Kesalahan Kata

ACKNOWLEDGEMENTS

First and foremost, I have to thank my research supervisor, Dr. Mumtaz Begun Mustafa of the Faculty of Computer Science and Information Technology at University of Malaya. Without her assistance and dedicated involvement in every step throughout the process, this research would have never been accomplished. I would like to thank you very much for your support and understanding over the past years.

I am grateful to my friends who participated in my presentations and supported me along the way.

Finally, I must express my very profound gratitude to my parents, my siblings, and my girlfriend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ASR	Automatic Speech Recognition
SS	Speech Separation
SR	Speech recognition
WER	Word Error Rate
SNR	Signal-to-Noise Ratio
MSE	Minimum Squared Error

CHAPTER 1 : INTRODUCTION

The approaches to building an automatic speech recognition system (ASR) have changed over the last decade. An ASR transforms audio signals to lexical or any other recognizable formats. Even though the precision of these systems has improved, it is far from measuring up to the capability of a human. Despite the lack of accuracy, ASR models are used frequently in automated typing, transcriptions, commands, etc. People with hearing losses are heavily dependent on the accuracy of this form of technology.

ASRs can be classified as human-computer interaction (HCI) tool, and the close competitors are keyboards (buttons), touch screens (touch), and even gesture recognizers. The later mentioned, buttons and touch system have done surprisingly well with regards to the accuracy. As such, ASR models need to perform equally well. The key factor that affects the accuracy of ASRs is the quality of the input audio signal. Separating the target speech from its noise is difficult; one that is still not perfect.

The process of separating the target signal from the noisy signal to achieve a lower word error rate (WER) for increased accuracy of an ASR model is called speech separation or speech segregation. Noises in the audio signal can originate from constant background noises, sudden sound bursts from horns, or even voices from other speakers. Noises can be classified as speech or non-speech. Technologies such as Mel-frequency cepstral coefficients MFCC, Ideal Binary Mask (IBM), Target Binary Mask (TMB) have made incredible breakthroughs in reducing non-speech noises. However, speech separation of signals tainted with other speech noise (voices) is a more challenging problem.

Deep Neural Networks (DNN) is one of the key technologies used to create robust and efficient ASR models. The recent trends consider speech separation as a supervised learning problem rather than a signal processing problem. Here, DNNs are heavily used to improve the accuracy of the system.

In real-life communication such as during conferences, meetings, interviews, or even in courtrooms, the audio signal of a speaker is tainted with voices from other speakers. It is common for the voices to overlap during conversations between the multiple target speakers. In such cases, it is important to separate the speech from both the target speakers from one single audio signal. This research looks closely into speech separation as a supervised learning problem using DNNs to create an ASR that is equipped to handle the (speech-noise based) speech separation.

1.1 Background

- The types of speech Separation and ASR environments (monaural vs. array-based)

There are two major types of speech separation and ASR models; namely monaural and array-based. These indicate the number of channels (microphone) used to record/train/test the two models. Monaural means only one channel (microphone) is used while in array-based multiple microphones placed at different strategic places to record the sounds. Monaural is more commonly used such as when talking over the mobile. Array-based setup is usually used in studios and stages where the source of the sound can be determined (for example on the stage). In such case, the microphones can be placed strategically around the user to get the best inputs. This research will focus on only monaural conditions as it is more pervasive in real-life scenarios.

- The type of environments where speech Separation or ASR is used

Speech Separation (SS) and ASR models can be developed for many environmental conditions to suit the needs of different applications. These systems can work in noisy or clean conditions. Some systems have more specificity by limiting the type of noise and its intensity level. Noises can be classified as voiced and non-voiced noise. Voiced noise is considered the noise coming from the speech of other people while unvoiced noise could be from any other sources. Non-voiced noise can be interfering noises coming from cars, bells, animals, etc.

- The types of users of Speech Separation or ASR

Speech Separation (SS) and ASR models can be for specific user or users. The two major types are speaker-dependent and independent. In speaker-dependent systems, the SS or ASR is only used by one person. If the system needs to handle voiced noise, then both the target speaker and all the interfering speakers need to be known. Speaker-independent systems are the most flexible and can be used by any user where neither the target speaker nor the interfering speaker must be known. Speech Separation had more success in separating speaker-dependent and target speaker-dependent systems. In speaker-independent systems, there is no way to identify the interfering speaker from the target speaker since the model of the system was modeled to recognize all the voices. This research will focus on speaker-dependent systems.

1.1.1 Automatic Speech Recognition (ASR)

Automatic speech recognition systems convert speech signals to lexical data. The speech signal is split into frames and features are extracted like the SS. These features are fed into a trained model to predict the mono-phone or triphones of each of the frames. A phoneme is one of the units of sound that distinguish one word from another in a language. When each frame is labeled as one phoneme then it is known as a mono-phone. In the case of the triphone, each frame is labeled as a tuple of three phonemes. The extra two phones are the immediately preceding and succeeding phonemes.

The machine learning models that have worked well in the past are the Support Vector Machine (SVM) and Gaussian Mixture Model (GMM,) which were used to predict the weight of the Hidden Markov models. With the recent advance of Deep Neural Network (DNN), a new generation of machine learning techniques is used for speech recognition.

The performance of the ASR depends on the noise and quality of the sound. As the input speech signal becomes increasingly noisy, the efficiency of the ASR model decreases (Zhang and Wang 2016; Wang et al. 2016). The initial attempt to counter this problem was to use noisy data while training the ASR models. Rajnoha (2014) demonstrated an increase in the accuracy in ASR with technique. However, training the model with too much noisy data will result in its ability to recognize clean speech to decrease. Therefore, training the DNN using noisy data has to be limited. Similar approaches were taken by Tu (2014;2016) to train the DNN.

1.1.2 Speech Separation

The goal of a speech Separation is to separate the clean speech signal from the noisy signal. Separating the speech of one speaker from a mixture of voices (voiced noise) is known as speaker separation (DeLiang Wang, 2017). There are broadly two groups of SS systems based on their training targets. A training target dictates what a model should learn and what its final output will be. Mapping-based SS uses the clean speech features as their training targets (i.e. the model learns to map the noisy speech features to clean speech features directly). Masking based training targets use a mask (relation between the clean to noisy speech) as their training target. In such cases, the model's output is a mask, which is then used along with the original noisy signal to calculate the clean signal.

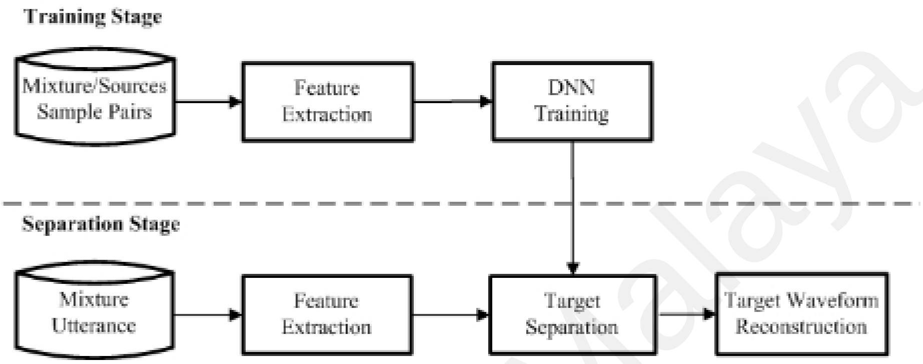


Figure 1.1: Speech separation framework (Tu, 2014)

Tu (2014) states that speech separation is divided into two stages as shown in Figure 1.1. At the training stage, the DNN training module is used to build a DNN model. In feature extractions, the input signal must be sliced into frames. These frames then must go through a process called feature extraction, which extracts important information from the raw speech signal frame. These frames are then used to train the model using different training targets (DNN training). The number of frames used to predict each slice is called the context window.

At the separation stage, the test input from the Mixture utterance is passed through a similar feature extraction stage. Next, the previously trained DNN model is used to separate the target signal from the noise.

Previously, Non-negative matrix (NMF) models were used to train the SS. Huang (2014) was the first to show that the neural networks (DNN and RNN) are more effective than NMF. He used both the DNN and Recurrent neural network (RNN) to develop a masking-based SS. While Huang's (2014) work was speaker-dependent, Du (2014) proposed a similar model for the target speaker-dependent approach.

Zhang and Wang (2016) proposed a deep ensemble network to address speaker-dependent as well as target-dependent separation. They used both masking and mapping based models to investigate the effect of context windows on features during training the model.

1.1.3 Joint SS and ASR model

The training model with noisy data still could not perform well in noisy conditions in Josef Rajnoha's (2014) work. Barker & Marxer (2001) proposed a non-voiced noise model using the Multi-channel Wiener filter (MWF) to reduce the background noise and preprocessing the input signal. However, this filtering system does not work well on unstable noise, such as voiced noise. Tu (2014) proposed that SS and ASR are combined. Unlike Barker & Marxer (2001), Tu 's model can handle both the voiced and non-voiced noise. Tu (2014) preprocessed the f using a SS, before feeding it to the ASR. The SS extracts the features from the noisy signal and predicts its features. The clean features are then reconstructed to a clean speech signal. The reconstructed/clean speech signal is fed into the ASR for another feature extraction to be labeled and predicted by the DNN model.

Delfarah (2017), shows that during the testing stage, feature extraction can take a long time (Delfarah and Wang, 2017). However, a major flaw of Tu Yanhui (2014) work, is that it uses two feature extraction components (one in SSS and one inside ASRS) that decrease the performance of the system in terms of the time taken to predict the output (Tu et al., 2014).

1.2 Motivation

The goal of speech separation should be to solve the 'cocktail party problem'. Humans have the unique ability to focus on source of sound while automatically ignoring others as noise; just like in a cocktail party, there are many noises all around, yet we can successfully focus on only one person's voice.

Human’s ability to focus on a single voice heavily depends on the type of noise and its intensity (loudness) of the audio signal. The ratio of the level of clean signal and noise signal is known as the Signal to Noise Ratio (SNR).

As of today, many techniques and methods have been used to develop automatic speech recognition (ASR) model. Over the years the accuracy of ASR has improved however it is still far from perfect. ASR model works very well and is able to convert sound to text well when the sound is clean and has been recorded in controlled environments; however in noisy environments the accuracy drops. In real-life scenarios, ASR is rarely used in controlled environments. Thus, the need for a better ASR model that can accurately convert speech to text is in noisy conditions is important.

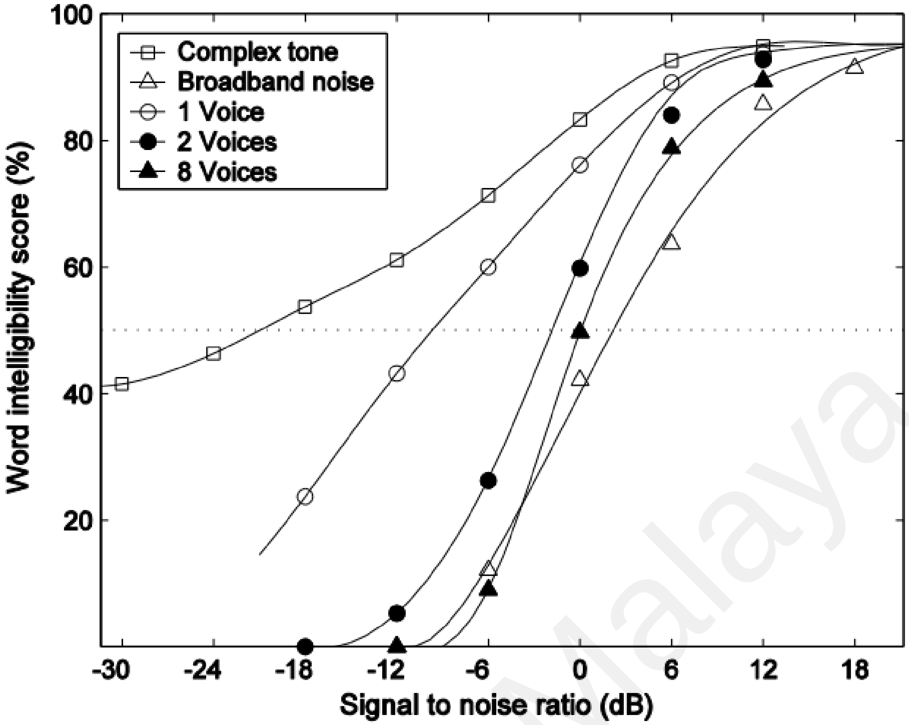


Figure 1.2: Word intelligibility score with respect to Signal to Noise (SNR) for different kinds of interference (Wang and Chen, 2017)

Figure 1.2 shows the intelligibility score of humans (recognizing words) to the SNR. As shown in figure 1.2, it is challenging for humans to understand sounds when tainted with broadband noises. If 0 SNR (50% noise) is considered, humans are poor at recognizing sounds with broadband noise (non-speech) as compared with voices (speech noise). However, this trend reverses in the case of an ASR model. To date, ASR models are more efficient at separating broadband noise as compared to voice noises. It raises a research gap to increase the efficiency of speech separation tainted with other speech noise.

1.3 Problem Statement

1.3.1 Optimum features for joint Speech Separation and ASR model

The latest trends show that the joint Speech Separation (SS) and ASR models perform well in separating speech from voiced noises (Tu',2016). Wang (2017) reviewed all major speech separation systems and concluded that join speech separation shows more accurate results. Tu’s (Tu, 2016) model outperforms its predecessors because both the SS and the ASR is joint together to create a robust model with better recognition of speech. However, the constraint of this approach is that the same features must be used throughout the system (both the SS and the ASR). This is because the speech separator and the ASR are joined together in Tu’s model. As such, the features used by the SS and the ASR has to be the same. Researches have been conducted on features used in speech separation and automatic speech recognizer separately (Delfarah, 2017). Different speech features were used on the SS to find which feature yields the most accurate results. However, there remains a gap in finding out the most optimum features for joint SS and ASR models. It is crucial to find the feature that works best for the joint systems to increase the accuracy of the model and make it even more efficient.

This issue was raised because the optimum features in an SS might not be the same for the ASR (Delfarah, 2017). In such a case, a tradeoff must be made to identify the features that yield the most accurate results for the combined joint SS and ASR model.

1.3.2 Performance of joint Speech Separation and ASR Model at different SNR levels

As the SNR of the signal decreases (noise increases), the accuracy of the ASR model starts to fall. In the real environment Barker (2001), ASR models need to deal with noisy signals. As such, it is important to test a model at a very low SNR to get an accurate evaluation. Tu's

(2016) joint SS and ASR model was evaluated as low as -6dB, which is not adequate, which means that Tu's (2016) model is not evaluated at a much lower SNR levels.

1.4 Objectives

The main aim of this research is to devise a method for identifying the most influencing speech features towards increasing the recognition accuracy of joint SS and ASR model. The specific objectives are:

1. To identify speech features used in the existing speech separation model for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal).
2. To develop a method for experimenting the most influencing speech features based on the identified speech features in objective (1) using joint speech separation and the ASR model.
3. To evaluate and compare the performance of joint Speech separation and ASR model against the speech separation model using the identified features at different signal-to-noise ratios.

1.5 Research Questions

1. What are the speech features used in the existing SS model for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal)? (Objective 1)

2. What are the method(s) for experimenting the most influencing speech features -using the joint Speech separation and the ASR model at different signal-to-noise ratio? (Objective 2)
3. How is the performance of joint Speech separation and ASR model against the speech separation model using the identified features at different signal-to-noise ratios? (Objective 3)

1.6 Scope

This research focuses on monaural speech separation rather than array-based. It is not always possible to set up multiple microphones at calculated lengths, which is why a monaural speech separation is more robust and portable.

All training will be using the English language and English spoken words. This research focuses on the noise that is similar to previous speech separation works for voice noises (Huang, 2014) (Zhang, 2016). So the SNR of the data prepared will only depend on two sound sources; target speaker and interfering speaker.

Tu (2016) also shows that a SI ASR model performs poorly compared to the target speaker-independent (TSD) system. The reason is; by limiting the conditions of the environment (in this case the target speaker) a better WER can be reached. From this, it can be inferred that if both the target and the interfering speaker are known (speaker-dependent), then the WER can be improved further. However, in Tu's (2016), the joint SS and ASR model only experiment with the speaker-dependent system. This is why both the target and the interfering speaker will be known to yield a better and clearer result in our research. Therefore this research will focus on both target speaker-dependent and interfering speaker-dependent systems.

1.7 Research Methodology

The current research study proposes a research methodology that composes of four main steps. The first one aims to review the existing research works on Speech separation and joint speech separation and ASR models. In particular, a few features were applied for speech separation in previous studies. Besides, this research also reviews the previous evaluation techniques that were used in speech separation studies. The next step is to develop a data-set that is suitable for our study. Since the number of experiments the researcher plan to perform is high; the database should be moderate in terms of size. The third step represents the most significant part: the development of speech separation followed by the joint speech separation and ASR model. It starts by developing a speech separation model using the training data set. Next, a basic ASR is developed which is then added to the speech separation to build the joint speech separation and ASR model. Phase three is repeated using each feature (selected in phase two). The fourth and last phase of our methodology aims to conduct a comparative study between different models built in phase three for each feature.

1.8 Significance of research

As shown in the above literature, the feature selection used to train a model can affect the accuracy of the model. Tu's (2016) model has yet to be experimented with different features to find the optimum feature. Such an experiment should help to understand the model better and identify which feature gives the most accurate results. In real-life scenarios, noise can impact the accuracy of a model as well hence. Therefore looking at the accuracy of the Joint

SS and ASR model at different SNR levels is important to identify which features work best in which environments.

1.9 Structure of the thesis

Chapter-2: This chapter describes comprehensive information about this research through the review of literature. It describes the techniques and models of speech separation, ASR models and in depth study of existing feature extraction methods for both.

Chapter-3: This chapter describes all the necessary stages of this research that were carried out, such as research problem and solution, data-set collection, design and development, and evaluation method, which help to meet the research goals.

Chapter-4: This chapter describes how the experiment was developed, such as configurations of the speech separation and ASR, tools and training models that were used.

Chapter-5: This chapter focuses on the evaluation and results of the speech separation and the joint speech separation and ASR model.

Chapter-6: This chapter summarizes and concludes this study. Besides that, this chapter describes the research contributions, the research limitations, and the future works related to this research.

CHAPTER 2 : Literature Review

2.1 Overview of this chapter

This chapter gives an overview of the components and techniques used in speech separation and ASR models. It discusses existing works and methods used in previous studies to build ASR and speech separation systems. Following that it looks further into previous attempts to join ASR and speech separation for more accurate results. This chapter also reviews the features and evaluation techniques in previous speech separation and ASR models.

2.2 Basic Speech separation Components

A description of a basic speech separation built by Yanhui et al. (2014) was made in chapter 1 (Section 1.1.2). The framework is divided into two stages: training and separation as depicted in Figure 2.1. The components in each of the stages are as follow:

Training:

1. Mixture/Sources Sample Pairs
2. Feature Extraction
3. DNN Training

Separation:

1. Mixture Utterance
2. Feature Extraction
3. Target Separation
4. Target Waveform Reconstruction

Today there is a range of different speech separation, each of them uses different **approaches**. These included:

1. Target User
2. Channels
3. Feature extractions
4. Training Targets

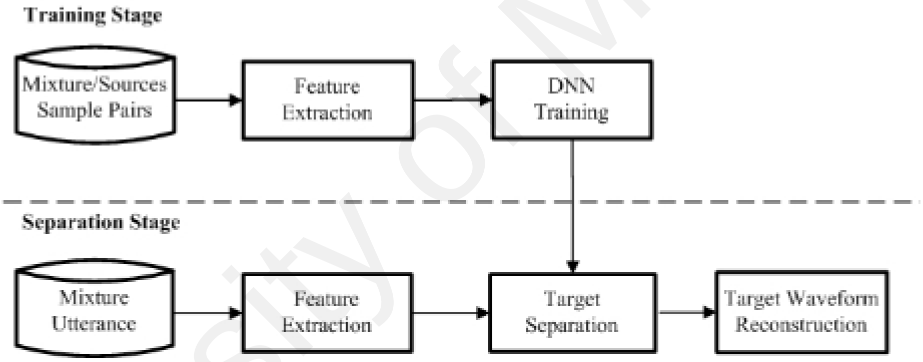


Figure 2.1: Speech separation framework (Tu, 2014)

The design of the components of a speech separation will depend on the techniques we have used. Each technique can influence one or more components in a speech separation. Table 2.1 shows the techniques and the influence of the components used in Yanhui et al. (2014) study.

Table 2.1: Relation between different approaches of designing speech separation and their effects on the components

Approaches	**Stages**	**Components**
Target User	Training	Mixture/Sources Sample Pairs
	Separation	Mixture Utterance
Channels	Training	Mixture/Sources Sample Pairs
		DNN Training
	Separation	Mixture Utterance
Feature Extraction	Training	Feature Extraction
	Separation	Feature Extraction
Training Targets	Training	DNN Training
	Separation	Target Separation
		Target Waveform Reconstruction

2.2.1 Target User

Speech separation can be a speaker-dependent system or speaker-independent system where all the speakers are known, and training is carried out using their voices speaker-independent: where neither the target speaker nor the interfering speaker's voice is available during training and target-dependent speaker, where only the speech of the target speaker is used to for training (Jun, 2014). The models used in speaker-dependent and target speaker-dependent are very similar. Their key difference lies in the data used to train the DNN model. In the former, both the target speaker and the interfering speaker's voice is used to train and test the DNN. While in the latter only the target speaker remains constant but the interfering speaker may be multiple and the interfering speaker used during testing differs from the training.

In this research, the components Mixture/Sources Sample Pairs and Mixture Utterance will be influenced, based on the technique used for the target user. This is because both these components are used to prepare training/separation data. Therefore the type of user (Target user) influences the development of the dataset.

2.2.2 Channels

Co-channel speech separation refers to a system that simultaneously separates both the target and interfering speech (Du, 2014). Figure 2.2 and 2.3 shows the two common structures of a co-channel and a single-channel speech separation. By referring to figure 2.2, the input layer only contains a single channel (the mixed speech of two users), while the output is separated into two sources. Figure 2.3 shows that the input and the output have only one set of nodes in each layer. There is no parallel set of nodes in the output layer since the model is trained to isolate the target user's speech.

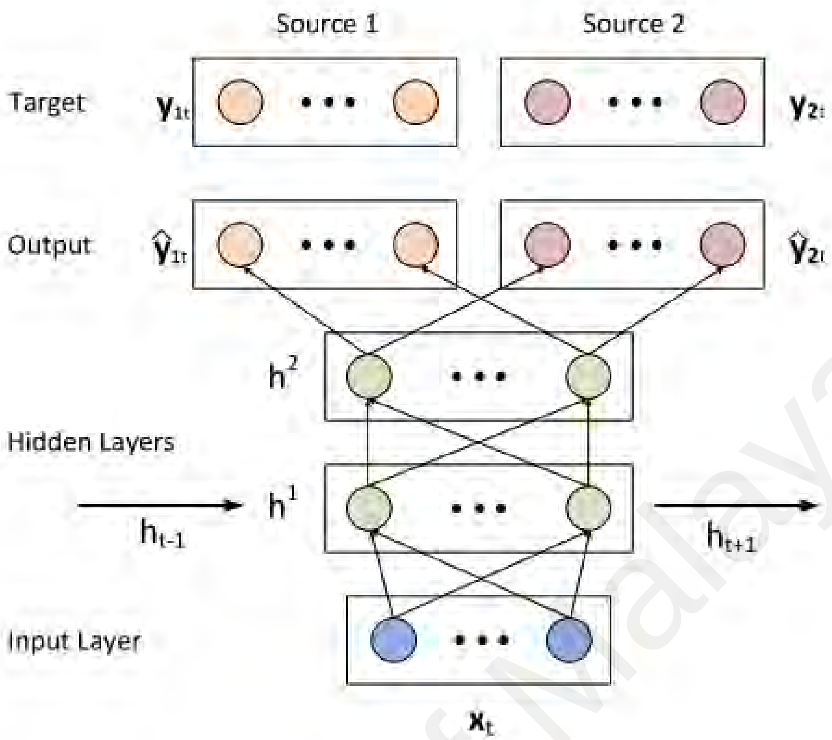


Figure 2.2: Co-channel Speech separation (Huang, 2014)

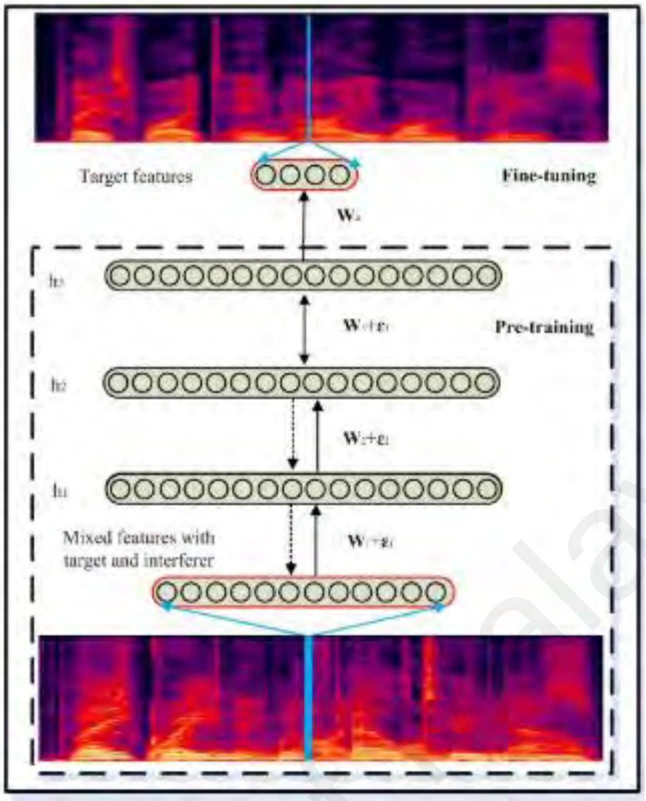


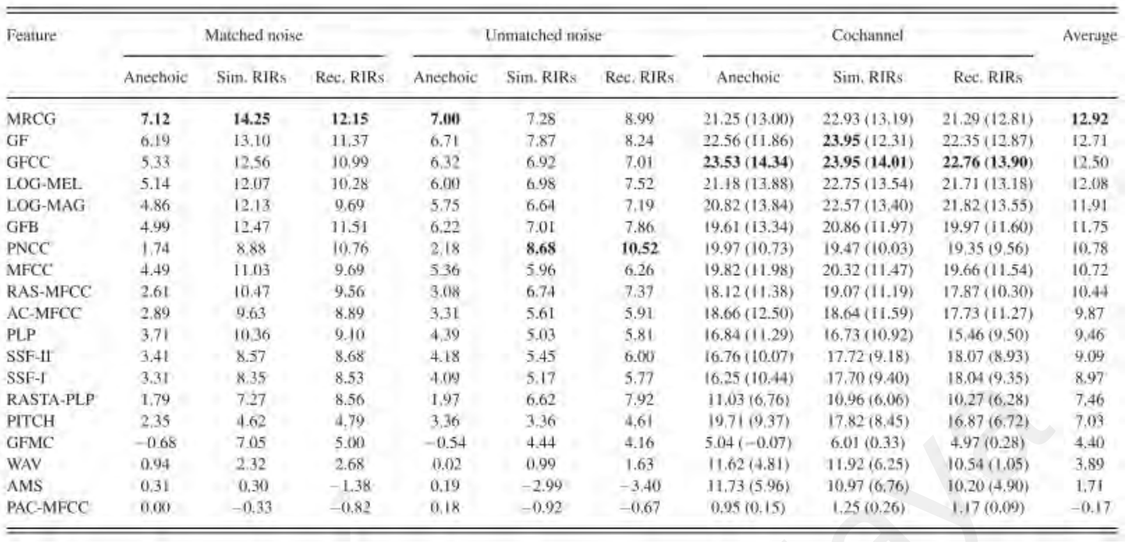
Figure 2.3: Single channel speech separation (Du, 2014)

The type of channels used to build a speech separation will influence the Mixture/Sources Sample Pairs and Mixture Utterance since these stages are used for data preparation. It will also influence the design of the DNN Training. As shown in Figures 2.2 and 2.3, the Neural Network design for single-channel differs from co-channel speech separation.

Single-channel and multichannel can also be related to the array-based and monaural speech separation. As explained in chapter 1 in array-based speech separation multiple microphones can be used to record the speech which can result in different multiple channels. So in this case array-based approach is appropriate for multichannel speech separation. Monaural speech separation indicates that there is only one source, i.e one microphone. This can translate to single-channel speech separation.

2.2.3 Feature Extraction of speech signal in speech separation

Feature extraction is one of the main processes that dictate the accuracy of the speech separation. It converts the audio signal into a string of frames which are represented by a suitable matrix (features). In recent studies, the following features were used namely; STFT (Zhang, 2016), LOG-MEL (Wang, 2017), and LOG-MAG (Du, 2014), Eric Healy (2017) introduced combined features for speech separation however the accuracy had little or no improvement. Masood Delfarah (2017) conducted a detailed study of different features in masking based speech separation. It dictates the importance of selecting the correct feature extraction in any speech separation system and how much it could affect the accuracy of the output. For example in Delfarah and Wang (2017), it can be observed that the STOI score can range from 12.92 (most accurate) to -0.17 (least accurate) just by changing the type of features used in the same speech separation system (Refer to Figure 2.4).



Feature	Matched noise			Unmatched noise			Cochannel			Average
	Anechoic	Sim. RIRs	Rec. RIRs	Anechoic	Sim. RIRs	Rec. RIRs	Anechoic	Sim. RIRs	Rec. RIRs	
MRCG	**7.12**	**14.25**	**12.15**	**7.00**	7.28	8.99	21.25 (13.00)	22.93 (13.19)	21.29 (12.81)	**12.92**
GF	6.19	13.10	11.37	6.71	7.87	8.24	22.56 (11.86)	**23.95** (12.31)	22.35 (12.87)	12.71
GFCC	5.33	12.56	10.99	6.32	6.92	7.01	**23.53 (14.34)**	**23.95 (14.01)**	**22.76 (13.90)**	12.50
LOG-MEL	5.14	12.07	10.28	6.00	6.98	7.52	21.18 (13.88)	22.75 (13.54)	21.71 (13.18)	12.08
LOG-MAG	4.86	12.13	9.69	5.75	6.64	7.19	20.82 (13.84)	22.57 (13.40)	21.82 (13.55)	11.91
GFB	4.99	12.47	11.51	6.22	7.01	7.86	19.61 (13.34)	20.86 (11.97)	19.97 (11.60)	11.75
PNCC	1.74	8.88	10.76	2.18	**8.68**	**10.52**	19.97 (10.73)	19.47 (10.03)	19.35 (9.56)	10.78
MFCC	4.49	11.03	9.69	5.36	5.96	6.26	19.82 (11.98)	20.32 (11.47)	19.66 (11.54)	10.72
RAS-MFCC	2.61	10.47	9.56	3.08	6.74	7.37	18.12 (11.38)	19.07 (11.19)	17.87 (10.30)	10.44
AC-MFCC	2.89	9.63	8.89	3.31	5.61	5.91	18.66 (12.50)	18.64 (11.59)	17.73 (11.27)	9.87
PLP	3.71	10.36	9.10	4.39	5.03	5.81	16.84 (11.29)	16.73 (10.92)	15.46 (9.50)	9.46
SSF-II	3.41	8.57	8.68	4.18	5.45	6.00	16.76 (10.07)	17.72 (9.18)	18.07 (8.93)	9.09
SSF-I	3.31	8.35	8.53	4.09	5.17	5.77	16.25 (10.44)	17.70 (9.40)	18.04 (9.35)	8.97
RASTA-PLP	1.79	7.27	8.56	1.97	6.62	7.92	11.03 (6.76)	10.96 (6.06)	10.27 (6.28)	7.46
PITCH	2.35	4.62	4.79	3.36	3.36	4.61	19.71 (9.37)	17.82 (8.45)	16.87 (6.72)	7.03
GFMC	−0.68	7.05	5.00	−0.54	4.44	4.16	5.04 (−0.07)	6.01 (0.33)	4.97 (0.28)	4.40
WAV	0.94	2.32	2.68	0.02	0.99	1.63	11.62 (4.81)	11.92 (6.25)	10.54 (1.05)	3.89
AMS	0.31	0.30	−1.38	0.19	−2.99	−3.40	11.73 (5.96)	10.97 (6.76)	10.20 (4.90)	1.71
PAC-MFCC	0.00	−0.33	−0.82	0.18	−0.92	−0.67	0.95 (0.15)	1.25 (0.26)	1.17 (0.09)	−0.17

Figure 2.4: Performance of features in SS (Delfarah and Wang, 2017)

The features used in the speech separation will influence the design of the feature extraction component in both the training and separation stage. This is because these two components convert the input sound signal to speech features.

2.2.4 Training Targets

The DNNs are trained using supervised learning. In the training stage, the noisy speech signals are segmented into smaller frames and a string of features are extracted from those frames (feature extraction). These features serve as inputs for the DNN. However, as DNNs are supervised learning technique; a training target has to be provided. A training target is used as the output from which a DNN can learn. Hence the DNN compares the input frame and the training target to modify its weight and develop a way to map it.

Two major categories of training targets have emerged namely, the mapping-based showed in Figure 2.5 and the masking-based speech separation (Wang, 2017), as depicted in Figure 2.6. The difference between the two techniques lies in the training targets used during training. In mapping based speech separation, the training target is the clean speech features; this means the DNN directly predicts a stream of cleaned features.

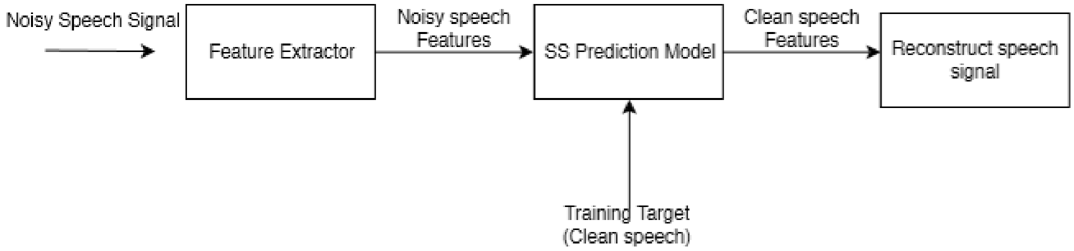


Figure 2.5: Mapping based speech separation

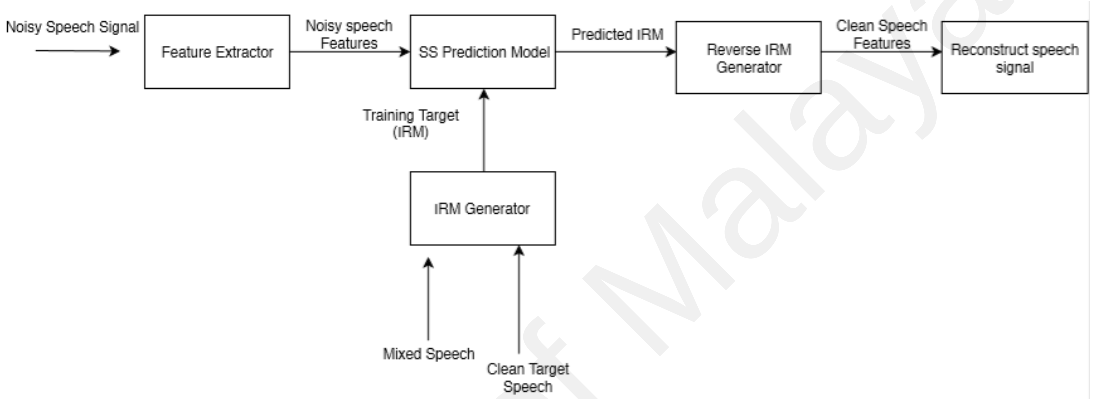


Figure 2.6: Masking based speech separation

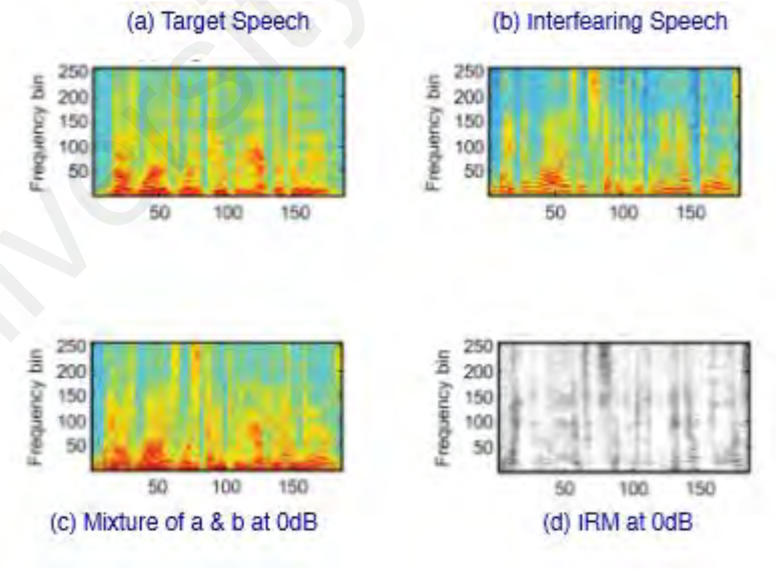


Figure 2.7: Spectrogram of target, interferer, mixture and IRM (Zhang, 2016)

Mapping-based speech separation uses clean and noisy speech of the same speech to identify the relation between them. The noisy speech signal is passed through the feature extraction which outputs a list of features split into small frames. The features are fed as input into the SS prediction model (usually DNN is used as prediction models in this case). Training the model requires training targets. In this case, the training targets are the speech features of the same frames; but clean. The model then tries to map the noisy speech feature to the clean features (training model). Once the model is trained the training targets are no longer required. Once trained, the model can be fed with noisy speech features which in-turn will map it to its (predictive) clean speech features, which are later reconstructed, frame by frame to the clean speech signal.

Similar to mapping based speech separation, masking based speech separation also uses training targets. However, the training targets used in masking based speech separation is different. Instead of using the clean speech as the training targets, masking-based training targets use a variation of the clean speech such as IBM or IRM. The IRM or IBM is derived from clean and noisy speech. Two extra components are added in masking based speech separation namely, the IRM generator and the reverse IRM generator. The task of IRM generator component is to get a derived representation of the clean speech which in this case is IRM. The reverse IRM generator component gets the predicted clean IRM as input which then reverses to its original speech features.

Figure 2.7 shows 4 spectrographs. Figure 2.7 (a) is the target speech which is used as the training target for mapping based speech separation. 2.7 (b) is the interfering speech and both the target speech and interfering speech is merged to create a mixture which is shown in figure 2.7 (c). The mixture is used as an input in both mapping and masking based speech separation. Figure 2.7 (d) is the IRM which is derived from the clean speech in Figure 2.8 (b) and it is used as the training target for masking-based speech separation.

Zhang (2016) discusses the advantages of masking and mapping techniques and concludes that:

(i) “The masking-based approach is more effective in utilizing the clean training speech of a target speaker.”

(ii) “The mapping-based method is less sensitive to the SNR variation of a training corpus.”

(iii) “Given a training corpus with a fixed mixture SNR and plenty of clean training speech from the target speaker, the mapping and masking-based methods tend to perform equally well.”

The components of DNN Training are used to train the model and Target separation is the trained DNN model from the training stage. Both of these components use the training targets which is why the type of training target will influence its design as shown in Table 2.1.

Target Waveform Reconstruction is used to rebuild the clean speech from clean speech features. Depending on the training targets used the reconstruction has to be altered. For example, if IRM training targets are used then the output of the DNN model will be an IRM feature that has to be reconstructed. This will influence the design of the reconstruction component.

2.2.5 Previous works on speech separation

Table 2.2 shows the recent works on the speech separation system in monaural conditions, built for speaker-dependent (SD), and target speaker-dependent (TSD) speech separation system based on deep learning techniques.

Table 2.2: Summary of speech separation models used in previous researches

Ref	**Training Target**		**Feature Extraction**	**User Type**	**Channel**
(Huang, 2014)	Mapping + Masking layer	Clean Speech	STFT, LOG-MEL	SD	Co channel
(Huang, 2015)	Mapping + Masking layer	Clean Speech	STFT, LOG-MEL	SD	Co channel
(Jun, Du, 2014)	Mapping	Clean Speech	LOG-MAG	TSD, SD	Single channel
(Jun, 2014)	Mapping	Clean Speech	LOG-MAG	TSD, SD	Co Channel
(Zhang, 2016)	Masking	IRM	STFT	TSD, SD	Single
(Healy, 2017)	Masking	IRM	Combined	SD	Single

From the literature review conducted on Speech separation, it is clear that both the masking and the mapping based models are common. Masking-based models use the IRM as training targets instead of the IBM.

LOG-MAG, LOG-MEL, and STFT are all commonly used. Researchers seem to prefer speaker-dependent systems, this could be used to the fact that it’s much easier to evaluate such a system. There have been works done on both the co-channel and single-channel speech separation. However, the single-channel is more applicable in real life due to the limitation of the environmental setup. Co-channel microphones have to be set up in a calculated position for it to work best which is why in real-life scenarios it is more common to have single-channel microphones.

2.3 Automatic Speech Recognition (ASR)

Automatic speech recognition uses speech signals to analyze and predict the series of lexis. In other words, it converts speech signals to test data. Gupta (2016) concludes that a typical ASR model contains:

(i) **Feature Extraction**: Extract valuable information from a given signal

(ii) **Decoder**: Combine the prediction of Pronunciation, Acoustic and language model to make a final prediction

(iii) **Pronunciation Models**: Different words may have different pronunciations, especially since there are many different accents for the same word.

(iv) **Acoustic Models**: Predicts the word/phoneme/tri-phone of a set of feature

(v) **Language Models**: Predicts the most likely word spoken based on the grammar and the language context. This research will only be looking into the acoustic model, which means other models including the decoder will not be part of this research.

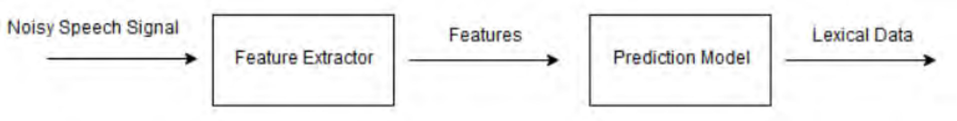


Figure 2.8: ASR model

A traditional ASR system has two stages; firstly a set of features will be extracted from the input signal (feature extraction), secondly, a prediction model will be used to train a model that will categorize a set of features.

The most common method for evaluating the ASR model is the word error rate (WER) and symbol error rate (SER). A lower WER means higher accuracy and vice versa.

2.3.1 Feature Extraction of ASR model

Feature extraction is the first step of the ASRC component. It converts the audio signal into a string of frames which are represented by a suitable matrix (features). Two commonly used features are log-mel filter bank (LMFB) and Mel-frequency cepstral coefficients (MFCC) (Wei Han, 2006). The techniques used to extract the features will affect the WER of an ASR. The output of this step is a string of numbers known as features for each frame of the speech signal.

2.3.2 Prediction Model (Acoustic model)

The second step of a traditional ASR is the prediction model. This model will ultimately predict a label for the frames from feature extraction. The feature extraction component splits the signal into shorter frames and outputs a list of features for each of the frames. The job of the prediction model is to assign a phoneme label to each of these frames. Phonemes are units of sound that can be used to distinguish one word from another. Therefore, the prediction model attempts to predict the sound being made on each of the frames. Once all the frames are labeled, the phonemes can be concatenated together. The concatenated phonemes will spell out a word which the user was trying to utter. The machine learning models that have worked well in the past are SVM and GMM. With the recent advance of DNNs, a new generation of machine learning techniques are used for speech separation. The current state of the art models used in ASRC are:

1. DNN-HMM (J. Pan, 2012)
2. RNN (Z. C. Lipton, 2015)
3. CNN (Z. C. Lipton, 2015)

2.4 Joint SS and ASR Model

Tu (2016) proposed another combined speech separation and recognition model eliminating the need for feature extraction performed twice as well as the signal reconstruction. Delfarah's (2017) work shows feature extraction may take a long time, this is a significant improvement as the accuracy of the system was not affected. Tu (2016) merged the SS and ASR models into one and uses just one feature extraction component (Tu 2016). The initial step of the joint SS and ASR model is the feature extraction (LMFB) component, eliminating the need for resynthesizing the clean speech since the clean features can be used directly as inputs for the acoustic model. The clean features are then used in a standard DNN-HMM acoustic model to predict the lexical data. In the recognition stage; only one feature extraction component is used.

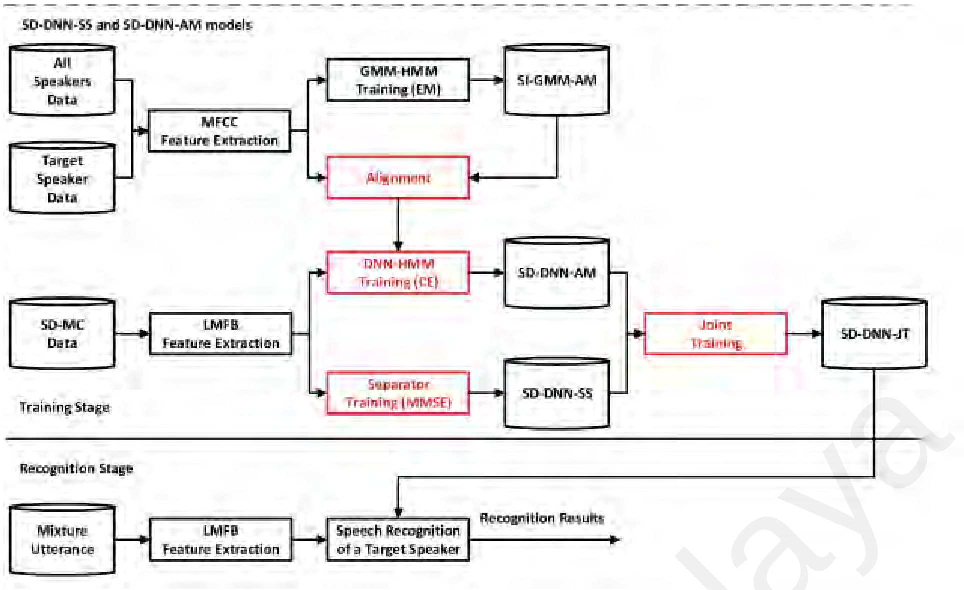


Figure 2.9: joint SS and ASR model (Tu, 2016)

Figure 2.9 depicts the joint SS and ASR model as proposed in Tu (2016). Initially, Tu (2016) mixed the target and interfering speaker data to synthesize a dataset. A standard three-layer DNN was used as the separator. The mapping-based separator was trained using both the clean and noisy data. Next, the audio signals are labeled/aligned with the correct phonemes. To do so, Zhang and Glass (2009) proposed the noisy data aligned using its original clean data. A DNN-HMM model was used to develop the ASR model, instead of using mono-phones triphones. The previously labeled data (during the alignment phase) was used to train the DNN-HMM model.

2.5 List of Features used in speech separation

The objective of this research is to identify the speech features used in the existing speech separation model. This section provides an overview of some of the speech features used in speech separation and ASR.

2.5.1 Short-time Fourier transform (STFT)

The Fourier transform (FT) decomposes a signal into the frequencies that make it up. Hence a FT will result in a list of frequency and intensity of a given signal. When a signal is sliced into smaller portions and FT is computed on each of the slices separately; then it is known as the Short-time Fourier transform (STFT).

2.5.2 Log-Mag

Log magnitude is a log operation performed over the magnitude of STFT.

2.5.3 Log-Power Mag

Log Power Mag is the result of a squared STFT followed by a log operation.

2.5.4 Log-Mel

In Log-Mel the STFT is processed by a mel-filterbank followed by a log operation. The human auditory system perceives the audio frequencies in a non-linear manner. To mimic this non-linearity of the human ear, a Mel-filterbank was introduced. The Mel-filterbank transforms the audio signal into how a human ear would interpret it. Figure 2.10 shows a mel filter bank.

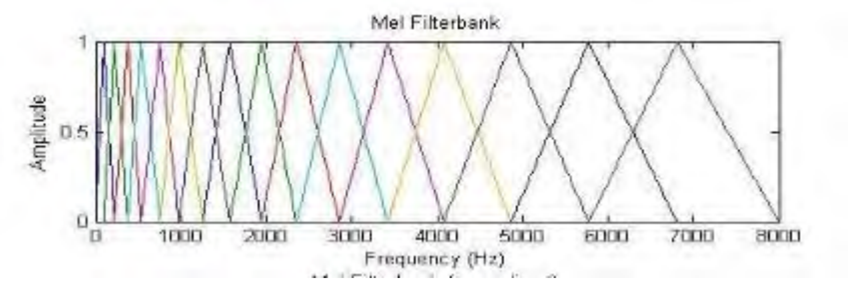


Figure 2.10: A Mel filter bank

2.5.5 Gama-tone Filter bank

A Gama-tone filter bank is quite similar to a Log-Mel filter bank that uses the same principle of human sound perception. However, the filter of the Gamma-tone is smoothened out to give a more gradual continuity in the results. The STFT is pass through a Gama-tone filter bank to compute the features as depicted in Figure 2.12.

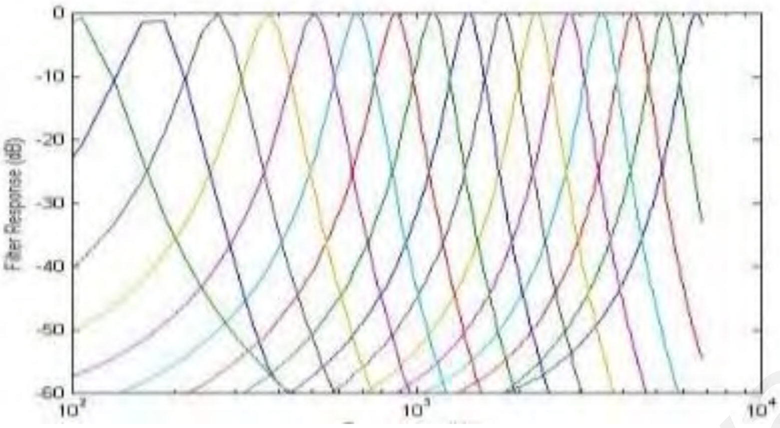


Figure 2.11: A Gama-tone filter bank

2.5.6 Mel frequency cepstral coefficient (MFCC)

Mel frequency cepstral coefficient (MFCC) is a derivation of the log-Mel features. First, the log-mel features are extracted, followed by a discrete cosine transformation (DCT). Here the DCT acts as an inverse short-time Fourier transform (ISTFT). However, the resultant coefficients are no longer in the frequency domain but rather in a cepstral domain due to the extra log operation performed when calculating the Log-Mel features.

2.5.7 Gamatone frequency cepstral coefficient (GFCC)

Gamatone frequency cepstral coefficient (GFCC) is similar to MFCC but instead of being a derivation of Log-Mel features, it is a derivation of Gamatone filter features.

2.5.8 Amplitude modulation spectrum (AMS)

To compute the amplitude modulation spectrum (AMS), first, the full-wave rectifier envelop is generated and multiplied by a factor of four. Next, the signal is sliced into frames and the

Hann function is executed on each frame. Following that, a 256 Point FT is computed on each frame (STFT). The results are multiplied by 15 triangular-shaped windows that are uniformly centered in the range of 15.6 Hz and 400 Hz.

2.5.9 Power-Normalized Cepstral Coefficients (PNCC)

To compute the Power-Normalized Cepstral Coefficients (PNCC) feature, STFT is first integrated with gammatone filter bank. Then, the asymmetric noise suppression procedure detects a lower envelope of the filtered spectrum as the noise floor. This lower envelope is then utilized to perform the masking on the noisy spectrum to remove the background and linear noises. The masked spectrum is compressed by the fifteenth-root operation, and finally, DCT to yield the PNCC features.

2.5.10 Relative Spectral - Perceptual Linear Predictive (RASTA - PLP)

PLP is based on a short-term speech spectrum which is modified by the psychophysically spectral transformation. However, PLP is vulnerable when it is modified by the frequency response of the communication channel. Humans are not as vulnerable to this, therefore a relative spectral methodology is used to make the PLP features more robust and tolerable to the frequency response of communication.

2.6 Features used in speech separation in previous studies

The effect of features for speech separation has been investigated by Wang (2014). Wang's study uses Rasta-PLP, MFCC, and COMB to investigate the accuracy of speech separation with regards to two parameters; matched/unmatched sound and pre-trained and non-pre-trained Restricted Boltzmen model (RBM). The SNR level of the noise was kept constant at

0dB. Voiced sounds from the opposite genders were used as noise for the interfering speaker. Han (2011) used a combination of pith and AMS to investigate the accuracy of masking based SVM speech separation at SNR levels -5 dB and 0 dB only. Unvoiced sounds were used as the noise in his experiments. Wang (2014), used another combination of AMS, Rasta-PPL, MFCC, and GF on speech separation models to investigate the effects of different training targets such as IBM, IRM, FFT-Mag, etc at SNR levels -5 dB and 0 dB. Unvoiced sounds were used as the noise in his experiments. Chen (2014), proposes a new combined feature that includes RAS-MFCC, GF, and PNCC, and compares the accuracy of speech separation at 0 dB for unmatched sounds. Except for Wang's (2013) work, most of the research focused on unvoiced sounds. The effect of SNRs was not investigated thoroughly as many of the existing experiments focused on noise levels between 0 dB and 5 dB SNR.

For the voiced speech separation joint SS and ASR models, Tu (2014) used the log power spectrum and Tu (2016) used log-mel filter-banks (LMFB). However, in either of these papers, the reasons for selecting these features were not explained. These works focus more on the model rather than the features. However, the type of features used in SS and ASR models can greatly affect the accuracy as shown in Masood (2017). However, such studies have not been conducted using the joint SS and ASR models. Given that the joint SS and ASR model shows good accuracy for noisy speech recognition, it is important to investigate the best features for the joint system.

Wang Y. (2017), Kolbæk (2017), Zhang (2016), and Huang (2011) have used short-time Fourier transform features (STFT) in their research. Despite being an old feature, STFT is used as the 'go-to' feature for speech separation. STFT is the computed FFT for each of the split frames. Chen (2017) and Hershey (2016) used Log magnitude (log-mag) features. Log-

Mag is a derivation from STFT by performing a log operation on the STFT. Du (2016), Tu (2014), Wang (2017), and Du (2014) used Log power spectrum (log-power).

Log-power is just as commonly used as STFT in voiced speech separation. Log-power is another derivation of STFT, where STFT is powered by two followed by log operation. Log-power seems to be more commonly used than log mag since the extra power operation helps to distinguish small changes in frequency bins with close values that might assist in distinguishing between the noise and clean signal. Po-Sen Huang (2011) and Po-Sen Huang (2015) used Log Mel features. Log mel features mimic the human ear by using mel-frequencies. Log mel has shown good results in other parts of ASR research. However, it is unknown as to why it is not used frequently in speech separation studies. Other features used in speech separation include AMS, Rasta-PLP, MFCC GF, GFCC, and PNCC.

A literature review was conducted to search for all speech separation (voiced noise) using deep learning. Table 2.3 shows the speech features used in the existing literature:

Table 2.3: Speech features used in existing researches

Ref	STFT	Log-mel	Log-power	Log-mag	AMS	RASTA-PLP	MFCC	GF	GFCC	PNCC
Huang (2014)	🟩	🟩								
Huang (2015)		🟩								
Du. (2014)			🟩							
Tu.(2014)			🟩							
Du.(2014)			🟩							
Zang (2016)	🟩									
Wang (2017)			🟩							

Healy (2017)					🟩	🟩	🟩	🟩		🟩
Hu (2013)									🟩	
Hershey (2016)				🟩						
Chen (2017)				🟩						
Wang Y. (2017)	🟩									
Kolbak (2017)	🟩									

Based on Table 2.3, the two most popular choices of features for speech separation are STFT and log-power. These are some of the oldest features which performed well. Log-mel and Log-mag are the third and fourth most common features used. Log-mel is a derivation from STFT which uses an extra mel filter.

2.7 Signal to Noise Ratio (SNR) used for evaluating speech separation and ASR models

Researches have been conducted to evaluate new speech features and models. Each of them has used different ranges of SNR to test the respective models and features as shown in Figure 2.13.

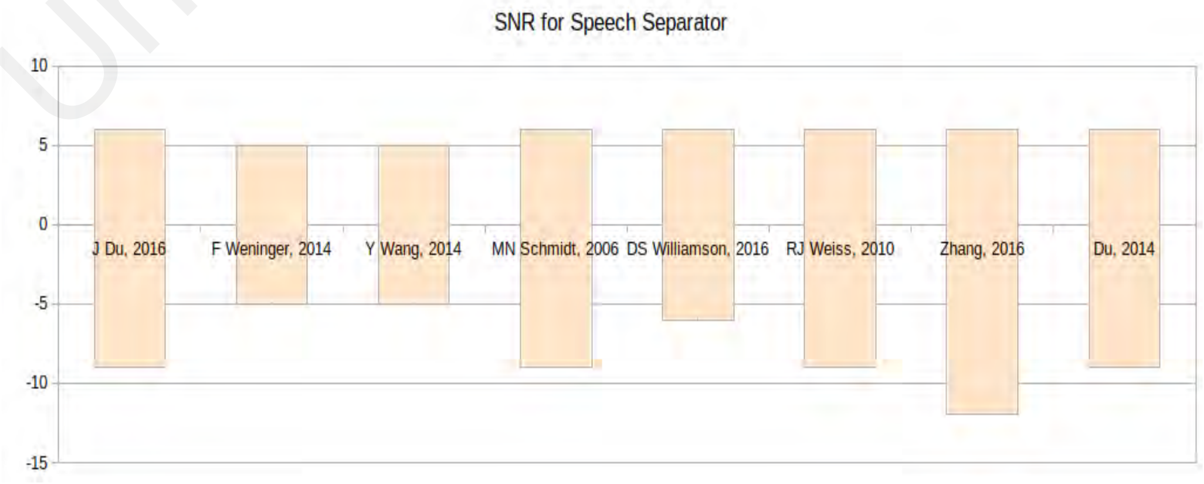


Figure 2.12: SNR ranges used in previous researches in speech separation

For SS, the range of SNR, the highest upper bound used is +6 and the lowest lower bound used is -12 to evaluate a speech separation. Du (2016), Weringer (2014), and Wang (2014) used the upper bound of SNR +5 while Schmidt (2006), Williamson (2016), Weiss (2010), Zhang (2016), and Du (2014) used an upper bound of +6. The lower bound varies more in comparison to the upper bound. This might be due to the limitation of the respective models used in each of the studies. However on average, the upper and lower bound is between +5.75 and -8.

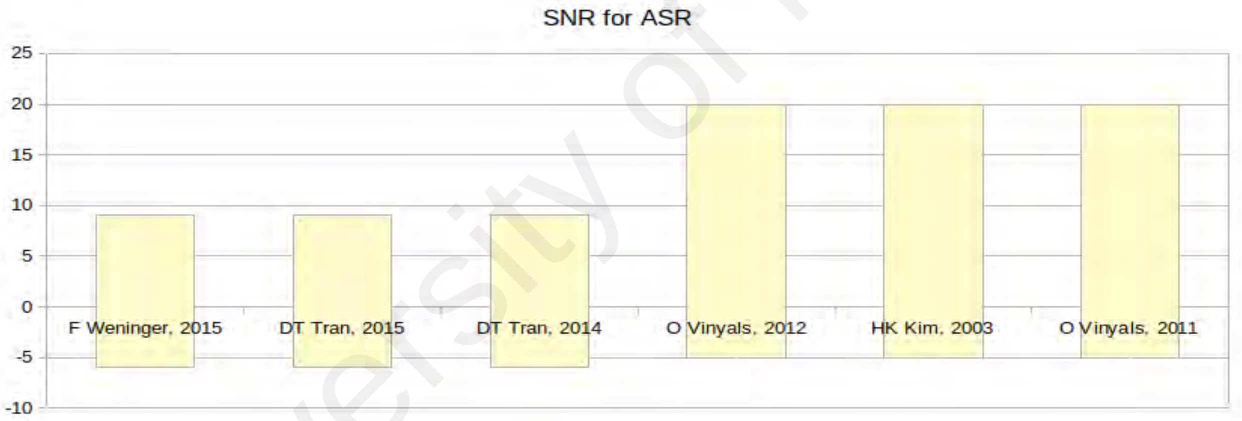


Figure 2.13: SNR ranges used in previous researches in ASR

Unlike speech separation, the ranges of SNR used to evaluate ASR are wider, with the upper-bounds of SNR used are +9 by F Weninger (2015), Tran (2015), Tran (2014), and +20 by Vinyals (2012), Kim (2003) and Vinyals (2011) averaging +14.5. The lower bounds of SNR used are -6 and -5, averaging -5.5.

Table 2.4: Summary of SNR ranges used in previous researches

	Average Upper Bound	Average Lower Bound
Speech separation	+5.75	-8.00
ASRs	+14.5	-5.50

Speech separations are designed to reduce noise which is why testing separators at a low SNR is very important. This might explain why the average upper bound of speech separation is much lower than the ASR. ASR models are designed to decode the lexical data in the speech signal. Unlike speech separation, its primary objective is not to decode noisy signals. This would explain why the upper bound of the ASR model is higher than separator since ASR models tend to be tested with much cleaner data.

2.8 Research Problems and Solutions

Tu (2016), proposed a joint SS and ASR model. Tu's work was focused on the mechanics of joining the two speech separator and ASR model to create a joint SS and ASR. However, the model was not evaluated to identify features that resulted in the most accurate results. In Delfarah's (2017), it was concluded the type of speech features can influence the accuracy of speech separation. Therefore, in this research, the focus is on evaluating the joint SS and ASR model will different speech features. This research aims to identify the features that yield the best results for both the speech separator and the ASR model thus adding on Tu's (2016) model to improve its accuracy. Moreover, the work in Tu (2016) can be further evaluated using different SNR levels. This will not only produce the best features to use but also indicate the best features to be used at different SNR levels. By understanding the best features at different SNR levels, the findings can be applied in different real-life environments to obtain optimum recognition accuracy. Therefore finding the optimum features at different SNR will provide an in-depth understanding of Tu's (2016) model and depict the best ways to use this model to increase accuracy.

CHAPTER 3 : RESEARCH METHODOLOGY

The objective of this research is to identify the most influencing speech feature toward improving the recognition accuracy of the joint SS and ASR model at a different signal-to-noise ratio (SAR). To do this, a joint SS and ASR model has to be developed and an experimental setup has to be designed to evaluate the model at different SAR levels.

3.1 Research methodology flow

To achieve the research objective, the following method is proposed:

i. Identify Features: In the literature review speech features used in the existing speech separation model for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal) have been identified.

ii. Prepare Dataset: The dataset required to train and test the Join SS and ASR model at different SNR are prepared.

iii. Develop / Train Join SS and ASR model: Tu (2016) joint SS and ASR is developed for 10 features.

iv. Evaluate joint SS and ASR model: Based on the results the models are evaluated at different SNR

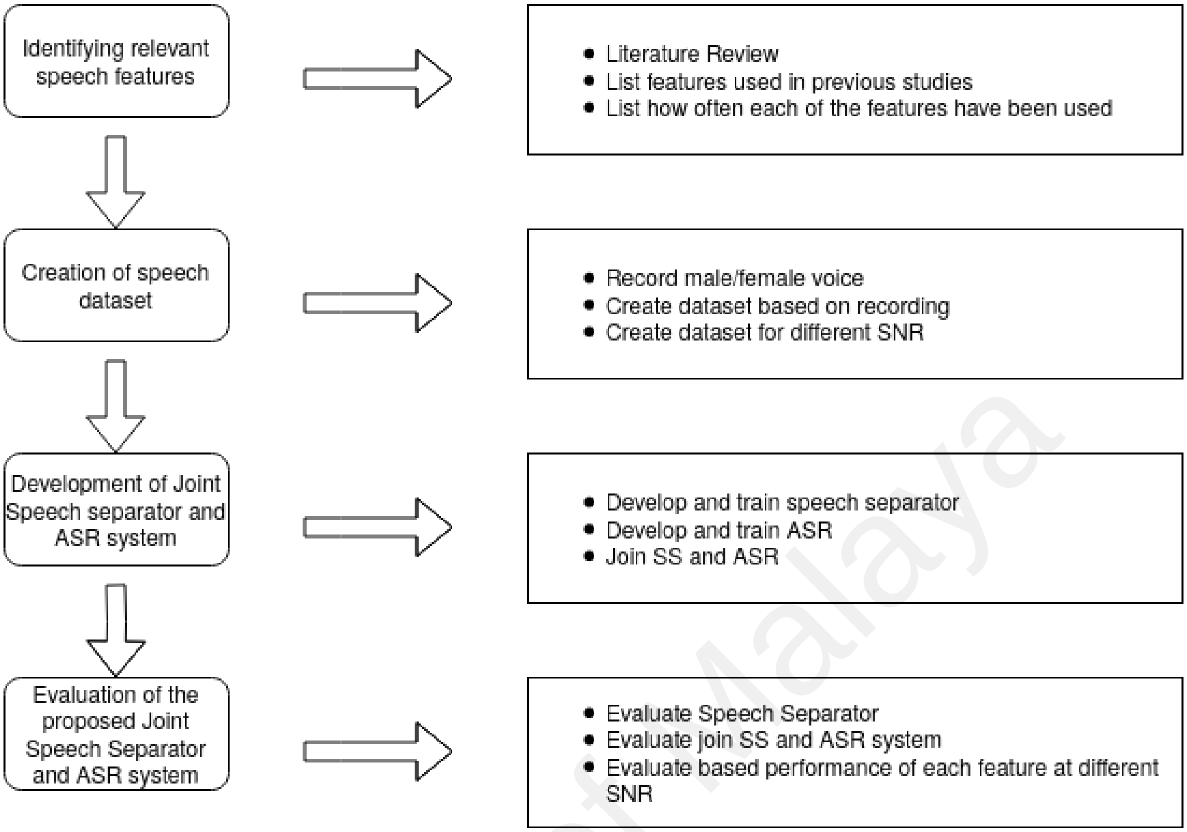

Identifying relevant speech features
Literature Review
List features used in previous studies
List how often each of the features have been used
Creation of speech dataset
Record male/female voice
Create dataset based on recording
Create dataset for different SNR
Development of Joint Speech separator and ASR system
Develop and train speech separator
Develop and train ASR
Join SS and ASR
Evaluation of the proposed Joint Speech Separator and ASR system
Evaluate Speech Separator
Evaluate join SS and ASR system
Evaluate based performance of each feature at different SNR

Figure 3.1: Research Flowchart

3.2 Identifying relevant speech features

The joint speech separation and ASR model is evaluated using different speech features to identify the most influencing features for the system. All speech features used in previous studies of speech separation were identified. Since Tu's (2016) joint speech separation and ASR model used DNN to train its model, only studies that use DNN speech separation were considered. Below is the list of speech features that were identified:

1. Short-time Fourier transform (STFT), Huang (2014)
2. Log magnitude (Log-Mag), Hershey (2016)
3. Log Power Mag (Log Power Mag), Du. (2014)
4. Log mel-filterbank (Log-Mel), Huang (2015)

5. Gama-tone filter bank (GF), Healy (2017)
6. Mel frequency cepstral coefficient (MFCC), Healy (2017)
7. Gamatone frequency cepstral coefficient (GFCC), Hu (2013)
8. Amplitude modulation spectrum (AMS), Healy (2017)
9. Power-Normalized Cepstral Coefficients (PNCC), Healy (2017)
10. Relative Spectral - Perceptual Linear Predictive (RASTA - PLP), Healy (2017)

3.3 Creation of speech dataset

The joint speech separation and ASR model were evaluated using different features but the same data set. In this section, the creation of the required dataset will be discussed. The dataset is required to have the following characteristics:

1. **A target-dependent and interfering dependent speaker**: Tu (2016) shows that a SI ASR model performs poorly compared to the target speaker-independent (TSD) system. The reason is; by limiting the conditions of the environment (in this case the target speaker) a better WER can be reached. From this, it can be inferred that if both the target and the interfering speaker are known (speaker-dependent), then the WER can be improved further. Therefore this research will focus on target-dependent and interfering dependent speaker joint speech separation and ASR model.
2. **SNR levels of -10,-5, 0 , 5 and 10**: As discussed in section 2.7 the average SNR of previous studies for speech separation and ASR can range from +5.75 to -8.00 and +14.5 to -5.5 which is shown in table 2.4. To get a more complete picture for both the ASR and speech separation component of the joint speech separation and ASR model a range from -10 to + 10 was selected.

3. **Small vocabulary**: This research compares the accuracy of a single model; joint speech separation and ASR model against different speech features. Therefore it is not important to have a dataset with large vocabulary but rather a smaller vocabulary dataset should be sufficient to compare the results.
4. **Male/Female voice:** To have a more complete dataset both male and female voices are needed. Both the candidates are non – native speakers within the age range of 20 – 30.
5. **Multiple repetitions of same words:** To train the DNN well on different stress and tone of the same word, every word must be recorded multiple times. This will allow the DNN to recognize the words even if it is spoken with a different tone or stress.

As a dataset that meets all the above criteria was not found a new dataset was created. Two speaker's voice was recorder; one female and one male of the age of 25 and 24 respectively. Ten words (digits) were recorded (0 to 9), which were repeated 50 times by each speaker. The recording session took place in a quiet room with a microphone for the speakers to speak. All the recordings were recorded at a 16 kHz sample rate.

3.4 Development of the Joint Speech separation and ASR model

The developed the joint model adopted from Tu (2016). Tu’s development architecture does not break down the steps for feature extraction and training the speech separator and ASR. This is because he used only one feature for each. However, in this study, the feature extraction had to be separated from the training stage since a different set of features are to be used for each of the experiments.

Initially the voices of two speakers were recorded and the desired datasets was prepared for two speakers at SNR levels -10,-5, 0,5, 10 (refer to section 3.3 for details). 20 models of joint SS and ASR were developed for 10 features and 2 speakers (refer to table 2.3, chapter 2).

Figure 3.1 depicts the framework used in this research adopted from Tu (2016) to develop the 20 joint SS and ASR models of 10 speech features from 2 speakers respectively. Evaluating the accuracy of each of the models will help in identifying the most influencing speech feature in improving the recognition accuracy. The models are tested at SNR levels -10, -5, 0, 5, 10. The lowest SNR used to test the models is -10, which means that the recognition accuracy of the models in very noisy environments can be evaluated.

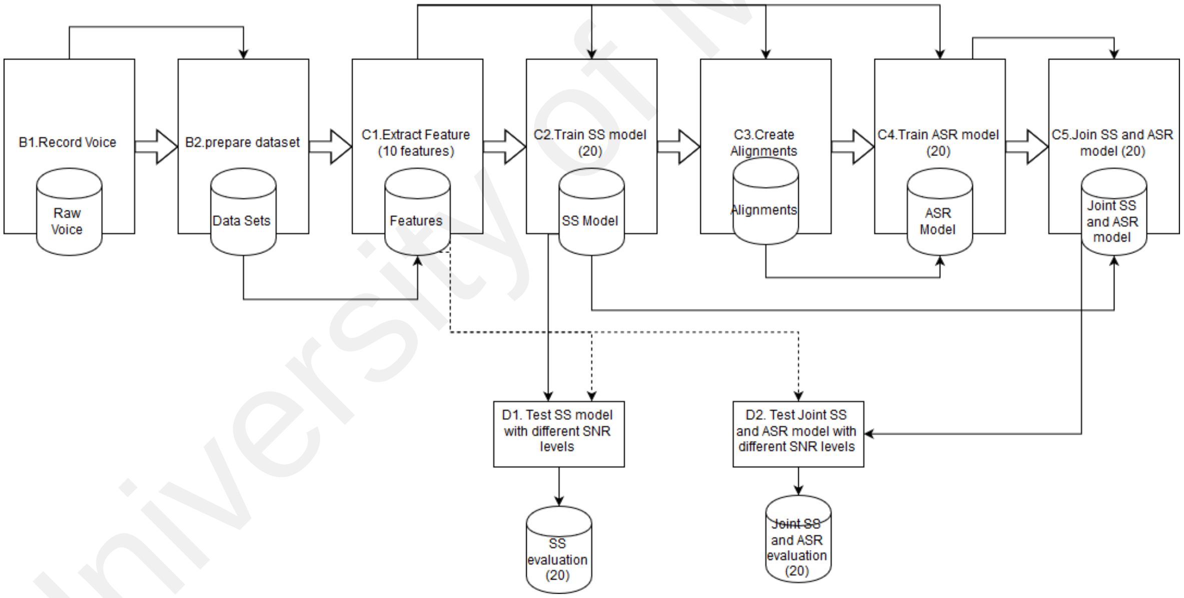

B1.Record Voice
Raw Voice
B2.prepare dataset
Data Sets
C1.Extract Feature (10 features)
Features
C2.Train SS model (20)
SS Model
C3.Create Alignments
Alignments
C4.Train ASR model (20)
ASR Model
C5.Join SS and ASR model (20)
Joint SS and ASR model
D1. Test SS model with different SNR levels
D2. Test Joint SS and ASR model with different SNR levels
SS evaluation (20)
Joint SS and ASR evaluation (20)

Figure 3.2: The framework used to develop joint SS and ASR model adopted from Tu 2014

3.4.1 Feature extraction

The audio of the dataset collected (refer to table 3.1) was split into smaller frames of approximately 10 ms. Next, the selected 10 features were used to extract the features from each of the frames. The configuration of each of the features such as dimensions, frame length, hop length, and a window was based on the existing works on speech separation. Section 4.1.3, chapter 4 explains the feature extraction stage in detail.

3.4.2 Train the Speech separation (SS) Model

This research is replicating the joint SS and ASR model (Tu, 2016). Tu (2014) used DNN to develop the speech separation. The SS model was trained using a standard 2 hidden layered DNN. The dimension of the input and output later varied based on the dimensions of the feature. The two hidden layers consist of 1024 neurons each with LeakyReLU activation functions and an alpha value of 0.1. Since this is a regression model the output layer uses a linear activation function. The loss function used to estimate the gradient is Minimum squared error (MSR) along with an 'Adam' optimizer. The context window was set to 1 for all of the models. Keres (a python library) was the key framework used to build this model. Section 4.1.4, chapter 4 explains the training stage in detail.

3.4.3 Creation of Alignment

The creation of alignment is the process of labeling the training set. A training dataset consist of a list of the speech signals with its original transcripts. In feature extraction, the speech signal is split into frames. In this stage, these frames have to be labeled. Each of the frames has to be labeled with a triphone. Triphones are a set of 3 phonemes. A phoneme is limited to only one sound however a triphone includes 3 sounds. The previous sound and the next sound is also concatenated to give greater clarity and this makes up a triphone. Therefore each of the frames is assigned a triphone to complete the training dataset. This process is known as Alignment.

3.4.4 Train ASR Model

The labeled alignments and the features are used to train the model. The model consists of three hidden layers with 200 neurons each. Stochastic gradient descent (SGD) was used as an optimizer with an initial learning rate of 0.02 and a final learning rate of 0.04 respectively. The context window was set to 1. The structure of the DNN including the number of layers and its neuron count was replicated from Tu's, 2016 Joint speech separation and ASR model. SGD, learning rate range, and context window rate are the default and most commonly used settings in Keres.

3.5 Evaluation

In this section, the evaluation method of speech separation and ASR within the joint speech separation and ASR will be discussed. The speech separator's task is to make sure that the speech features are cleaned. It makes sure that the speech features of the target speaker are extracted from the mixture of target and interfering features. Therefore the accuracy of a speech separation will depend on its ability to clean the speech features. The ASR's task is to transcribe what the target speaker is saying and ignore the interfering noises. Hence the accuracy of the ASR will depend on how many words it can accurately transcribe.

3.5.1 Evaluation of the Speech separation model

Short-term objective intelligibility (STOI) gives a value for how clean a sound is to the human ear (quality of the sound) (Delfarah and Wang 2017). However, for this research, that is irrelevant since the output of the speech separation will not be heard by a human but rather its output will be passed to an ASR model. Instead, the reduction of the minimum square error (MSE) is more important since that will bring the value of the noisy features closer to the clean features. Hence the ASR model should be able to decode them more efficiently. Therefore, the evaluation metrics of the speech separation will be the percentage reduction of the MSE of the predicted features. Given the clean feature, noisy feature, and predicted feature (output of the SS model) is represented as F-clean, F-noisy, F-predicted; the SS model will be evaluated using the following equation:

MSE (F-clean, F-noisy) – MSE (F-clean, F-predicted) / MSE(F-clean, F-noisy) * 100

3.5.2 Evaluation of the joint Speech separation and ASR model

The accuracy of the model is evaluated using Word Error Rate (WER). The WER is calculated using the following equation:

$$WER = \frac{S + D + I}{N}$$

formula 1

In which:

S is the number of substitutions (miss recognition of one word for another),

D is the number of deletions (words missed by the recognition system),

I is the number of insertions (words introduced into the text output by the recognition system),

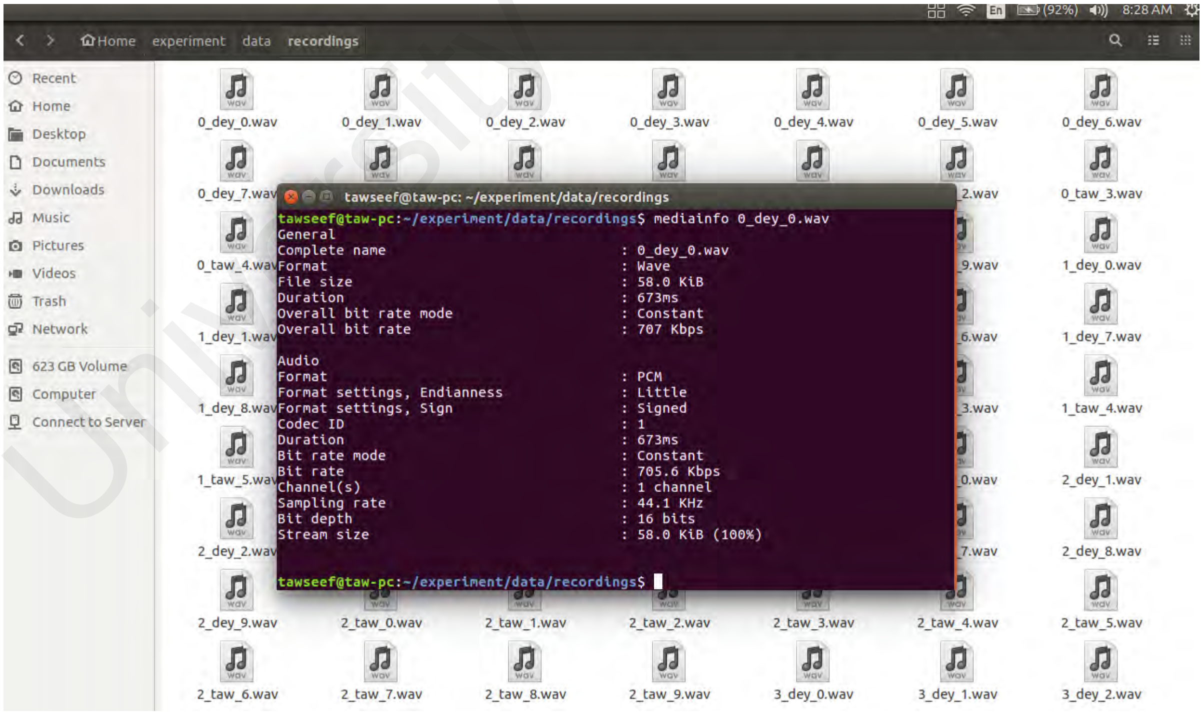
N is the number of words in the reference.

CHAPTER 4 : DEVELOPMENT OF THE JOINT SPEECH SEPARATION AND ASR MODEL

4.1 Creation of Speech Dataset

4.1.1 Speech recording

The voices were recorded in a quiet room to reduce the noise with a microphone. A transcript was prepared for the users containing random numbers from 0 to 9. The user was given 800 ms to speak each of the words written in the transcript. Once the recording was complete, each of the recorded voices was manually checked to make sure it matched with the transcript. The voices were stored in Wav format with a sampling rate of 44.1 kHz.



```
tawseef@taw-pc:~/experiment/data/recordings$ mediainfo 0_dey_0.wav
General
Complete name                            : 0_dey_0.wav
Format                                   : Wave
File size                                : 58.0 KiB
Duration                                 : 673ms
Overall bit rate mode                    : Constant
Overall bit rate                         : 707 Kbps

Audio
Format                                   : PCM
Format settings, Endianness              : Little
Format settings, Sign                    : Signed
Codec ID                                 : 1
Duration                                 : 673ms
Bit rate mode                            : Constant
Bit rate                                 : 705.6 Kbps
Channel(s)                               : 1 channel
Sampling rate                            : 44.1 KHz
Bit depth                                : 16 bits
Stream size                              : 58.0 KiB (100%)


tawseef@taw-pc:~/experiment/data/recordings$ 
```

Figure 4.1: The samples of the recorded voice

4.1.2 Prepare dataset

The recorded voice of one male and one female was used to prepare the data sets. The first 40 utterances from both speakers were used to synthesize the training data at SNRs -10, -5, 0, 5, and 10. The remaining ten were used for synthesizing testing data. The volume of data at different SNR levers was kept constant so as not to produce any biased results. The length of the train data was 1, 5, and 10 words per utterance. All the test data were sentences with ten words. Table 4.1 shows the data synthesized:

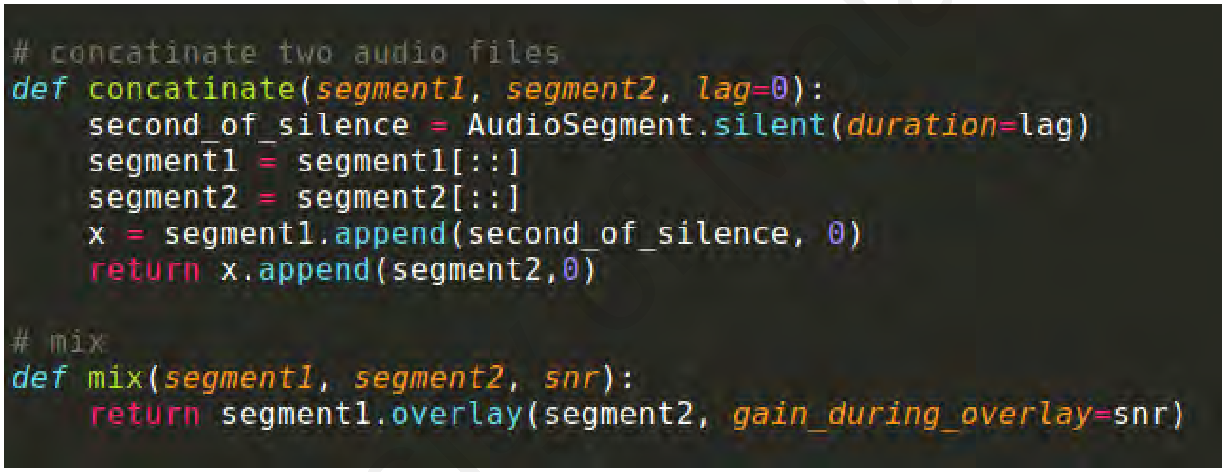
Table 4.1: Summary of Training/Test data

Name	Purpose	SNR	No. of words in each utterance
Speaker: 1 train	train	-10, -5, 0, 5, 10	1, 5, 10
Speaker: 2 train	train	-10, -5, 0, 5, 10	1, 5, 10
Speaker: 1 test (-10)	test	-10	10
Speaker: 1 test (-5)	test	-5	10
Speaker: 1 test (0)	test	0	10
Speaker: 1 test (+5)	test	+5	10
Speaker: 1 test (+10)	test	+10	10
Speaker: 2 test (-10)	test	-10	10
Speaker: 2 test (-5)	test	-5	10
Speaker: 2 test (0)	test	0	10
Speaker: 2 test (+5)	test	+5	10
Speaker: 2 test (+10)	test	+10	10

The dataset was prepared using a python script using a pydub library to overlay the voice of the users at different SNRs. The audio was randomly selected an overlay of the target speaker voices and interfering speaker. To build each of the conditions the recorded voices had to be concatenated or overlayed.

Concatenation was used to add a few voice samples together to create a longer (For example 10 word) audio. The words were chosen at random. The append function of the pydub library was used for this purpose.

The overlay was used to create different SNV level. The interfering speaker's voice was overplayed using the required SNR. The overlay function of the Audio Segments in the pydub library was used to create the different SNR levels as depicted in Figure 4.2 below.



```
# concatinate two audio files
def concatinate(segment1, segment2, lag=0):
    second_of_silence = AudioSegment.silent(duration=lag)
    segment1 = segment1[::]
    segment2 = segment2[::]
    x = segment1.append(second_of_silence, 0)
    return x.append(segment2,0)

# mix
def mix(segment1, segment2, snr):
    return segment1.overlay(segment2, gain_during_overlay=snr)
```

Figure 4.2: Code to concatenate and mix audio to prepare the dataset

4.1.3 Feature Extraction

The features were extracted by using the inbuilt functions of different libraries. The processed testing and training data from section 4.1.2 was used as inputs to convert the speech signals to speech features. Table 4.2 shows the Feature extraction implementation tools/libraries used in this research.

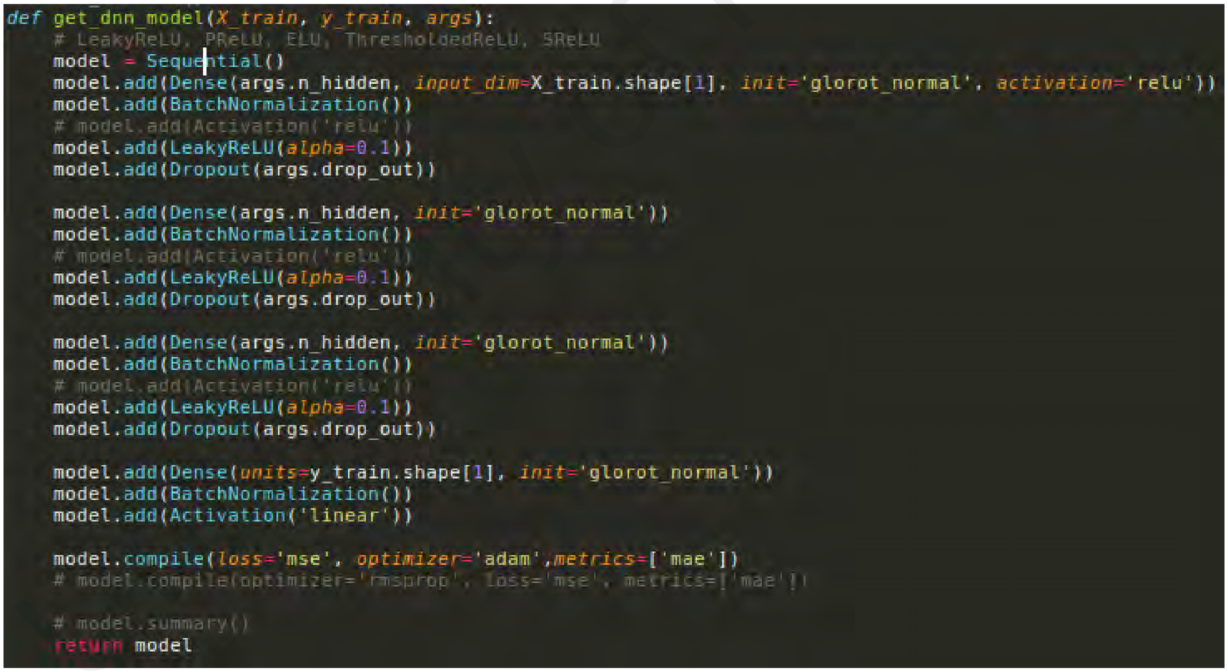
Table 4.2: Feature extraction implementation tools/libraries

Features	Platform	Libraries
STFT	python	numpy, scipy

Log Power	python	numpy, scipy
Log Magnitude	python	numpy, scipy
Log Mel	python	numpy, scipy
MFCC	python	numpy, scipy
GF	python	speech_utils, feature_extractor, scipy
GFCC	python	speech_utils, feature_extractor, scipy
PNCC	matlab	PNCC
RASTA-PLP	matlab	Rastaplp

4.1.4 Train speech separation model

A python script was used to build all the SS models. Sklearn was used to calculate the MSE while keras was used to build the DNN from scratch. Tenserflow was used to utilize the GPU to speed up the training process as depicted in Figure 4.3.

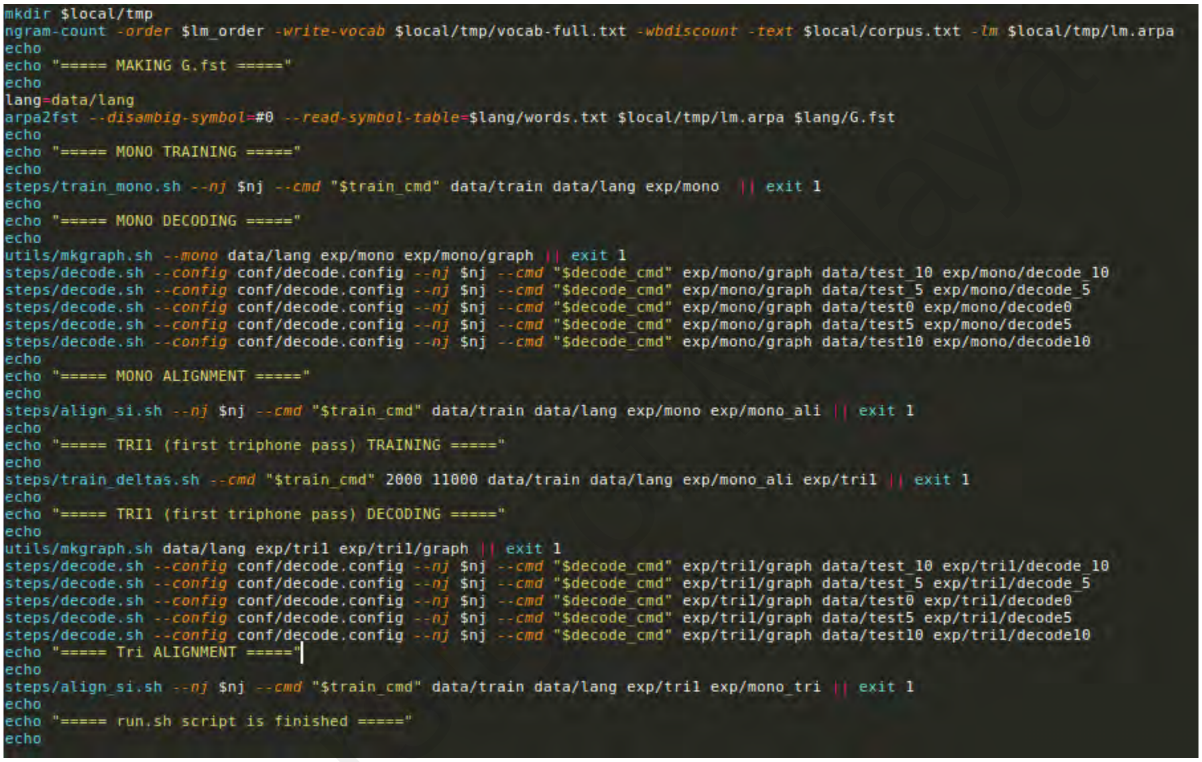


```
def get_dnn_model(X_train, y_train, args):
    # LeakyReLU, PReLU, ELU, ThresholdedReLU, SReLU
    model = Sequential()
    model.add(Dense(args.n_hidden, input_dim=X_train.shape[1], init='glorot_normal', activation='relu'))
    model.add(BatchNormalization())
    # model.add(Activation('relu'))
    model.add(LeakyReLU(alpha=0.1))
    model.add(Dropout(args.drop_out))

    model.add(Dense(args.n_hidden, init='glorot_normal'))
    model.add(BatchNormalization())
    # model.add(Activation('relu'))
    model.add(LeakyReLU(alpha=0.1))
    model.add(Dropout(args.drop_out))

    model.add(Dense(args.n_hidden, init='glorot_normal'))
    model.add(BatchNormalization())
    # model.add(Activation('relu'))
    model.add(LeakyReLU(alpha=0.1))
    model.add(Dropout(args.drop_out))

    model.add(Dense(units=y_train.shape[1], init='glorot_normal'))
    model.add(BatchNormalization())
    model.add(Activation('linear'))

    model.compile(loss='mse', optimizer='adam',metrics=['mae'])
    # model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])

    # model.summary()
    return model
```

Figure 4.3: Code to create the DNN model

4.1.5 Create Alignments

Python was used along with the kaldi engine to create the alignment using forced alignment method. Kaldio_io library was used to access the kaldi command line interface through python to create the alignments as shown in Figure 4.4 below.

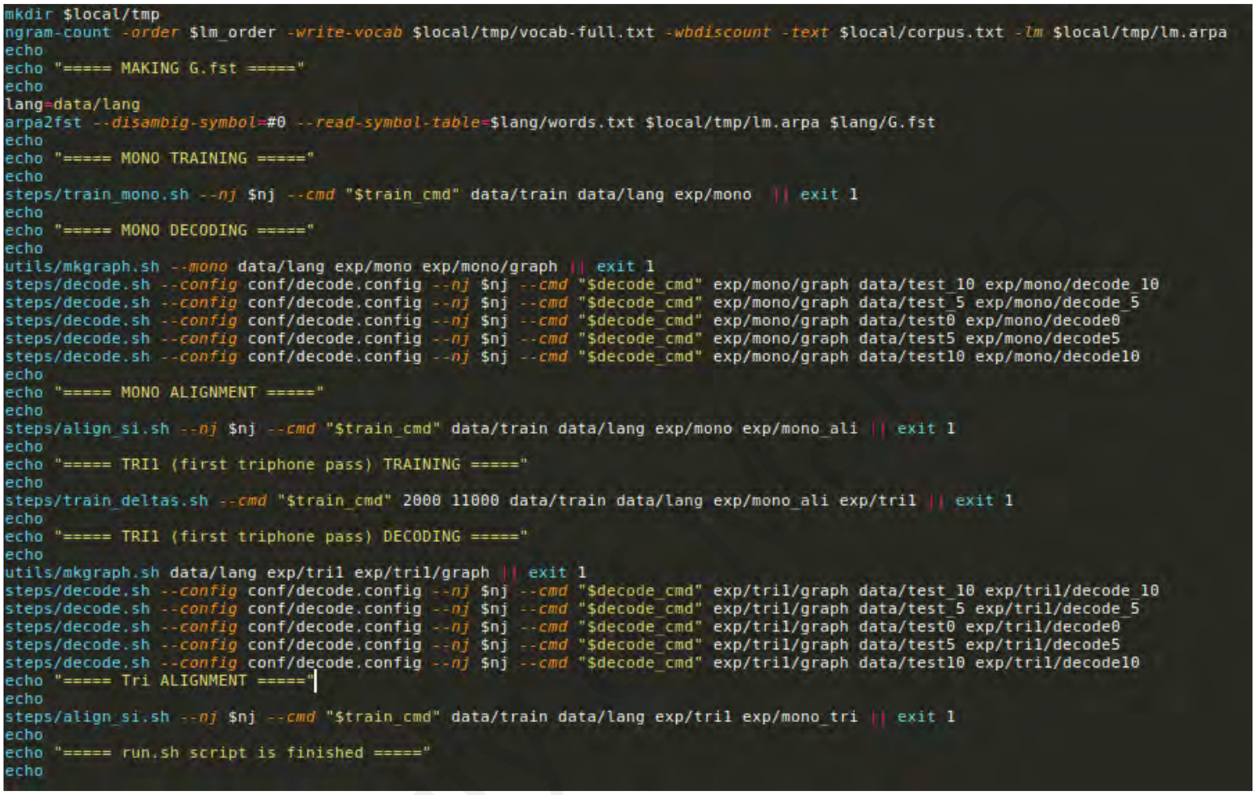


```
mkdir $local/tmp
ngram-count -order $lm_order -write-vocab $local/tmp/vocab-full.txt -wbdiscount -text $local/corpus.txt -lm $local/tmp/lm.arpa
echo
echo "===== MAKING G.fst ====="
echo
lang=data/lang
arpa2fst --disambig-symbol=#0 --read-symbol-table=$lang/words.txt $local/tmp/lm.arpa $lang/G.fst
echo
echo "===== MONO TRAINING ====="
echo
steps/train_mono.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/mono  || exit 1
echo
echo "===== MONO DECODING ====="
echo
utils/mkgraph.sh --mono data/lang exp/mono exp/mono/graph || exit 1
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test_10 exp/mono/decode_10
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test_5 exp/mono/decode_5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test0 exp/mono/decode0
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test5 exp/mono/decode5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test10 exp/mono/decode10
echo
echo "===== MONO ALIGNMENT ====="
echo
steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/mono exp/mono_ali || exit 1
echo
echo "===== TRI1 (first triphone pass) TRAINING ====="
echo
steps/train_deltas.sh --cmd "$train_cmd" 2000 11000 data/train data/lang exp/mono_ali exp/tri1 || exit 1
echo
echo "===== TRI1 (first triphone pass) DECODING ====="
echo
utils/mkgraph.sh data/lang exp/tri1 exp/tri1/graph || exit 1
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test_10 exp/tri1/decode_10
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test_5 exp/tri1/decode_5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test0 exp/tri1/decode0
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test5 exp/tri1/decode5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test10 exp/tri1/decode10
echo "===== Tri ALIGNMENT ====="
echo
steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/tri1 exp/mono_tri || exit 1
echo
echo "===== run.sh script is finished ====="
echo
```

Figure 4.4: Code to create alignments

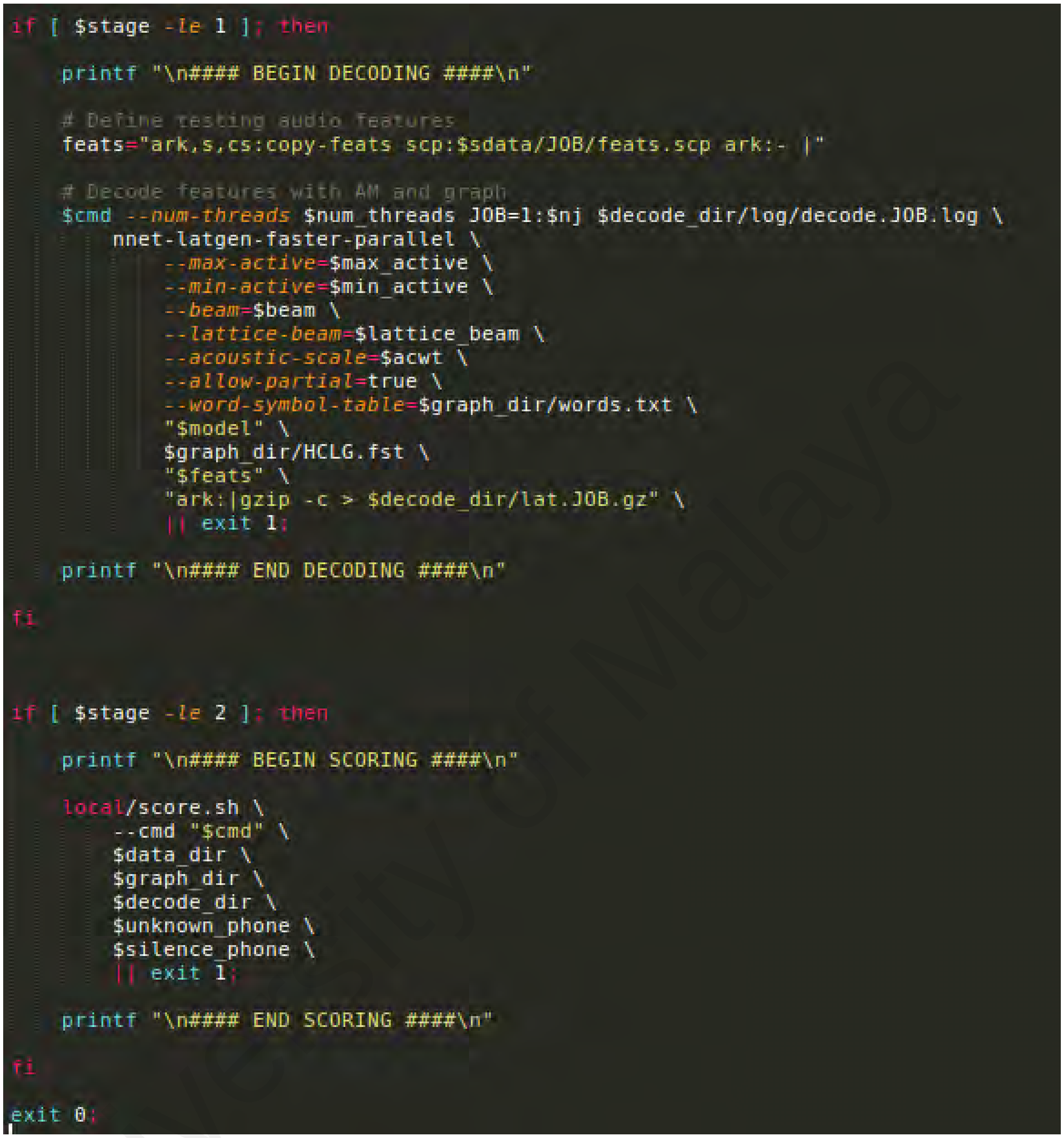
4.1.6 Train ASR model

All the ASR models were developed using nnet2 of kaldi. Kaldi was configures with tenserflow to increase the processing speed. Source code modification was needed to the kaldi nnet2 decoder to be used in python as depicted in Figure 4.5 below.



```
mkdir $local/tmp
ngram-count -order $lm_order -write-vocab $local/tmp/vocab-full.txt -wbdiscount -text $local/corpus.txt -lm $local/tmp/lm.arpa
echo
echo "===== MAKING G.fst ====="
echo
lang=data/lang
arpa2fst --disambig-symbol=#0 --read-symbol-table=$lang/words.txt $local/tmp/lm.arpa $lang/G.fst
echo
echo "===== MONO TRAINING ====="
echo
steps/train_mono.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/mono  || exit 1
echo
echo "===== MONO DECODING ====="
echo
utils/mkgraph.sh --mono data/lang exp/mono exp/mono/graph || exit 1
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test_10 exp/mono/decode_10
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test_5 exp/mono/decode_5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test0 exp/mono/decode0
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test5 exp/mono/decode5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/mono/graph data/test10 exp/mono/decode10
echo
echo "===== MONO ALIGNMENT ====="
echo
steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/mono exp/mono_ali || exit 1
echo
echo "===== TRI1 (first triphone pass) TRAINING ====="
echo
steps/train_deltas.sh --cmd "$train_cmd" 2000 11000 data/train data/lang exp/mono_ali exp/tri1 || exit 1
echo
echo "===== TRI1 (first triphone pass) DECODING ====="
echo
utils/mkgraph.sh data/lang exp/tri1 exp/tri1/graph || exit 1
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test_10 exp/tri1/decode_10
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test_5 exp/tri1/decode_5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test0 exp/tri1/decode0
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test5 exp/tri1/decode5
steps/decode.sh --config conf/decode.config --nj $nj --cmd "$decode_cmd" exp/tri1/graph data/test10 exp/tri1/decode10
echo "===== Tri ALIGNMENT ====="
echo
steps/align_si.sh --nj $nj --cmd "$train_cmd" data/train data/lang exp/tri1 exp/mono_tri || exit 1
echo
echo "===== run.sh script is finished ====="
echo
```

Figure 4.5: Code to prepare the ASR model

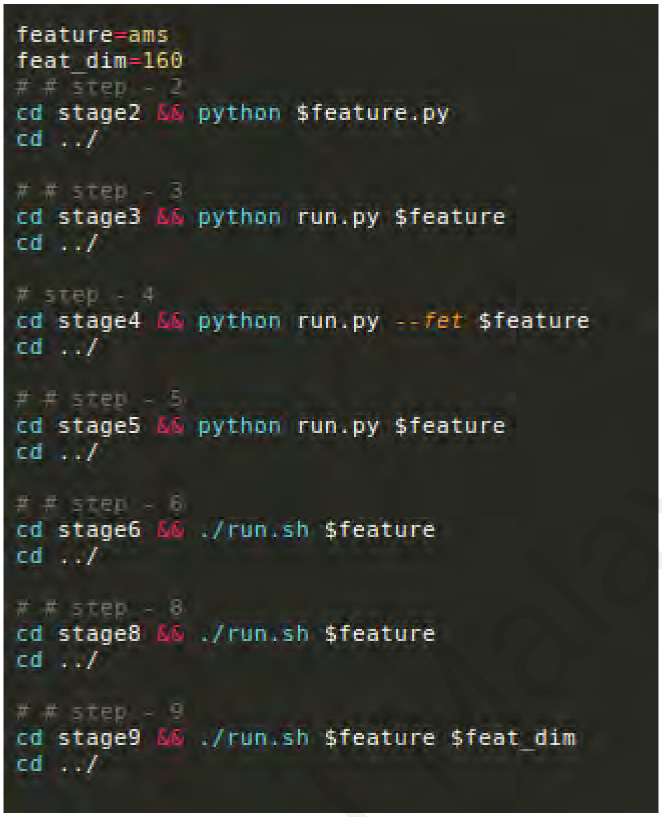


```
if [ $stage -le 1 ]; then

    printf "\n#### BEGIN DECODING ####\n"

    # Define testing audio features
    feats="ark,s,cs:copy-feats scp:$sdata/JOB/feats.scp ark:- |"

    # Decode features with AM and graph
    $cmd --num-threads $num_threads JOB=1:$nj $decode_dir/log/decode.JOB.log \
        nnet-latgen-faster-parallel \
            --max-active=$max_active \
            --min-active=$min_active \
            --beam=$beam \
            --lattice-beam=$lattice_beam \
            --acoustic-scale=$acwt \
            --allow-partial=true \
            --word-symbol-table=$graph_dir/words.txt \
            "$model" \
            $graph_dir/HCLG.fst \
            "$feats" \
            "ark:|gzip -c > $decode_dir/lat.JOB.gz" \
            || exit 1;

    printf "\n#### END DECODING ####\n"

fi


if [ $stage -le 2 ]; then

    printf "\n#### BEGIN SCORING ####\n"

    local/score.sh \
        --cmd "$cmd" \
        $data_dir \
        $graph_dir \
        $decode_dir \
        $unknown_phone \
        $silence_phone \
        || exit 1;

    printf "\n#### END SCORING ####\n"

fi

exit 0;
```

Figure 4.6: Code to train the ASR Model

4.1.7 Joint speech separation and ASR model

Python along with unix bash script was used to join the two SS and ASR models to create a unified Joint SS and ASR model as depicted in Figure 4.7 below.



```
feature=ams
feat_dim=160
# # step - 2
cd stage2 && python $feature.py
cd ../

# # step - 3
cd stage3 && python run.py $feature
cd ../

# step - 4
cd stage4 && python run.py --fet $feature
cd ../

# # step - 5
cd stage5 && python run.py $feature
cd ../

# # step - 6
cd stage6 && ./run.sh $feature
cd ../

# # step - 8
cd stage8 && ./run.sh $feature
cd ../

# # step - 9
cd stage9 && ./run.sh $feature $feat_dim
cd ../
```

Figure 4.7: Code to join the speech separation and ASR model

As shown in figure 4.7 the implementation of the joint speech separation and ASR model is divided into 9 steps. The output of each of the step acts as an input for its subsequent steps. These output/input are stored locally in a folder and is referenced as directory paths within the code. This section will discuss the tasks of each of these steps:

Step-1: In this step, the data is prepared. Both the training dataset and the testing dataset is developed for both speaker 1 and speaker 2 in this stage using the pre-recorded voices of both the users. This step is run only once which is why it is not part of the code. The same datasets are going to be used for all the 20 experiments.

Step-2: This stage is the feature extraction stage. The features are extracted from all the clean, mixed, and interfering speeches. The dataset prepared in step-1 is used to extract the features.

Step-3: This step is used to concatenate the features for all the datasets to prepare the input of the Speech separation model to train it. This step is repeated for each of the features i.e. 10 features.

Step-4: In this step, the speech separation model is trained. As explained earlier, the speech separation uses DNN as an acoustic model. Therefore the feature dataset prepared in step 3 is used to train and test the speech separation.

Stage-5: At this stage, the output from the speech separation in step-4 is merged, prepared, and restructured to be used as an input for the Kaldi ASR model.

Step-6: At this stage, the empty template of the ASR model is prepared. Therefore 20 empty templates are prepared which are ready to be trained. This template is an empty Kaldi template that is yet to be trained.

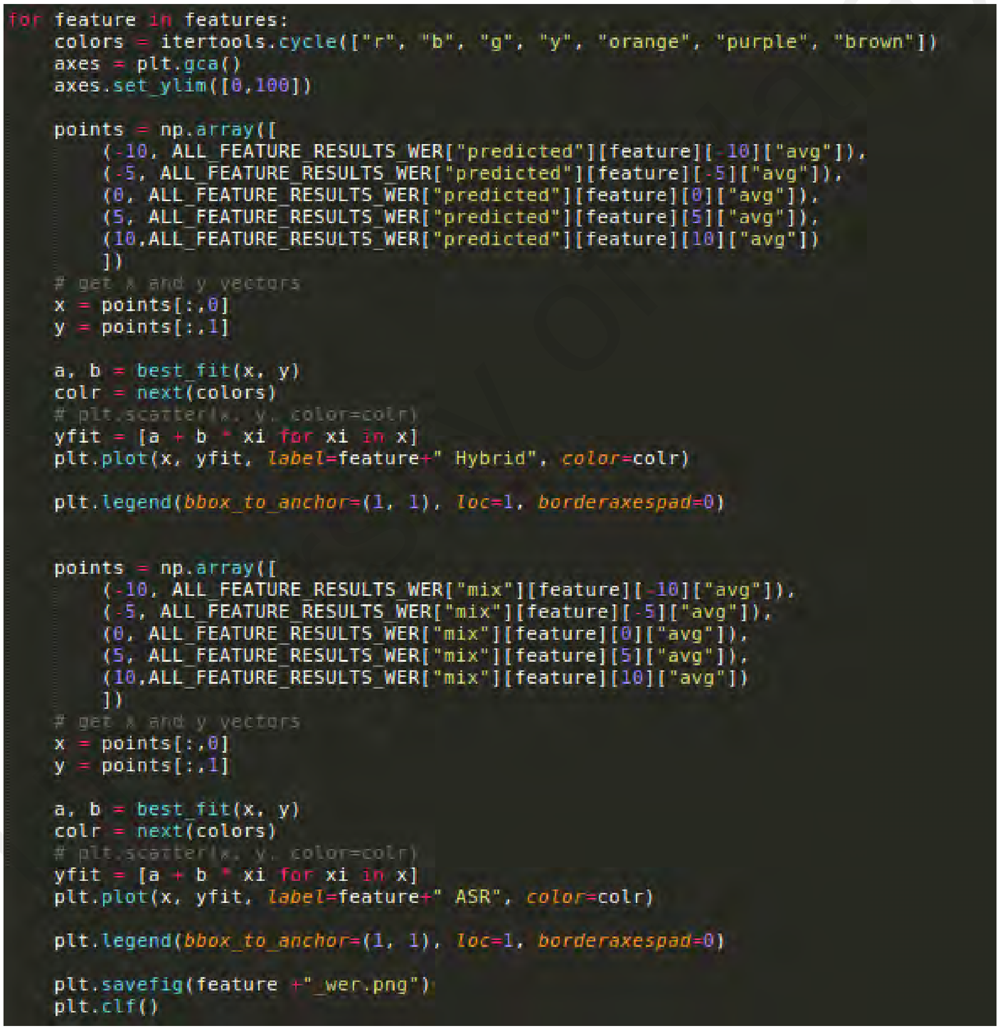
Step-7: At this stage, the training data from step 5 is aligned. This is the alignment stage of the ASR as explained earlier. At this stage, the features are to be labeled with the appropriate triphones. These labeled training set will be later used to train the ASR model.

Step-8: At this stage, the Kaldi ASR model is trained using the empty models prepared in step-6 using the data in step -7. This step takes the longest time to run since the ASR training is a slow process.

Step-9: At this stage, all the 20 speech separation and 20 Joint speech separation and ASR model is evaluated. The testing data prepared in step -5 for the ASR and step-3 for speech separation is used to evaluate the models.

4.1.8 Evaluation of the developed model

A Python script was used to evaluate the models. Sklearn and spacy were used to calculate the MSE while kaldi_io was used to calculate the WER of the two models. Matplotlib was used to plot the graphs in an interface and export them as image files, as described by the code shown in Figure 4.8 below.



```python
for feature in features:
    colors = itertools.cycle(["r", "b", "g", "y", "orange", "purple", "brown"])
    axes = plt.gca()
    axes.set_ylim([0,100])

    points = np.array([
        (-10, ALL_FEATURE_RESULTS_WER["predicted"][feature][-10]["avg"]),
        (-5, ALL_FEATURE_RESULTS_WER["predicted"][feature][-5]["avg"]),
        (0, ALL_FEATURE_RESULTS_WER["predicted"][feature][0]["avg"]),
        (5, ALL_FEATURE_RESULTS_WER["predicted"][feature][5]["avg"]),
        (10,ALL_FEATURE_RESULTS_WER["predicted"][feature][10]["avg"])
        ])
    # get x and y vectors
    x = points[:,0]
    y = points[:,1]

    a, b = best_fit(x, y)
    colr = next(colors)
    # plt.scatter(x, y, color=colr)
    yfit = [a + b * xi for xi in x]
    plt.plot(x, yfit, label=feature+" Hybrid", color=colr)

    plt.legend(bbox_to_anchor=(1, 1), loc=1, borderaxespad=0)


    points = np.array([
        (-10, ALL_FEATURE_RESULTS_WER["mix"][feature][-10]["avg"]),
        (-5, ALL_FEATURE_RESULTS_WER["mix"][feature][-5]["avg"]),
        (0, ALL_FEATURE_RESULTS_WER["mix"][feature][0]["avg"]),
        (5, ALL_FEATURE_RESULTS_WER["mix"][feature][5]["avg"]),
        (10,ALL_FEATURE_RESULTS_WER["mix"][feature][10]["avg"])
        ])
    # get x and y vectors
    x = points[:,0]
    y = points[:,1]

    a, b = best_fit(x, y)
    colr = next(colors)
    # plt.scatter(x, y, color=colr)
    yfit = [a + b * xi for xi in x]
    plt.plot(x, yfit, label=feature+" ASR", color=colr)

    plt.legend(bbox_to_anchor=(1, 1), loc=1, borderaxespad=0)

    plt.savefig(feature +"_wer.png")
    plt.clf()
```

Figure 4.8: Code for evaluating the models

4.2 Experimental Design

Ten sets of features were extracted and a joint SS and ASR model was implemented for each. All configurations for training/testing data and structure of the joint SS and ASR model was kept constant. The only varying points were the features (used for training and testing) and the dimension of the input and output layers of the DNN, which has to correspond to the dimensions of the features. Once the training was completed the testing was done at different SNR using the appropriate evaluation metrics.

4.2.1 Feature Extraction

Table 4.3 shows the selected features and dimensions of the features,

Table 4.3: Feature extraction configurations

Features	Dimensions (size)	Frame Length (ms)	Hop length (ms)	Window Function	Configurations
STFT	257	32	8	hamming	N/A
Log Power	257	32	8	hamming	Power = 2
Log Magnitude	257	32	8	hamming	N/A
Log Mel	26	25	1	None	n-filter = 26
MFCC	13	25	1	None	n-filter = 26
GF	64	25	1	hamming	N/A
GFCC	31	25	1	hamming	N/A
PNCC	26	25	1	hamming	N/A
RASTA-PLP	21	25	1	hamming	N/A
AMS	160	25	1	None	N/A

4.2.2 Alignments

GMM-HMM alignment was performed using Kaldi forced alignment. 13 coefficient MFCC features were used to create the triphone alignments with the following configurations:

(i) first_beam=10.0

(ii) beam=13.0

(iii) lattice_beam=6.0

4.2.3 Training models

It can be seen that a different configuration can yield varying accuracy for each feature. However, to conduct an unbiased comparison, trade-offs had to be made via trial and error and a single basic configuration had to be chosen.

The data for each speaker consists of five different SNR levels which are used to extract the respective features. These features are used to train the SS Models to predict clean features. The predicted clean features were then used to train the Acoustic model to categorize the frames into tri-phones. The concatenation of the two models is collectively referred to as a joint SS and ASR model as shown in Table 4.4.

Table 4.4: Speech separation and joint SS and ASR models' summary

SNR Train Data	Feature Extraction	SS Model	ASR model	joint SS and ASR model
Speaker1-train	STFT	SS Speaker1 STFT	AM Speaker1 STFT	joint Speaker1 STFT
Speaker1-train	Log Power	SS Speaker1 logpow	AM Speaker1 logpow	joint Speaker1 logpow
Speaker1-train	Log Magnitude	SS Speaker1 logmag	AM Speaker1 logmag	joint Speaker1 logmag
Speaker1-train	Log Mel	SS Speaker1 logmel	AM Speaker1 logmel	joint Speaker1 logmel
Speaker1-train	MFCC	SS Speaker1 mfcc	AM Speaker1 mfcc	joint Speaker1 mfcc
Speaker1-train	GF	SS Speaker1 gf	AM Speaker1 gf	joint Speaker1 gf
Speaker1-train	GFCC	SS Speaker1 gfcc	AM Speaker1 gfcc	joint Speaker1 gfcc
Speaker1-train	PNCC	SS Speaker1 pncc	AM Speaker1 pncc	joint Speaker1 pncc
Speaker1-train	RASTA-PLP	SS Speaker1 rastaplp	AM Speaker1 rastaplp	joint Speaker1 rastaplp
Speaker1-train	AMS	SS Speaker1 ams	AM Speaker1 ams	joint Speaker1 ams
Speaker2-train	STFT	SS Speaker2 STFT	AM Speaker2 STFT	joint Speaker2 STFT
Speaker2-train	Log Power	SS Speaker2 logpow	AM Speaker2 logpow	joint Speaker2 logpow
Speaker2-train	Log Magnitude	SS Speaker2 logmag	AM Speaker2 logmag	joint Speaker2 logmag

Speaker2-train	Log Mel	SS Speaker2 logmel	AM Speaker2 logmel	joint Speaker2 logmel
Speaker2-train	MFCC	SS Speaker2 mfcc	AM Speaker2 mfcc	joint Speaker2 mfcc
Speaker2-train	GF	SS Speaker2 gf	AM Speaker2 gf	joint Speaker2 gf
Speaker2-train	GFCC	SS Speaker2 gfcc	AM Speaker2 gfcc	joint Speaker2 gfcc
Speaker2-train	PNCC	SS Speaker2 pncc	AM Speaker1 pncc	joint Speaker1 pncc
Speaker2-train	RASTA-PLP	SS Speaker2 rastaplp	AM Speaker1 rastaplp	joint Speaker1 rastaplp
Speaker2-train	AMS	SS Speaker2 ams	AM Speaker1 ams	joint Speaker1 ams

4.2.4 Experiments

After the training of the 20 models (2 speaker x 10 features), each of the models will be evaluated using five different levels of SNR data. The final evaluation will be carried out by averaging the evaluation matrices of the two speakers for each of the features. Table 4.5 display the total number of experiments conducted for each of the features:

Table 4.5: Summary of experiments conducted

SNR Test Data	SS Model (for each of the 10 features)	Joint Model (SS+ASR) (for each of the 10 features)
Speaker: 1 test (-10)	SS Speaker1 feature(x)	joint Speaker1 feature(x)
Speaker: 1 test (-5)	SS Speaker1 feature(x)	joint Speaker1 feature(x)
Speaker: 1 test (0)	SS Speaker1 feature(x)	joint Speaker1 feature(x)
Speaker: 1 test (+5)	SS Speaker1 feature(x)	joint Speaker1 feature(x)
Speaker: 1 test (+10)	SS Speaker1 feature(x)	joint Speaker1 feature(x)
Speaker: 2 test (-10)	SS Speaker2 feature(x)	joint Speaker2 feature(x)
Speaker: 2 test (-5)	SS Speaker2 feature(x)	joint Speaker2 feature(x)
Speaker: 2 test (0)	SS Speaker2 feature(x)	joint Speaker2 feature(x)
Speaker: 2 test (+5)	SS Speaker2 feature(x)	joint Speaker2 feature(x)
Speaker: 2 test (+10)	SS Speaker2 feature(x)	joint Speaker2 feature(x)

As explained in section 4.1.7 stage -9 is the evaluation stage. The experiments begin once all of the 20 models are trained and developed. Table 4.4 displays the template of all the experiments that were conducted. The test data originated from the dataset prepares earlier from two speakers at SNR levels -10, -5, 0, 5, and 10. The joint SS and ASR model contains a component speech separation. The joint speech separation and ASR model and the speech separation are evaluated separately. In table 4.4 'x' denotes one of the 10 features. Since 2 speakers at different SNR level is used, a total of 10 datasets are to be evaluated. To evaluate the speech separation, all the 10 datasets are evaluated for each of the 10 features which add up to 100 evaluations for the speech separation. Subsequently, the joint speech separation and ASR model is evaluated using the same 10 dataset for each of the 10 features, which adds up to another 10 evaluations for the joint speech separation and ASR model.

CHAPTER 5 : RESULTS

5.1 Results of the Speech separation Model

Table 5.1 shows the MSE (Minimum Square Error) of the Speech Separation (SS) model at different SNR levels. The Male columns are the performance of the SS when extracting the male features and vice versa for the female column. Base refers to the MSE of the noisy features, while ‘Predicted’ refers to the MSE of the predicted which is cleaner than the noisy features. It is expected that the ‘predicted’ MSE to be better than the base MSE.

Table 5.1: Average improved MSE of the speech separation

features	**SNR**	**male_base**	**male_predicted**	**female_base**	**female_predicted**	**avg_improved**
stft	**-10**	28179885.07	26941331.11	22195176.95	16523781.69	14.97
stft	**-5**	25077504.77	27916704.28	15732721.17	13909193.26	0.13
stft	**0**	23274042.52	28461178.90	11466626.49	11426332.04	-10.96
stft	**5**	32990422.56	36148860.71	28350803.22	11736449.95	24.51
stft	**10**	92778012.77	80448007.29	137706033.94	37198876.54	43.13
log_power_mag	**-10**	9.36	5.20	8.34	5.43	39.61
log_power_mag	**-5**	8.35	5.08	7.25	4.72	36.98
log_power_mag	**0**	7.89	5.12	6.74	4.25	36.01
log_power_mag	**5**	8.61	5.24	7.49	3.98	42.99
log_power_mag	**10**	11.28	5.45	10.25	3.90	56.80
log_mag	**-10**	2.34	1.23	2.08	1.38	40.42
log_mag	**-5**	2.08	1.13	1.81	1.21	39.14
log_mag	**0**	1.97	1.09	1.68	1.09	39.73
log_mag	**5**	2.15	1.09	1.87	1.02	47.18
log_mag	**10**	2.82	1.13	2.56	1.00	60.21

gf	**-10**	0.91	0.53	0.70	0.39	42.77
gf	**-5**	0.81	0.52	0.57	0.30	41.11
gf	**0**	0.77	0.50	0.52	0.25	43.58
gf	**5**	0.91	0.50	0.69	0.22	56.47
gf	**10**	1.43	0.53	1.28	0.21	72.74
gfcc	**-10**	83.01	48.37	65.34	42.92	38.01
gfcc	**-5**	75.14	48.64	54.49	36.92	33.75
gfcc	**0**	76.20	46.41	54.06	31.52	40.39
gfcc	**5**	100.07	43.99	79.13	28.31	60.13
gfcc	**10**	170.35	46.67	156.10	28.89	77.04
log_mel	**-10**	7.36	3.86	5.88	3.34	45.38
log_mel	**-5**	6.61	3.73	5.03	2.69	45.04
log_mel	**0**	6.49	3.82	4.88	2.23	47.62
log_mel	**5**	7.61	4.05	6.05	2.01	56.67
log_mel	**10**	10.62	4.35	9.20	2.00	68.60
mfcc	**-10**	186.49	97.36	138.69	90.43	41.29
mfcc	**-5**	161.09	90.20	111.47	73.77	38.91
mfcc	**0**	132.30	84.53	86.59	59.22	33.85
mfcc	**5**	101.77	78.33	64.08	48.22	23.89
mfcc	**10**	72.67	72.60	43.96	40.03	4.51
PNCC	**-10**	0.69	0.42	0.52	0.34	36.43
PNCC	**-5**	0.60	0.38	0.41	0.28	33.98
PNCC	**0**	0.49	0.33	0.31	0.23	29.05
PNCC	**5**	0.38	0.28	0.22	0.19	20.91
PNCC	**10**	0.28	0.24	0.15	0.15	7.46
RASTA-PLP	**-10**	0.67	0.52	0.86	0.62	25.12
RASTA-PLP	**-5**	0.48	0.49	0.67	0.52	9.98
RASTA-PLP	**0**	0.30	0.44	0.47	0.42	-17.60
RASTA-PLP	**5**	0.17	0.41	0.29	0.35	-79.17
RASTA-PLP	**10**	0.09	0.42	0.16	0.30	-228.23
AMS	**-10**	0.00	0.00	0.00	0.00	30.96
AMS	**-5**	0.00	0.00	0.00	0.00	16.91
AMS	**0**	0.00	0.00	0.00	0.00	6.87
AMS	**5**	0.00	0.00	0.00	0.00	60.55
AMS	**10**	0.00	0.00	0.00	0.00	90.90

Figure 5.1 depicts the percentage of noise cleaned by the SS model for each feature. This is the aggregated visualization of table 5.1. The speech separation was able to clean the speech signal for all the features except for ransta-plp. As shown in figure 5.1, the speech separation disrupts the input features of rasta-plp.

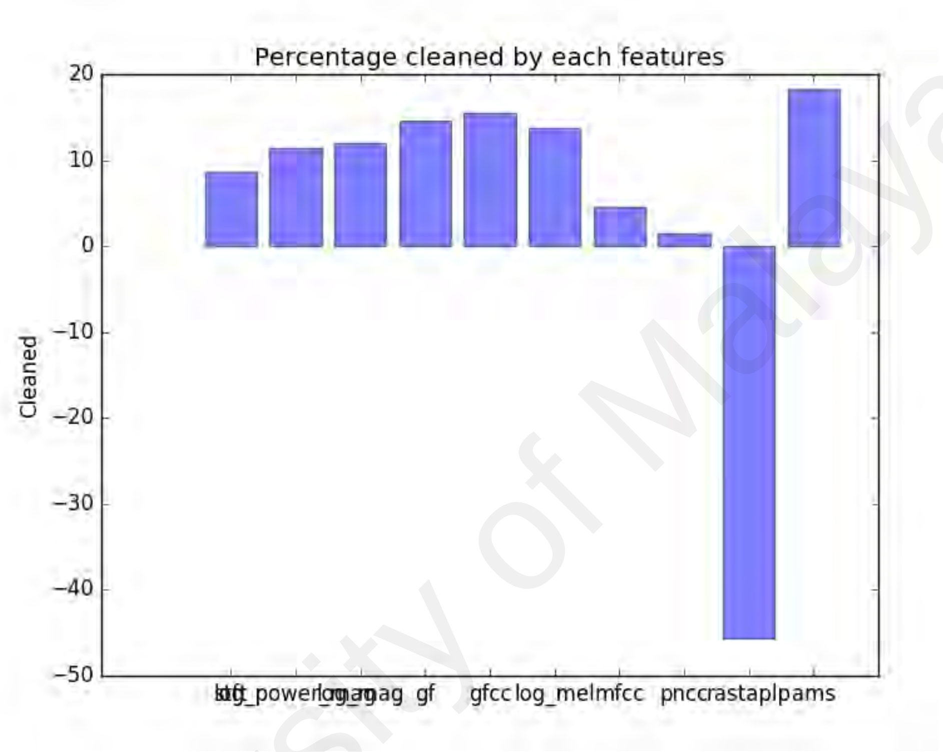


Figure 5.1: Percentage of noise cleaned in each feature by the speech separation

5.2 Results of the Joint Speech Separation and ASR model

Table 5.2 shows the Word Error Rate (WER) of the joint SS and ASR model. The table displays the average WER for male and female voice experiments at different SNR levels.

Table 5.2: Word Error Rate (%) of each of the Joint Speech separation and ASR model in different SNR

features	**SNR**	**joint _male_WER (%)**	**joint _female_ WER (%)**	**joint _avg_ WER (%)**
stft	-10	22	18	20
stft	-5	14	8	11
stft	0	14	0	7
stft	5	4	0	2
stft	10	6	0	3
log_power_mag	-10	46	28	37
log_power_mag	-5	30	20	25
log_power_mag	0	18	12	15
log_power_mag	5	14	8	11
log_power_mag	10	12	8	10
log_mag	-10	50	30	40
log_mag	-5	32	18	25
log_mag	0	20	10	15
log_mag	5	18	12	15
log_mag	10	10	8	9
gf	-10	10	16	13
gf	-5	10	6	8
gf	0	10	6	8
gf	5	10	2	6
gf	10	0	0	0
gfcc	-10	4	22	13
gfcc	-5	8	12	10
gfcc	0	8	6	7
gfcc	5	2	4	3
gfcc	10	2	2	2
log_mel	-10	18	16	17
log_mel	-5	22	10	16
log_mel	0	10	6	8
log_mel	5	8	0	4
log_mel	10	8	0	4
mfcc	-10	30	14	22
mfcc	-5	20	12	16
mfcc	0	12	6	9
mfcc	5	8	4	6
mfcc	10	2	0	1
PNCC	-10	24	22	23
PNCC	-5	18	6	12
PNCC	0	6	4	5
PNCC	5	4	2	3
PNCC	10	4	0	2

RASTA-PLP	-10	62	44	53
RASTA-PLP	-5	44	18	31
RASTA-PLP	0	32	6	19
RASTA-PLP	5	22	0	11
RASTA-PLP	10	12	0	6
AMS	-10	90	90	90
AMS	-5	90	90	90
AMS	0	90	90	90
AMS	5	90	90	90
AMS	10	90	90	90

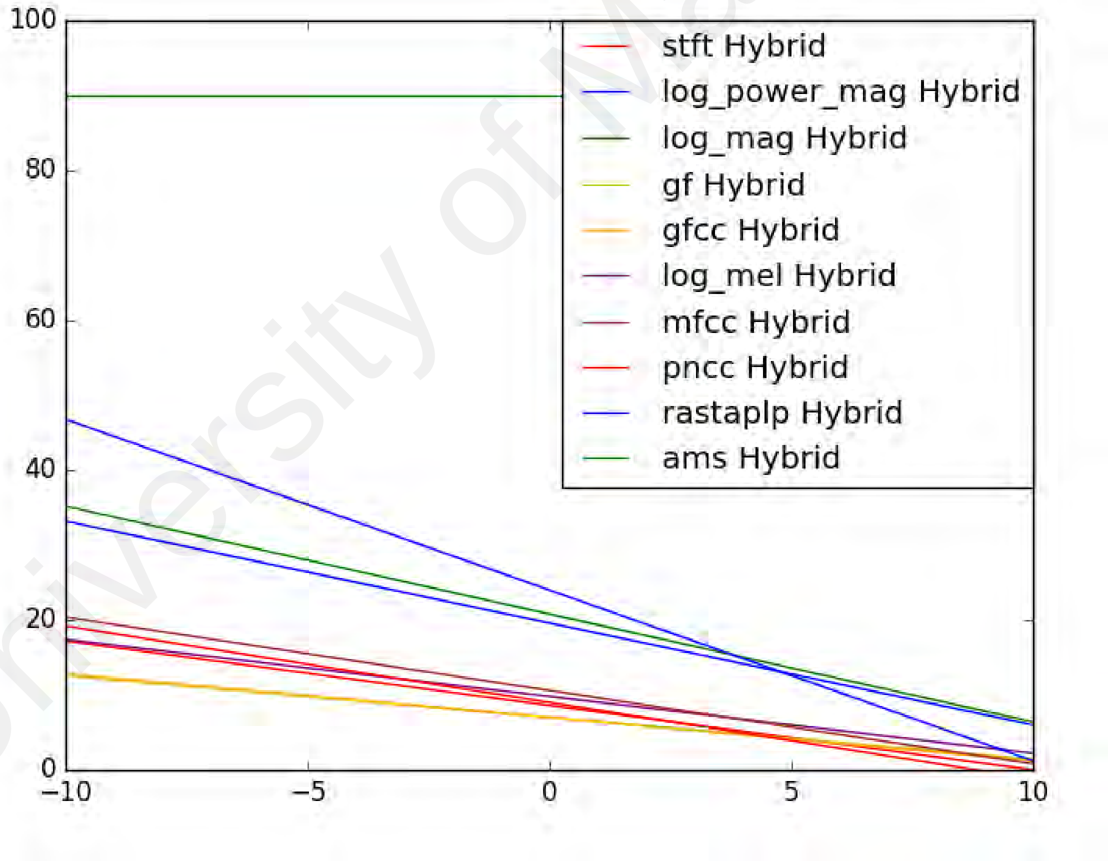


Figure 5.2: SNR vs WER for each feature

Figure 5.2 shows the WER of each of the features at different SNR levels. In general, for all the features the WER will reduce as the SNR increases. This is because the SNR increases when the level of interfering noise decreases which reduces the amount of miss translation (WER) of the model.

5.3 Discussion

5.3.1 Comparison of the results between speech separation and the ASR model

It can be seen that when using the GFCC feature the SS model can reduce the noise by 15%, which is the highest percentage out of the 7 features experimented. MFCC only manages a 2% reduction in noise. The rest of the features show MSE improvement from 9% to 14%. It can be argued that GFCC yields the best recognition and MFCC the lowest, based on the percentage of reduction in noise.

However, when evaluating the joint SS and ASR model, it can be seen that MFCC shows a better result at high SNR (10 dB) than GFCC, while, at -10 dB, GFCC shows much better accuracy. This shows that, while the SS model is not effective with MFCC, the ASR model shows a much better performance. Log_mag and log_power_mag shows the worst accuracy at -10 dB. It is interesting to see the even though STFT is the most basic feature, it can produce a good recognition accuracy at both low and high SNR levels. This would explain why STFT is the 'go-to' feature in many of the previous researches.

5.3.2 The most influencing speech feature(s) for speech separation model

Table 5.3 shows the most influencing features for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal) for the Speech separation model used in the Joint Model at different signal-to-noise ratios.

Table 5.3: Average MSE at different SNR of speech separation

SNR	stft	log_power_mag	log_mag	gf	gfcc	log_mel	mfcc	PNCC	RASTA-PLP	AMS	Average MSE
-10	14.97	39.61	40.42	42.77	38.01	I	41.29	36.43	25.12	30.96	35.50
-5	0.13	36.98	39.14	41.11	33.75	45.04	38.91	33.98	9.98	16.91	29.59
0	-10.96	36.01	39.73	43.58	40.39	47.62	33.85	29.05	-17.60	6.87	21.95
5	24.51	42.99	47.18	56.47	60.13	56.67	23.89	20.91	-79.17	60.55	29.32
10	43.13	56.80	60.21	72.74	77.04	68.60	4.51	7.46	-228.23	90.90	24.57

SNR: -10

SNR -10 indicates extremely high interfering noise. The average percentage of speech cleaned is 35%. STFT performs very low at this SNR level. log_power_mag, log_mag, gf, gfcc, log_mel, mfcc, PNCC, RASTA-PLP, and Amplitude modulation spectrum (AMS) displays similar results from 30% to 45%. However, despite being a naïve feature, log_mel performance is the best with 45% cleaned speech at SNR -10.

SNR: -5

SNR -5 indicates high interfering noise. At this SNR level, the interfering noise is louder than the target speech. The average improvement at SNR -5 is 29%. STFT shows the least improvement of 0.134% at this level. RASTA-PLP and AMS also show low improvement with 9% and 16%. log_power_mag, log_mag log_mag, gf, gfcc, log_mel, mfcc, and pncc recorded improvements ranging from 35% to 45%. Log_mel has the best results at SNR -5 with a 45% improvement.

SNR: 0

At SNR 0, the interfering noise is matched with the loudness of the target speech. At this level, the average improvement for all features is 21%. STFT and RASTA-PLP reported deteriorating results indicating that STFT is not suitable to be used as a feature in the speech separation at this SNR level. However, it might perform better if the raw features are fed into the ASR model, skipping the separation stage. AMS performance is also poor at only 16% improvement. log_power_mag, log_mag log_mag, gf, gfcc, log_mel, mfcc, and pncc shows improvements ranging from 35% to 47%. Log_mel displays the best results at SNR 0 with a 47% improvement.

SNR: 5

At SNR 5, the target speech is louder than the interfering sound. This is a more realistic level found in an everyday environment. At this level, the average improvement is 29%. RASTA-PLP reported deteriorating results at this level. However, STFT shows a better improvement at this level as compared to SNR 0. PNCC and MFCC display improvements of 20% to 30%. log_power_mag, log_mag log_mag, gf, gfcc, log_mel, and AMS displays improvements ranging from 40% to 60%. AMS shows the best improvement at this stage with 60% improvement.

SNR 10

At SNR 10, the interfering sound is minimal. At this level, the average improvement is 24%. RASTA-PLP still shows negative improvements and PNCC and MFCC perform under 10%. STFT, log_power_mag, log_mag, gf, gfcc, and log_mel show varying improvements all of which are over 40%. AMS shows the best improvements at this SNR level with 90%.

5.3.3 The most influencing speech feature(s) for the Joint Speech separation and ASR model

Table 5.4 presents the WER of each feature in terms of percentage. The most influencing features for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal) for the Joint Model at different signal-to-noise ratio are highlighted in green which are the gf, gfcc, pncc, and stft.

Table 5.4: Word Error Rate (%) of Speech Features for the Joint Model

SNR	Stft (%)	log_power_mag (%)	log_mag (%)	Gf (%)	Gfcc (%)	log_mel (%)	mfcc (%)	PNCC (%)	RASTA-PLP (%)	AMS (%)	Average (%)
-10	20	37	40	13	13	17	22	23	53	90	32.8
-5	11	25	25	8	10	16	16	12	31	90	24.4
0	7	15	15	8	7	8	9	5	19	90	18.3
5	2	11	15	6	3	4	6	3	11	90	15.1
10	3	10	9	0	2	4	1	2	6	90	12.7
Average	4.3	9.8	10.4	3.5	3.5	4.9	5.4	4.5	12.0	90.0	

In this section, the WER of the features in each of the SNR level will be discussed. Since AMS is an amplitude modulation based feature, it performed poorly at all SNR level. Due to this, AMS will be excluded from further discussion.

SNR -10

The average WER at SNR -10 is 32.8. RASTA-PLP, log_power_mag, and log_mag performed poorly with WER as compared with the average value. Stft, gf, gfcc, log_mel, mfcc, and PNCC performed better than the average. GF and GFCC perform equally well with a WER of 13.

SNR -5

The average WER at SNR -5 is 24.4. Except for log_power_mag, log_mag, and RASTA-PLP, all of the features performed better than the average WER. GF performs the best with the lowest WER of only 8.

SNR 0

The average WER at SNR 0 is 18.3. With the exception to RASTA-PLP, all the other features performed better than the average WER. GF performs best with a WER of only 8.

SNR 5

The average WER at SNR 5 is 15.1. All the features (except AMS) scored a WER below 15. STFT is the best at this level with WER of 2.

SNR 10

The average WER at SNR 10 is 12.7. All the features (except AMS) scored a WER below 10. GF is the best at this level with a WER of 0.

5.4 Summary

As shown in section 5.3.1 it can be seen that an improvement of MSE in the speech separation model does not always guarantee a better accuracy in the ASR model. The accuracy of the ASR depends on the type of features that are being used. In table 5.3 it can be seen that log mel speech features yields the most improvement on the percentage of speech features on average. However, when using the complete joint speech separator and ASR model, table 5.4 depicts that GF and GFCC yields the best average results. Therefore it can be concluded that Tu's (2016) joint speech separator and ASR model should be using either GF or GFCC for best results instead of log mel which was originally used by Tu (2014;2016).

CHAPTER 6 : CONCLUSION AND FUTURE RESEARCH

6.1 Overview

This chapter provides the overall conclusion of the research objectives stated in chapter 1. The main aim of this research is to devise a method for identifying the most influencing speech features towards increasing the recognition accuracy of joint speech separation and ASR model. From the experiments, it can be concluded that the joint speech separation and ASR model show a varying degree of Word Error Rate (WER) based on the sound features used. This concludes that the proposed method in identifying the most suitable feature is important for the joint model to perform well. As the Signal-to-Noise Ratio (SNR) of the input decreases, the WER rate increases in each of the features. It was also apparent that some features work better in low SNR while others work better in high SNR. This concludes that based on the environment a suitable sound feature should be selected.

6.1 Meeting of Research Aims and Objectives

This section discusses how each of the research objectives and questions is fulfilled.

6.1.1 Fulfilling Objective 1 and Question 1

Research Objective 1: To identify speech features used in the existing speech separation model for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal).

Research Question: What are the speech features used in the existing speech separation model for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal)?

To identify the features, past researches on speech separation were reviewed to identify different types of features used to build speech separation models. Table 1.3 summarizes the list of features from past studies. In section 2.5, these features used in speech separation have been discussed in detail. Following that, section 2.6, provides the list of speech features used in the existing speech separation model for separating the speech of the target speaker (clean signal) from the interfering speaker's speech (noisy signal). It also shows the frequency of each of the features used in different studies. The two sections address the first objective and the first research question.

6.1.2 Fulfilling Objective 2 and Question 2

Research Objective 2: To develop a method for experimenting the most influencing speech features based on the identified speech features in objective (1) using joint speech separation and the ASR model.

Research Question: What are the method(s) for experimenting the most influencing speech features -using the joint Speech separation and the ASR model at different signal-to-noise ratio?

In Section 3.1, a simplified framework that can be applied to develop a method for experimenting with the most influencing speech feature using joint Speech separation and ASR model was presented. In section 3.2, each of the phases is explained and the techniques used to develop the phases are discussed. The key stages used to develop the system are Record Voice, prepare dataset, extract features, train speech separation, create alignments, train ASR to join speech separation and ASR. Section 4.1, presents the detail of the implementation of the framework, elaborating on the steps taken to implement the framework. Python code with the help of Kaldi and Keras was used to implement the joint

SS and ASR model. Section 4.2 displays the detailed configurations used to build the framework. The speech separation and ASR model was built for both male and female speakers, which were trained using the 10 features listed in objective 1. In total 20 (2 male/female x 10 features) joint SS and ASR models were built. The above sections address the second objective and the second research question.

6.1.3 Fulfilling Objective 3 and Question 3

Research Objective 3: To evaluate and compare the performance of joint Speech separation and ASR model against the speech separation model using the identified features at different signal-to-noise ratios.

Research Question: How is the performance of joint Speech separation and ASR model against the speech separation model using the identified features at different signal-to-noise ratios?

Section 2.7 of chapter 2, discusses the evaluation techniques that can be used to measure the accuracy of the joint Speech separation and ASR model and speech separation model. To evaluate how well the speech separation can clean the speech features and extract the clean speech features, MSE was used. The MSE indicates how different the output of the speech separation is from its original clean speech. To evaluate how well joint speech separation and ASR model can transcribe what the target user is speaking, WER was used. Word Error Rate (WER) computed how many words the model was able to identify correctly. By utilizing the evaluation models (i.e MSS and WER), the two models have been evaluated using the features listed in objective 1 and the framework developed in objective 2. The results of this evaluation are shown in Tables 5.3 and 5.4. These address the third objective and third research question.

6.2 Conclusion

Chapter 2 lists the features used in the previous studies. STFT, log power mag, log mag, gf, gfcc, log mel, mfcc PNCC, RASTA-PLP and AMS are the features that were used in previous studies. Chapter 4 contains the details of developing joint speech separation and the ASR model. Using the models developed in chapter 4, the 10 features were evaluated.

As shown in table 6.1 the most influencing speech feature can be seen by GF. At a high SNR GF performs almost flawlessly with a WER of only 0. The average minimum WER is 4 (excluding AMS). STFT, GF, GFCC, Log_mel, MFCC PNCC have all been able to score below or equal to 4. Thus, proving to be a viable feature to be used at a higher SNR level. On the other hand log_power_mag, log_mag, RASTA-PLP, and AMS have all score high WER which indicates that the best performance at high SNR is not adequate.

The average maximum WER (excluding AMS) is 26.4. STFT, GF, GFCC, Log_mel, MFCC, PNCC have all scores below the average and so these features could be used at low SNR levels. Log_power_mag, Log_mag, RASTA-PLP, and AMS have all scores above the average which indicates that these should not be used at low SNR.

The slope indicates the gradient of the line of each feature. It is the relation between WER and SNR level. As expected as the SNR increases (becomes less noisy) the WER (Word error rate) decreases. This is why the slope for all the features is negative. As the slope decreases, it indicates a faster drop of the WER. Generally, if the Joint speech separation and ASR Model is used in an environment where the SNR level can vary then a higher slope is preferred. This will ensure that the performance of the model remains as stable as possible throughout varying SNR levels. In such cases, GFCC, GF, and STFT could be a viable choice. In the case where the SNR level of the environment can be predetermined than the values shown in table 6.2 can be used.

Table 6.1: Min, Max and slope Word Error Rate (WER) of features

SNR	stft	log_power_mag	log_mag	gf	gfcc	log_mel	mfcc	PNCC	RASTA-PLP	AMS
Minimum	2	10	9	0	2	4	1	2	6	90
Maximum	20	37	40	13	13	17	22	23	53	90
Average	4.3	9.8	10.4	3.5	3.5	4.9	5.4	4.5	12.0	90
Slope	-0.85	-1.35	-1.55	-0.65	-0.55	-0.65	-1.05	-1.05	-2.35	~

Based on the results in chapter 5 we can conclude that the following features should provide the lowest WER at different SNR levels as indicated by Table 6.2

Table 6.2: Best performing features by SNR level

SNR	-10	-5	0	5	10
Features	GF, GFCC	GF	PNCC	STFT	GF

6.3 Research Outcomes and Contributions

This research summarizes all the speech features used before in speech separation. It also counts the number of studies where each of the features has been used in. This summary indicates a general trend and preference of features that are being used in the field of speech separation.

Tu (2016), proposed a joint Speech Separation and ASR model. However, the model was not evaluated to identify features that resulted in the most accurate results. The evaluation in chapter 5 shows the details of the best features to be used in joint speech separation and the ASR model.

By understanding the most influencing speech features at different SNR levels, the findings can be applied in different real-life environments to obtain optimum recognition accuracy. This research divides its evaluation of the features based on SNR level. This uncovers which feature to use in different environments.

The joint speech separation and ASR model is a new and improved model, which is yet to be thoroughly tested. This research thoroughly tests this model against other previously used speech features. In doing so, this model can be configured with the appropriate features to maximize its accuracy. Moreover, this research also evaluates the model against features at different SNR levels. Figure 5.2 shows the relationship between the SNR and WER of each of the features. In doing so it helps to decide which features to use environments where the SNR level can be determined.

Moreover, this research also evaluates the speech separation and the Joint Speech separation and ASR separately for each of the features. This helps us understand the efficiency of the speech separation within a Joint speech separation and ASR model.

6.4 Limitations and Future Research

From findings in chapter 5, it can be concluded that different features excel at different SNR levels. The joint Speech separation and ASR model should be configured with the appropriate features based on the SNR of the input for the best results. The nature of the environment should determine the SNR level. However, this has to be predetermined for the model to work. It might be possible to determine the SNR levels of the environment beforehand to auto-configure the appropriate feature to use in a different language. In such a case, the model could be used more easily and broadly.

Reference

Barker, J., & Cooke, M. (2001). Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise. Proc. Eurospeech. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.5392&amp;rep=rep1&amp;type=pdf

Chen, J., Wang, Y., & Wang, D. (2014). A feature study for classification-based speech separation at low signal-to-noise ratios. IEEE/sACM Transactions on Audio Speech and Language Processing, 22(12), 1993–2002. https://doi.org/10.1109/TASLP.2014.2359159

Chen, Z., Luo, Y., & Mesgarani, N. (2017). Deep attractor network for single-microphone speaker separation. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2(1), 246–250. https://doi.org/10.1109/ICASSP.2017.7952155

Delfarah, M., & Wang, D. (2017). Features for Masking-Based Monaural Speech Separation in Reverberant Conditions. IEEE/ACM Transactions on Audio Speech and Language Processing, 25(5), 1085–1094. https://doi.org/10.1109/TASLP.2017.2687829

Du, J., Tu, Y., Dai, L., & Lee, C. (2016). A Regression Approach to Single-Channel Speech Separation Via High-Resolution Deep Neural Networks. 1–13. https://doi.org/10.1109/TASLP.2016.2558822

Du, J., Tu, Y., Xu, Y., Dai, L., & Lee, C.-H. (2014). Speech separation of a target speaker based on deep neural networks. Proceedings 12th International Conference on Signal Processing, ICSP 2014, 473–477. https://doi.org/10.1109/ICOSP.2014.7015050

Gupta, H., & Gupta, D. (2016). LPC and LPCC method of feature extraction in speech recognition. 498–502.

Healy, E. W., Delfarah, M., Vasko, J. L., Carter, B. L., & Wang, D. (2017). An algorithm to increase intelligibility for hearing-impaired listeners in the presence of a competing talker. The Journal of the Acoustical Society of America, 141(6), 4230–4239. https://doi.org/10.1121/1.4984271

Hu, K., & Wang, D. (2013). An Unsupervised Approach to Cochannel. 21(1), 122–131.

Huang, P.-S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2015). Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation. 23(12), 1–12. https://doi.org/10.1109/TASLP.2015.2468583

Huang, P.-S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2011). DEEP LEARNING FOR MONAURAL SPEECH SEPARATION Po-Sen. Acta Physica Polonica B, 42(1), 33–44. https://doi.org/10.5506/APhysPolB.42.33

Huang, P.-S., Kim, M., Hasegawa-Johnson, M., & Smaragdis, P. (2014). Deep learning for monaural speech separation. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). doi:10.1109/icassp.2014.6853860

Hershey, J. R., Z. Chen, J. Le Roux and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 31-35, doi: 10.1109/ICASSP.2016.7471631.

Jun, D., Yanhui, T., Yong, X., Lirong, D., & Chin-hui, L. (2014). Speech Separation Based on Improved Deep Neural Networks with Dual Outputs of Speech Features for Both Target and Interfering Speakers. Icsp, 473–477. https://doi.org/10.1109/ISCSLP.2014.6936615

Kolbaek, M., Yu, D., Tan, Z. H., & Jensen, J. (2017). Multi-talker Speech Separation with Utterance-level Permutation Invariant Training of Deep Recurrent Neural Networks. IEEE International Workshop on Machine Learning for Signal Processing, MLSP, 2017-Septe, 1–6. https://doi.org/10.1109/MLSP.2017.8168152

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. 1–38. https://doi.org/10.1145/2647868.2654889

Pan, J., Liu, C., Wang, Z., Hu, Y., & Jiang, H. (2012). Investigation of Deep Neural Networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMS in acoustic modeling. 2012 8th International Symposium on Chinese Spoken Language Processing, ISCSLP 2012, 301–305. https://doi.org/10.1109/ISCSLP.2012.6423452

Tu, Y., Du, J., Dai, L., & Lee, C. (2016). A Speaker-Dependent Deep Learning Approach to Joint Speech Separation and Acoustic Modeling for Multi-Talker Automatic Speech Recognition Georgia Institute of Technology. 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), 1–5. https://doi.org/10.1109/ISCSLP.2016.7918432

Tu, Y., Du, J., Xu, Y., Dai, L., & Lee, C. (2014). Speech Separation Based on Improved Deep Neural Networks with Dual Outputs of Speech Features for Both Target and Interfering Speakers. 250–254.

Wang, D. L., Brown, G. J., Zhang, X. X. L., Wang, D. L., Williamson, D. S., Member, S., Wang, D. L. (2016). On Training Targets for Supervised Speech Separation. IEEE Transactions on Neural Networks, 21(5), 1475–1487. https://doi.org/10.1109/72.761727

Wang, D., & Chen, J. (2017). Supervised Speech Separation Based on Deep Learning: An Overview. 1–27. https://doi.org/10.1109/TASLP.2018.2842159

Wang, Y., Du, J., Dai, L., & Lee, C. (2017). A Maximum Likelihood Approach to Deep Neural Network Based Nonlinear Spectral Mapping for Single-Channel Speech Separation. Interspeech, 1178–1182. https://doi.org/10.21437/Interspeech.2017-830

Wang, Y., Du, J., Dai, L.-R., & Lee, C.-H. (2017). A gender mixture detection approach to unsupervised single-channel speech separation based on deep neural networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(7), 1535–1546. https://doi.org/10.1109/TASLP.2017.2700540

Wang, Y., Narayanan, A., & Wang, D. (2014). On Training Targets for Supervised Speech Separation. 22(12), 1849–1858. https://doi.org/10.1109/TASLP.2014.2352935

Han, W., Chan, C, Choy, C. , & Pun, K. (2006). An efficient MFCC extraction method in speech recognition. 2006 IEEE International Symposium on Circuits and Systems, 4. https://doi.org/10.1109/ISCAS.2006.1692543

Yanhui, T., Jun, D., Yong, X., Lirong, D., & Chin-hui, L. (2014). Deep Neural Network Based Speech Separation for Robust Speech Recognition. 2014 12th International Conference on Signal Processing (ICSP), 532–536. https://doi.org/10.1109/ICOSP.2014.7015061

Zhang, X. L., & Wang, D. (2016). A deep ensemble learning method for monaural speech separation. IEEE/ACM Transactions on Audio Speech and Language Processing, 24(5), 967–977. https://doi.org/10.1109/TASLP.2016.2536478

Zhang, Y., & Glass, J. R. (2009). Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams. 398–403.

Weninger, F., Hershey, J. R., Le Roux, J., & Schuller, B. (2014). Discriminatively trained recurrent neural networks for single-channel speech separation. 2014 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2014, 577–581. https://doi.org/10.1109/GlobalSIP.2014.7032183

Schmidt, M. N., & Olsson, R. K. (2006). Single-channel speech separation using sparse non-negative matrix factorization. 2614–2617.

Weiss, R. J., & Ellis, D. P. W. (2010). Speech separation using speaker-adapted eigenvoice speech models. Computer Speech & Language, 24(1), 16–29. https://doi.org/https://doi.org/10.1016/j.csl.2008.03.003

B, F. W., Erdogan, H., Watanabe, S., Vincent, E., Roux, J. Le, & Hershey, J. R. (2015). Latent Variable Analysis and Signal Separation. 9237, 91–99. https://doi.org/10.1007/978-3-319-22482-4

D. T. Tran, E. Vincent and D. Jouvet, "Nonparametric Uncertainty Estimation and Propagation for Noise Robust ASR," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 11, pp. 1835-1846, Nov. 2015, doi: 10.1109/TASLP.2015.2450497.

D. T. Tran, E. Vincent and D. Jouvet, "Fusion of multiple uncertainty estimators and propagators for noise robust ASR," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, 2014, pp. 5512-5516, doi: 10.1109/ICASSP.2014.6854657.

Vinyals, O., Ravuri, S. V, & Povey, D. (2012). “Revisiting recurrent neural networks for robust ASR” International Computer Science Institute , Berkeley , CA , USA EECS Department , University of California - Berkeley , Berkeley , CA , USA. 4085–4088.

Hong Kook Kim and R. C. Rose, "Cepstrum-domain acoustic feature compensation based on decomposition of speech and noise for ASR in noisy environments," in IEEE Transactions on Speech and Audio Processing, vol. 11, no. 5, pp. 435-446, Sept. 2003, doi: 10.1109/TSA.2003.815515.

O. Vinyals and S. V. Ravuri, "Comparing multilayer perceptron to Deep Belief Network Tandem features for robust ASR," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, 2011, pp. 4596-4599, doi: 10.1109/ICASSP.2011.5947378.