A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECHNIQUE USING COMPOSITE SPEECH MODELS

BY

MAK WEN XUAN

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2020

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECHNIQUE USING COMPOSITE SPEECH MODELS

Academic Session: JAN 2020

I MAK WEN XUAN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
10, Lorong Sembilang 28,
Seberang Jaya,
13700 Perai, Pulau Pinang

Dr. Aun Yichiet
Supervisor's name

Date: 24/04/2020

Date: 24/04/2020

A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECHNIQUE USING COMPOSITE SPEECH MODELS

BY

MAK WEN XUAN

A PROPOSAL

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled "**A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECHNIQUE USING COMPOSITE SPEECH MODELS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : MAK WEN XUAN

Date : 24/04/2020

ACKNOWLEDGEMENT

I would like to express my sincere gratitude towards my final year project supervisor, Dr. Aun Yichiet for involving me in this project and enabling me to pursue knowledge in the domain of Acoustics, Speech and Signal Processing. Moreover, Dr. Aun had been extremely resourceful by providing guidance to certain deep learning and audio processing topics, as well as finding access to a research machine equipped with GPUs to make this project possible.

Secondly, I wanted to say thank you to my supportive friends and family for their unending love and everlasting encouragement throughout the entire process. They helped me in building a stronger mental fortitude. Last but not least, I would like to thank a very special person in my life, Tan Sze Mei, for standing by my side during difficult times, her unconditional love prompts me to push myself to be a better man. I definitely couldn't have completed this report without her.

ABSTRACT

Speech separation techniques are commonly used for selective filtering of audio sources. Early works apply acoustic profiling to discriminate against multiple audio sources. Meanwhile, modern techniques leverage on composite audio-visual cues for a more precise audio source separation. With visual input, speakers are firstly recognized for their facial features, then voice-matched for corresponding audio signal filtering. However, existing speech separation techniques do not account for off-screen speakers when they are actively speaking in these videos. This project aims to design a robust speaker-aware speech separation pipeline to accommodate speech separation for off-screen speakers. The pipeline essentially performs speech separation in a sequential fashion, starting from (1) audio-visual speech separation for all visible speakers, then (2) performing blind source separation on residual audio signal to determine off-screen speech. Two independent models are designed, namely an audio-only and an audio-visual model, which is then merged together to form a pipeline that performs comprehensive speech separation. The outcome of the project is a data type agnostic speech separation technique that demonstrates robust filtering performance regardless of input types.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

y	Noisy Input Speech
n	Background Noise
s	Clean Target Speech
$\hat{s}$	Estimated Clean Target Speech
$\hat{a}(y\|\theta)$	Masking Function Estimator
a^*	Target Mask
$\hat{a}$	Estimated Mask
θ	Parameters
D	Objective Function
t	Time
f	Frequency
S	Source Signal Sequence
x	Source Signal at time t
y	Mixed Signal Sequence
X	Source Signal Magnitude Spectra at time t
Y	Mixed Signal Magnitude Spectra
$\lvert S_{t,f} \rvert$	Magnitude Response
$\theta_{S_{t,f}}$	Phase Response
$\Re$	Real Component of Complex Number
$\Im$	Imaginary Component of Complex Number
e	Acoustic Signal Error
$l2$	Squared Error
A	Audio Spectrogram
J	Loss Function
T	Audio Sequence
γ	Regularization Constant
y_x	x-th Clean Speech Input
$\tilde{y}_x$	x-th Predicted Speech Output
M	Complex Ideal Ratio Mask
$\mathbb{R}$	Real Numbers
∞	Infinity

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
GMM	Gaussian Mixed Model
DNN	Deep Neural Network
CNN	Convolutional Neural Network
MRI	Magnetic Resonance Imaging
PNC	Passive Noise Control
ANC	Active Noise Control
RNN	Recurrent Neural Network
SNR	Signal-to-Noise Ratio
PSA	Phase-sensitive Spectrum Approximation
LSTM	Long Short-term Memory
BLSTM	Bidirectional Long Short-term Memory
STFT	Short-time Fourier Transform
iSTFT	Inverse Short-time Fourier Transform
PIT	Permutation Invariant Training
MSE	Mean Squared Error
VAD	Voice Activity Detection
VLAD	Vector of Locally Aggregated Descriptors
SDR	Signal-to-Distortion Ratio
SIR	Signal-to-Intereference Ratio
SAR	Signal-to-Artifact Ratio
FPS	Frames Per Second
MTCNN	Multi-task Convolutional Neural Network
cRM	Complex Ratio Mask
cIRM	Ideal Complex Ratio Mask
ReLU	Rectified Linear Unit
GPU	Graphics Processing Unit
API	Application Program Interface
STOI	Short Term Objective Intelligibility
PESQ	Perceptual Evaluation of Speech Quality
MIR	Music Information Retrieval

Chapter 1 Introduction

1.1 Problem Statement and Motivation

The human brain has an innate capability to concentrate its auditory attention towards a specific sound source, while effectively filtering out other sounds. This is described as the 'cocktail party effect', where an individual is capable of concentrating on a single conversation even though he/she is inside a noisy room. As to how the human brain achieves such a feat, the question remains unanswered. Yet, there's research that describes that when a speaker's facial features are available, the listener is capable of resolving perceptual ambiguity even though the circumstances are unfavorable. (Ma et al. 2009; Golumbic et al. 2013)

Automatic speech separation is the task of separating a noisy input audio signal into its individual denoised/enhanced audio signals. In order to obtain a reasonable solution, usually the algorithm requires a priori knowledge of the clean audio source or special configuration of multiple microphones, which is not a feasible way. When attempting to computationally recreate the cocktail party effect, the task becomes non-intuitive as mixed speech signals usually overlap one another, in addition to being non-linear signals. This problem is subsequently coined with the term 'cocktail party problem'.

There are often scenarios where conversations are inaudible within a video clip, e.g. heated political debates, where multiple speakers are speaking simultaneously, with the addition of non-speech background noises, e.g. ringing smartphones. These mixed sound signals result in the degradation of speech quality and interpretation. There already exist audiovisual methods that are able to separate mixed sound signals within video clip, in which the model is able to produce a number of enhanced audio output corresponds to the number of specified speakers.

These audiovisual models are able to precisely isolate different speeches because they utilize visual information in the form of speaker's facial features to 'match' the corresponding speech signals in the spectrogram. Yet, these audiovisual models are only effective in isolating mixed speech for when speakers show their faces in a recognizable manner. Due to the fact that these models depend heavily on information extracted from facial features, it isolates distinct speech signal for each detected speaker in the video. The rest of the sound sources are effectively categorized as background

noises, even though the ‘residual signal’ may contain speech signals from other speakers that did not appear inside the video. This prompted the motivation for creating an audiovisual neural network model that is able to detect and isolate each and every speech signal which corresponds to an individual speaker, regardless of its appearance in the video.

1.2 Project Scope

This project aims to build upon existing audio-visual neural networks that is able to separate mixed speech signals that corresponds to the number of speakers detected. The model should be able to distinguish speech signals between multiple speakers, regardless if they appear on- or off-screen, as well as categorizing all non-speech signals as background noise and suppressing them.

1.3 Project Objectives

The objectives of this project are:

i. To design a speaker-aware speech separation pipeline that is able to distinguish different speech signals that corresponds to the number of detected speakers regardless of speaker visibility.
   a. To design a data collection and preprocessing technique to build a dataset for training a speaker-independent audiovisual speech separation neural network.
   b. To build an audio-visual model to isolate speech for visible speakers in a video clip
   c. To build an audio-only model to enhance speech for non-visible speakers in a video clip

ii. To investigate state-of-the-art speech separation techniques and their relative performances.

iii. To evaluate the performance of the proposed speaker-aware speech separation pipeline.

1.4 Impact, Significance and Contribution

The outcome of this project is a novel speech separation neural network model that is capable of distinguish different speech signals that corresponds to the number of detected speakers, including those that are speaking off-screen. The result obtained is

significant, considering this feat had not been achieved by any previously reviewed literature. By achieving the project objectives, the boons will be threefold: (a) a large-scale, preprocessed dataset will be made available for training speaker-independent audio-visual networks; (b) separated audio sources can serve as input for automatic speech recognition and video transcription purposes; (c) enhance interaction quality between human and machine in a multi-speaker scenario.

1.5 Background Information

1.5.1 Introduction to speech separation

Speech separation is the effort of isolating a target speaker's voice from its background interference (Wang and Chen, 2018). When speech separation is conducted through a medium that contains multiple sound sources, the problem is better known as the cocktail party problem, as describe by Cherry (1953). However, speech separation is not to be confused with other terms such as speech recognition or speaker recognition. Speech recognition is the task of recognizing spoken words or phrases, then converting them into a format compatible with computers, whereas speaker recognition is the task of recognizing an unknown speaker based on characteristics derived from his/her speech signals.

1.5.2 Speech separation using deep learning

Traditional signal processing technique implement shallow structured architectures that contain either one or two layers for non-linear feature transformation, such as support vector machines (SVMs) or Gaussian mixture models (GMMs). Usually, these architectures work fine for simple and constrained signal problems, but their inability to process complex signals such as human speech prompted the use of deep learning techniques, which has a more layered and deep architecture to support information extraction from these complex signals. There are two methods that employ such techniques, namely audio-based and audiovisual-based methods. In audio-based methods, a noisy audio signal serves as an input to a neural network, which then produces a clean audio signal that contains only the target speaker's voice, effectively suppressing other sound signals altogether (this implies other speakers as well as background noise). In contrast to audiovisual-based methods, the neural network takes in two inputs: a noisy audio signal and images of the corresponding speaker. It then outputs a clean audio signal as well, that represents the target speaker's voice only.

1.5.3 Deep neural network

The moniker neural network derives its name from the neural connections of the human brain, its basic building blocks known as neurons, analogous to the biology term. These artificial neurons can be said to act as mathematical functions, which take in several inputs and produces an output, much similar to its biological counterparts. The function contained within a neuron are generally coined as activation function.

Artificial neurons can be arranged in a way that they form a layer, and when these layers are stacked together, they would form what is so-called a neural network. Outputs from a layer of neurons will be fed as inputs for the next layer of neurons, forming a complex chain which mimics the inner workings of a human brain. This is why usually neural network with more layers are able to better represent complex functions, but the number of layers should always be determined according to the complexity of the problem

.

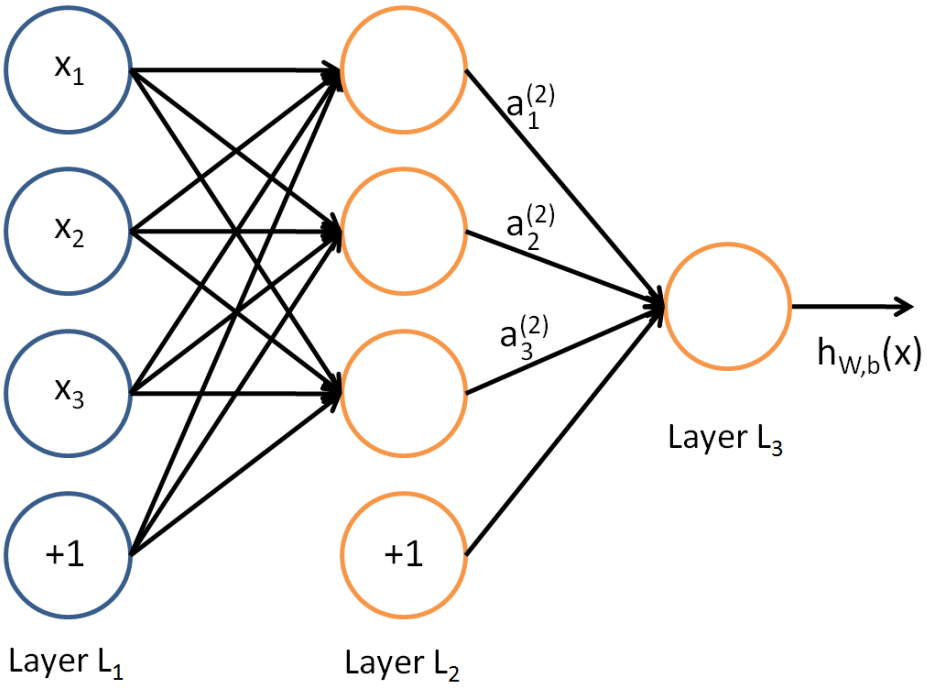


Figure 1.1: Layers of CNN

The input layer is located at the leftmost layer, whereas the output layer is located at the rightmost layer. The layers in between are known as the hidden layer, because its computes intermediate values that are not visible throughout training phase. When a neural network possesses more than one hidden layer, it is referred as a Deep Neural Network (DNN).

1.5.4 Convolutional neural network

Convolutional neural networks (CNN) is a variant of neural network which specializes in image processing and classification. As the name suggests, a CNN's hidden layers are made up of convolutional layers, with pooling layers that interweave between them. These layers differ from the traditional layers in a way that the activation functions are replaced with convolution and pooling functions instead. Not only that, a CNN's hidden layers also consist of pooling layers and fully-connected layers.

The next few paragraphs will elaborate briefly upon the common hidden layers for CNN. Convolution is meant by taking two images, one as "input" and the other as "filter". The filter is applied over the input in a sliding window fashion to produce an output. Imagine a 3x3 filter matrix that overlays a 3x3 input matrix, which its dot product will be computed and stored, according to Figure 1.2. This process will witness the filter matrix sliding over every possible block of the input matrix that it convolves. After the filter convolves with the entire input, what's left will be a matrix of dot products, to be passed on to the next layer for another iteration of convolution.

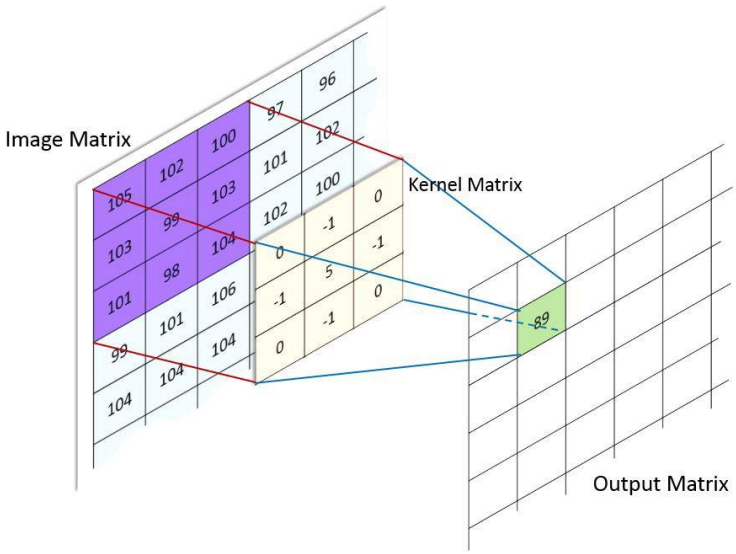


Figure 1.2: Spatial Convolution

On the other hand, pooling is a said to be a sample-based discretization process. An input representation is down-sampled to reduce its dimensionality, and allow researchers to make assumptions regarding the features contained in the sub-region bins. There are two types of pooling, namely max and min pooling. Max pooling is selecting the maximum value from the sub-region and min pooling is the complete inverse.

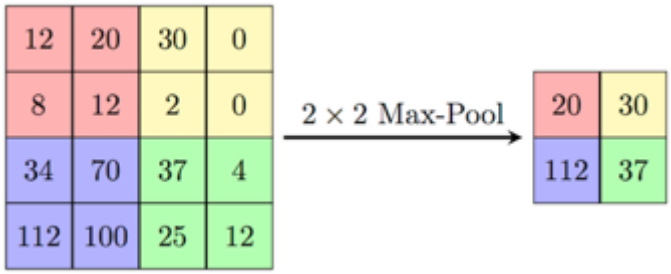


Figure 1.3: Max Pooling

1.5.5 Temporal and spectral features in speech processing

In essence, temporal features represent features in the time domain, whereas spectral features represent features in the frequency domain. Usually, temporal features are easily extracted from speech signals to interpret its properties, such as amplitude, sound level, signal energy, zero crossing rate, etc. On the other hand, spectral features are derived by converting the speech signal into the frequency domain through Fourier Transform, for example spectral density, spectral flux, spectral roll-off, spectral centroid, etc. Since these features are linked to the frequency domain, it can be used to determine the pitch, note, rhythm, melody of speech and much more.

1.5.6 Issue with noise pollution

After the industrial revolution, humans are heavily reliant on industrial equipment to perform daily functions. However, the increased use of industrial equipment comes with increased production of acoustic noise. One prominent example involves the magnetic resonance imaging (MRI) system, a medical equipment that utilizes magnetic fields and radio waves to visualize the insides of a body. When in use, the MRI scanner generates a rhythmic thumping noise, which ranges from 110 to 120 decibels (dB) (approximately the noise level of a live concert). These noise pollutions provide an unpleasant auditory experience, which could possibly affect the health of human hearing, digestive system, nervous system, endocrine system, etc. (Møller 1984, George and Panda 2013). Thus, the field of acoustic noise cancellation emerged as a result. Figure 1.4 illustrates noise intensity according to decibel, as well as its relevant examples.

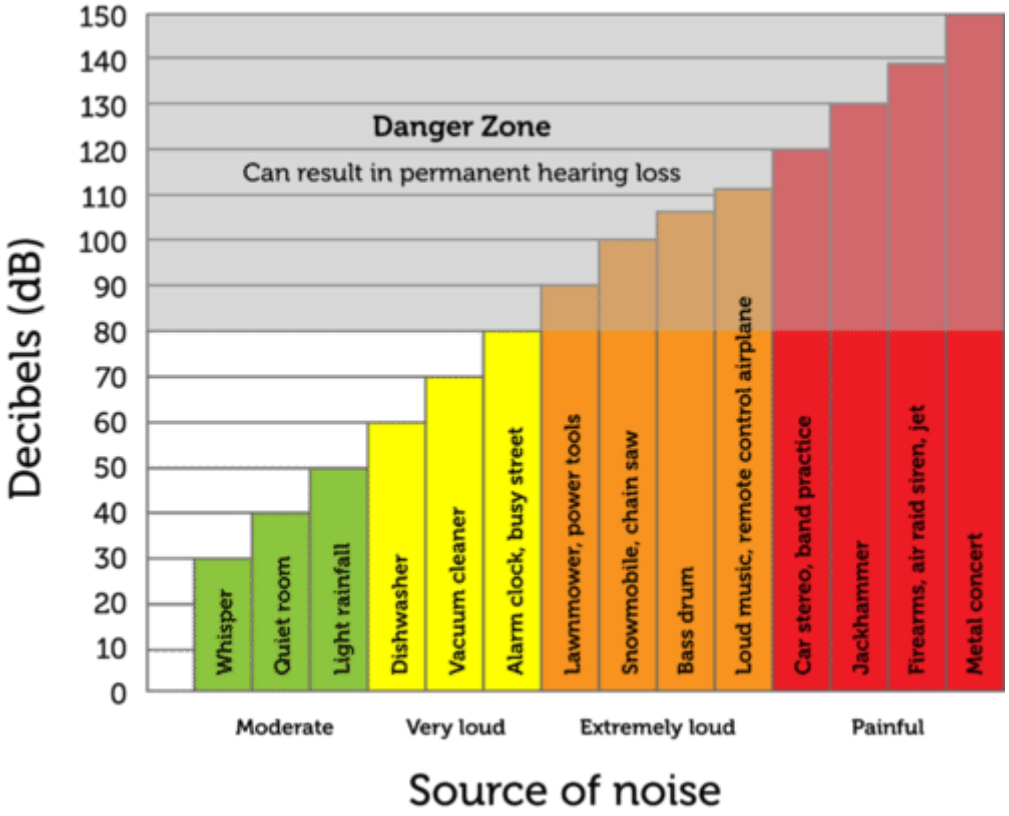


Figure 1.4: Noise Intensity Chart

1.5.7 Wide ranges of acoustic noise

The typical hearing range of human is 20 to 20,000 Hertz (Hz), although there might be considerable variation between each individual. Acoustic noise can occur on anywhere on the range of audible frequencies; these noises can be classified as low, middle and high frequency. Low frequency sound ranges from 10 to 200 Hz, and they often sound like a low rumble to the human ear. Due to the fact that hearing becomes gradually less sensitive as frequency decreases, these frequencies are being felt as vibrations rather than being heard as sounds. Middle frequency sound ranges between 500 to 2000 Hz, where human speech can be intelligently determined. The frequency range of 2000 to 5000 Hz is where human hearing is most sensitive, that is why most alarms and sirens fall within these frequency range. However, it is impossible to determine the direction of the sound at this frequency range. Finally, high frequency sound ranges between 8 to 22 kHz. These frequencies mimic a ‘sizzle’ sound when being transmitted over the air. Although sounds with frequencies over 16kHz can hardly be heard, but are not completely inaudible. When exposed long enough, these sounds could induce hearing loss, tinnitus, dizziness and nausea. Figure 1.5 below describes

the audible frequency range and provides some examples to help conceptualize sounds with their corresponding frequency ranges.

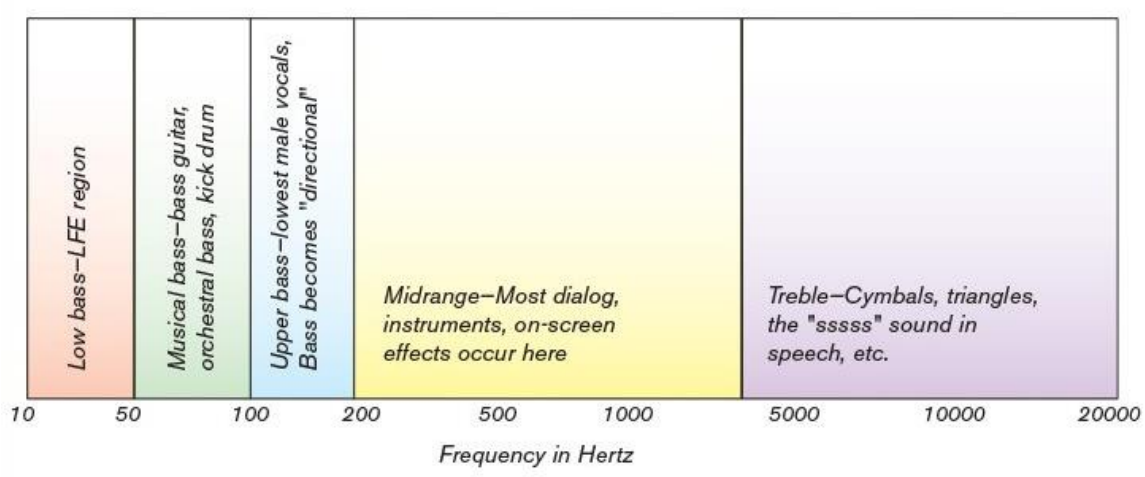


Figure 1.5: Audible Frequency Range and Examples

1.5.8 Acoustic noise cancellation

Acoustic noise cancellation can be categorized into two types: passive and active. Passive noise control (PNC) takes an orthodox approach, which relies on various sound-absorbing materials such as (1) earplugs or earmuffs to block unwanted noise from entering the ear canal; (2) acoustic foams to absorb ambient noise from one's surroundings. Although these techniques are able to cancel out a wide range of noise in the audio spectrum, the materials are relatively bulky and expensive to acquire. Plus, they are inefficient when it comes to cancelling low frequency noise. Thus, a novel technique was proposed to overcome this limitation, called active noise control (Lueg 1936). The technique requires the use of a microphone to record surrounding noise as reference, as well as digital signal processing techniques to generate anti-noise, which is audio waves with the same amplitude, but with inverted phases. A loudspeaker then broadcasts the anti-noise, effectively cancelling each other out through destructive interference of acoustic waves. ANC and PNC complement each other in a way that ANC attenuates low frequency noise in an efficient manner, but performs abysmally when cancelling out high frequency noise.

1.5.9 Two categories of noise

Broadband noise, also known as wideband noise, possess energy that is scattered over a relatively broad range in the audio spectrum. Conversely, narrowband noise possess

energy that is dispersed over a relatively narrow range in the audio spectrum. When specifying narrowband sources, it is also important to specify the frequency at which it occurs. Typical broadband sources are noise from a propulsion system (steam engine), whereas typical narrowband sources can be found for various pieces of machinery (motors, pumps). However, there's no definitive approach to determine whether a certain frequency range should be classified as broadband or narrowband signals. Figure 1.6 and Figure 1.7 are illustrations of broadband and narrowband noise in an amplitude-frequency graph respectively.

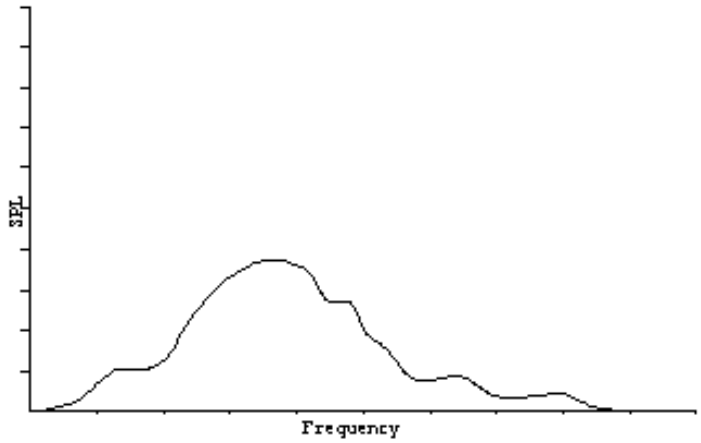


Figure 1.6: Broadband Noise

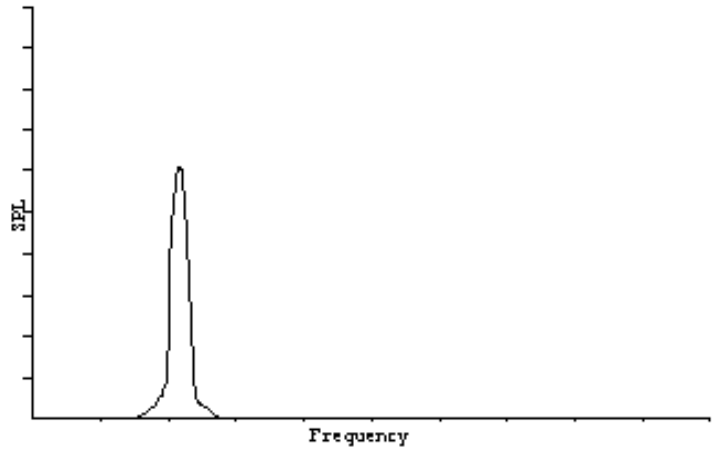


Figure 1.7: Narrowband Noise

Chapter 2 Literature Review

In the field of audio processing, the task of speech separation is well-studied, with extensive research spanning decades. Yet, it remains ill-posed. Despite recent advances with deep learning techniques, single-channel multi-source speech separation is still considered a difficult problem. This is true even with multi-channel audio inputs. Previous literatures had attempted to tackle such problems, but given the experiments are conducted with specific configurations, the flexibility of the solution against real-world scenarios is effectively limited (Girin, Gannot and Li 2018)

2.1 Time-frequency masks

With speech separation tasks, usually a masking function is used to estimate a time-frequency mask. This mask is then multiplied with the frequency-domain feature representation (a.k.a. spectrogram) of a noisy input speech to produce an estimate of clean target speech. Despite that, the estimated masks need to be applied onto spectrogram frames one at a time, because speech signals usually fluctuate and are non-linear in nature.

Thus, discrete Fourier transform are computed on the time-domain signals for each windowed frame, to obtain a complex-valued short-time spectrum of noisy input speech $y_{f,t}$, background noise $n_{f,t}$, and clean target speech $s_{f,t}$. Given the masking function $\hat{a}_{f,t}$, the clean target speech $\hat{s}_{f,t}$ can be estimated with the formula of

$$\hat{s}_{f,t} = \hat{a}_{f,t} \cdot y_{f,t}$$

When training a supervised learning speech separation network (noisy input speech and clean target speech are given), the minimization of an objective function will proceed to train an estimator $\hat{a}(y|\theta)$ for the masking function $\hat{a}_{f,t}$:

$$\hat{\theta} = \arg min_{\theta} \sum_{f,t} D(\hat{a}_{f,t})$$

where θ represents parameters and $\hat{a}$, y represent variables for all time-frequency bins. There are two types of objective functions, namely mask approximation and signal approximation. Mask approximation objective function measures error between target mask a^* and estimated mask $\hat{a}$, whereas signal approximation objective function measure error between target clean speech s and estimated speech $\hat{a}y$. The respective formulas are simply:

$$D_{ma}(\hat{a}) = D(a^*||\hat{a})$$
$$D_{sa}(\hat{a}) = D(s||\hat{a}y)$$

2.2 Audio-based speaker-dependent speech separation methods

Audio-only methods of speech separation have always existed as a fundamental problem in audio signal processing. A recently published review paper written by Wang and Chen (2018) gave a comprehensive overview of recent methods that tackle audio-only speech separation based on deep learning. The following paragraphs shall discuss two techniques deployed for the task of speech separation.

2.2.1 Phase sensitive spectrum approximation

There are several researchers that proposed different techniques onto recurrent neural networks (RNN) in modern literature. In particular, Erdogan et al. (2015) developed a phase-sensitive spectrum approximation objective function that considers both amplitude and phase error of noisy input in its calculations. The goal of an objective function is to calculate complex short-time spectrum errors between the speech estimate and clean target speech. When this error is computed and reduced, the signal-to-noise ratio (SNR) will be significantly improved in the speech estimate. Historic objective function solely takes the amplitude error of noisy input into account, in which the reconstructed speech estimates still preserves the noisy input phase. When phase error is present in the reconstructed speech estimate, its amplitude will differ from the clean target speech amplitude, which deteriorates audio quality. However, when phase error is considered in the objective function's calculations, the distortion in amplitude is compensated by shrinking the estimated mask, indirectly improve the speech estimate SNR. The phase-sensitive spectrum approximation (PSA) objective function is as below:

$$E^{PSA}(\hat{a}) \sum_{f,t} \left|\hat{a}_{f,t} y_{f,t} - s_{f,t}\right|^2$$

2.2.2 Vanishing/Exploding gradients

An error gradient represents the magnitude and direction that is used to update the network weights, which is calculated during the training phase. It undergoes backpropagation up until the initial layer. However, gradients from deeper layers have to go through multiple matrix multiplications due to the chain rule. When approaching

shallower layers, the gradients could diminish or accumulate. A gradient could shrink exponentially until it vanishes (vanishing gradient problem), or it could grow exponentially until it explodes (exploding gradient problem). Updating the network weights with it could result in an unstable network (either the model is unable to learn, or the model crashes).

2.2.3 Long short-term memory and its variant

The reasoning for deploying a recurrent neural network is its capability to recognize previous state sequences and thus, utilize the context information provided to derive better results. Yet, RNNs face the problem of 'vanishing or exploding gradients' during its back-propagation (Weninger et al. 2015). To alleviate this, long short-term memory (LSTM) structures are implemented as a quick and dirty fix. It is designed to capture long-term dependencies in previous state sequences, which regular RNNs are unable to do so. The way LSTM work is that it replaces the hidden nodes of RNNs with memory cells equipped with input, output and forget gates, which controls information flow in and out of the cells. Furthermore, memory cells are also capable of modifying the scalars and vectors stored within them. Even though backpropagation through time is performed, the gradient never vanishes or blows up because the recurrent connection from each memory cell to itself is just 1 (Erdogan et al. 2015). In attempts to further improve the network, Erdogan et al. experimented with bidirectional LSTM network (BLSTM). As the term suggest, BLSTM has recurrent connections in dual directions (forward and backward). Thus, it could utilize contextual information from both sides of the state sequence. Results from the literature had shown that BLSTM successfully achieved improvements over its LSTM baseline.

2.3 Audio-based speaker-independent speech separation methods

2.3.1 Deep clustering

Hershey et al. (2016) proposed a novel framework termed 'deep clustering', which clusters and separates distinct speech signals with spectrogram embeddings that are discriminatively-trained. Audiovisual speech separation problems can be viewed as segmentation problems, where a set of speech signal 'elements' is formulated through an indexed set of features, each carrying a chunk of signal information. These elements can be represented as pixels for images, or time-frequency coordinates for speech

signals. They are then segmented into groups or partitions by assigning respective group or partition labels.

Clustering is technically different to segmentation in the sense that clustering calculates domain-independent pairwise point relations based on simple objective functions, whereas segmentation relies on complex processing of training examples with given segment labels. Yet, clustering methods can be applied to segmentation problems as well. The literature proposed a partition-based segmentation method that learns each input elements' embeddings to determine its correct labels through simple clustering methods, which is more computationally efficient. The framework uses an objective function to derive embedding features such that distances between embeddings of elements within a partition is minimized whereas distances between embeddings of elements in different partitions are maximized. Given that the network uses a fixed-dimensional output, all partitions and their permutations can be implicitly represented. Partition-based segmentation can then be applied using simple and computationally efficient clustering algorithms, such as k-means. The estimated partition is then used to build a time-frequency mask, which is then applied onto the source signal to produce a spectrogram, which is further inverted using inverse STFT (iSTFT) to obtain the separated speech signal.

2.3.2 Label permutation problem

Single-channel speech separation is also known as monaural speech separation, in which the end goal is to estimate distinct speech signals in a linearly mixed audio signal retrieved using only with a single microphone. To better understand the label permutation problem, some baseline notations have to be defined.

S is the source signal sequences in the time domain as $x_s(t), s = 1, \dots, S$
whereas $y(t) = \sum_{s=1}^{S} x_s(t)$ is the mixed signal sequence.

When calculating the Short-time Fourier transformation (STFT) of these source signals, the corresponding outputs are $X_s(t, f)$ and $Y(t, f) = \sum_{s=1}^{S} X_s(t, f)$ for each time t and frequency f, respectively. In speech separation problems, $Y(t, f)$ is given, and the end goal is to estimate $X_s(t, f)$ for each source signals. However, in the transformation process, only the magnitude of mixed spectrum $|Y(t, f)|$ is available.

With such limited information, it is impossible to estimate accurate combinations of source signals $|X_s(t,f)|$, because it is possible that there are infinite combinations that could overlap to form $|Y(t,f)|$. Moreover, since the trained model estimates a set of masks $M_s(t,f)$ for each source signals in a simultaneous manner, there's nothing that dictates the order of output vector between mixed signal and masks. In other words, the permutation of output masks is unknown. This is described as the label permutation problem (Hershey et al. 2016; Yu et al. 2017).

2.3.3 Permutation invariant training

Permutation invariant training (PIT) is said to eliminate the label permutation problem entirely (Yu et al. 2017; Kolbaek et al. 2017). The mechanism behind this is that the source signals are given to the model as a set, in lieu of an ordered list as described in above paragraph. Under any circumstances, the training result will be consistent. The algorithm first calculates all the possible assignments ($S!$) of reference signals to each estimated source. By taking a single reference signal $|X_s|$ and its estimated source $|\tilde{X}_s|$, the combined pairwise mean squared error (MSE) can be calculated for all of its possible permutations. Each permutation and its respective calculation results in a value known as the total MSE. Subsequently, permutation with the lowest MSE is selected and undergo reduction through the optimization of model weights. This performs two actions at the same time: label assignment as well as error evaluation.

Although the computational complexity for permutation of speakers is factorial in nature, the computational complexity for pairwise MSE is only quadratic, meaning that PIT is suitable for separating speech for numerous speakers, e.g. $S \geq 2$. In PIT, a meta-frame is defined as X successive frames, which is congregated to exploit contextual information. An output meta-frame consisting of Y successive frames can be estimated with just a single input meta-frame; by shifting the input meta-frame by one or more frames repeatedly, speech separation can be carried out. As stated in the PIT training criterion, the permutation of output-to-speaker will remain constant for a given output meta-frame. However, when different output meta-frames are considered, there's the possibility that the permutation may change. It is suggested that in order to achieve better performance, a speaker-tracing algorithm may be integrated into the PIT framework.

2.4 Audiovisual-based speaker-dependent speech separation methods

There already exists a multitude of literatures that describes the workings of audiovisual-based methods (Rivet et al. 2014), without using neural networks for the task. More recent literatures had started to deploy neural networks that utilize both auditory and visual information to tackle speech separation problems. The literature reviewed in this section will elaborate audiovisual methods that are speaker-dependent, which implies that the neural network models are trained for each target speaker. This largely limits the flexibility of the speech separation models when imposed on real-world scenarios.

2.4.1 Multi-task encoder-decoder system

A CNN-based audiovisual encoder-decoder system had recently been proposed by Hou et al. (2018) that uses multi-task learning for speech enhancement tasks. The term encoder-decoder system can be described as processing input data with different modalities individually with separate CNNs, afterwards fusing them together to form a joint network for further computations. The term multi-task learning is mentioned here because the model will have to process heterogenous information to learn joint multimodal features. By providing visual data in both input and output during model training, along with noisy speech at input and clean speech at output, it serves as a constraint for the model. Compared to previous audiovisual speech separation literature, which deploys DNN for audio data and CNN for visual data respectively, Hou et al. takes a completely novel approach by adopting CNNs to process data from audio and visual streams. There's a clear distinction in this approach. Rather than feeding speech signals to a DNN, the speech signals are converted into the time-frequency domain, then transformed into a spectrogram sequence to be fed into a CNN. It is believed that such unique audiovisual encoder-decoder network design was not deployed in any literature in relevant research fields.

There are additional findings for the literature as well. For instance, early fusion of audio and visual streams experienced a decrease in performance in comparison to late fusion. In relation to audio and visual features, early fusion meant concatenation of features before going through convolution and pooling operations, whereas late fusion meant the other way around. Not only that, the research also found high correlation between speech and lip shapes as it is an effective auxiliary feature in voice activity

detection (VAD). When the multimodal inputs contain mismatched visual features such as incorrect lip sequences, the performance of the audiovisual model dropped significantly, which is reflected by increased losses.

2.4.2 Noise invariant training

Another recent literature features Gabbay et al. (2017) proposing a speaker-dependent audiovisual neural network model for speech separation tasks. The novelty in the research is the deployment of a training method known as 'noise-invariant training'. It made use of a unique audiovisual dataset compiled from scratch, which the background noises in the video is mixed synthetically with additional voices of the target speaker. The resulting video would technically contain multiple voices, yet the voices all belong to a single person. It should be noted that there's only a single instance of the target speaker within the video. The rationale behind this research is that although DNNs can effectively differentiate between sources of unique speech, its limitation lies within variance of a single speech source (Isik et al. 2016; Chen 2017).

By training the network with the presence of such videos, it prompts the model to better exploit visual features as well as generalizing well to different noise types. The neural network architecture also represents that of an encoder-decoder system, deploying dual CNNs that each takes in audio and visual input respectively, which is then encoded to form a shared embedding. Subsequently, the shared embedding is decoded by entering transposed convolutional layers to form the end product, which is a spectrogram representing the enhanced speech. The network is trained to minimize the mean square error l_2 loss among output spectrogram and ground truth spectrogram. It is crucial for the model to be trained with both input modalities, else the noise-invariant training method would fail, as stated by the authors.

2.5 Audiovisual-based speaker-independent speech separation methods

2.5.1 Face embeddings as visual representation

A recent literature by Ephrat et al. (2018) proposed the first speaker-independent audiovisual model, that is able to isolate distinct speech signals from mixed speech signals, inclusive of background noise. Moreover, it is capable of performing high quality speech separation with real-world scenarios, which was not addressed by previous speech separation literatures, possibly due to model rigidity. A real-world

scenario can be represented with video with mixed speech, such as screaming children, noisy bar, undisputed interview etc. The proposed model from the literature is proven to outperform modern audio-only speech separation techniques in terms of mixed sounds. When compared to existing audiovisual speech separation techniques (which are speaker-dependent), the proposed model is able to outperform those them as well, albeit by a small margin. The term speaker-dependent meant that a dedicated model had to be trained for a target speaker. However, the model is only able to perform speech separation for that particular target speaker that it is specifically trained for. Conversely, the term speaker-independent is meant that the model is able to perform speech separation regardless of who the target speaker is. In addition to that, it could also separate speech in languages that were originally not present inside the training dataset.

2.5.2 Estimating speech spectrogram with lip regions

Inspired by the film *The Conversation*, Afouras et al. (2018) conducted research to isolate individual speakers from a multi-speaker environment with visual information from the target speaker's lip region. Traditional speech enhancement methods only dealt with the refinement of magnitude of noisy input signal, whereas the phase of noisy input signal is used separately for signal reconstruction. Usually this approach works well for high SNR scenarios, but when SNR decreases, the noisy input signal phase becomes a bad approximation of the ground truth signal phase (Fu et al. 2017). Hence, a deep neural network is trained to predict magnitude and phase of the denoised speech spectrogram. The proposed technique is unique in several ways, such that: (a) spectrograms are treated not as images, but as temporal signals with frequency bins as channels, allowing a deeper network with a large number of parameters to be built; (b) a soft mask is applied onto the noisy input signal for filtering, rather than synthesizing the clean output signal from scratch, which is found to be more effective; (c) a phase enhancing sub-network is introduced to the conventional audiovisual speech enhancement network architecture.

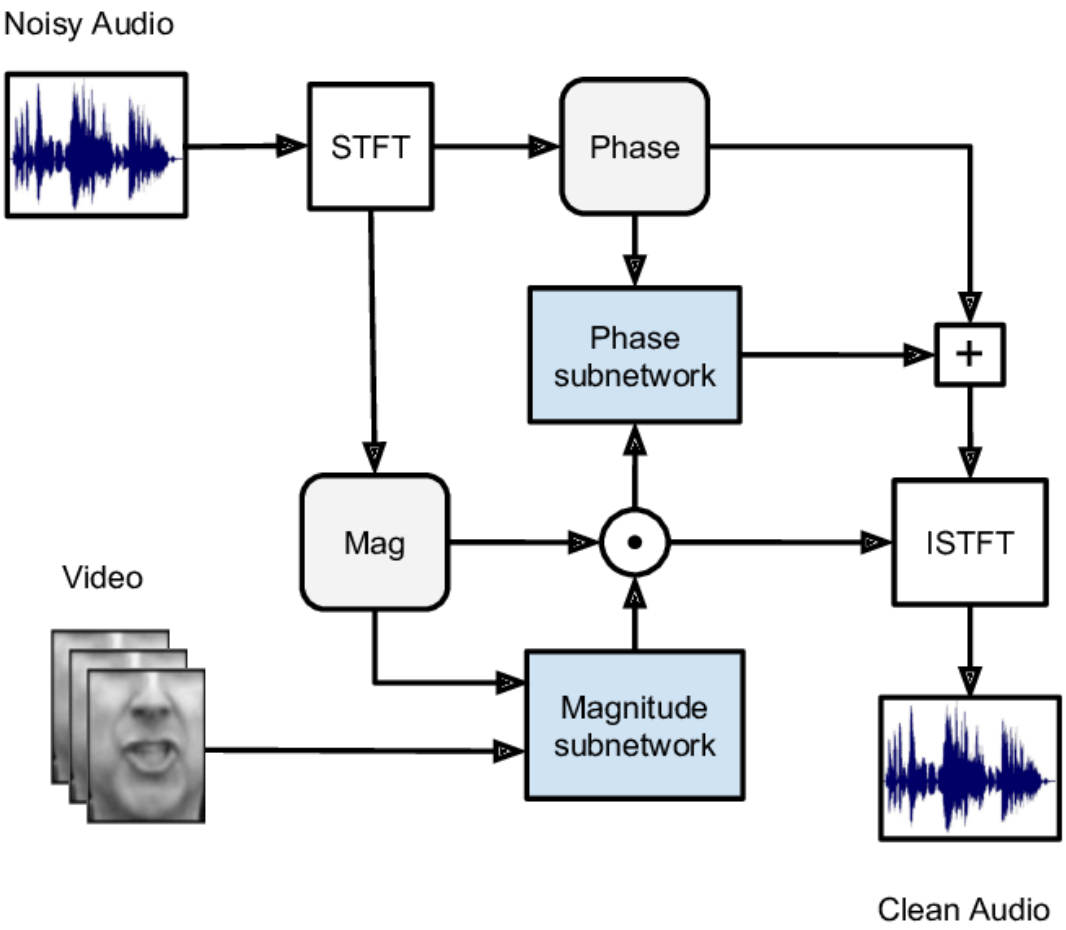


Figure 2.1: Speech Enhancement with Magnitude and Phase

The proposed architecture is novel in the way that it consists of two modules: a magnitude subnetwork and a phase subnetwork. First, a noisy signal undergoes STFT to produce two spectrograms, one for magnitude and the other for phase. Each noisy spectrogram will be received by their respective subnetworks. With noisy video as the secondary input, the magnitude subnetwork outputs a soft mask, which is then multiplied with the noisy magnitude spectrogram (element-wise) to produce an enhanced magnitude spectrogram. Next, the enhanced magnitude spectrogram along with the noisy phase spectrogram is fed into the phase subnetwork to produce a phase residual. Afterwards, the residual is added to the noisy phase spectrogram, which generates an enhanced phase spectrogram. Finally, both enhanced magnitude and phase spectra undergoes iSTFT to be transformed into clean audio waveform.

2.6 On/off-screen audio source separation

Another concurrent research conducted by Owens and Efros (2018) suggested that when audio and visual events occur simultaneously, there is a possibility both events originate from a single event. A neural network was first trained with self-supervised learning to determine whether the audio signals are temporally aligned to the video frames. The representation has numerous applications, namely sound source localization, audiovisual action recognition and on/off-screen audio source separation. Since this project emphasizes heavily on speech separation, the following paragraphs shall discuss mainly on the on/off-screen audio source separation.

A 3D multisensory CNN is proposed for the fused audiovisual network, in addition to having an early-fusion design. It is hypothesized that the early fusion of both modalities is crucial in predicting whether audio sounds and video frames are temporally aligned, before proceeding to the task of audio source separation. The input is represented as a Time x Height x Width volume. Firstly, the visual stream undergoes temporal downsampling through a series of 3D convolution and pooling operations. Conversely, the audio stream is processed by a series of strided 1D convolution operations. At this point, both modalities will have the same sampling rate. The learned features will be concatenated to form a multisensory representation that undergoes additional 3D convolution and pooling operations.

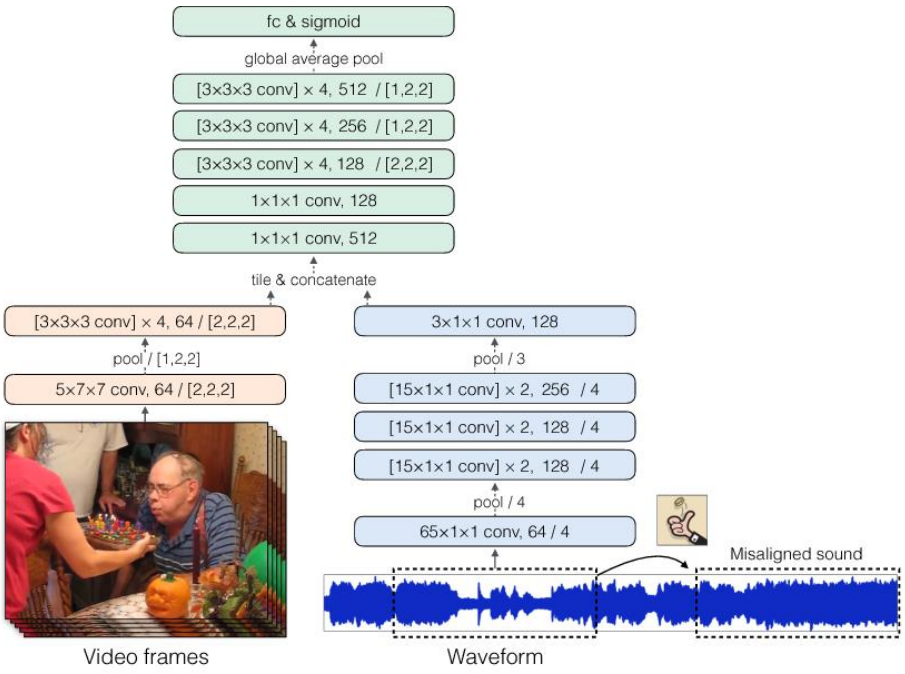


Figure 2.2: Network Architecture of Multisensory CNN

By visually masking different speakers in a video (a single visible speaker in a two-speaker video), the model is able to demonstrate the separation of on- and off-screen sounds, effectively preserving the speech signal of the visible speaker. The model performs audio source separation on raw video; no data preprocessing or labelling is required. The typical audiovisual network is augmented with a *u*-net encoder-decoder, which maps the mixed audio signals according to its on- and off-screen components. Linear interpolation is used to match video features with the audio sampling rate. Then, these video features are spatially mean-pooled and tiled over the frequency domain. Consequently, the Time x Height x Width output from the three-dimensional CNN is transmuted to fit into the two-dimensional encoder as Time x Frequency input. That way, the *u*-net can receive visual information properly. Due to the large number of frequency channels in the input spectrogram, a pair of convolutional layers is added to compensate for it. The model hereby predicts the magnitude and phase of the log-spectrogram.

2.7 Associate faces with voices

In the early development stages for audiovisual schematics, there was a straightforward problem to tackle: how does the deep learning algorithm associates a particular audio source with a particular face in any given video? The challenge of this problem is threefold: the diversity of speakers, the appearance of multiple people and awkward camera angles relative to the speaker's face. Hoover et al. (2017) is able to propose a system that is speaker and environment independent, meaning that no a priori information is required, such as number of speakers or spatial signals. The system is able to achieve an accuracy of 71%, according to the literature.

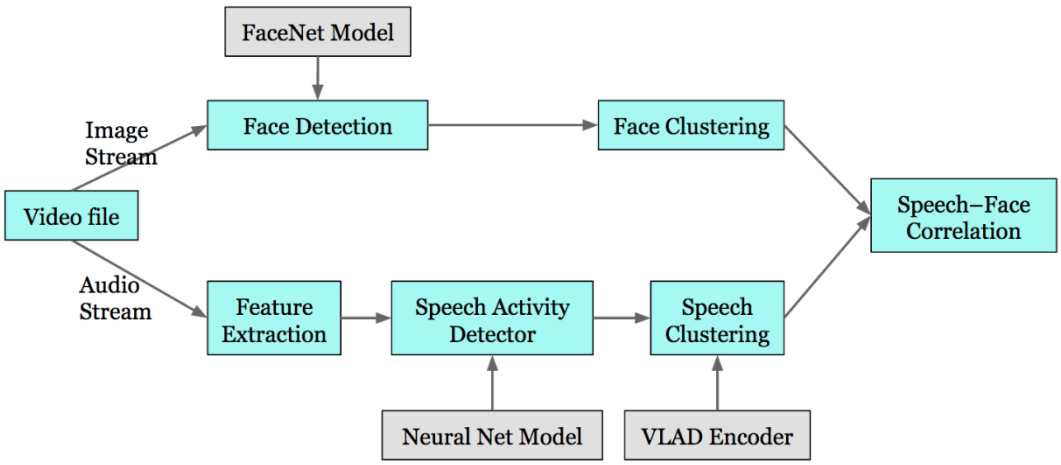


Figure 2.3: Audio-Visual Data Processing Pipeline

The combination of trivial audio and visual information extracted from a video could potentially introduce a significant indicator towards connecting speech signals to a speaker's face. The information extracted is defined as 'trivial in a sense that it is unable to identify the active speaker; yet the extraction techniques are state-of-the-art. The processing of image and audio streams are outlined in Figure 2.3 as separate branches. Both data streams will eventually combine in the rightmost process as output.

2.7.1 Image stream processing

The literature deploys a pre-trained CNN model called FaceNet (Schroff et al. 2015), which creates a 128-dimensional embedding for images that contains a recognizable face. FaceNet boasts a recognition accuracy of approximately a hundred percent on the LFW dataset. The CNN model is capable of detecting human faces by virtue of using a fixed threshold to cluster face points within a neighborhood. This cluster is also called a face cluster.

2.7.2 Audio stream processing

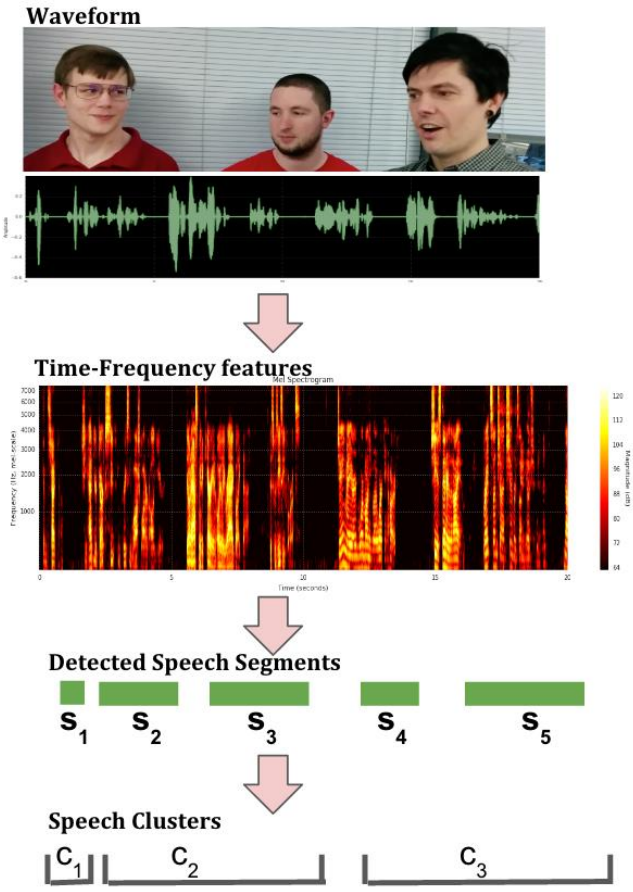


Figure 2.4: Stages of Audio Data Pipeline

For audio signals to serve as input to the downstream stages, it must be converted to the frequency domain, as shown in Figure XXX. A speech detection algorithm creates various speech segments through analyzing the time-frequency representation of audio signals that indicates speech activity. The speech segments are under the assumption that they solely belong to a single speaker. The next phase involves the process of clustering speech segments, which is better known as speaker diarization. The literature takes a conservative diarization technique – under-clustering, meaning more clusters than speakers. Firstly, a fixed size embedding is computed for each speech segment, which is then represented in the embedding space as a point through a technique known as Vector of Locally Aggregated Descriptors (VLAD) (Jegou et al. 2010). A hierarchical agglomerative clustering algorithm is then used on these speech segments, where a similarity threshold is determined to warrant conservative cluster merging.

2.7.3 Mapping voices to faces

In the final stage, information from the audio and visual stream are merged together to determine the speaker. It is assumed that face clusters from the visual stream are perfect, meaning that they contain only images of a single individual, and speech clusters from the audio stream also contain speech segments of a single individual, albeit not the same respective individual. The merging process is determined whether which face cluster often cooccurs with a speech cluster. For any given speech cluster, a map is created for the face cluster that appears the most in the corresponding video frames. After all speech clusters are mapped to its respective face clusters, this implies that the final stage is complete – mapping voices to faces.

2.8 Dataset difference and weakness

All existing audiovisual datasets only features a small number of speakers, showcase limited vocabulary, or not publicly available. For instance, the CUAVE dataset features 36 subjects pronouncing numerals ranging from zero to nine over and over, which amount to a total of 1800 audio samples. Secondly, the TCD-TIMIT dataset comprises of 60 subjects reciting sentences from the TIMIT dataset, which amount to around 200 videos for each subject. Thirdly, the Mandarin sentence dataset comprises of 320 video recordings of a native speaker reading sentences in the Mandarin language, which covers complete range of phonemic characters. Lastly, the Lip Reading Sentences dataset features diverse subjects and large vocabulary, yet the dataset is not publicly

available. To train a model that is truly robust and adaptive, a better dataset had to be compiled from scratch that is sufficiently large and diverse in order to achieve better speech separation performance.

2.9 Short-time Fourier Transform

Polar coordinates such as magnitude and phase are used to enhance the STFT of mixed speech, given equation:

$$S_{t,f} = |S_{t,f}|e^{i\theta_{S_{t,f}}}$$

where $|S_{t,f}|$ denotes the magnitude response, $\theta_{S_{t,f}}$ denote the phase response of STFT at time *t* and frequency *f*. The time-frequency unit of the STFT representation is a complex number. Magnitude and phase responses can be derived directly when a pair of real and imaginary components are given, as such:

$$|S_{t,f}| = \sqrt[2]{\Re(S_{t,f})^2 + \Im(S_{t,f})^2}$$

$$\theta_{S_{t,f}} = tan^{-1}\frac{\Im(S_{t,f})}{\Re(S_{t,f})}$$

Alternatively, STFT can be expressed with Cartesian coordinates as well, with the use of complex exponential expansion. The derivation of equations are as follows:

$$S_{t,f} = |S_{t,f}|\,cos\left(\theta_{S_{t,f}}\right) + i|S_{t,f}|\,sin\left(\theta_{S_{t,f}}\right)$$

$$\Re(S_{t,f}) = |S_{t,f}|\,cos\left(\theta_{S_{t,f}}\right)$$

$$\Im(S_{t,f}) = |S_{t,f}|\,sin\left(\theta_{S_{t,f}}\right)$$

Therefore, the magnitude and phase spectra can be enhanced implicitly through enhancement of the time-frequency unit of STFT representation (Williamson, Wang and Wang, 2016).

Chapter 3 Proposed Method/Approach

3.1 Methodology

To realize the project within projected timeframe, a methodology has been proposed.

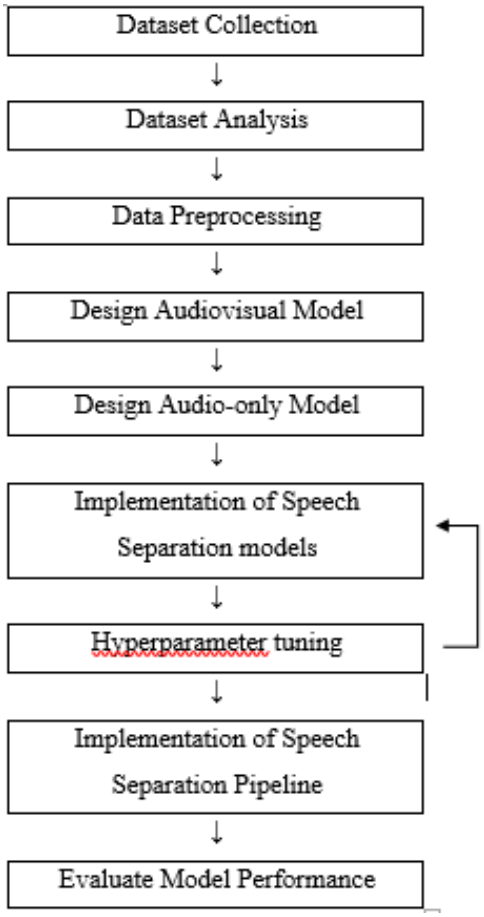


Figure 3.1: Research Methodology

During the early phases of project methodology, multiple datasets are downloaded from the Internet. The list of datasets is chosen according to the datasets used in the literature reviewed in Chapter 2, respectively. These datasets are usually clean, plus they are available to the public for model training of their own. Proceeding to the dataset analysis phase, where datasets go through a thorough screening process to determine whether they fit the training data criteria in term of audio clarity, language diversity, speaker count, speaker visibility, video resolution etc. Next, the data undergo preprocessing to fit requirements for training the deep learning model. Such process involves splitting the video into audio and visual streams, followed by data transformation and formatting.

Moving on to the design and implementation phase, an audiovisual model is designed using schematics from existing literature as references. The design for audio-only model follows shortly afterwards. With the architecture of both models all laid out, these models are implemented by training them with the aforementioned preprocessed data. After each implementation phase, the models will be evaluated using a test set prepared beforehand. If the outcome is not desirable, amendments will be made to both models, such as hyperparameter tuning or altering dataset size.

In the final phases of project methodology, a speech separation pipeline will be created to incorporate both models together to provide a novel functionality, as per stated in the project objectives. In order to assess the performance of the proposed speech separation pipeline, a pre-defined evaluation metric will be used to determine it. Additionally, the results will also be compared against other state-of-the-art speech separation techniques.

3.2 Tools and technologies used

In the course of this project, Python is selected as the programming language of choice for the proposed deep learning model. TensorFlow, an open source machine learning platform is used in this project as well. Through TensorFlow, the Keras functional API is called to provide deep learning libraries for creating neural networks. Several Python libraries are used for data preprocessing as well, such as *ffmpeg* and *youtube-dl* for downloading YouTube videos from the AVSpeech dataset, *librosa* for normalizing audio files, *MTCNN* for face detection and frame cropping, *keras-facenet* for creating face embeddings. Last but not least, visualization tools are taken from 2 GitHub repository, *Audio-visual_speech_separation_basic* and *Speech-denoise-Autoencoder*, for spectrogram plotting and signal visualization purposes.

3.3 Standard evaluation metrics for model performance

To determine whether the proposed deep learning model achieves satisfactory results, qualitative analysis will be conducted. The following evaluation metrics from the BSS Eval toolbox measure performance according to the energy ratios expressed in unit decibels (Vincent et al. 2006). It should be noted that the original source signals must be available to serve as ground truths.

(a) Signal-to-distortion ratio (SDR):

$$SDR := 10 \cdot log_{10} \frac{\left\| s_{target} \right\|^2}{\left\| e_{interf} + e_{noise} + e_{artif} \right\|^2}$$

(b) Signal-to-interferences ratio (SIR):

$$SIR := 10 \cdot log_{10} \frac{\left\| s_{target} \right\|^2}{\left\| e_{interf} \right\|^2}$$

(c) Signal-to-noise ratio (SNR):

$$SNR := 10 \cdot log_{10} \frac{\left\| s_{target} + e_{interf} \right\|^2}{\| e_{noise} \|^2}$$

(d) Signal-to-artifacts ratio (SAR):

$$SAR := 10 \cdot log_{10} \frac{\left\| s_{target} + e_{interf} + e_{noise} \right\|^2}{\left\| e_{artif} \right\|^2}$$

where s_{target} is the ground truth, e_{interf} is the interferences, e_{noise} is the additive noise and e_{artif} is the algorithmic artifacts. Before signal reconstruction, any undesirable signal that contaminates the desirable signal is considered as additive noise. After signal reconstruction, these noises will be reclassified as interference or artifacts. Interferences arise as a result of signal mis-separation, e.g. hearing residual background music after extracting vocals from a song. Artifacts occur by virtue of the reconstruction algorithm, e.g. flawed STFT phase estimation. Usually there will be trade-offs between the two.

By including certain terms at the numerator, the formula makes the calculations independent of that particular term. The presence of e_{interf} in SNR makes it independent of SIR. Likewise, the presence of e_{interf} and e_{noise} in SAR makes it independent of SNR and SIR. For the proposed model, SDR will be employed since functions as a global performance measure that takes all error terms into account. Considering that speech signals fluctuate across the temporal domain, the perceived separation quality will also vary accordingly. A finite length centered window is

defined and the local windowed signals are computed respectively. These values can be visualized through the plotting of a cumulative histogram.

Intuitively, input signal that has a low SDR should achieve a larger improvement, and vice versa. This is because signals with low SDR indicates that the ground truth signal is weak, relative to its error terms. Hence, the speech separation model by right should eliminate most of error terms, effectively strengthening the ground truth signal – in turn increasing the SDR.

3.4 Dataset

This project made use of AVSpeech; a large-scale audio-visual dataset specifically put together for the purpose of training speaker-independent audiovisual neural networks (available at https://looking-to-listen.github.io/avspeech/download.html). The dataset consists entirely of speech clips, free from any background interference. As a result, the visible face and audible sound in the video are said to correspond to a single individual.

The dataset comprises approximately 1500 hours' worth of video clips, each with varying length, ranging from 3 to 10 seconds long. Moreover, the dataset boasts diversity in speaker ethnicity, spoken language as well as face poses. The reason behind this is that the dataset is collected from YouTube channels of university lectures, TED talks, how-to videos etc. Video clips from such channels usually involve only a single speaker, in addition to having high video and audio quality.

The process of acquiring the dataset involves downloading two csv files from the AVSpeech repository, which represents the training and testing set data respectively. Both files contain video segment annotations, except that the training set contains around 270k videos while the testing set contains around 22k videos. The format of the csv files is as follows:

YouTube ID	Start segment	End segment	X coordinate	Y coordinate
u5MPyrRJPmc	108.24	111.24	0.849219	0.305556
H1ulMfj5wRY	112.32	116.94	0.1125	0.345833
GNRPRH-E-sI	30.2302	38.171467	0.333594	0.494444
VvcwAGkSy2o	240.2	253.366667	0.491667	0.372222
XoboRQKD-KY	219.088533	222.088533	0.239844	0.248611
hSN3RZh3iL8	54.429667	57.724511	0.525781	0.266667
bR80JEaW4LY	86.878456	89.964878	0.359896	0.325
O1cI99Imta4	72.8	75.8	0.295312	0.307407
ewkFvB89_AE	278.244	281.414	0.648438	0.287037
SGJz8ysQXlQ	160.026533	164.597767	0.472917	0.337037

Figure 3.2: Dataset Attributes and Instances

The first three attributes are used to manipulate the YouTube URL to determine the correct video and segment length. The coordinates serve as a reference point of the speaker's face in the video. When these videos are downloaded, their frame size are normalized so that the coordinate values remain in a fixed range. Finally, it should be noted that the testing set contains disjoint speakers.

3.5 Data Preprocessing

3.5.1 Visual data preprocessing

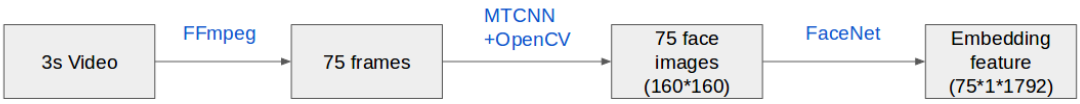


Figure 3.3: Visual Data Preprocessing Pipeline

Video clips are downloaded from the dataset through FFmpeg, an open source software that provides tools to process multimedia content. Given specific parameters, FFmpeg trims each video to be exactly 3 seconds long, having a frame rate of precisely 25 frames per second (FPS). The resulting frames for each video clip are 75 frames. Next, these frames are fed into a multi-task convolutional neural network (MTCNN) for face detection. A 160*160-pixel face image is extracted from each frame with the guidance of the face center coordinates provided in the AVSpeech dataset. Lastly, a pretrained facial recognition model - FaceNet is used to extract face embeddings for each of the

detected face images. These images are then transferred to a convolutional layer which is not spatially varying. The corresponds to the lowest layer in the dilated CNN. (1792) By doing so, the embeddings discard irrelevant variation between images such as illumination, while retaining relevant information for facial recognition.

3.5.2 Audio data preprocessing

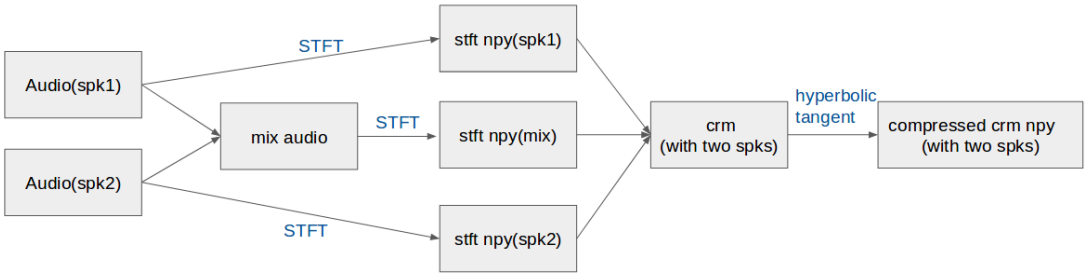


Figure 3.4: Audio Data Preprocessing Pipeline

On the other hand, STFT is calculated for the video soundtrack, which is also partitioned into 3-second audio segments. STFT splits the audio signal into shorter segments, so that signal properties can be analyzed at a particular point in time. The evolution of audio frequency over time can also be scrutinized with finer detail. The resulting output is a matrix representation that contains complex numbers. Since complex-valued output cannot be visualized directly, a log function is applied on the output to produce a heat-map, which is also known as a spectrogram. In more layman's terms, computing STFT of the input audio signal will produce a spectrogram image for the dilated CNN.

3.5.3 Synthetic speech mixtures

For the purpose of generating a two-speaker speech corpus, a pair of clean speech is selected from the AVSpeech dataset, each featuring different speakers. By merging two waveforms into a singular waveform, a synthetic speech mixture is produced, such that:

$$Mix_i = AVS_j + AVS_k$$

where AVS_j and AVS_k are two separate, clean speech samples, Mix_i is the resultant synthetic speech mixture. Variant of the dataset can be created through further addition of non-speech background noise (obtained from the AudioSet dataset) into the speech mixtures, such that:

$$Mix_i = AVS_j + AVS_k + 0.3 * AS_l$$

where AS_i represents non-speech audio, with its amplitude scaled-down to replicate the impression of background noise. Nonetheless, each dedicated model requires its own unique speech corpus. This implies the production of different combinations of speech mixture and non-speech background noise to cater for each model. For instance, a model that takes 3 visual streams for input demands the speech mixture to contain 3 speakers, so on and so forth. The addition of background noise is an optional choice.

3.6 Network Architecture

Figure 3.5 provides a general overview of the proposed network. Each of the network modules will be further elaborated in detail. For the sake of clarity, the network will have dual visual streams for two visible speakers in any given video. The same weights will be shared across convolutional layers between all visual streams.

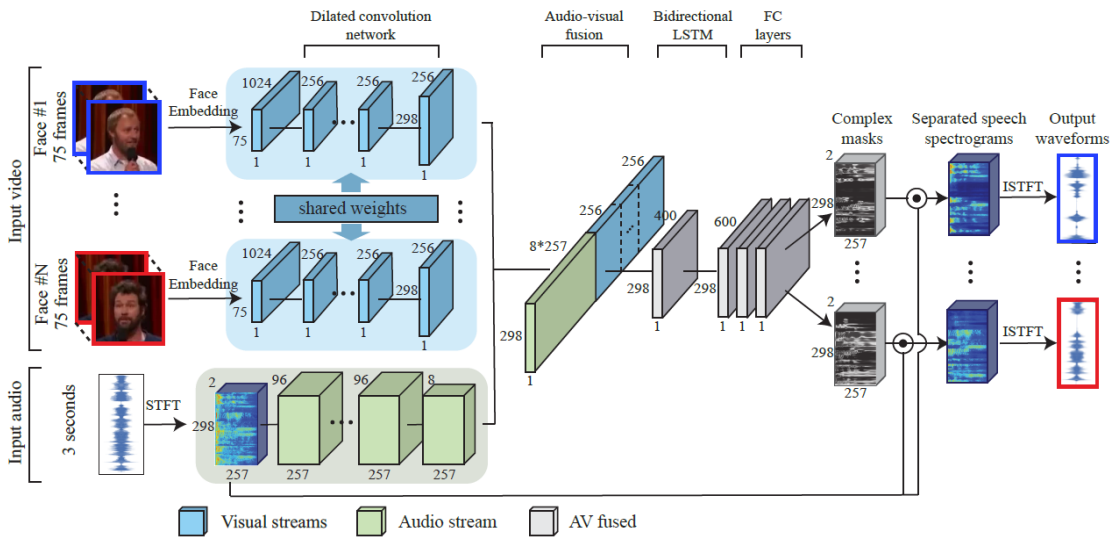


Figure 3.5: Proposed Audio-Visual Network Architecture

Since there's two data streams (audio and visual) in the network, two separate dilated convolution networks are used to learn their respective features and representations. Both networks differ from each other in terms of layers and parameters.

Data processing is different across modalities too. Visual data undergo spatial convolutions and dilations over the temporal axis to better determine when the speakers are actually speaking. On the other hand, audio stream is processed as-is since its dimensions are lesser. A difference exists between the sampling rate of audio and video

signal. In order to complement the spectrogram sampling rate, the visual stream output will be upsampled to 100Hz. Interpolation is performed on each visual feature in the temporal dimension using simple nearest neighbor algorithm. Afterwards, a joint audiovisual representation is created through the concatenation of learned audio and visual features. Subsequently, a bidirectional LSTM and three FC layers further process the joint audiovisual representation. The squared error (L2) serves as the loss function between clean spectrogram and the mask-filtered spectrogram. .

3.7 Expected Output

In the end, the model will output a multiplicative time-frequency spectrogram mask, that clearly describe the relationship between target speech to its background noise. If multiple speakers are detected in the video, the model will output masks for each respective speaker. The output mask refers to a complex ratio mask (cRM). The ultimate goal of the model would be to estimate a mask that closely resembles the complex ideal ratio mask (cIRM), which is capable of reconstructing clean speech from mixed speech perfectly through complex multiplication (this means that the multiplication involves real and imaginary components). Afterwards, iSTFT is performed onto the clean spectrogram to obtain its denoised waveform.

3.8 Audiovisual model specifications

The audio stream goes through a total of 15 convolutional layers, whereas the visual stream goes through a total of 6 convolutional layers. Additionally, a Rectified Linear Unit (ReLU) layer is interweaved between each convolutional layer. Batch normalization is also performed after each convolutional layer. Dropout is not integrated into the architecture under the impression that a large dataset is used, thus the model would not suffer from overfitting. Batch normalization, in theory, should suffice for regularization effects.

3.9 Preprocessing specifications

Audio files collected from the dataset are resampled to 16kHz, STFT is performed using FFT size of 512, hop length of 10ms and Hann window of length 25ms, which produces an audio feature of 257 x 298 x 2 scalars. Furthermore, input/output audio spectrogram A will undergo power-law compression with the formula $|A|^p$ with p = 0.3. As mentioned before, the input visual stream will receive a total of 75 face embeddings,

extracted from 3 second videos with 25 FPS. The discrepancy between face embeddings will be compensated through a process of elimination or replication.

3.10 Loss function

To compute the difference between actual and predicted targets, a loss function is defined as below:

$$J_{MSE} = \frac{1}{2}\sum_{t=1}^{T}(\|\tilde{y}_{1_t} - y_{1_t}\|_2^2 + \|\tilde{y}_{2_t} - y_{2_t}\|_2^2)$$

where y_{1_t} and y_{2_t} represent the inputs (clean speech), $\tilde{y}_{1_t}$ and $\tilde{y}_{1_t}$ represent the predicted outputs (enhanced speech) and $t = 1, \dots, T$, where T is the sequence length.

The DNNs minimize the loss function when targets have similar spectra. However, the approach leads to reduce SIR during model testing, since ambiguous spectral features will 'bleed' partially across inputs (Huang et al. 2015). The issue is resolved by adding a regularization constant to the loss function.

The formula for the custom loss function can be elaborated as:

$$J_{MSE} = \frac{1}{2}\sum_{t=1}^{T}(\|y_{1_t} - \tilde{y}_{1_t}\|^2 + \|y_{2_t} - \tilde{y}_{2_t}\|^2 - \gamma\|y_{1_t} - \tilde{y}_{2_t}\|^2 - \gamma\|y_{2_t} - \tilde{y}_{1_t}\|^2)$$

where γ represents the regularization constant. The custom loss function penalizes interference from other inputs, allowing higher SIR to be achieved while maintaining SDR and SAR.

3.11 Derivation of complex ideal ratio mask

In research conducted by Williamson et al. (2016), given the complex spectrum of clean speech $S_{t,f}$ and noisy speech $Y_{t,f}$, the complex ideal ratio mask $M_{t,f}$ is derived as follows:

$$S_{t,f} = M_{t,f} * Y_{t,f}$$

where '*' denotes complex multiplication, taking into account the fact that the terms above are complex numbers. These terms could also be expressed in rectangular form:

$$S = S_r + iS_i$$

$$M = M_r + iM_i$$

$$Y = Y_r + iY_i$$

where subscripts *r* denotes a real number and *i* denotes an imaginary number. Although time *t* and frequency *f* are not defined in the notation, but the equations are subjected to each time-frequency unit. Using the above equations, the real and imaginary components of clean speech can be derived as such:

$$S_r + iS_i = (M_r + iM_i) * (Y_r + iY_i)$$
$$S_r = M_r Y_r - M_i Y_i$$
$$S_i = M_r Y_i + M_i Y_r$$

The complex ideal ratio mask's real and imaginary components can be obtained with further derivations:

$$M = \frac{Y_r S_r + Y_i S_i}{{Y_r}^2 + {Y_i}^2} + i\frac{Y_r S_i - Y_i S_r}{{Y_r}^2 + {Y_i}^2}$$
$$M_r = \frac{Y_r S_r + Y_i S_i}{{Y_r}^2 + {Y_i}^2}$$
$$M_i = \frac{Y_r S_i - Y_i S_r}{{Y_r}^2 + {Y_i}^2}$$

It is possible that values of the real and imaginary components range from $(-\infty, \infty)$, since M_r and $M_i \in \mathbb{R}$. Thus, the complex ideal ratio mask is compressed with a hyperbolic tangent, as shown below:

$$cIRM_x = K\frac{1 - e^{-C \cdot M_x}}{1 + e^{-C \cdot M_x}}$$

where *x* denotes either real or imaginary components. The compression bounds the mask values within $[-K, K]$ while C controls its steepness. Williamson et al. conclude the selection of values $K = 10$ and $C = 0.1$ due to its empirical performance. When recovering an uncompressed mask estimate, the following inverse function is applied on the DNN output O_x, as shown below:

$$\widehat{M}_x = -\frac{1}{C}\log\left(\frac{K - O_x}{K + O_x}\right)$$

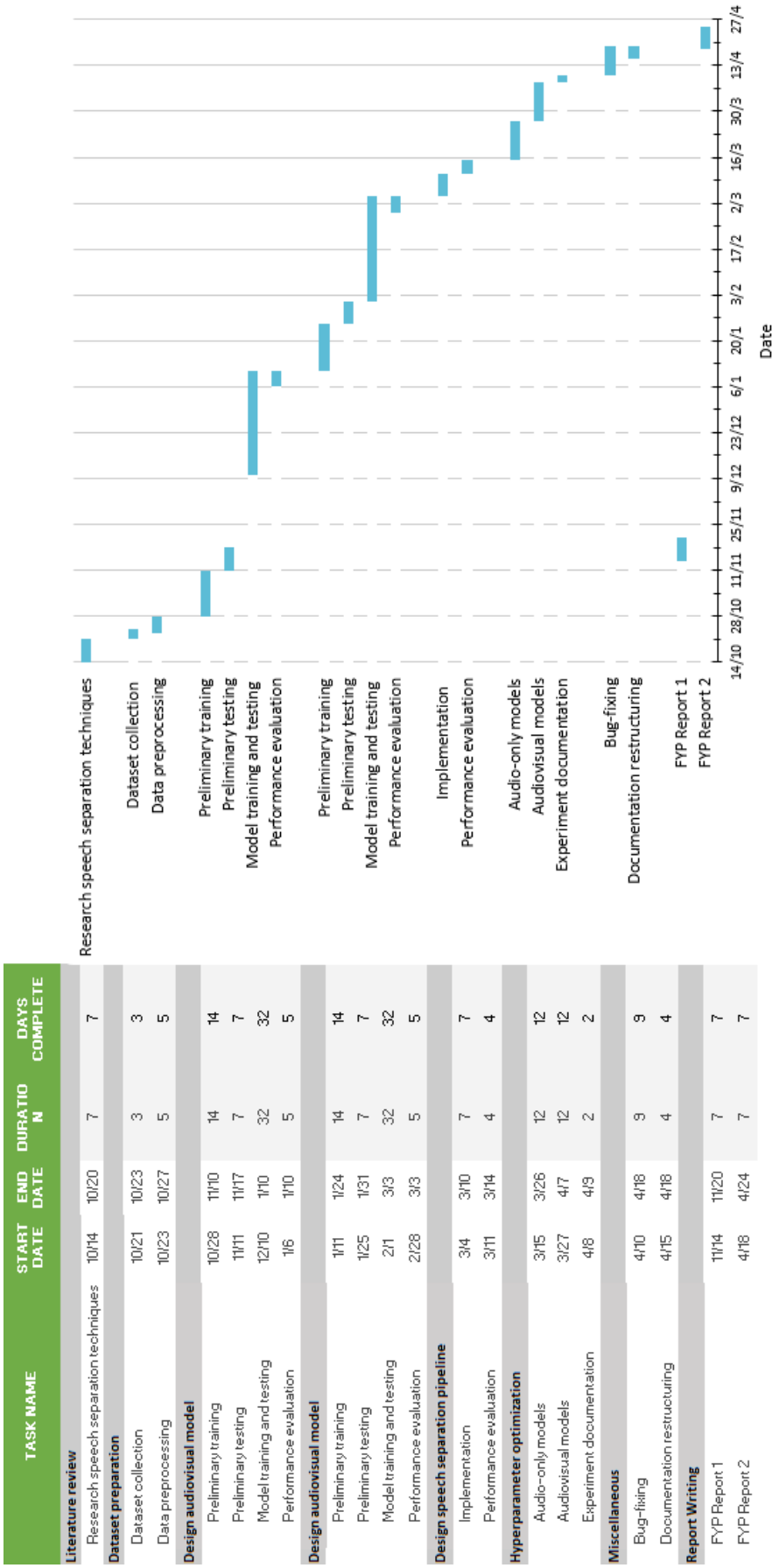
3.12 Implementation issues and challenges

Several issues had arisen during the implementation phase of the neural networks. In an attempt to replicate results described by Ephrat et al. (2018), the same dataset employed in the literature was selected to eliminate possible variables in training configurations. However, the dataset contains thousands of hours of video segments,

just by training the neural network alone could take up to several weeks of computations. This prolonged method of training does not align with the project timeline; hence it is deemed infeasible.

Secondly, the movement of data across the speech separation pipeline is speculated to not be streamlined. The first stage of the pipeline describes the neural network receiving input from audio and visual streams, perform mask estimation, then output a number of audio files corresponding to the number of speakers, plus an additional audio file for background interference. At the second stage, only the audio file for background interference should be fed into the neural network for further computations. Additional conditions have to be set in place to ensure that the correct audio files are selected for the second stage, while the rest of the audio files are appended to the final output file.

3.13 Timeline



TASK NAME	START DATE	END DATE	DURATION	DAYS COMPLETE
Literature review				
Research speech separation techniques	10/14	10/20	7	7
Dataset preparation				
Dataset collection	10/21	10/23	3	3
Data preprocessing	10/23	10/27	5	5
Design audiovisual model				
Preliminary training	10/28	11/10	14	14
Preliminary testing	11/11	11/17	7	7
Model training and testing	12/10	1/10	32	32
Performance evaluation	1/6	1/10	5	5
Design audiovisual model				
Preliminary training	1/11	1/24	14	14
Preliminary testing	1/25	1/31	7	7
Model training and testing	2/1	3/3	32	32
Performance evaluation	2/28	3/3	5	5
Design speech separation pipeline				
Implementation	3/4	3/10	7	7
Performance evaluation	3/11	3/14	4	4
Hyperparameter optimization				
Audio-only models	3/15	3/26	12	12
Audiovisual models	3/27	4/7	12	12
Experiment documentation	4/8	4/9	2	2
Miscellaneous				
Bug-fixing	4/10	4/18	9	9
Documentation restructuring	4/15	4/18	4	4
Report Writing				
FYP Report 1	11/14	11/20	7	7
FYP Report 2	4/18	4/24	7	7

Figure 3.6: Project Gantt Chart

Chapter 4 Model Implementation

4.1 Experimental setup

The proposed audio-only and audiovisual models will be trained on a workstation operating on Ubuntu 18.04.3 LTS, equipped with two GPUs (Nvidia GeForce RTX 2080 Ti). All Python scripts are executed through the terminal without needing to specify additional parameters to the scripts used.

Firstly, *AV_train.py* sets the foundation for the audiovisual model. The script links necessary training and validation datasets to be fed into the model, as well as providing custom parameters and hyperparameters for tweaking purposes. After setting all the required configurations, the script compiles and trains the model via the Keras functional API.

These are the default hyperparameters that will be used throughout the training phase:

i. Number of epochs: 100
ii. Number of speakers: 2
iii. Batch size: 2
iv. Optimizer: Adam
v. Gamma loss: 0.1
vi. Beta loss: 0.2
vii. Learning rate:
    a. 5e-5 for audio-only model
    b. 1e-5 for audiovisual model
viii. Learning rate scheduler: Reduce learning rate by a factor of 5 per 10 epochs

The proposed audiovisual model only supports isolation of 2 visible speakers in any given video clip. If speech separation were to be performed on video clips that involved more speakers, a dedicated model should be trained separately. The same restrictions also apply to the audio-only model. The modifications can be made easily by tweaking the ‘number of speakers’ hyperparameter before model training.

4.2 Building training and validation dataset

Different variants of synthetic speech mixtures can be created by specifying the ‘number of speakers’ parameters in the Python script *build_audio_database.py*. The

script executes a series of functions in order to compile the training and validation dataset. In this project, the number of speakers is set to 2.

Firstly, all downloaded audio files undergo power-law compression, STFT as well as expansion of complex data into true real and imaginary numbers, essentially transforming them into *numpy* file format (.npy) to be stored as ground truth/reference source. In relation to the number of speakers, equal-partitioning is performed for the audio files accordingly. Subsequently, permutations of the audio files are generated across each partition.

For further clarification, imagine a dataset consisting of 100 .npy files to be partitioned for a 2-speaker model training dataset. As a result of equal-partitioning, the first partition consisted of the first 50 files, while the second partition consisted of the last 50 files. The *nth* file from the 1st partition will be mixed with the *mth* file from the 2nd partition, where $n = 1, \dots, N$ and $m = 1, \dots, M$ (N, M denotes the partition size) to form the synthetic speech mixtures. Additionally, STFT spectrograms for the ground truths and synthetic speech mixtures are generated as well.

Proceeding to the next step, the script computes the cRM for both the ground truths and synthetic speech mixtures. Using STFT spectrograms generated from the previous step, the real and imaginary components of the cRM are computed and stored accordingly. Finally, the whole dataset is shuffled and split into training and validation set with a ratio of 9:1. The sets are stored in the form of text file, namely *dataset_train.txt* and *dataset_val.txt*.

Initially, a total of 600 video clips are selected from the AVSpeech dataset for download and pre-processing. However, only 328 video clips are eventually used to build the training and testing dataset, due to unforeseen reasons such as (a) the video link is invalid/video no longer available; (b) the speaker is not visible throughout the video clip; (c) the video clip ends prematurely. According to the above restrictions, the remaining 328 video clips are split into two partitions to perform mixing. The resultant dataset consists of $164\ x\ 164 = 26896$ video clips, and will be used to train the

finalized speech separation models. As a side note, a smaller dataset that consists of $19\ x\ 164 = 3116$ video clips is generated for the purpose of conducting experiments.

4.3 Spectrogram analysis

Spectrograms are intermediate data that the speech separation model uses to perform forward propagation, backpropagation, and to calculate the loss function and prediction accuracy. In this section, the spectrograms are visualized to give a better understanding on how the model works, and to demonstrate that the project objectives of building the speech separation models had been fulfilled.

Before forward propagation, clean audio signals will be transformed into spectrograms via STFT. In the proposed model, there will be two spectrograms for each train sample. Figure 4.1 illustrates the audio spectrograms that will be fed into the model as input and reference sources. The leftmost figure represents the mixed speech (input), the center figure represents clean speech for the 1st speaker and the rightmost figure represents clean speech for the 2nd speaker (reference sources).

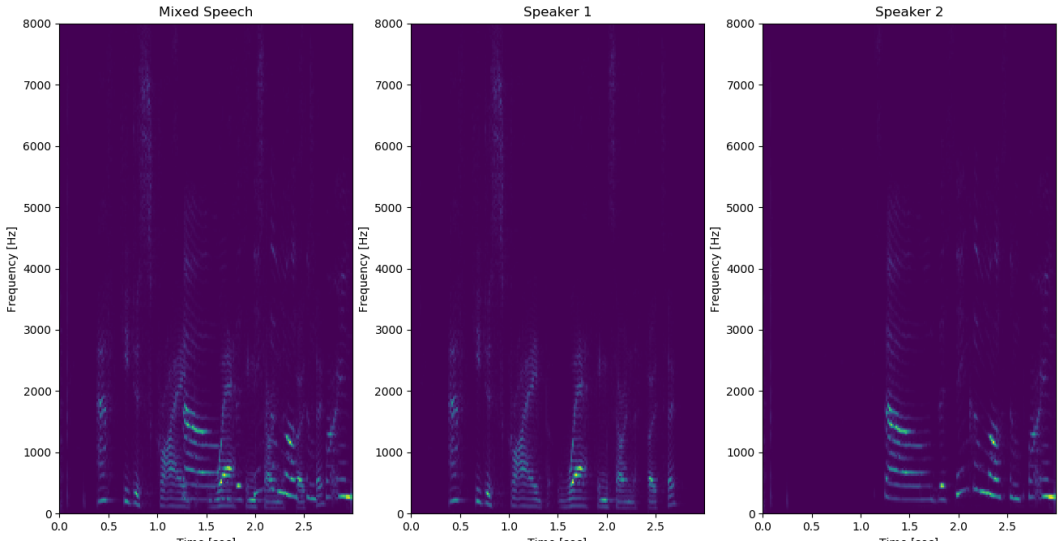


Figure 4.1: Spectrograms of mixed speech and two clean speeches

During forward propagation, the model performs prediction to generate the estimated audio spectrograms based on the mixed speech spectrogram. Figure 4.2 represents the estimated audio spectrograms, which will be compared with the reference sources to calculate the training loss and perform back-propagation. The leftmost figure depicts the mixed speech (same as Figure 4.4), the center figure depicts estimated speech for the 1st speaker and the rightmost figure depicts estimated speech for the 2nd speaker (estimated sources).

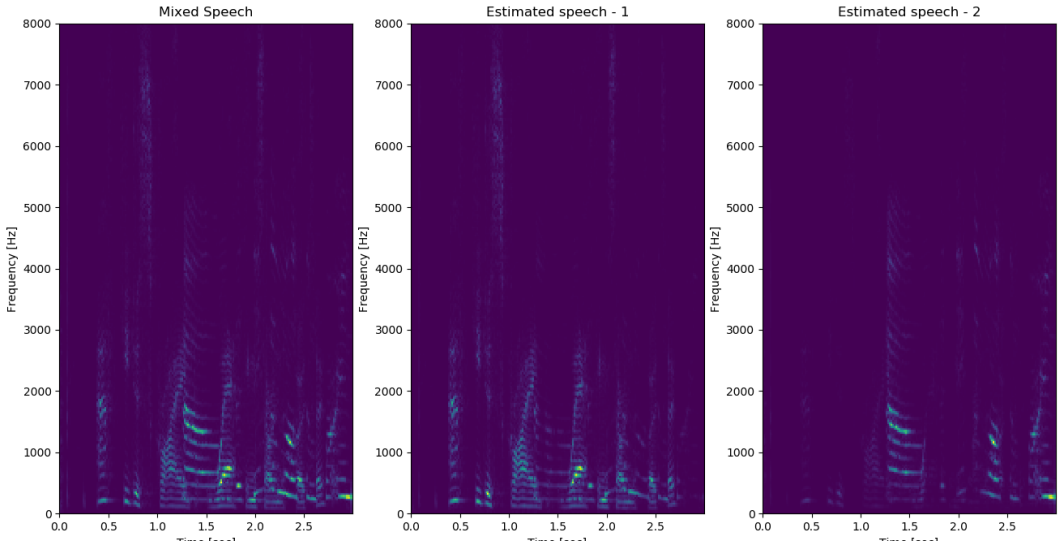


Figure 4.2: Spectrograms of mixed speech and two estimated speeches

In conclusion, the model is operating as intended, meaning that it can learn to separate the mixed speech spectrogram to produce two estimated speech spectrograms. However, the model still requires more finetuning since the estimated audio signals did not reach satisfactory levels yet.

Chapter 5 Experiments and Results

5.1 Hyperparameter optimization

This section elaborates on finding out the best configuration of hyperparameters for the speech separation pipeline models. The experiment is setup by training different variations of the speech separation model, then comparing their relative performance via loss, accuracy and pre-defined evaluation metric. For the control group, a baseline model is trained according to its default hyperparameter configurations, while for the experimental groups, the models are trained by tweaking a single hyperparameter for each experimental group. This implies that each modification of the baseline model will result in its unique training process. All models in this experiment are trained on the same dataset to eliminate undesirable variables, which refers to the manually crafted, smaller dataset mentioned in Chapter 4.

The hyperparameter list is compiled according to the hyperparameter's perceived impact on model performance. The chosen hyperparameters are:

- learning rate
- mini-batch size
- optimization algorithm

Hyperparameters that normally does not need tuning, such as Adam hyperparameters β_1, β_2 will be kept constant for this experiment. Aside from that, model parameters such as number of layers and number of hidden units will be excluded from the hyperparameter list as well. The experiment deploys one of the main strategies for searching the best hyperparameter configuration – Grid Search. It is a naïve approach that simply evaluates every possible hyperparameter configuration. However, this approach suffers from the curse of dimensionality, thus the searching space is limited to at most 2 dimensions to simplify the process.

The standard Grid Search workflow involved defining a N dimension grid for each configurable hyperparameter and the range of possible values for these hyperparameters, followed by searching through all the possible configurations to determine the best hyperparameter combination.

5.2 Optimization findings for audiovisual model

Table 5.1 below states the selected range of values for each hyperparameter, as well as the training loss and prediction accuracy for each audiovisual model variation. The first row represents the default hyperparameter configuration for model training. The following rows represent the variations, which are highlighted in yellow. The best performing model (according to training loss and prediction accuracy) will be highlighted in green.

ID	Learning Rate	Mini-batch size	Optimizer	Loss	Accuracy	Epoch
1	1e-5	2	Adam	0.48189	0.6291	16
2	1e-4	2	Adam	0.39610	0.6793	20
3	1e-6	2	Adam	0.57667	0.5718	19
4	1e-5	1	Adam	0.70369	0.5106	1
5	1e-5	4	Adam	0.48169	0.6282	20
6	1e-5	2	SGD	1.04144	0.5002	14
7	1e-5	2	RMS	0.48432	0.6282	16

Table 5.1: Comparison between audiovisual model variations

A higher learning rate allows the model to converge quicker, converge continuously and achieve a higher accuracy to outperform its competition. Convergence rate for the other model variations either started to stagnate after a few epochs of training, or fluctuated between a narrow range. This implies that they are not suitable for the task of speech separation, or there are better hyperparameter alternatives in comparison. Even so, there are several notable mentions, the mini-batch size of 4, as well as the RMS optimizer are able to achieve similar training loss and prediction accuracy in contrast with the default configuration.

5.3 Graph analysis for audiovisual model

Figures 5.1, 5.2 and 5.3 below depicts the training loss over epochs between the default hyperparameter configuration and its variations to visualize how different hyperparameters will affect audiovisual model convergence.

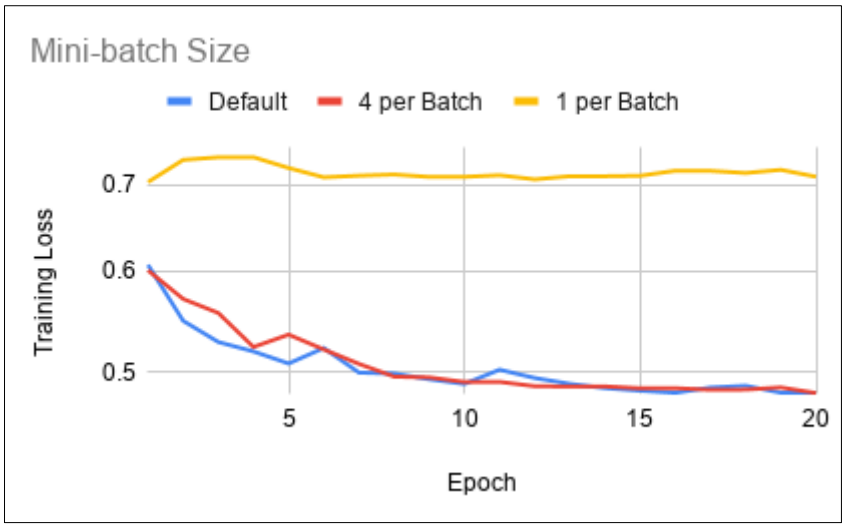


Figure 5.1: Training loss respective to mini-batch size (audiovisual)

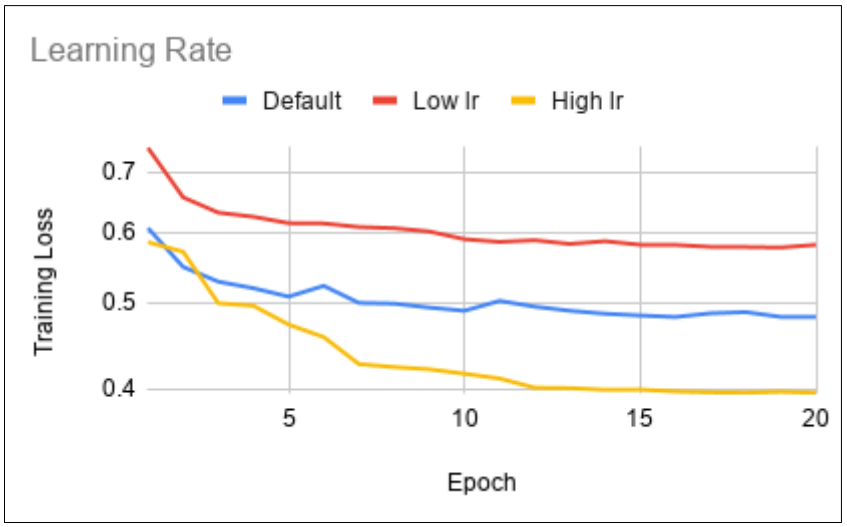


Figure 5.2: Training loss respective to learning rate (audiovisual)

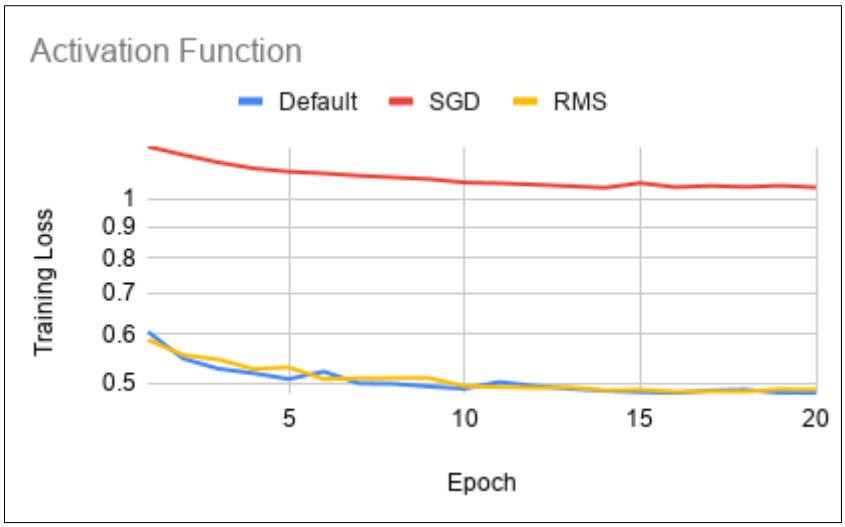


Figure 5.3: Training loss respective to activation function (audiovisual)

The trend in mini-batch size and the activation function graphs are rather similar, each comprised of one hyperparameter that failed to converge to the optimum, as well as one hyperparameter that obtained a similar model convergence trajectory with the default configuration. Hence, information from these graphs carry limited weight in deciding which hyperparameters are better.

However, in the learning rate graph, the drop for training loss is different for each learning rate. The line with the high learning rate had a steeper decline slope compared to the default and low learning rate. Regarding the project's speech separation method, setting a learning rate of 1e-4 is arguably better for model convergence, rather than a learning rate of 1e-5 or 1e-6.

5.4 Optimization findings for audio-only model

Table 5.2 below also states the selected range of values for each hyperparameter, as well as the training loss and prediction accuracy for each audio-only model variation. The table structure and format remain unchanged.

ID	Learning Rate	Mini-batch size	Optimizer	Loss	Accuracy	Epoch
1	5e-5	2	Adam	0.40793	0.6720	20
2	5e-4	2	Adam	0.43073	0.6578	20
3	5e-6	2	Adam	0.47421	0.6311	20

4	5e-5	1	Adam	0.41845	0.6657	20
5	5e-5	4	Adam	0.42931	0.6595	20
6	5e-5	2	SGD	0.79876	0.5001	20
7	5e-5	2	RMS	0.40815	0.6714	20

Table 5.2: Comparison between audio-only model variations

Setting a high learning rate for model training this time did not achieve the rate of convergence and prediction accuracy akin to the audiovisual model. In addition to that, the audio-only models are able to converge up until the final epoch, unlike the audiovisual models. The model with default configuration and the model with RMS optimizer had seemingly attained greater results in contrast to the other model variations. Since the training loss and prediction accuracy between both models are extremely identical, the mere difference in the training loss is said to be negligible. Lastly, the underperforming models may also indicate that there are better hyperparameter alternatives to choose from.

5.5 Graph analysis for audio-only model

Figures 5.4, 5.5 and 5.6 below also depicts the training loss over epochs between the default hyperparameter configuration and its variations to visualize how different hyperparameters will affect audio-only model convergence. The sequence of figures remains unchanged.

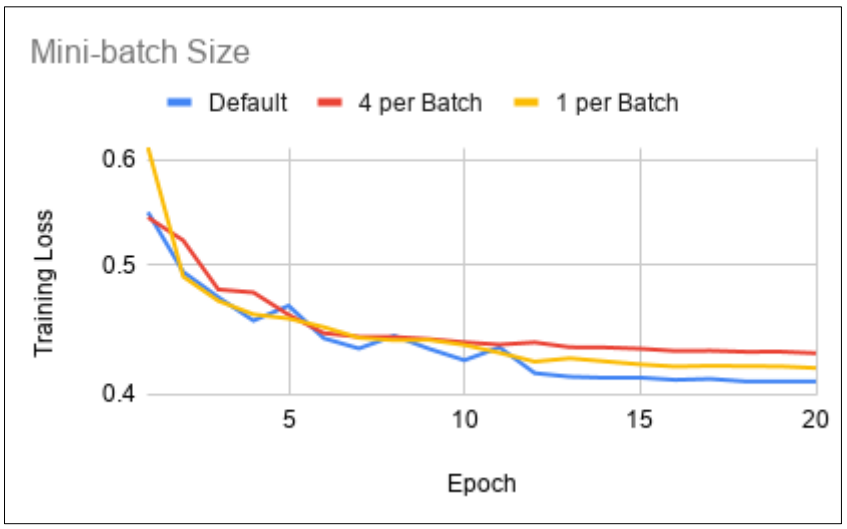


Figure 5.4: Training loss respective to mini-batch size (audio-only)

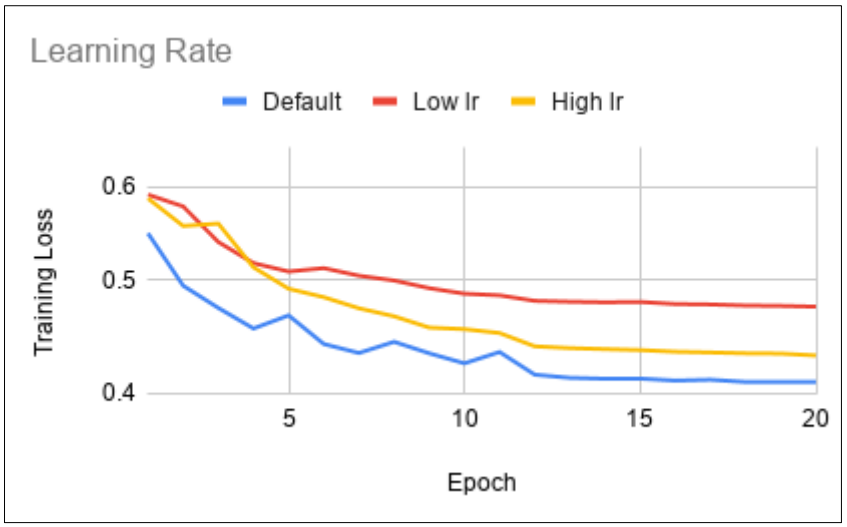


Figure 5.5: Training loss respective to learning rate (audio-only)

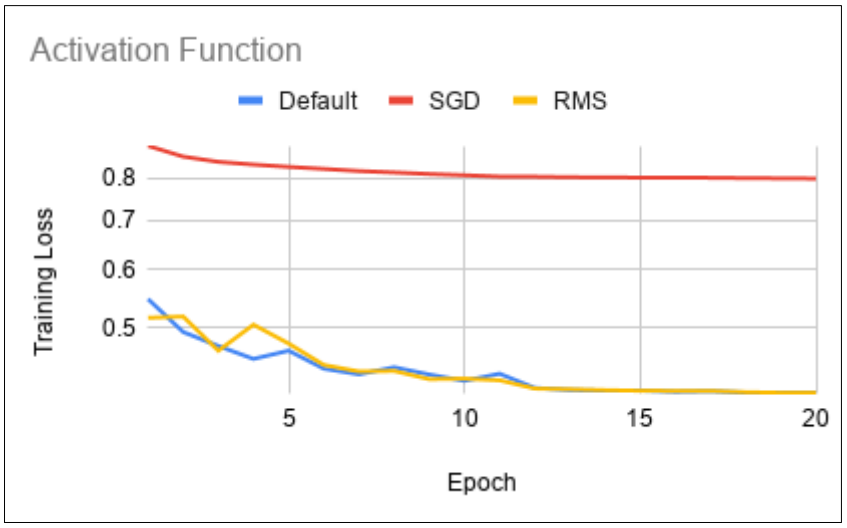


Figure 5.6: Training loss respective to activation function (audio-only)

According to the mini-batch size graph, the default batch size, which is 2 samples per batch, converged the most among the other batch sizes. The graph differs from its audiovisual counterpart in several ways. Firstly, the audio-only models are able to converge regardless of batch size, although this is not the case for audiovisual models. Moreover, manipulating the batch size resulted in different convergence rate. This implies that there exist certain nuances in training audiovisual model versus audio-only model when it comes to batch sizes.

The breakdown for the learning rate graph is rather straightforward, whereby the default learning rate of 5e-5 obtained the lowest training loss. The default learning rate is the better option, given the fact that other learning rates (5e-4, 5e-6) had a slower reduction of training loss over time.

Last but not least, the activation function graph for audio-only model closely resembles the corresponding graph for audiovisual model. The implementation of Adam or RMS optimizer in the context of speech separation brings little to no difference.

5.6 Building test data and its variants

After 100 epochs of training, the best-performing audiovisual speech separation model is used to perform inference on various single channel speech separation tasks, which comprise of unique mixtures of clean speech. The following paragraphs elaborate about the generation procedure for the test dataset.

To determine the robustness of the speech separation models, test data is constrained with a set of specifications. The evaluation phase will take the permutations between gender, spoken languages and accents into account for qualitative analysis. The process of generating train data and test data are identical, by retrieving the video and audio tracks from YouTube as the source. Table 5.3 below represents the specifications of test data.

ID	Category (of synthetic mixtures)
1	Same gender, language and accent
2	Same gender, language, but different accent
3	Same gender, different languages
4	Mixed gender, same language and accent
5	Mixed gender, same language but different accent
6	Mixed gender, different languages

Table 5.3: Test data specifications

As mentioned before, the training data for the speech separation models comprised of solely three-second long video clips. However, to improve the perceived clarity during evaluation, the synthetic mixtures were created by concatenating back-to-back three-

second long video clips to form a nine-second long video clip for each category. The rationale behind this is that from a listener's perspective, he/she might not be able to make sense of the underlying video context from a three-second long video clip. A nine-second long video clip is just enough for a listener to interpret spoken sentences in the synthetic mixtures. Similar to the training data, the test data are sourced from YouTube videos as well.

5.7 Calculate SDR score with MIR_EVAL

Subsequently, after model inference, the separated speech quality is evaluated by calculating the SDR improvement through an open-source Python library, *mir_eval*. The library's original purpose is to compute commonly-used metrics to measure performance of Music Information Retrieval (MIR) algorithms. However, there is an existing submodule dedicated towards source separation algorithm (adapted from BSS Eval toolbox). The functions within the submodule compare between extracted sources and reference sources to measure the perceptual quality of separated speech, instead of music.

Before moving on to the test data evaluations, a few experiments were conducted beforehand to establish a baseline for the SDR score, such as which range of values may indicate good improvements and vice versa. In order to compute the SDR score, the reference and estimated sources are required as input. Since the project emphasizes on speech separation between 2 mixed audio tracks, the inputs for each experiment will also contain 2 audio tracks. The resultant output from the algorithm is an array that encapsulates the SDR, SIR and SAR score for each audio track.

The first experiment employed the same audio track as the reference and estimated source to determine the upper limit for the SDR score. Conversely, the second experiment employed two distinct audio tracks as the reference and estimated source to determine the lower limit for the SDR score. Subsequent experiments used clean and predicted audio tracks sourced from a recent literature (Tian et al., 2019) as the reference and estimated source. The SDR score for state-of-the-art techniques shall establish a realistic baseline for the project. Table 5.4 shows the resultant SDR score for each experiment.

Experiment configurations	SDR score	
	Track #1	Track #2
Same audio track as estimated source	290.748	263.430
Distinct audio track as estimated source	-22.681	-23.485
Predicted audio track as estimated source (Sample #1)	6.400	15.034
Predicted audio track as estimated source (Sample #2)	6.385	10.717
Predicted audio track as estimated source (Sample #3)	4.130	6.293

Table 5.4: Baseline SDR score for state-of-the-art technique

From the experiments, the upper limit of SDR score is well off into the range of three-digit numbers, which occurs under the condition that the speech separation algorithm is able to emulate a perfect, clean split for any given mixed audio track. Meanwhile, the lower limit of SDR score is within the range of negative values, which implies that the two audio tracks are not correlated in any way.

Based on the subsequent experiments, the state-of-the-art speech separation algorithm (Deep Audio Prior) obtained SDR scores ranging approximately from 4 to 15. It is observed that the predicted audio tracks that are split from the same mixed audio track did not have the same SDR scores.

5.8 Qualitative analysis on synthetic mixtures

In the final phase of the project, the test data is fed into the trained audiovisual model to perform inference. This section is dedicated to fulfill the project objective of evaluating the performance of the proposed speaker-aware speech separation pipeline. For each synthetic mixture, the audiovisual model outputs two audio files that represent the estimated speech. Afterwards, the clean speech files and the estimated speech files are used to calculate the SDR scores.

For each language and accent permutations, 2 samples are inferred; for each gender combination (male-female, male-male, etc.), 6 samples are inferred. The entire testing dataset is used for evaluation, which sums up to 18 samples. In fact, since a single test sample is created through the concatenation of 3 train samples, the actual testing dataset size is precisely 54 samples. The tables below represent the calculated SDR scores for the synthetic mixtures.

Category	SDR score			
	Test sample #1		Test sample #2	
Same language and accent	-0.5466	-2.7088	-3.2558	-0.1202
Same language, different accent	-1.5273	-1.9252	-1.0208	-1.7508
Different languages	-0.2402	2.8471	-1.5555	0.5143

Table 5.5: SDR score for mixed-gender mixtures

Category	SDR score			
	Test sample #1		Test sample #2	
Same language and accent	-2.2120	-0.0851	-1.8216	-0.6644
Same language, different accent	-2.8402	-0.6627	-4.3587	0.2489
Different languages	1.8083	-3.5607	-0.7703	1.8609

Table 5.6: SDR score for same-gender (male) mixtures

Category	SDR score			
	Test sample #1		Test sample #2	
Same language and accent	-0.7847	-2.5193	0.3496	-3.0342
Same language, different accent	-1.7700	-1.1645	-1.1545	-2.2667
Different languages	0.9811	-4.2493	-2.9967	0.2021

Table 5.7: SDR score for same-gender (female) mixtures

According to the results above, the mixed-gender and same-gender mixtures all have the same trend for SDR scores. Firstly, the audiovisual model seemingly performed better speech separation when it comes to synthetic mixtures with different spoken

languages. However, for synthetic mixtures with the same spoken language, the audiovisual model is unable to perform a clearer speech separation. Secondly, the existence of different accents for the same spoken language did not help in gaining better performances by the audiovisual model. Thirdly, both the same-gender mixtures (male and female) had worse cumulative SDR scores compared to mixed-gender mixtures. This finding is supported by Delfarah and Wang (2017) and Hershey et al. (2016), which stated that model performance would plummet when separating same-gender speech mixtures.

5.8.1 In-depth discussion

Ultimately, the SDR scores for the synthetic mixtures fell below expected range of values. When comparing with the SDR baseline, the obtained results remains inconclusive. There are several possible reasons for this scenario:

- Insufficient amount of training time allocated to the speech separation models.
- Poor choice of hyperparameters and its range of values to train the speech separation models.
- The training dataset is unable to represent the task complexity for speech separation due to inadequate data diversity, which involves multiple gender, language and accent.

It is speculated that with a larger and more diverse dataset, it would better represent the complexity for tasks such as speech separation. Then, the proposed model would be eventually be able to perform gender-robust speech separation. Thus far, the test cases above merely perform speech separation on artificially mixed videos, and does not accurately reflect real-world scenarios. Even so, the results remained unsatisfactory. The original plan for comparison with previous works in audiovisual speech separation and enhancement were unable to proceed. Hence, the project objective of investigating state-of-the-art speech separation techniques and their relative performances was only partially fulfilled.

Chapter 6 Conclusion

6.1 Project review

The problem of speech separation has long plagued the domain of audio signal processing. Fortunately, recent breakthrough in deep learning had witnessed the use of audio and visual modality in convolutional neural networks for isolating and enhancing speech signals. Although these techniques are great at exploiting visual information to support speech separation tasks, they do not account for off-screen speakers. Hence, tackling speech separation become more sophisticated as visual information is not always available for every speaker. The motivation for addressing this problem stems from real-world scenarios where mixed speech in video clips would remain imperceptible although it only features one or two speakers. Enhanced speech from off-screen speaker(s) may potentially provide additional context towards interpreting social interactions.

This project proposed a speech separation pipeline that leverages the availability of visual information to perform speech isolation for visible speakers using convolutional neural network, then piping the residual audio signals into a deep neural network for another phase of speech enhancement, using a blind source separation algorithm.

Last but not least, due to certain limitations imposed on the speech separation models, the proposed project is unable to come to fruition. However, every phase of the project methodology had been completed, from the dataset collection phase to the model evaluation phase. Furthermore, additional experiments had shown that the trained model is able to separate mixed speech, albeit not as cleanly as intended. This implies that with more dedicated computational time, model performance is bound to improve.

6.2 Future work

Subsequent developments of the proposed model comprise of performance evaluation against modern speech separation techniques. The proposed model may be trained and tested on several established open-source datasets such as the GRID corpus, Mandarin sentences corpus, TCD-TIMIT dataset et cetera. Performance evaluation of the proposed model may also utilize metrics defined in recent literatures, such as Short Term Objective Intelligibility (STOI) (Taal et al., 2010) and Perceptual Evaluation of Speech Quality (PESQ) (Rix et al., 2001).

Other than that, the development of audio-only speech separation neural network may be resumed in the future as well. The audio-only network serves as the indispensable, secondary module to the robust user-aware speech separation pipeline. The conception of the audio-only network will strictly follow the implementation phases of the audio-visual network to eliminate inconsistencies in the speech separation pipeline.

Moreover, the test cases may be expanded to include videos that resembles real-world scenarios, e.g. video footage of a noisy restaurant, vlogs of social media influencers. Edge cases are considered as well, e.g. identical twins as separate speakers, to ensure a more extensive coverage of possible test cases.

BIBLIOGRAPHY

Afouras, T., Chung, J. and Zisserman, A. (2018). The Conversation: Deep Audio-Visual Speech Enhancement. Interspeech 2018.

Chen, Z., 2017. Single Channel auditory source separation with neural network (Doctoral dissertation, Columbia University).

Cherry, E. (1953). Some Experiments on the Recognition of Speech, with One and with Two Ears. The Journal of the Acoustical Society of America, 25(5), pp.975-979.

Delfarah, M. and Wang, D. (2017). Features for Masking-Based Monaural Speech Separation in Reverberant Conditions. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(5), pp.1085-1094.

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., Freeman, W. and Rubinstein, M. (2018). Looking to listen at the cocktail party. ACM Transactions on Graphics, 37(4), pp.1-11.

Erdogan, H., Hershey, J., Watanabe, S. and Le Roux, J. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Fu, S., Hu, T., Tsao, Y. and Lu, X. (2017). Complex spectrogram enhancement by convolutional neural network with multi-metrics learning. 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP).

Gabbay, A., Shamir, A. and Peleg, S. (2018). Visual Speech Enhancement. Interspeech 2018.

George, N. V., & Panda, G., 2013. Advances in active noise control: A survey, with emphasis on recent nonlinear techniques. Signal Processing, 93(2), pp.363–377.

Girin, L., Gannot, S. and Li, X. (2018). Multimodal behavior analysis in the wild. Academic Press, pp.53-78.

Golumbic, E., Cogan, G., Schroeder, C. and Poeppel, D. (2013). Visual Input Enhances Selective Speech Envelope Tracking in Auditory Cortex at a "Cocktail Party." Journal of Neuroscience, 33(4), pp.1417-1426.

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

Hoover, K., Chaudhuri, S., Pantofaru, C., Slaney, M. and Sturdy, I., 2017. Putting a face to the voice: Fusing audio and visual signals across a video to determine speakers. arXiv preprint arXiv:1706.00079.

Hou, J., Wang, S., Lai, Y., Tsao, Y., Chang, H. and Wang, H. (2018). Audio-Visual Speech Enhancement Using Multimodal Deep Convolutional Neural Networks. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(2), pp.117-128.

Huang, P., Kim, M., Hasegawa-Johnson, M. and Smaragdis, P. (2015). Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(12), pp.2136-2147.

Isik, Y., Roux, J., Chen, Z., Watanabe, S. and Hershey, J. (2016). Single-Channel Multi-Speaker Separation Using Deep Clustering. Interspeech 2016.

Jegou, H., Douze, M., Schmid, C. and Perez, P. (2010). Aggregating local descriptors into a compact image representation. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Kolbaek, M., Yu, D., Tan, Z. and Jensen, J. (2017). Multitalker Speech Separation With Utterance-Level Permutation Invariant Training of Deep Recurrent Neural Networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 25(10), pp.1901-1913.

Lueg, P.,1936. Process of Silencing Sound Oscillations. US Patent 2043416.

Ma, W., Zhou, X., Ross, L., Foxe, J. and Parra, L. (2009). Lip-Reading Aids Word Recognition Most in Moderate Noise: A Bayesian Explanation Using High-Dimensional Feature Space. PLoS ONE, 4(3), pp.233-252.

Møller, H., 1984. Physiological and Psychological Effects of Infrasound on Humans. Journal of Low Frequency Noise, Vibration and Active Control, 3(1), pp.1–17.

Owens, A. and Efros, A.A., 2018. Audio-visual scene analysis with self-supervised multisensory features. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 631-648).

Rivet, B., Wang, W., Naqvi, S. and Chambers, J. (2014). Audiovisual Speech Source Separation: An overview of key methodologies. IEEE Signal Processing Magazine, 31(3), pp.125-134.

Schroff, F., Kalenichenko, D. and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

Vincent, E., Gribonval, R. and Fevotte, C. (2006). Performance measurement in blind audio source separation. IEEE Transactions on Audio, Speech and Language Processing, 14(4), pp.1462-1469.

Wang, D. and Chen, J. (2018). Supervised Speech Separation Based on Deep Learning: An Overview. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(10), pp.1702-1726.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. and Schuller, B. (2015). Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR. Latent Variable Analysis and Signal Separation, pp.91-99.

Williamson, D., Wang, Y. and Wang, D. (2016). Complex Ratio Masking for Monaural Speech Separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(3), pp.483-492.

Yu, D., Kolbaek, M., Tan, Z. and Jensen, J. (2017). Permutation invariant training of deep models for speaker-independent multi-talker speech separation. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.241-245.

Taal, C.H., Hendriks, R.C., Heusdens, R. and Jensen, J., 2010, March. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing* (pp. 4214-4217). IEEE.

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

Tian, Y., Xu, C. and Li, D., 2019. Deep Audio Prior. *arXiv preprint arXiv:1912.10292*.

APPENDIX A: Final Year Project Source Code

GitHub Link: https://github.com/alexmak916/FYP2

README.md

Final Year Project 2

Title: A Robust Speaker-aware Speech Separation Technique using Composite Speech Models

Disclaimer: This project borrows reference from another repository - Google Audiovisual Model.

Dependencies

The project runs on Python 3.6. Please download and pip install packages in *requirements.txt*

Instructions

Dataset

To prepare the dataset, navigate into the **data** folder and run the following command:

```
python download_dataset.py
```

Several configurations can be set in the *download_dataset.py* script, such as number of video clips and normalizing audio.

For audiovisual models, navigate to **model/pretrain_model** and generate the face embeddings. Run the following command:

```
python pretrain_load_test.py
```

Then, rename the output folder to **face_emb** and move it to **data/video**.

Model Training

To train the audio-only speech separation model, navigate to **model/model_v1** and run the following command:

```
python AO_train.py
```

Model Inference

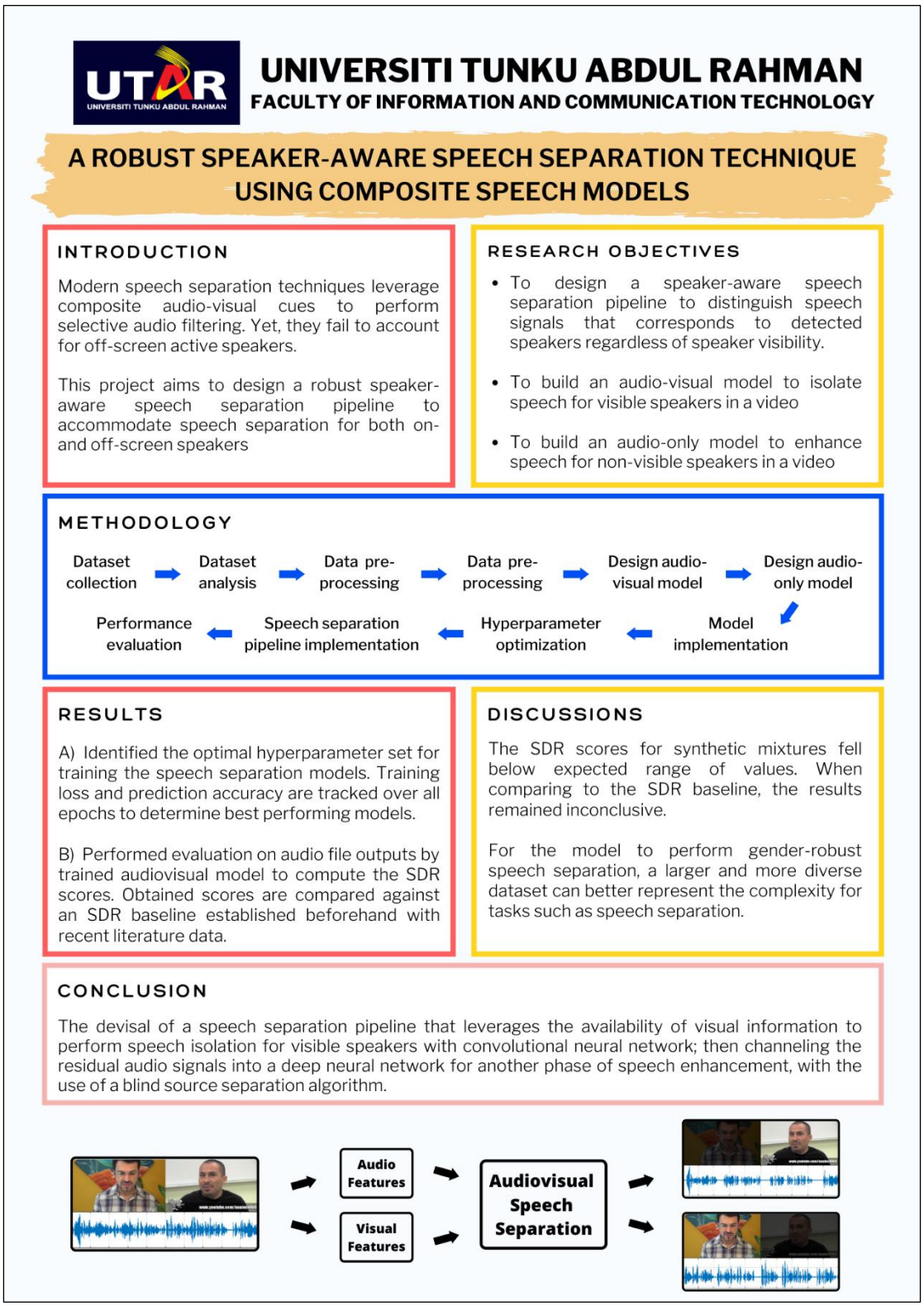
Copy and paste the saved H5 model file into **saved_AV/AO_models** folders. For testing data, change the **dl_from_training** variable to **False** in the *download_dataset.py* script.

Afterwards, modify the file path in the *AV/AO_predict_video.py* script and run the following command:

```
python AO_predict_video.py
python AV_predict_video.py
```

The estimated speech files will be in the respective **pred** folder.

APPENDIX B: Poster



APPENDIX C: Final Year Project Biweekly Report

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: Mak Wen Xuan 16ACB04621	
Supervisor: Dr. Aun Yichiet	
Project Title: A Robust Speaker-Aware Speech Separation Technique Using Composite Speech Models	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

i. Caught up with current project status.
ii. Prepare required files for hyperparameter optimization experiments.

2. WORK TO BE DONE

i. Verify that the TensorFlow library is compatible since its upgrade.
ii. Upload all the required files to cloud storage.

3. PROBLEMS ENCOUNTERED

i. Library dependencies and version conflicts.
ii. Required to book time slots for the research machine.

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: Mak Wen Xuan 16ACB04621	
Supervisor: Dr. Aun Yichiet	
Project Title: A Robust Speaker-Aware Speech Separation Technique Using Composite Speech Models	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

i. Uploaded training files for audiovisual model to Google Drive.
ii. Completed audiovisual model training for hyperparameter optimization.

2. WORK TO BE DONE

i. Proceed with audio-only model training for hyperparameter optimization.
ii. Determine hyperparameters for audio-only model training.

3. PROBLEMS ENCOUNTERED

i. Connection timeout in Google Colab.
ii. File size upload limit in Google Drive.

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 6**
Student Name & ID: Mak Wen Xuan 16ACB04621	
Supervisor: Dr. Aun Yichiet	
Project Title: A Robust Speaker-Aware Speech Separation Technique Using Composite Speech Models	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

i. Uploaded training files for audio-only model to Google Drive.
ii. Completed audio-only model training for hyperparameter optimization.

2. WORK TO BE DONE

i. Finalize the experiment findings.
ii. Capture the training process for speech separation models.

3. PROBLEMS ENCOUNTERED

iii. Connection timeout in Google Colab.
iv. File size upload limit in Google Drive.

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: Mak Wen Xuan 16ACB04621	
Supervisor: Dr. Aun Yichiet	
Project Title: A Robust Speaker-Aware Speech Separation Technique Using Composite Speech Models	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

i. Completed hyperparameter optimization for the speech separation models.
ii. Recorded the training loss and accuracy in a spreadsheet.

2. WORK TO BE DONE

i. Generate the testing dataset and its variants.
ii. Start writing content for FYP2 report.

3. PROBLEMS ENCOUNTERED

i. No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 10**
Student Name & ID: Mak Wen Xuan 16ACB04621	
Supervisor: Dr. Aun Yichiet	
Project Title: A Robust Speaker-Aware Speech Separation Technique Using Composite Speech Models	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

i. Generated the testing dataset for model evaluation.
ii. Drafted a simple outline for Chapter 4, 5 and 6 for the report draft.

2. WORK TO BE DONE

i. Perform evaluation on the audiovisual model.
ii. Add more content to the draft outline.

3. PROBLEMS ENCOUNTERED

i. Movement Restriction Order discontinued access to research machine.
ii. Retrieving the saved files on the research machine.

4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 12**
Student Name & ID: Mak Wen Xuan 16ACB04621	
Supervisor: Dr. Aun Yichiet	
Project Title: A Robust Speaker-Aware Speech Separation Technique Using Composite Speech Models	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

i. Performed evaluation on the trained audiovisual model.
ii. Finalized the outline for Chapter 4, 5 and 6 for the report draft.

2. WORK TO BE DONE

i. Create PowerPoint slides for the upcoming presentation.
ii. Finish revision for the report draft and check for plagiarism.

3. PROBLEMS ENCOUNTERED

i. Learn how to load the audio files as numpy array in Google Colab.
ii. Ensure the audio files' integrity and sequence.
iii. Hard to arrange consultation due to Movement Restriction Order.

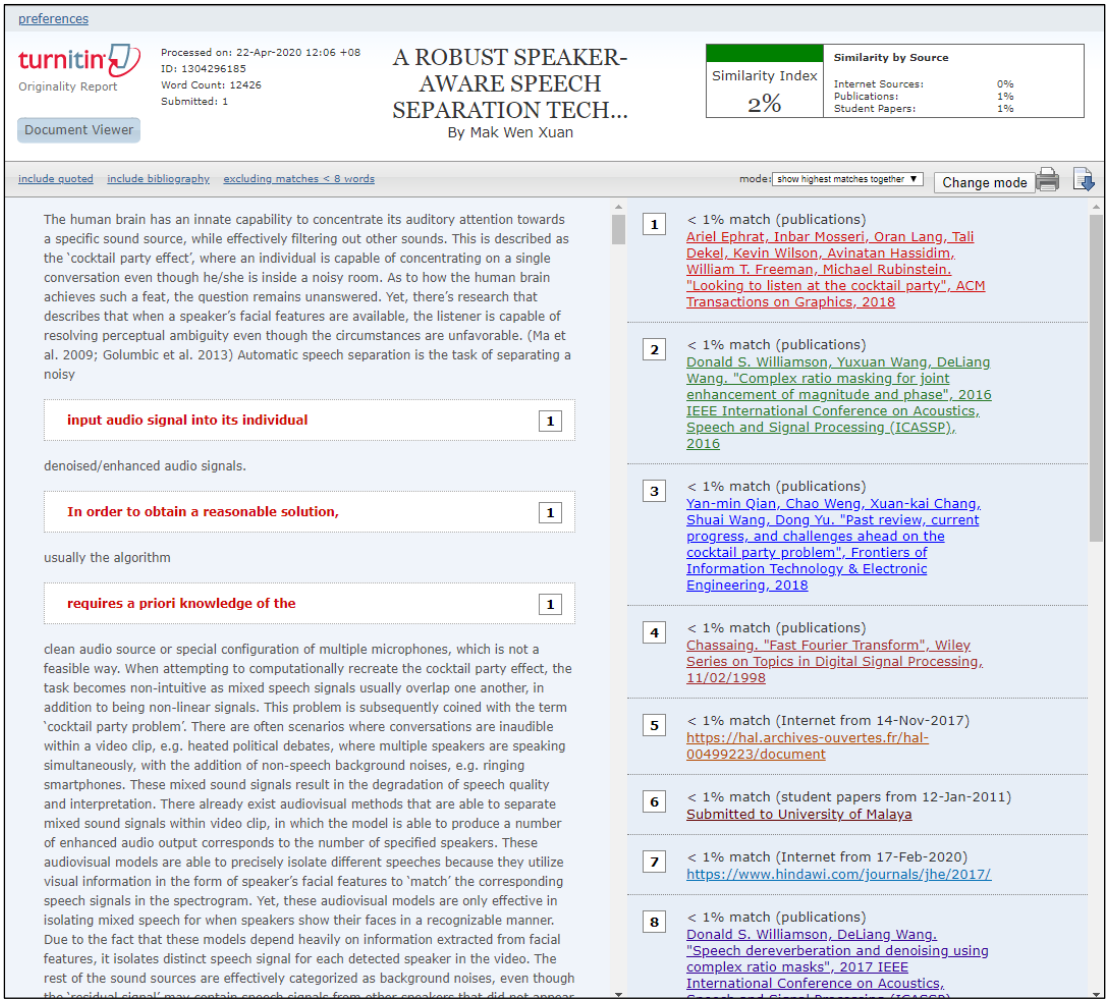
4. SELF EVALUATION OF THE PROGRESS

Self-assigned tasks are completed within expected timeframe.

Supervisor's signature

Student's signature

APPENDIX D: Plagiarism Check Result



preferences

turnitin Originality Report

Processed on: 22-Apr-2020 12:06 +08
ID: 1304296185
Word Count: 12426
Submitted: 1

A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECH...
By Mak Wen Xuan

Similarity Index 2%

Similarity by Source
Internet Sources: 0%
Publications: 1%
Student Papers: 1%

Document Viewer

include quoted include bibliography excluding matches < 8 words

mode: show highest matches together Change mode

The human brain has an innate capability to concentrate its auditory attention towards a specific sound source, while effectively filtering out other sounds. This is described as the 'cocktail party effect', where an individual is capable of concentrating on a single conversation even though he/she is inside a noisy room. As to how the human brain achieves such a feat, the question remains unanswered. Yet, there's research that describes that when a speaker's facial features are available, the listener is capable of resolving perceptual ambiguity even though the circumstances are unfavorable. (Ma et al. 2009; Golumbic et al. 2013) Automatic speech separation is the task of separating a noisy

input audio signal into its individual 1

denoised/enhanced audio signals.

In order to obtain a reasonable solution, 1

usually the algorithm

requires a priori knowledge of the 1

clean audio source or special configuration of multiple microphones, which is not a feasible way. When attempting to computationally recreate the cocktail party effect, the task becomes non-intuitive as mixed speech signals usually overlap one another, in addition to being non-linear signals. This problem is subsequently coined with the term 'cocktail party problem'. There are often scenarios where conversations are inaudible within a video clip, e.g. heated political debates, where multiple speakers are speaking simultaneously, with the addition of non-speech background noises, e.g. ringing smartphones. These mixed sound signals result in the degradation of speech quality and interpretation. There already exist audiovisual methods that are able to separate mixed sound signals within video clip, in which the model is able to produce a number of enhanced audio output corresponds to the number of specified speakers. These audiovisual models are able to precisely isolate different speeches because they utilize visual information in the form of speaker's facial features to 'match' the corresponding speech signals in the spectrogram. Yet, these audiovisual models are only effective in isolating mixed speech for when speakers show their faces in a recognizable manner. Due to the fact that these models depend heavily on information extracted from facial features, it isolates distinct speech signal for each detected speaker in the video. The rest of the sound sources are effectively categorized as background noises, even though

1 < 1% match (publications)
Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, Michael Rubinstein, "Looking to listen at the cocktail party", ACM Transactions on Graphics, 2018

2 < 1% match (publications)
Donald S. Williamson, Yuxuan Wang, DeLiang Wang, "Complex ratio masking for joint enhancement of magnitude and phase", 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016

3 < 1% match (publications)
Yan-min Qian, Chao Weng, Xuan-kai Chang, Shuai Wang, Dong Yu, "Past review, current progress, and challenges ahead on the cocktail party problem", Frontiers of Information Technology & Electronic Engineering, 2018

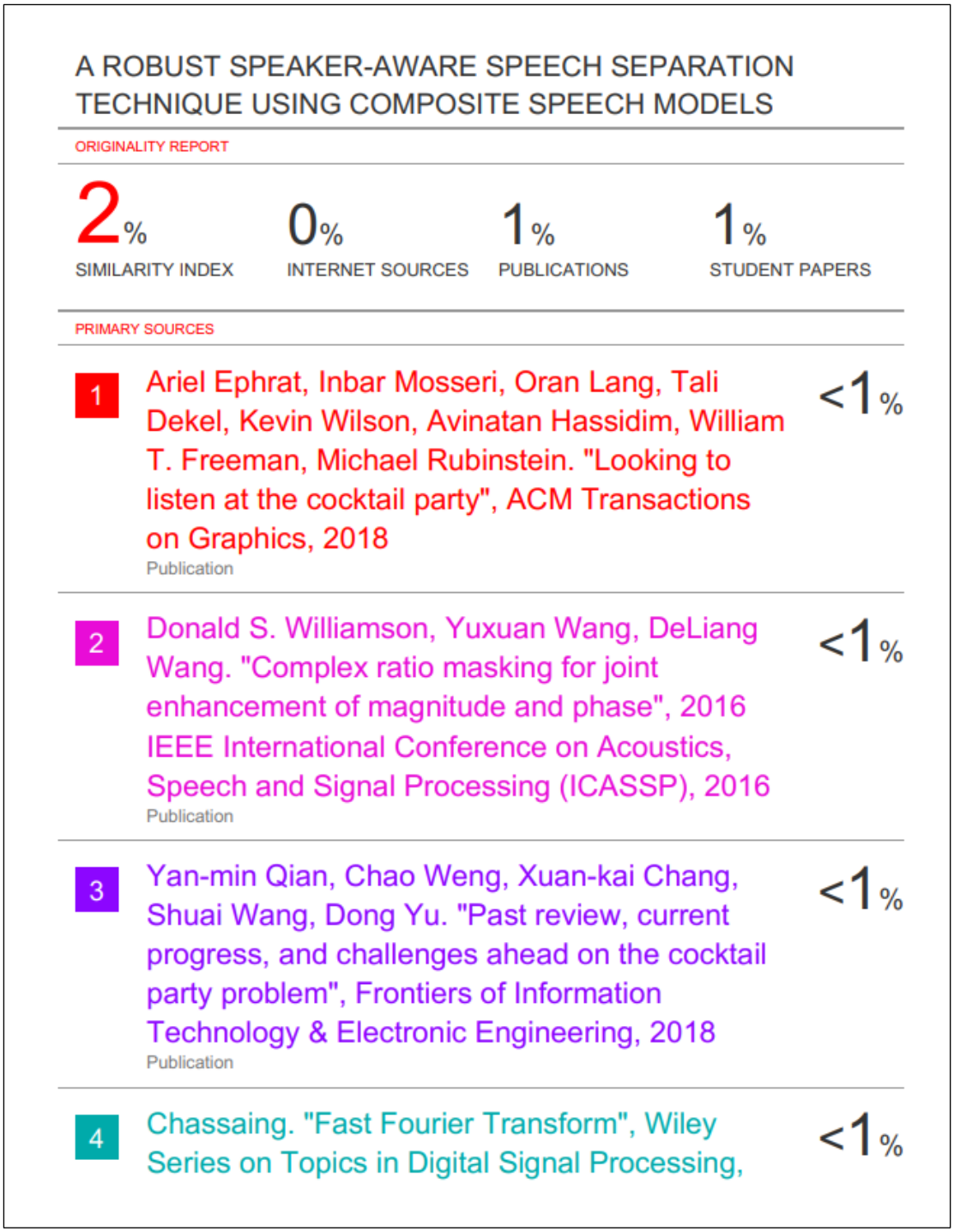
4 < 1% match (publications)
Chassaing, "Fast Fourier Transform", Wiley Series on Topics in Digital Signal Processing, 11/02/1998

5 < 1% match (Internet from 14-Nov-2017)
https://hal.archives-ouvertes.fr/hal-00499223/document

6 < 1% match (student papers from 12-Jan-2011)
Submitted to University of Malaya

7 < 1% match (Internet from 17-Feb-2020)
https://www.hindawi.com/journals/jhe/2017/

8 < 1% match (publications)
Donald S. Williamson, DeLiang Wang, "Speech dereverberation and denoising using complex ratio masks", 2017 IEEE International Conference on Acoustics,



A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECHNIQUE USING COMPOSITE SPEECH MODELS

ORIGINALITY REPORT

2%	0%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, Michael Rubinstein. "Looking to listen at the cocktail party", ACM Transactions on Graphics, 2018 Publication	<1%
2	Donald S. Williamson, Yuxuan Wang, DeLiang Wang. "Complex ratio masking for joint enhancement of magnitude and phase", 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016 Publication	<1%
3	Yan-min Qian, Chao Weng, Xuan-kai Chang, Shuai Wang, Dong Yu. "Past review, current progress, and challenges ahead on the cocktail party problem", Frontiers of Information Technology & Electronic Engineering, 2018 Publication	<1%
4	Chassaing. "Fast Fourier Transform", Wiley Series on Topics in Digital Signal Processing,	<1%

Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	MAK WEN XUAN
ID Number(s)	16ACB04621
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONS)
Title of Final Year Project	A ROBUST SPEAKER-AWARE SPEECH SEPARATION TECHNIQUE USING COMPOSITE SPEECH MODELS

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 2 % **Similarity by source** Internet Sources: 0 % Publications: 1 % Student Papers: 1 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: DR. AUN YICHIET

Date: 24/04/2020

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB04621
Student Name	MAK WEN XUAN
Supervisor Name	DR. AUN YICHIET

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date: 24/04/2020	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. _____ (Signature of Supervisor) Date: 24/04/2020