Isolated Speech Analysis using Deep Learning Feature for Mobile-Assisted Language Learning Application

Tanakorn Tragoonrungsi
Parin Kobboon
Phum Vajanavinit

Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2018

KMITL-2019-IC-B-003-008

Thesis - Academic Year 2018

Bachelor of Engineering in Software Engineering
International College
King Mongkut's Institute of Technology Ladkrabang

Title: Isolated Speech Analysis using Deep Learning Feature for Mobile-Assisted Language Learning Application

Authors:

1. Tanakorn Tragoonrungsi Student ID: 58090017
2. Parin Kobboon Student ID: 58090023
3. Phum Vajanavinit Student ID: 58090029

Approved for Submission

....

(Dr. Montri Phothisonothai)
Advisor

Date ..8../.Jul./.2019

Acknowledgments

We would like to express our sincere gratitude to Dr.Montri Phothisonothai, our project advisor, for providing resources with all the tools and knowledge during the project. Every comment and suggestion is an important part of guiding our project in the right direction while driving forward. We would also like to thank the seniors for helping us during the difficult time and always motivating us to the end. There are several changes, incidents, and unexpected problems during the project development, without the help of our advisor or seniors the goal may not have been achieved. Last but not least, we would like to thanks our family for supporting us spiritually and mentally throughout this project and whoever read this report for spending your time and interest in our project.

Abstract

State-of-the-art in speech analysis system is focusing on various factors such as dialects or accents of the spoken language among different speakers, i.e., the speaker gender, as well as the speaker age. Our thesis proposes the development of a mobile application that analyzes the speaker's speech, determines the speaker's accent country, and help improve speaker accent. During analysis, the input audio goes through the process of pre-processing, feature extraction, classification, and assessment before the result will be returned. Then, with the help of deep learning approach, the mentioned process can possibly be improved in accuracy and correctness. Ultimately in this project, we tried to build a program which aspires to assist the speakers learning ability on their pronunciation and the correctness of their speech compared to native speakers.

Table of contents

List of figures

List of tables

Chapter 1
Introduction

1.1 Problem Descriptions

Nowadays, there are over hundreds of English accents. Having these various accents are important, they add personality to English, and without them, conversations would be tedious. Nevertheless, not all accents are easy for the average English speaker to understand. It is one of the biggest cause of communication breakdowns when two non-native speakers communicate in English [1]. Aside from affecting the communication between speakers, the accent does affect the Automatic Speech Recognition systems. Statistical analysis has identified gender and accent to be the most important factors of speaker variability affecting the fluency of Automatic Speech Recognition systems [2].

Unfortunately, learning to pronounce a word correctly can be one of the hardest parts of learning a language and mastering it might take up to 6 hours a day in private language classes, followed by several hours of personal study time. However, not all of us have the time and resources for this kind of training.

Our goal is to develop a deep learning model that is able to identify and classify a speaker by his or her predicted native language. In addition, we would want to give a suggestion on how to improve their accent compare to native speakers back to the speakers on. Furthermore, in the context of learning to pronounce a language, it may be helpful to verify automatically whether an applicants' accent corresponds to accents spoken in a region he claims, he is from. When an individual learns to speak a second language, there is a tendency to replace some syllables in the second language with more prominent syllables from his native language [3]. Thus, accented speech can be seen as the result of a language being filtered by a second language, and the analysis of accented speech may uncover hidden semblances among different languages.

1.2 Objective

The objective of this project is to design a mobile-based automated system for English language learner which can be categorized as follows:

- Develop a mobile application for users to record their audio and see the result.
- Display the input audio files in signals and file format.
- Process the audio file through pre-processing, feature extraction, and assessment.
  - Pre-processing approaches contain zero mean, normalization, fixed-length, power, window-based standard deviation, thresholding, and speech segmentation.
  - Feature extraction approaches contain spectrogram, mfcc, and formant frequency.
  - Feature extraction approaches contain zero-crossing rate, and time duration.
  - Assessment approaches contain dynamic time warping, time duration, and frequency.
- Collect enough data to implement deep learning into the process of classification.
- Return feedback to the user after assessing the input, with high-level feedback for the user to improve their pronunciation.

1.3 Scope of Work

The proposed system is designed to work under the scope as follows:

- The input audio (*.wav file) in the noiseless environment.
- The isolated word can be selected.
- Shows the speakers origin country
- Scored the input using expert system in a grading system (0-100 score)
- Monitor and analyze the speakers audio input

Chapter 2
Related Works

This chapter contains a closer look at basic concepts and related published work that are essential for the work in this thesis. Some of the works mentioned will be used as a base to be built on and to further expand the possible solution to the problem. In Figure 2.1, shows the relationship between these papers and how each one of them belongs to which categories.

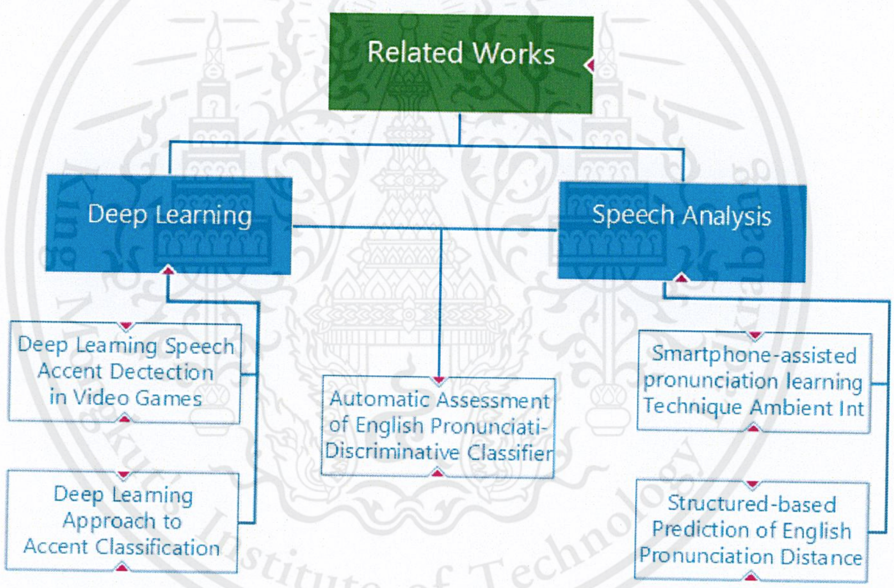


Figure 2.1: The diagram of related works.

2.1 Deep Learning for Speech Accent Detection in Video Games [5]

This research aims to help sociolinguists and analysts establish a critical study and content analytical findings for instance about stereotypical uses of speech accents, to better analyze who has what accent in video games, and what kind of language ideologies and social value judgments the use of accents in games. The paper presented a preliminary study on training a deep neural network to automatically label audio files with accents. They trained AlexNet on the Speech Accent Archive data and then applied it to audio files captured from a video game [5].

Proposed Method

In order to determine the best value for each parameter to use with the network, experiments are run with a range of values before selecting the set that gives the best test results. The paper is interested in four parameters, two related to the network: number of epochs and the size of the batch used; and two related to the size of the spectrograms: number of time windows and number of frequency filters. The input for AlexNet is a 227×227 image and the ranges tested for each parameter are: 10, 50, 100, and 200 epochs, 3, 5, 10, 50, and 100 batch sizes, 10, 25, 50, 75 and 113 number of windows and frequency filters and divided the audio files into 1, 3, and 5 seconds segment. The best parameters was: 50 epochs, 5 images per batch, 113 frequency filters, 50 time windows, with 3 seconds of audio segments. With these parameters the network had an average test accuracy of 61%.

Result

Once AlexNet was trained, they froze its weights and applied it to audio files captured from Dragon Age: Origins. They achieved an accuracy of 75% from training the 134 audio files with same parameters in Speech Accent Archive. Then the 17 audio files from the game were put tested and the network achieved the accuracy of 52.7% and 60% when the audio file does not have background noise.

2.2 Deep Learning Approach to Accent Classification [10]

Nowadays automatic speech recognition (ASR) systems are likely to be trained to perform best in American and British accents which have a unique, strong, and distinctive accent. For those who are not a native speakes tend to be forced to speak these accents in order to get ASR system to correctly recognize the speech. To give more suitable speech recognition model for alternative accents, the objective is to develop a deep learning model which is capable of classifying a speaker by his/her native language with an utterance as an input.

Proposed Method

In order to accurately predict the accent of a utterance, pre-processing is used to manipulate the audio signals using a peak detection library. The pre-processed signal is then extracted to Mel-frequency cepstral coefficients (MFCCs) which is a 2 dimensional array. The 4 methods proposed in the paper to predict the result consists of 2 traditional methods including Random Forests, and Gredient Boosting methods; and 2 deep neural network architectures including Convolutional Neural Network (CNN), and Multi-layer Perceptron (MLP). CNN tends to be the most effective methodology, since its layers preserve the spatial relationship between pixels by learning local patterns. After doing 10 fold cross validation, the test accuracy of the traditional methods is approximately 69% which is lower than deep neural network architectures achieving accuracy of 80 - 88%

2.3 Accent Classification Using Deep Belief Networks [3]

This research aims to develop an effective classifier for foreign accented English speech in order to determine the origins of the speaker. They created an accented spoken English corpus that consists of 30 speakers from 6 different countries including China, India, France, Germany, Turkey and Spain. They used MFCC as feature and Deep belief network (DBN) as classifier. The parameters of the DBN were determined by an iterative approach in which the node weight is updated according to the substitution error.

Proposed Method

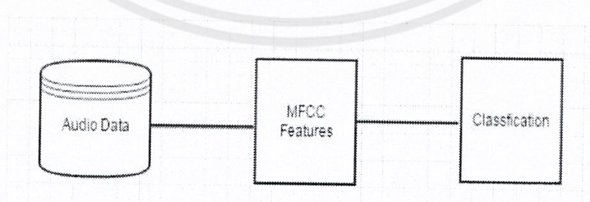


Figure 2.2: Methodology Overview [3].

For audio data, they created a 6 hours foreign accented speech database with 30 speakers from 6 countries. All the data were pre-processed with noise cancellation and normalization. They used 80% data for training and 20% data for testing. Then they used MFCC for the feature of the audio. 13 MFCC was extracted with frame size of 42ms, with 20ms overlapping windows. The DBN model has 2 hidden layers, and 1000 nodes per layer. The classifiers were trained by 2 countries (China, France), 4 countries (India, Germany, Spain and Turkey) and all 6 countries respectively.

Result

Classifier	**2 countries**	**4 countries**	**6 countries**
Random Forest	**67.68%**	**43.59%**	**33.01%**
K-NN	**72.01%**	**49.34%**	**40.16%**
Naïve Bayes	**57.66%**	**31.88%**	**22.96%**
Logistic Regression	**50.01%**	**27.01%**	**19.64%**
SGD Classifier	**49.51%**	**25.04%**	**20.64%**
Perceptron	**52.22%**	**24.94%**	**18.64%**
SVM	**66.58%**	**41.74%**	**32.64%**
DBN	90.20%	77.20%	71.90%

Figure 2.3: Comparison of classifiers [3].

In Figure 2.3, they achieved an accuracy of 90.2% for 2 accented datasets and 71.9% for 6 accented datasets. The result is much better than the other state-of-art methods such as SVM, k-NN and random forest which have around 40% accuracy.

2.4 Automatic Assessment of English Learner Pronunciation Using Discriminative Classifiers [8]

This research objective is to present a novel system for automatic assessment of pronunciation quality of English learner speech, based on deep neural network features and phoneme specific discriminative classifiers. The training of the neural network model was done with a large corpus of data both having native and non-native speaker and then used to extract phoneme posterior probabilities. A part of the corpus includes per phone teacher annotations, which allows training of two Gaussian Mixture Models(GMM), representing correct pronunciations and typical error patterns. The likelihood ratio is then obtained for each observed phone. Several learning models were evaluated on a large corpus of English-learning students, with a variety of skill levels, and aged 13 upwards.

Proposed Method

The proposed method describe a method for phone-level pronunciation error detection and for automatic assessment tools permeate teaching methodologies, reliable automatic assessment of English learner speech is of increasing interest.

In order to assess the pronunciation quality of an unknown student speaker a reference is describe using pronunciation spoken by a teacher. Then the system output a vector with quality assessment score for each phonemes. Teacher and Student will have the same word content, given the two recording a computation is done to for calculating the probability that the learner recording mimics the pronunciation in the teacher reference. The constant for every similar utterance of the learner and the computed probability can be approximated. Then after both student and teacher feature sequences was split into a phone-related sets, the problem can be turned into a binary classification problem where phoneme-level scores for each student are computed and serves as a threshold which depends on the degree of proficiency of the learners.

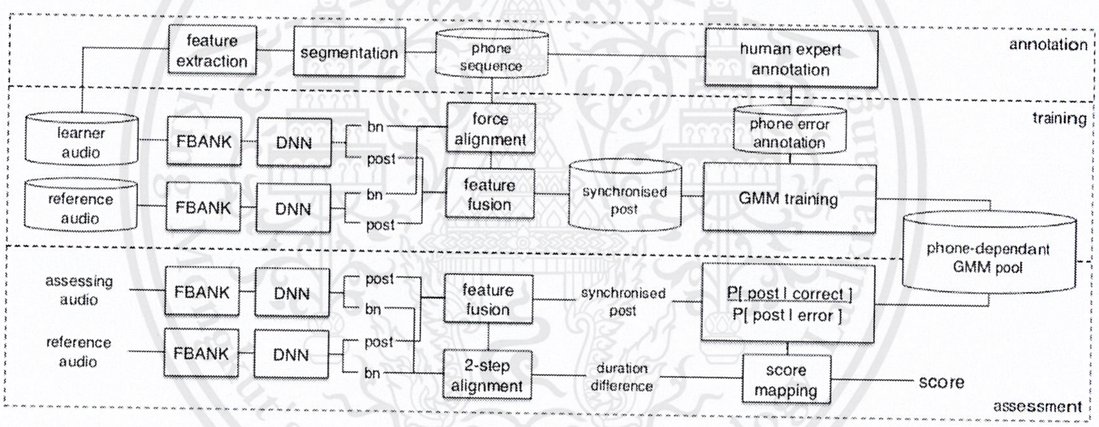


Figure 2.4: Pronunciation Evaluation Framework [9].

The pronunciation evaluation framework shown in Figure 2.4 display stages of annotation(top), training(middle) and assessment(bottom) with the method described implemented with the architecture outlined here as well.

Result

SYSTEM	U_t (gender)	U_s	MAPPING	FAR	MISS	CC	ACC	F-SCORE	NCE
Baseline [D+E+S+Δ]	INY51 (F)	$INA_{ph\ 1\text{-}2}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.443	0.365	0.443	0.695	-	0.008
DNN_{UK}+INY51	INY51 (F)	$INA_{ph\ 1\text{-}2}$	none	0.239	0.207	0.426	0.763	0.327	N.A.
DNN_{UK}+INY51	INY51 (F)	$INA_{ph\ 1\text{-}2}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.184	0.193	0.716	0.816	0.388	0.397
DNN_{UK}+INY51	INY51 (F)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.252	0.264	0.613	0.747	0.261	0.447
DNN_{UK}+INY51+D	INY51 (F)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.252	0.262	0.614	0.748	0.262	0.446
DNN_{UK}+INY51+D+Δ	INY51 (F)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.258	0.255	0.616	0.742	0.260	0.451
DNN_{UK+INT}+INY51+D+Δ	INY51 (F)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.297	0.275	0.581	0.704	0.229	0.559
DNN_{UK+INT}+MULTI+D+Δ	INY51 (F)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.288	0.284	0.576	0.712	0.232	0.473
DNN_{UK}+INY51+D+Δ	INY52 (M)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.250	0.241	0.574	0.751	0.504	0.371
DNN_{UK+INT}+INY51+D+Δ	INY52 (M)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.291	0.281	0.582	0.710	0.231	0.558
DNN_{UK+INT}+MULTI+D+Δ	INY52 (M)	$INA_{ph\ 3\text{-}6}$	reg.tree($INA_{ph\ 1\text{-}2}$)	0.278	0.277	0.587	0.722	0.240	0.470

Figure 2.5: Pronunciation Score [9].

Figure 2.5 shows the comparison of phone-level mispronunciation detection scores predicted by the system, S, and marked in the human annotators' reference, R. Top: baseline system. Middle: system development using a matched teacher voice for training and assessment stages of the framework. Bottom: Comparisons using a mismatched teacher voice.

A measure of the time-discrepancy between teacher and student phone durations(D), and their differential values($\triangle$). A pronunciation error resulted when mapped scores fell beneath a threshold whose level sets the system's strictness. Here, the rate of undetected mispronunciations (MISS) balanced the well-pronounced phones marked as errors (false alarm rate, FAR).

The evaluation framework was used to assess learner pronunciation in the (unseen) INAph 3-6 dataset. For this, the system outcome, S, was compared with the combined human reference, R, using six common information retrieval indices. Of these, CC, FAR and MISS were previously defined; additionally, Accuracy (ACC), F-score, and normalised cross-entropy (NCE) are displayed in 2.5. ACC gives the percentage of phones that were correctly assessed. The F-score combines recall and precision rates[16]. NCE concerns the mutual information between the correctness of the mispronunciation detection and the confidence score in making that decision[19]. For the perfect system, FAR and MISS values are low, and CC, ACC, F-score and NCE are high.

2.5 Smartphone-Assisted Pronunciation Learning Technique for Ambient Intelligence [7]

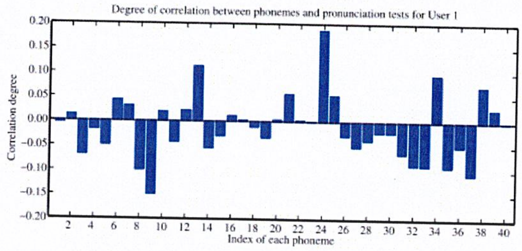
This research aims to develop Smartphone-assisted pronunciation learning technique (SAPT) which is capable of recommending suitable words for pronunciation improvement by analyzing user's speech to detect the pronunciation mistakes using a lightweight word recommendation approach. The system is designed to be able to use on a smartphone with low computational capacity by transferring the user's speech signals to an Internet of Things (IoT) member that can support speech recognition processing, resulting in a significant improvement on the applicability to ambient intelligence environment.

Proposed Method

Lightweight word recommendation technique in SAPT consists of seven steps. The technique starts with receiving speech from a user in audio signal format to test the spoken word and create a data set in the form of bag-of-phoneme model. The system then accesses correlation of the phonemes and mispronunciation. SAPT uses Google voice search (GVS) system to obtain a set of words for evaluating the pronunciation and a speech recognition system for the next process. After the calculation process of correlation, the system calculates importance value of each word and keep it for the next calculation method. The system is now able to calculate the selection probability for each word using the importance value from the previous calculation and eventually returns recommended selected words for the user to continue practicing.

Result

Degree of correlation between phonemes and pronunciation tests for User 1



Degree of correlation between phonemes and pronunciation tests for User 1. The top five phonemes are /f/ **(24),** /ɜ/ **(13),** /r/ **(34),** /v/ **(38), and** /b/ **(21).**

Top 20 words in *R* for User 1 based on selection probability (Top 10 phoenems: /f/, /ɜ/, /r/, /v/, /b/, /g/, /oʊ/, /ɔɪ/, /w/, **and** /ɔ/**).**

Rank	Words (Phonemes)	Prob.	Rank	Words (Phonemes)	Prob.
1	four [fɔr]	2.68%	11	cough k[ɔf]	0.83%
2	fork [fɔr]k	2.15%	12	for [f]ə[r]	0.79%
3	bird [bɜr]d	1.83%	13	Thursday θ[ɜr]zdeɪ	0.78%
4	grove [groʊv]	1.82%	14	worm [wɜr]m	0.69%
5	fog [fɔg]	1.78%	15	float [f]l[oʊ]t	0.62%
6	first [fɜr]st	1.62%	16	father [f]aðə[r]	0.54%
7	girl [gɜr]l	1.41%	17	row [roʊ]	0.51%
8	burger [bɜrg]ə[r]	1.14%	18	fill [f]ɪl	0.49%
9	folk [foʊ]k	1.02%	19	thirty θ[ɜr]ti	0.45%
10	robe [roʊb]	0.89%	20	where [w]e[r]	0.43%

Figure 2.6: The Result of Lightweight Word Recommendation Technique [8].

In Figure 2.6, the result of the Lightweight word recommendation technique indicates that the experimental user's top five incorrect pronunciation phonemes are /f/, /3/, /r/, /v/, and /b/. The result of incorrect phonemes is later used to compute a bag of phonemes based on selection probability as shown in the table.

2.6 Structure-based Prediction of English Pronunciation Distances & Its Analytical Investigation [6]

This research aims at creating a global and individual-basis map of English pronunciations to be used in teaching and learning World Englishes(WE) by applying Support Vector Regression(SVR) to help predict the pronunciation distance. Diversity of WE can be found in various aspects such as dialogue, syntax, pragmatics, lexical choice, spelling, pronunciation, etc. Among these kinds of diversity, this paper focuses on pronunciation. Creating this map mathematically requires a distance matrix in terms of pronunciation differences among all the speakers considered, and technically requires a method of predicting the pronunciation distance between any pair of the speakers.

Proposed Method

Structure-based Prediction of Pronunciation Distances

In order to accurately predict the pronunciation distance a speech structure method was proposed. By representing speech using speech structure, and proved that acoustic and non-linguistic variations involved in speech can become effectively unseen. The speech structure is invariant against any kind of continuous and convertible transform and this invariant is due to the transform-invariance of f-divergence as shown in Figure 2.7.

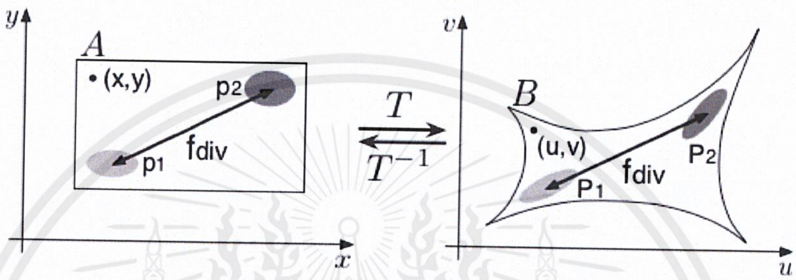


Figure 2.7: Transform-invariant of f-divergence [7].

where p1 (x) and p2 (x) are density functions of two distributions on measurable space X. g(t) is a convex function for t > 0. If take $\sqrt{t}$ as g(t), -log(f div) becomes the Bhattacharyya Distance (BD).

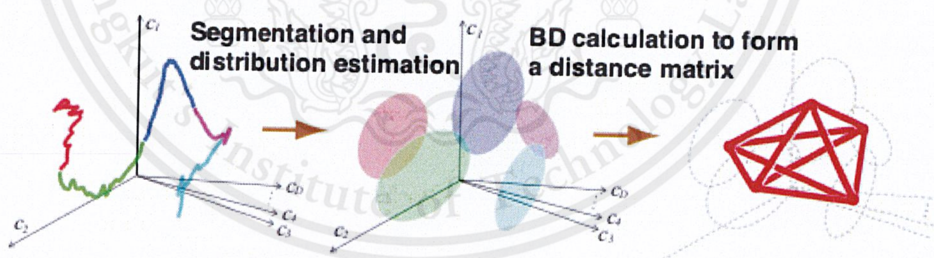


Figure 2.8: Procedure of BD [7].

Figure 2.8 shows the procedure of representing an utterance only by BDs. The utterance is a sequence of vectors and it is converted into a sequence of distributions. Here, any speech event is characterized as distribution. Then, the BD is calculated between any distribution pair and the resulting BD-based distance matrix is an invariance speech structure. When this representation is applied to pronunciation analysis, the matrix is called pronunciation structure.

Pronunciation Structure Calculation Procedure

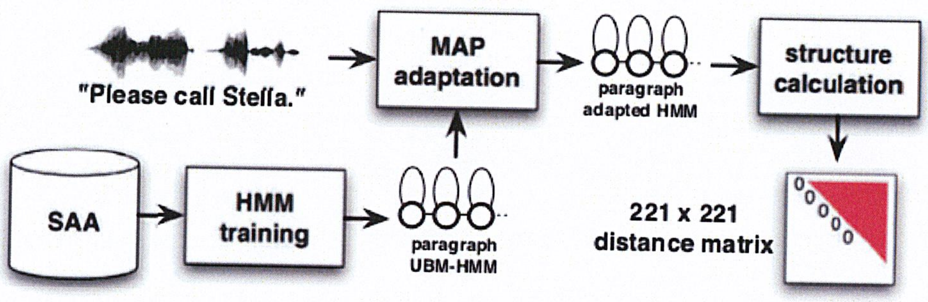


Figure 2.9: Procedure to Calculate the Pronunciation Structure [7].

Figure 2.9 shows the detailed procedure to calculate the pronunciation structure, where the Universal Background Model(UBM) was trained as paragraph-based Hidden Markov Model(HMM) by using all the 370 speakers. Result of comparing the data with the model structure will be determine by the pronunciation structure in form of a distance matrix. From there the two distance matrices of speakers S and T.

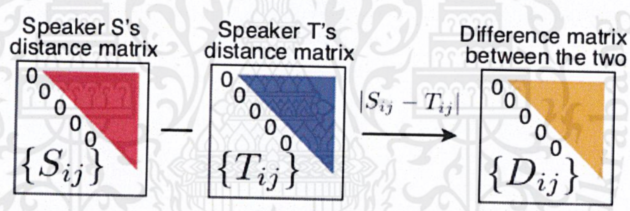


Figure 2.10: Difference Matrix Derived from Two Speakers' Matrices [7].

{Dij} were used as input features to Support Vector Regression(SVR) to predict the pronunciation distance and represent the differences between the two.

Result

An analytical investigations was done in order to understand how the proposed method works better. By using feature selection based on locality of speech contrasts and on phonetic attributes to investigate whether local contrasts contribute better than distant contrasts and how multiple phonetic classes affects the distance matrices. Figure 2.11 display the results of the investigations where the correlations that were obtained in the individual experiments are plotted.

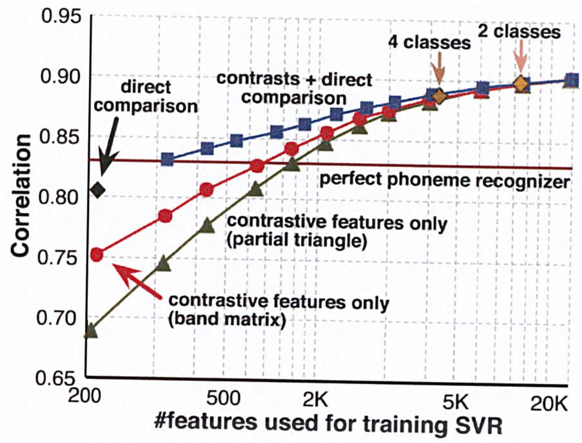


Figure 2.11: Correlations Obtained by Analytical Investigations [7].

Chapter 3
Background Knowledge

In order to be able to classify accents, it is necessary to understand how the underlying process of extracting information, pre-processing, feature extraction, classification, and assessment of speech works. This chapter discusses about background knowledge which introduces various concepts used in this project.

3.1 WAV File

A WAV file is a raw uncompressed lossless audio format which can take up quite a bit of space, coming in around 10 MB per minute with a maximum file size of 4 GB. The format uses containers to store audio data, track numbers, sample rate, and bit rate.

3.1.1 Sample Rate

The sample rate is the number of times the audio is sampled per second. For example, CD audio has a sample rate of 44100 Hz. This means that the audio is sampled 44100 times every second. It is measured in Hz or kHz and is mostly used for measuring frequency.

Nyquist Frequency

Nyquist Frequency is half of the sampling rate of a discrete signal processing system [11]. It is sometimes known as the folding frequency of a sampling system.

3.1.2 Bit Depth

Bit depth is the number of bits of information in each sample, and it directly corresponds to the resolution of each sample. It is the number of possible values to represent a sample of the signal with.

3.1.3 Audio Channel

Audio Channel refers to the independent audio signal which is collected or playback when the sound is recording or playback in different spatial position. Therefore, the number of channel is the amount of sound source when the sound is recording or the relevant speaker number when it is playback.In monaural sound, one single channel is used. It can be reproduced through several speakers, but all speakers are still reproducing the same copy of the signal.

3.1.4 Bit Rate

Bitrate refers to the number of bits or the amount of data that are processed over a certain amount of time. In audio, this usually means kilobits per second and its bitrate is calculated by:

$$kbitspersecond = sample\ rate \times bitdepth \times numberofchannels \tag{3.1}$$

3.2 Fourier Transform

The Fourier transform is a mathematical function that takes a time-based pattern as input and determines the overall cycle offset, rotation speed and strength for every possible cycle in the given pattern. The Fourier transform is applied to waveforms which are basically a function of time, space, or some other variable. Also it decomposes a waveform into a sinusoid and thus provides another way to represent a waveform.

3.2.1 Windowing concept

Signal is a non-stationary time variant signal. Windowing of a simple waveform like cos ωt causes its Fourier transform to develop non-zero values (commonly called spectral leakage) at frequencies other than ω.

3.2.2 Short-time Fourier transform(STFT)

The Short-Time Fourier Transform (STFT), is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time [6].

Continuous-time Short-time Fourier transform

The function to be transformed is multiplied by a window function which is nonzero for only a short period of time. The Fourier transform (a one-dimensional function) of the resulting signal is taken as the window is slid along the time axis, resulting in a two-dimensional representation of the signal.

$$STFT\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)\mathrm{e}^{-j\omega t}\,dt \tag{3.2}$$

Where w(τ) is the window function, commonly a Hann window or Gaussian window centered around zero, and x(t) is the signal to be transformed (note the difference between w and ω). X(τ,ω) is essentially the Fourier Transform of x(t)w(t-τ), a complex function representing the phase and magnitude of the signal over time and frequency.

Discrete-time Short-time Fourier transform

The data to be transformed could be broken up into chunks or frames (which usually overlap each other, to reduce artifacts at the boundary). Each chunk is Fourier transformed, and the complex result is added to a matrix, which records magnitude and phase for each point in time and frequency [6].

$$STFT\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n-m]\mathrm{e}^{-j\omega n} \tag{3.3}$$

Similarly to Continuous-time STFT, with signal x[n] and window w[n]. In this case, m is discrete and ω is continuous.

3.3 Spectrogram

A spectrogram is a visual way of representing the signal strength, or ”loudness”, of a signal over time at various frequencies present in a particular waveform.The magnitude squared of the STFT yields the spectrogram representation of the Power Spectral Density of the function:

$$spectrogram\{x(t)\}(\tau,\omega) \equiv |X(\tau,\omega)|^2 \tag{3.4}$$

3.4 Mel-frequency cepstral coefficients (MFCCs)

MFCC is an audio feature extraction technique which extracts parameters from the speech similar to ones that are used by humans for hearing speech, while at the same time, deemphasize all other information [3].

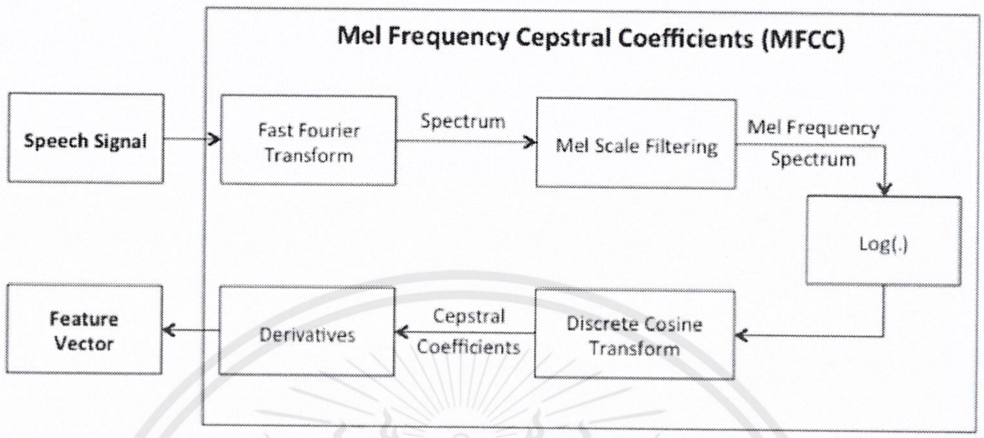


Figure 3.1: MFCC Steps [3].

In figure 3.1, the speech signal is first divided into time frames consisting of an arbitrary number of samples. After the windowing, Fast Fourier Transformation is calculated for each frame to extract frequency components of a signal in the time-domain. Then the logarithmic Mel-Scaled filter bank is applied to the Fourier transformed frame. The last step is to calculate Discrete Cosine Transformation of the outputs from the filter bank. For each speech frame, a set of MFCC is computed. This set of coefficients is called an acoustic vector which represents the phonetically important characteristics of speech and is very useful for further analysis.

3.5 Formant Frequency

A formant frequency is a concentration of acoustic energy around a particular frequency in the speech wave. A range of frequencies of a complex sound in which there is an absolute or relative maximum in the sound spectrum. It is the spectral peaks of the sound spectrum.

3.6 Deep Learning

Deep Learning is a sub-field of machine learning, dealing with structure and function of an artificial neutral network working similarly to how human brain nervous system structured where each neuron connected each other and passing information. Deep learning models work in layers with difference models each having its own structural and functional properties for information processing and communication patterns [14].

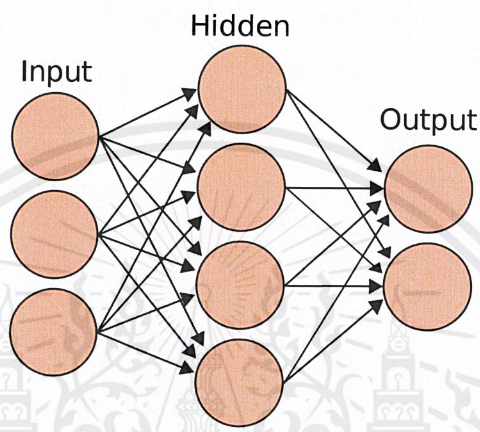


Figure 3.2: Multiple Layers within the Neural Networks.

A typical deep learning model will at least have three layers which are the input layer, hidden layer, and output layer with the number of hidden layer varies depending on method and purpose. Each layer accepts the information from previous and passes it on to the next one and each connection between neurons is associated with a weight. This weight dictates the importance of the input value where the initial weights are established randomly.

3.6.1 Weight

Weight is a value assign throughout the fully connected layers of the neural networks. Each neural will computes an output from the incoming input data from the receptive field in previous layer by applying a function to the input values. The vector of weights and a bias will be the factors determining which function will be applied to the input.

Learning in a neural network is make by making incremental adjustments to the biases and weights. As the neural network learns more about what kind of input data leads to which kind of output, the network adjusts the weights based on any errors in categorization that the previous weights resulted in. This is called training the neural network.

3.6.2 Activation Function

An activation function is an extremely essential component of the artificial neural networks. It is what decide whether a neuron should be activated or not and whether the information that the neuron is receiving is relevant for the given information or should it be ignored. Activation function is a non linear transformation that is done over the input signal then the transformed output will be further sent to the next layer of neuron as the input.

This neural networks are often applied to supervised learning problems with the training process involves adjusting the parameters or the weights and biases in order to decrease the error ratio of the output predicting model. The availability to adjust the model provide a flexibility allowing MLP to be applied to many categories of problem.

3.6.3 Convolutional Neural Networks (CNN)

A Convolutional Neural Networks is mostly known for its efficiency for solving problem involving image data as an input. CNN has an ability to develop an internal representation of a two-dimensional input for example an image. This allows the model to learn the pattern of positions of a result determining factors within a variant structures of data. CNN consists of input and output layer as well as the multiple hidden layers similar to MLP but with CNN the hidden layers will mostly comprise of convolutional layers, pooling layers, fully connected layers, and normalization layers.

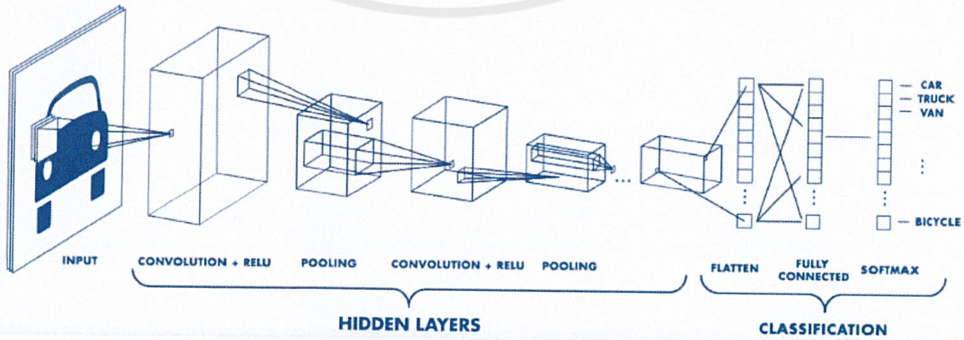


Figure 3.3: Convolutional Neural Networks Model [13].

Convolutional

Convolutional layer is the main layer for extracting features from an input by applying a convolution method that utilize a pre-construct filter to perform a mathematical operation on the input set of values with that particular filter and resulting in a smaller but recognizable representation before passing that result to the next layer.

Pooling

Pooling layers are responsible for reducing the number of parameters or dimensions of a larger input to be represented by a much smaller scale, depending on the type of pooling the result representing the output will differ for example Max pooling will take the largest value from the rectified features to represent that group of values then pass it to the next layer.

Fully Connected

Fully connected layer or FC layer act as a tying knots that will use the vector from the flattened data then connect all the neurons in one layer to every neurons in another layer. The purpose of FC layer is to detect specific global configurations of features that are detected by the lower layers in the network.

3.7 Client-Server Architecture

Client-server architecture is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requester called clients.

3.7.1 Client Mobile Application

React Native

React Native is a JavaScript framework for writing natively rendering mobile applications for iOS and Android. It is based on React JavaScript library for building user interfaces that target mobile platform. React Native applications are written using a mixture of JavaScript and XML-esque markup, known as JSX then React Native bridge will invokes the native rendering APIs in Objective-C(for iOS) or Java(for Android) to render using real mobile UI components

3.7.2 Server

Flask

Flask is a micro web framework written in Python and it is classified as a microframework because it does not require particular tools or libraries.

3.8 Fuzzy Logic

Fuzzy logic is a method of reasoning by trying to imitate reasoning ways of human. The principle of Fuzzy logic is the process of decision making based on intermediate possibilities between digital values True and False.

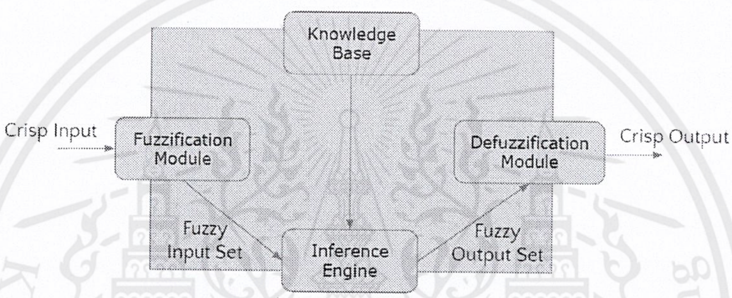


Figure 3.4: Fuzzy Logic Systems Architecture [20].

Fuzzification Module

transforms the system inputs, which are crisp numbers, into fuzzy sets ,for example, large positive, medium positive, small, medium negative, and large negative

Knowledge Base

This process stores IF-THEN rules provided by experts.

Inference Engine

This process simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.

Defuzzification Module

This process transforms the fuzzy set obtained by the inference engine into a crisp value.

3.9 Dynamic Time Warping (DTW)

DTW is an algorithm which is able to measuring similarity between two temporal sequences which may vary in speed. It can find similarity of a sample voice with other a template voice even if the voice is faster or slower. DTW can be used to analyse any data in a form of linear sequence by calculating an optimal match between two given sequences with certain restrictions.

$$dtw(x, y) = (\sum_{i=1}^{N} |x_i - y_i|^2)^{\frac{1}{2}} \tag{3.5}$$

where x and y are sequences, and N is the length of the sequence.

3.10 Sum of Absolute Differences (SAD)

SAD is an algorithm which is able to measure the similarity between two 2-dimensional arrays. The result can be obtained by calculating the absolute difference between each element in the template array and the sample array being used for comparison.

$$sad(A, B) = \sum_{i=1}^{N} \sum_{j=1}^{M} abs(A_{ij} + B_{ij}) \tag{3.6}$$

where A and B are two-dimensional arrays, and N and M are the number of rows and columns respectively.

Chapter 4
Methodology

This chapter discusses about how the entire process are done starting from input signal to pre-processing, feature extraction, classification, and the assessment. In order to be able to relate all the procedure together this section will summarize all the methods being used and what the result system will represent.

4.1 Proposed Method and System Overview

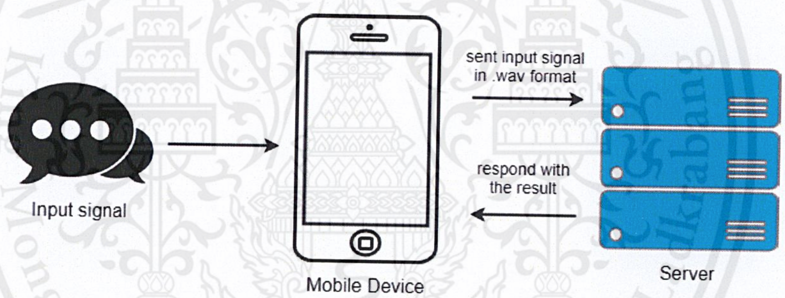


Figure 4.1: Client-server architecture of the proposed system.

The result of this research will be present using a system comprise with client-side and server-side, in a simple client-server architecture seen in Figure 4.1. The mobile application will be used to input user's voice signal then sent to the server in the form of a .wav file. After the server finished processing the input signal, it will respond the analyzed result back to the mobile application and desktop application.

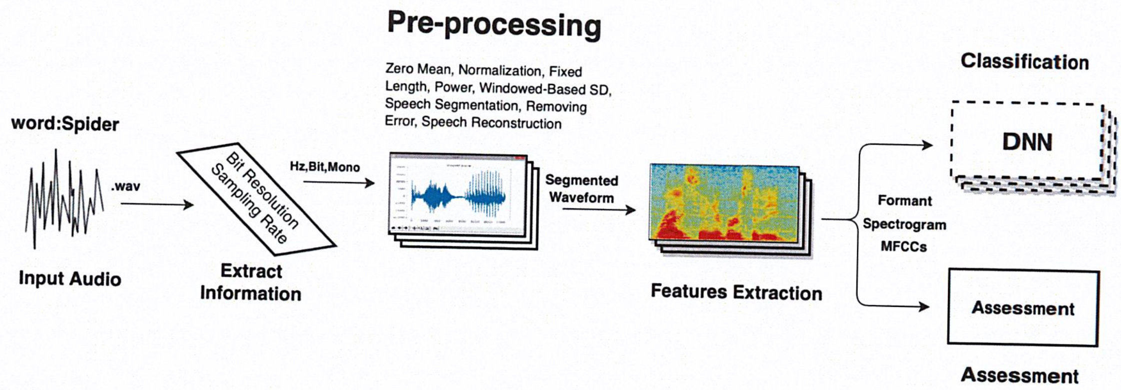


Figure 4.2: System overview of the proposed method.

In Figure 4.2, shows the entire processing method located in the server-side. The process start with receiving an audio from the speaker and extracting its information. Then the received audio is being pre-process into an acceptable signal meaning the time, frequency, loudness, and other factors will be adjusted equally. Next, the segmented waveform will be extracted into three features, Spectrogram, MFCC, and Formant Frequency. Out of the three features, one will be choose to later on feed into classification model using deep learning approach and used in speech assessment.

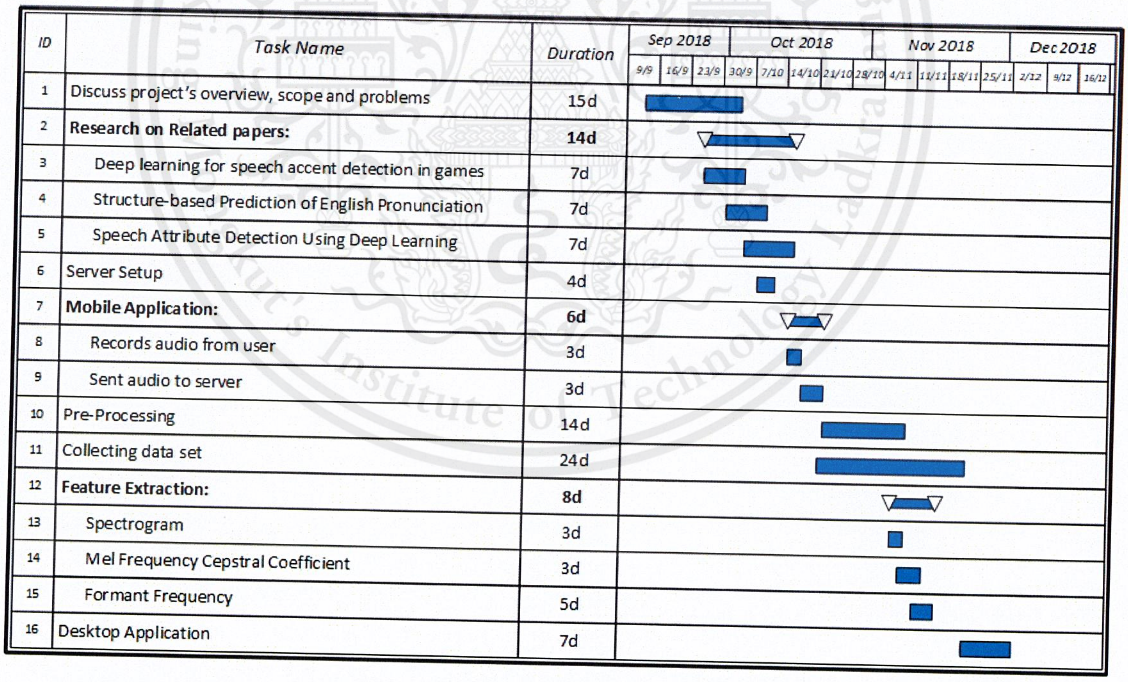


Figure 4.3: Gantt chart semester 1

In Figure 4.3 the main tasks for this proposed method is introduce with a gantt chart to visualized to time period needed to finalized the intended result from the research.

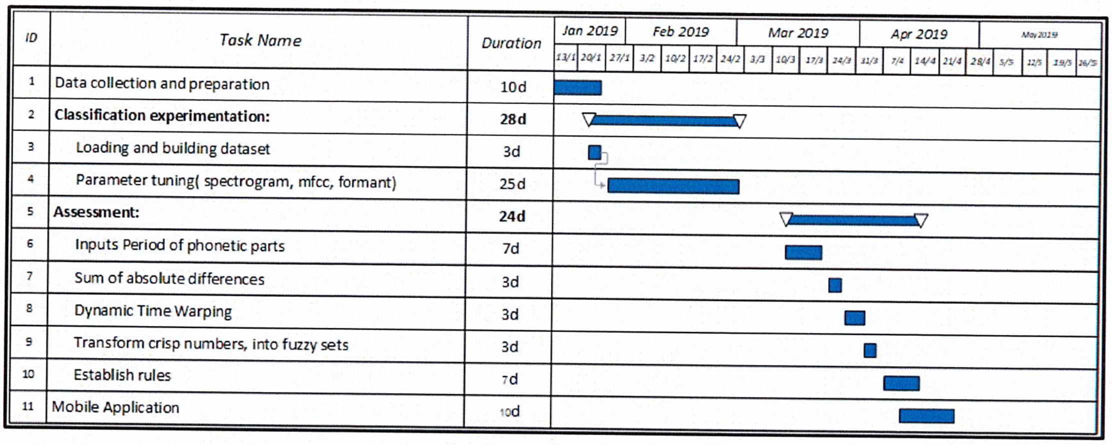


Figure 4.4: Gantt chart semester 2

In Figure 4.4 the main tasks for this proposed method is introduce with a gantt chart to visualized to time period needed to finalized the intended result from the research.

4.2 Pre-processing

Pre-processing is a technique to adjust raw audio signals from users, since, these signals can be different in time, frequency, and loudness before using the signals in the other steps.

4.2.1 Zero mean

This process is to make an average of a signal to become zero by subtracting each data in the signal by its population mean which is calculated by

4.2.2 Normalization

This process is to adjust the amplitude of a signal to be a normal form by dividing the data of the signal by its maximum value.

4.2.3 Fixed-Length

This process is to confine a length of data within a signal, the determined length will be considered based on the maximum length of a word in the example words.

4.2.4 Power

This process finds the power of a signal by power every data in the signal by two to identify which part of a signal is a spoken part.

4.2.5 Window-based Standard Deviation

The process of Window-Based SD is based on finding the range of standard deviation of the information through the length of the window, which is 256 point. With each point it will be used to find the standard deviation. Finding these signals and points could point out which part of the sound is actually a sound signal based on the standard deviation.

4.2.6 Thresholding

Thresholding is a process that partition the audio file into parts that have wanted parts and incorrect parts. When the SD is lower than threshold, it is considered to be an incorrect part but when it is higher, it is considered to be a wanted part.

4.2.7 Speech Segmentation

Speech segmentation is a process that separates the audio file into parts that have considered to be wanted parts and incorrect parts. The selection is decided by the standard deviation and threshold, if the standard deviation is higher than the threshold then it is considered to be a speech. On the contrary, if the standard deviation is lower than the threshold, it is considered to be a non-speech.

4.2.8 Removing Error

Removing error is a process in which the error in the segmentation part will be eliminated. The error list contains [0, 1, 0] and [0, 1, 1, 0]. In order to find the error part, we iterate through all data signal and exclusive or (XOR). If the set of data matches the error template, the set of data will be set to be 0.

4.3 Feature Extraction

For this proposed method a procedure of feature extraction is required to extract outstanding features from a pre-processed signal to form a two dimensional array with a unique characteristic. We decided to use three types of feature including Spectrogram, Mel-frequency cepstral coefficients (MFCC), and formant frequency, to characterize each signal.

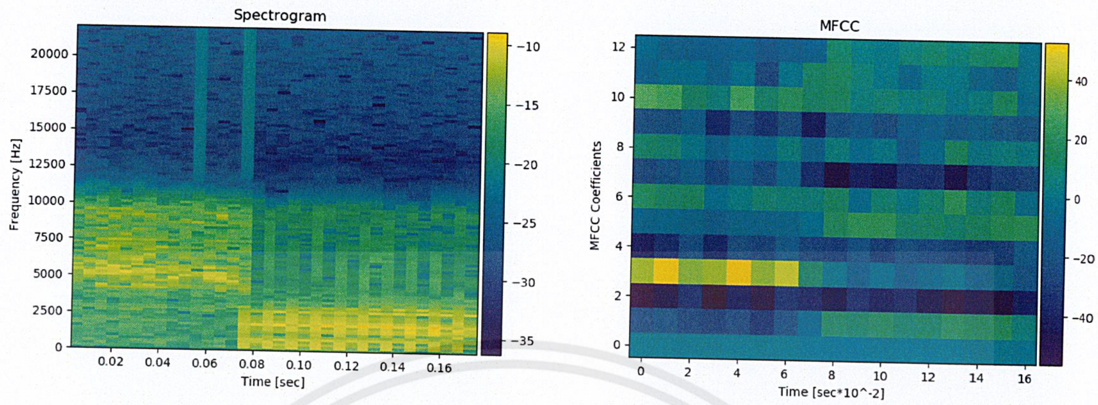


Figure 4.5: Example of spoken word “snack” plotted in spectrogram and MFCC domains

The procedure of feature extraction is then used on the pre-processed signal to generate two representation of features, the spectrogram which represents the signal strength over the time duration and MFCC which represents the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. By using these two-dimensional graph to represent the signal it can be pass to the neural network model as both two-dimensional array data and image data. All the representation is image format is the size of 640x480 pixels

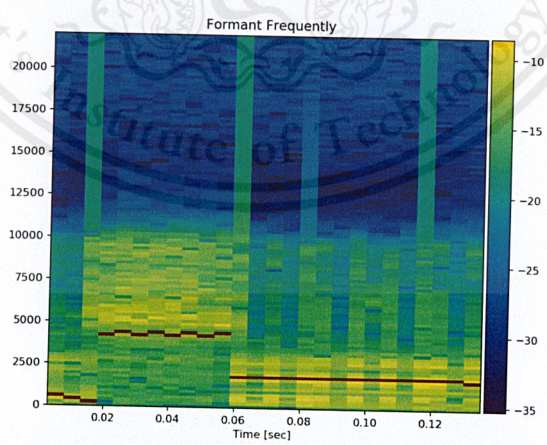


Figure 4.6: Example of F0 formant frequency plotted on top of spectrogram

The third feature, the formant frequency is extracted from the value of the two-dimensional array of spectrogram and can represent a concentration of acoustic energy around a particular frequency in the signal which in other word indicate the peak and subsequent value in each period of the domain. Commonly the formant frequency is denote by F0 as the highest peak or the point in which the highest concentration of energy is at a given time then F1 is considered second highest point and so on.

4.4 Data Preparation

The dataset that are selected to used was from the OSCAAR and it contained a scripted reading scenario in which participants clearly enunciated a scripted list of words one at a time. This was useful in the pre-processing step where we segmented out individual word utterances as separate audio clips from the original speech recording, producing many word-level utterances for us to perform further feature extraction from.

Other stimuli used in the current study were sentences selected from the Hoosier Database of Native and Non-native Speech. This database contains digital audio recording of native and non-native speakers of English reading words, sentences, and paragraphs.

Table 4.1: Number of audios for each category

Category	**Number of audios**
Native	2200
French	2150
German	1650
Mandarin	2200
Spain	2200
Japanese	2200
Korean	2200
Thailand	2195
Total	16995

As the dataset from Hoosier database consists of predominantly French, German, Japan, Korean, Mandarin, Spanish, and English native language speakers, with addition of Thai accent collected by ourselves. We decided to use these 8 as our accent classification task classes.

Table 4.2: Hoosier Database of Native and Non-Native Speech

Items	Tasks
160	Hearing in Noise Test for Children sentences
10	Digit words
48	Multi-syllabic Lexical Neighborhood Test words
50	Northwestern University-Children's Perception of Speech words
100	Lexical Neighborhood Test words
50	Lexical Neighborhood Sentence Test sentences
40	Pediatric Speech Intelligibility sentences
20	Pediatric Speech Intelligibility words
339	Bamford-Kowal-Bench sentences
150	Phonetically Balanced Kindergarten words
72	Spondee word
100	Word Intelligibility by Picture Identification words

The Hoosier Database of Native and Non-native Speech comprised of 27 total talkers of the mentioned 7 native language backgrounds each produced a total of English 1139 recordings in the tasks shown in 4.3.

Thai Accent

This additional accent is included both to explore a solution of a system specified to help Thai native learn other languages and to measure the impact on how a data which originate from a similar process but vary in variables and environment could affect the outcome when using deep learning approach.

The participants are Thai native individuals with background that does not affiliated with environment which could affect their pronunciation such as studied in international school since child or participated in any long period activities that interact with foreigner on a daily basis. As for the environment when recording we used a closed-area with no noise present and recorded using 20-20000 Hz of frequency response microphone to record.

4.5 Classification

The approach we used is by applying deep learning technique, we can create a based model references of pronunciation of words that will act as a classifier to classify, analyze, and assess the word or speech input pronunciation. This proposed method utilized neural networks model which is the CNN. Data that is used to train the deep learning model are the attribute features we extracted from the pronunciation sound of a word or speech with a corresponding spectrogram, MFCC, and formant frequency.

4.5.1 Data Segmentation

The data set obtained from feature extraction are the spectrogram, MFCC, and formant frequency which are used separately to train a different classification model in order to compare whether which feature in which format can represent a pronunciation of an accent with the highest accuracy from the deep learning perspective of a specific structure of a neural network.

Items included in the data preparation all have its own three features extracted and will have a representation of a two-dimensional graph in a form of image data in both color image and gray-scale image and a vector of values that represented those particular two-dimensional graph.

4.5.2 Model Training

The neural network is fed with the three types of feature each with multiple different parameters in order to archive the highest possible accuracy with each of the feature in particular when working with deep learning.

Table 4.3: Parameters of data for model training

Parameter	Values
Pre-processing Threshold	0.01, 0.05, 0.005
Image Size	128x48, 64x48, 28x28
Image Type	gray-scale, colored

All three data are fed to neural network in image data format with multiple values of parameters separately with formant frequency having an addition one-dimensional array representation as well. Those data are used to generate a corresponding classification models and tested whether which type of feature can yield the highest accuracy that could be archived with this data set.

Utilizing both classification models, an input signal could be characterize into the respective accent that the particular signal is closest to and with enough data set the system should be able to identify the gender of the speaker as well. While the experiments is done on multiple types of feature in multiple format the result system utilized only the one that yield the highest accuracy in prediction.

4.5.3 Predictive Model

The input will be classify into native and non-native pronunciation, for non-native pronunciation it will be further classify into one of the seven accents the input is closest to. The accents used within this research are British, French, German, Korean, Japanese, Mandarin, Spanish, and Thai. The result obtained from the predictive model will be the percentages of each accent of how closely related that particular accent is with the input signal with all percentages where all percentages summarize up to one hundred percent. After the classification with predictive model the input signal will be further process and analyze in order to assess the input pronunciation base on the the result from the processing and analyzing.

4.6 Assessment

Assessment is used for evaluating an audio signal that a user speaks using Fuzzy Logic System in order to receive score as percentage and to evaluate whether the input pronunciation is acceptable in casual communication level or not.

4.6.1 Features

Three features including Sum of Absolute differences (SAD), Period of Phonetic parts (PoP), and Dynamic Time Warping (DTW) are used as crisp inputs of the fuzzy logic system.

Sum of Absolute Differences (SAD)

This approach focuses on receiving the result by calculating SAD between the sample and template two-dimensional arrays. The arrays can be obtained by extract features of spectrogram from the signals and used that spectrogram as the arrays. However, this feature was later deducted to use in fuzzy logic system due to its fluctuated value.

Period of Phonetic Parts (PoP)

This approach is firstly to segment phonetic parts of the sample signal using pre-processing method. The result of PoP is the time (in seconds) of each sample segmented part deducted by each template segmented part. In segmentation process, solely one threshold could not provide us a suitable segmentation result. Thus, Adaptive thresholding method is used in this process which covers thresholds from 0.001 to 0.017. The ranging domain of PoP is from 0 to 1 second.

Dynamic Time Warping (DTW)

This approach is to extract features of MFCCs from a template signal and a sample signal. Following by measure similarity between temporal sequences using Dynamic time warping (DTW) in order to get a result as normalized distance. The ranging domain of DTW is from 100 to 400.

4.6.2 Fuzzy Logic System

Fuzzy Logic System is a system using fuzzy logic to produce acceptable but definite output in response to ambiguous inputs. Fuzzy logic is applied in the system to accept three features of assessment and produces the result in percentage as a feedback for a user.

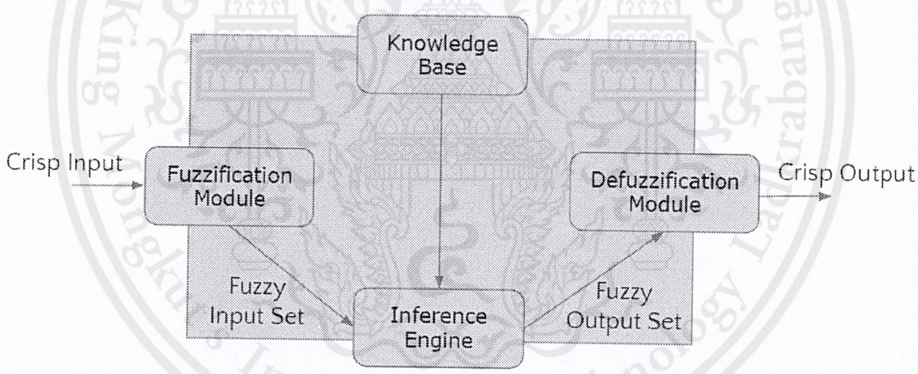


Figure 4.7: Architecture of Fuzzy Logic System

Fuzzy logic comprises of four components including fuzzification module, knowledge inference engine, and defuzzification module.

Fuzzification Module

In this module, the extracted features are used as crisp inputs. The module later generates membership functions for each crisp input. DTW and PoP as crisp inputs are first converted into the scale of 0 to 100 according to its ranging domain in order to use in the system.

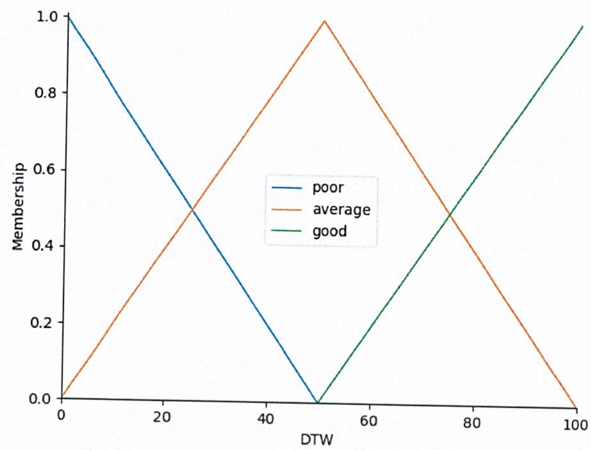


Figure 4.8: Membership function of DTW

In figure 4.8, Dynamic Time Warping (DTW) is converted into three sets of fuzzy due to the large boundary of its value including poor, average, and good.

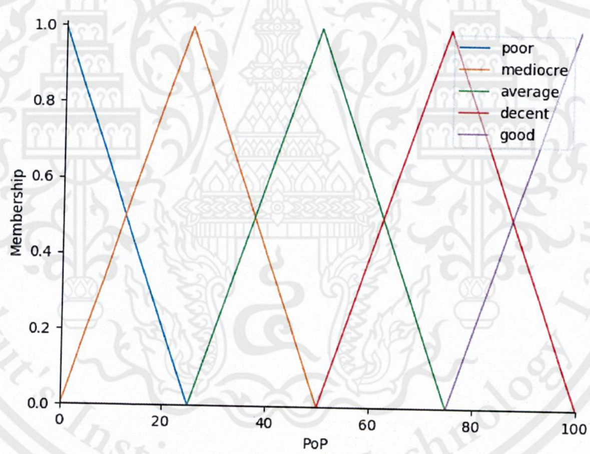


Figure 4.9: Membership function of PoP

In figure 4.9, Period of Phonetic Parts (PoP) is converted into five sets of fuzzy since its value can distinguish the range of similarity better including poor, mediocre, average, decent, and good.

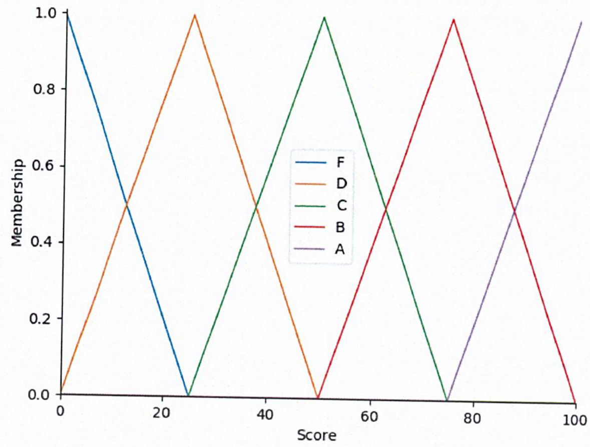


Figure 4.10: Membership function of Score

In figure 4.10, The membership function of score is composed of five sets of fuzzy indicating grades obtained after the calculation process including A to F grading scale.

Knowledge Base

DTW \ PoP	Poor (0 - 20)	Mediocre (20 - 40)	Average (40 - 60)	Decent (60 - 80)	Good (80 - 100)
Poor (0 - 33.33)	F	F	D	D	D
Average (33.33 - 66.66)	C	C	B	A	A
Good (66.66 - 100)	C	B	A	A	A

Figure 4.11: Rules table

This part stores IF-THEN rules as shown in figure 4.11 which is a matrix of DTW values versus PoP values that the system is expected to provide. The rule can be derived as, for example, IF DTW = (Good) AND PoP = (Poor) THEN Score = (C).

Inference Engine

This process proposes membership functions using triangle function and related linguistic variables in order to find the value of every membership function of inputs.

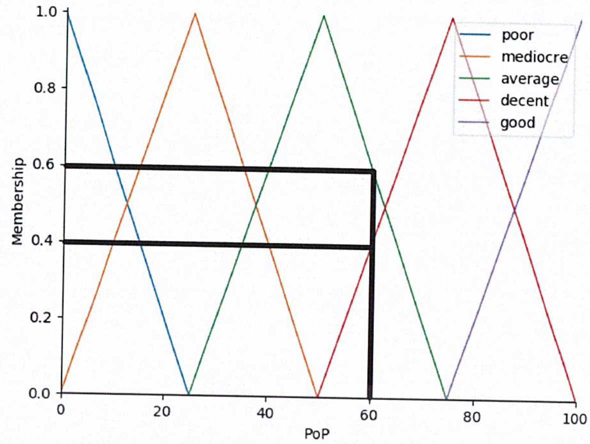


Figure 4.12: Value mapping in PoP membership function

In figure 4.12, the example value of PoP is 60. The results are the values of membership which can be obtained by looking at the intersections of triangle functions and PoP value. In this case, the results are 0.6 of average and 0.4 of decent.

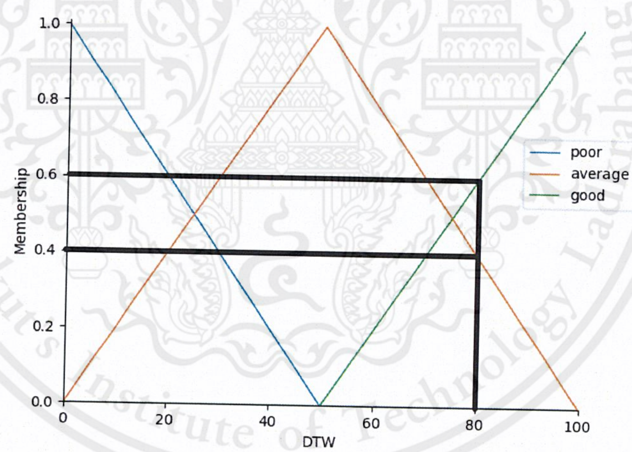


Figure 4.13: Value mapping in DTW membership function

In figure 4.12, the example value of DTW is 80. The results also are the values of membership. In this case, the results are 0.4 of average and 0.6 of good.

DTW = 80

poor	0
average	0.4
good	0.6

PoP = 60

poor	0
mediocre	0
average	0.6
decent	0.4
good	0

Figure 4.14: Results of Fuzzy Values of PoP and DTW

In figure 4.14, the table shows the result obtained in the form of table. the next process is to evaluate two fuzzy values using rules from knowledge base. According to the example result in figure 4.14, only average, good in DTW, and average, decent in PoP will be used in rule evaluation.

Membership Value of DTW	Membership Value of PoP	Grade	Membership Value of Grade
average = 0.4	average = 0.6	B	0.4
average = 0.4	decent = 0.4	A	0.4
good = 0.6	average = 0.6	A	0.6
good = 0.6	decent = 0.4	A	0.4

Figure 4.15: Result of rule evaluation

In figure 4.15, grades are obtained regarding the table of knowledge base and values are obtained according the rule. For example, Fuzzy values of DTW and PoP are 0.4 of average and 0.6 of average respectively. According to the rule table in figure 4.11, we can derive that IF DTW = (Average) AND PoP = (Average) THEN Score = (B) and AND operator means to find a minimum. Thus, the membership value of grade is min(0.4, 0.6) which is 0.4.

Defuzzication Module

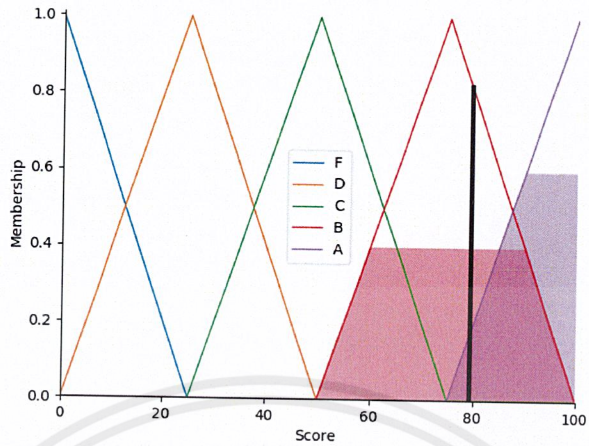
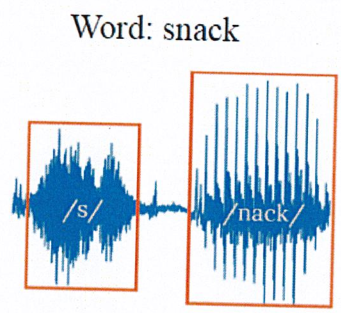


Figure 4.16: Score of membership function after evaluation

Each value obtained from evaluation is the value of membership which can be represented in each grading area. In order to find the result, centroid estimation which is a calculation to find the centre area of a graph is used to compute the result. Therefore, the result for this example is approximately 79.39.

4.7 Feedback

Feedback is used for providing extra information of how good a user speaks comparing with a template signal using Time Duration feedback.



	/s/	/nack/
Time Duration	-0.5 s	+0.7 s
Frequency	-12 Hz	+20 Hz

Figure 4.17: Example of a feedback provided of the word 'Snack'

4.7.1 Time Duration

This provides the information about the time differences of every extracted phoneme from a speech between an audio signal that a user speaks comparing with a template signal. Each phoneme duration is displayed in second with minus duration representing the pronunciation being short by that duration and plus being the vice versa. This information is derive from the PoP procedure and is the exact value that used for assessing the pronunciation.

Chapter 5
Implementation

5.1 Development Tools

The system is implemented in Python via PyCharm 2018.1. Various libraries were used during the implementation as shown in Table 5.1. The development and experimental were conducted on personal computer with the following specification: Intel 1151 Core i5-8400 CPU 2.80 GHZ, 24 GB of RAM, and 64-bit operating system.

Library	Version	Purpose
Tensorflow	1.13.1	Machine learning library
Keras	2.24	High level neural network API written in Python
SciPy	1.10	Data Management and Computation
PythonSpeechFeature	0.6	MFCCs and filterbank energies
Matplotlib	3.0.0	Plotting library which produces figures
Flask	1.0.2	Provides RESTful request dispatching
Numpy	1.15.2	Fundamental package for scientific computing

Table 5.1: Libraries used

5.2 Pre-processing

Before the input signal can be extract to get the outstanding features of each input, various methods are used to pre-process the raw audio input from user to adjust the data that contain unequivalent circumstances such as the different in time duration, frequency, and loudness.

Methods that were used are Zero mean for turning an average of an audio signal to become zero, Normalization for adjusting the amplitude of the signal, Fixed-Length to confine the length of data of a signal, Power method to identify the spoken part of a signal, Window-based Standard Deviation for identifying signals and points that point out which signal is actually a sound signal, Thresholding to partitioning the audio file, and Speech Segmentation process for separating wanted parts from unwanted parts.

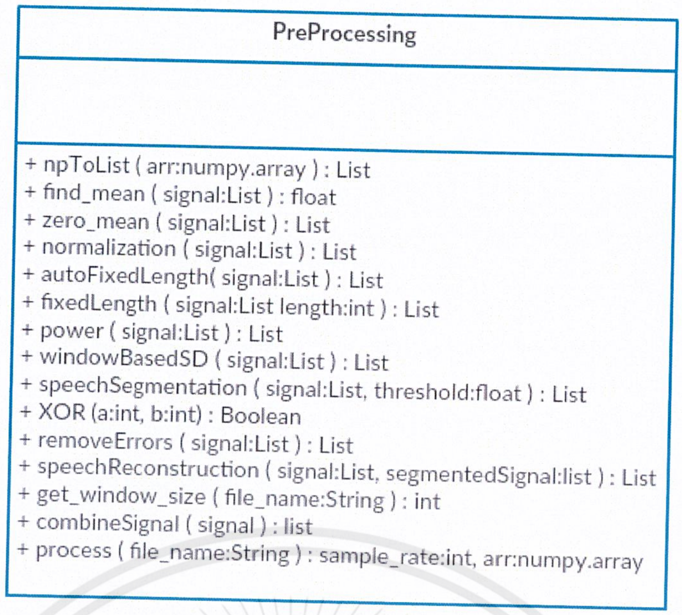


Figure 5.1: Pre-processing

The class diagram above shows the overview of how pre-processing methods are implemented.

5.2.1 Zero mean

Algorithm 1 Zero Mean

procedure ZERO MEAN($signal\_data$)
    $total \leftarrow 0$
    for each *signal* in signal_data **do**
        $total \leftarrow total + signal$
    $mean \leftarrow \frac{total}{\text{the length of signal\_data}}$
    for each *signal* in signal_data **do**
        $signal \leftarrow signal - mean$
    return $signal\_data$

5.2.2 Normalization

Algorithm 2 Normalization

```
procedure Normalization(signal_data)
    max_signal ← signal[0]
    for each signal in signal_data do
        if signal > max_signal then
            max_signal ← signal
    for each signal in signal_data do
        signal ← signal / max_signal
    return signal_data
```

5.2.3 Fixed-Length

Algorithm 3 Fixed Length

```
procedure FIXED_LENGTH(signal_data, length)
    if the length of signal_data > length then
        return array of [0 to length] of signal_data
    else if the length of signal_data < length then
        N ← length - the length of signal_data
        for i ← 0 to N do
            Append 0 into signal_data
    return signal_data
```

5.2.4 Power

Algorithm 4 Power

```
procedure Power(signal_data)
    new_signal ← An empty array
    for each signal in signal_data do
        temp_signal ← signal^2
        Append temp_signal into new_signal
    return new_signal
```

5.2.5 Window-based Standard Deviation

Algorithm 5 Window-based Standard Deviation

procedure Window-based Standard Deviation($signal\_data$)
    $new\_signal \leftarrow$ An empty array
    $length \leftarrow$ The length of signal_data
    for $i \leftarrow 0$ to $length$ by 256 **do**
        $mean \leftarrow$ Find mean of signal_data between [i to i + 256]
        $total \leftarrow 0$
        for $j \leftarrow$ to $i + 256$ **do**
            $total \leftarrow total + (signal[j] + mean)^2$
        $SD \leftarrow \sqrt{\frac{total}{256}}$
        Append SD into new_signal
    return new_signal

5.2.6 Speech Segmentation

Algorithm 6 Speech Segmentation

procedure Speech Segmentation($signal\_data, threshold$)
    for each *signal* in signal_data **do**
        if signal > threshold **then**
            $signal \leftarrow 1$
        else
            $signal \leftarrow 0$
    return $signal\_data$

5.2.7 Removing Error

Algorithm 7 Removing Error

```
procedure Removing Error(signal_data)
    errors ← An array of [0, 1, 0] and [0, 1, 1, 0]
    length ← The length of signal_data
    for each error in errors do
        error_length ← The length of error
        for i ← 0 to length − error_length + 1 do
            isError ← True
            for j ← 0 to error_length do
                if error[j] XOR signal_data[i + j] equals to 1 then
                    isError ← False
            if isError is True then
                for j ← 0 to error_length do
                    signal_data[i + j] ← 0
    return signal_data
```

5.2.8 Speech Reconstruction

Algorithm 8 Speech Reconstruction

```
procedure Speech Reconstruction(signal_data, segmentedSignal)
    new_signal ← An empty array
    segmentedSignal_length ← the length of segmentedSignal
    for i ← 0 to segmentedSignal_length do
        if segmentedSignal[i] equals to 0 then
            for j ← i * 256 to i * 256 + 256 do
                signal_data[j] ← 0
    for each signal in signal_data do
        if signal not equals to 0 then
            Append signal into new_signal
    return new_signal
```

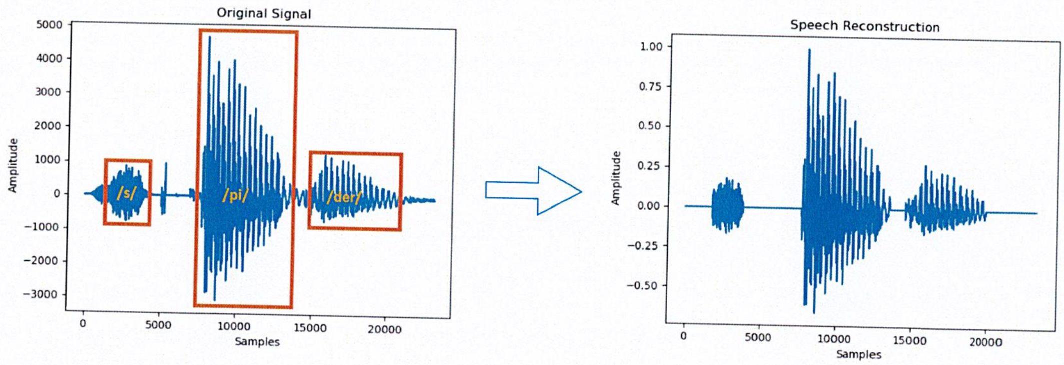


Figure 5.2: The original signal before and after pre-process

5.3 Feature Extraction

Using the pre-processed data, the outstanding features of the input can be extracted more accurately and with those features, a one-dimensional and two-dimensional array each with its own unique characteristic can be created. There are three types of features representation that were used which are Spectrogram, Mel-frequency cepstral coefficients (MFCC), and Formant Frequency. By using the mentioned representation labeled with the necessary information, it could be use as a data set for training the classification model using deep learning approach. Furthermore it can be used for assessment.

5.3.1 Spectrogram

For the implementation of spectrogram, we decided to use Scipy library to generate using scipy.signal.spectrogram function in which its parameters and returns are shown below. The input value of x is a segmented signal data which we obtain from our pre-processing approaches. The results achieved from the function are later plotted in a graph.

Table 5.2: Parameters table of scipy.signal.spectrogram [17]

Parameter	Type	Input	Description
x	array	processed_signal	Time series of measurement values.
fs	float	44100	Sampling frequency of the x time series
window	string,tuple,array	'tukey',0.25	Desired window to use

nperseg	int	None	Length of each segment
nooverlap	int	None	Number of points to overlap between segments
nfft	string	None	Length of the FFT used
detrend	string,function,false	'constant'	Specifies how to detrend each segment
return_onesided	boolean	True	Return a one-sided spectrum if True
scaling	density,spectrum	'desnity'	Selects between computing the power spectral density an power spectrum
axis	int	-1	Axis along which the spectrogram is computed
mode	string	'psd'	Defines what kind of return values are expected

Table 5.3: Return table of scipy.signal.spectrogram [17]

Return	**Type**	**Description**
t	ndarray	Array of sample frequencies
t	ndarray	Array of segment times
Sxx	ndarray	Spectrogram of x

5.3.2 Mel-frequency cepstral coefficients (MFCCs)

For the implementation of MFCCs, we decided to use Python_Speech_Features library to generate using python_speech_features.base.mfcc function in which its parameters and returns are shown in Table 5.4. The input value of signal is a segmented signal data which we obtain from our pre-processing approaches. The results achieved from the function are later plotted in a graph as same as the spectrogram.

Table 5.4: Parameters table of python_speech_features.base.mfcc[18]

Parameter	Type	Input	Description
signal	array	processed_signal	The audio signal from which to compute features
samplerate	float	44100	The sample rate of the signal we are working with
winlen	float	0.025	The length of the analysis window in seconds
winstep	float	0.01	The step between successive windows in seconds
numcep	int	13	The number of cepstrum to return
nfilt	int	26	The number of cepstrum to return
nfft	int	512	The number of cepstrum to return
lowfreq	float	0	Lowest band edge of mel filters
highfreq	float	22050	Highest band edge of mel filters
preemph	float	0.97	Apply pre emphasis filter with preemph as coefficient
ceplifter	float	22	Apply a lifter to final cepstral coefficients
appendEnergy	boolean	True	If this is True, the zeoth cepstral coefficient is replaced with lof of total frame energy
winfunc	funcion	None	The analysis window to apply to each frame

Table 5.5: Return table of python_speech_features.base.mfcc [18]

Return	Type	Description
mfcc_feat	numpy arry	Each row holds 1 feature vector

5.4 Formant Frequency

For the implementation of formant frequency, the data received from scipy.signal. spectrogram was used for finding the frequency when n^{th} maximum of amplitude occurs. According to the parameter table shown in Table 5.2, f and Sxx which are frequencies and spectrogram data respectively will be used as a parameters of the function and n as n^{th} maximum of amplitude.

Algorithm 9 Formant Frequency

```
procedure Formant Frequency(spectrogram_data, frequencies, nth_of_formant)
    new_signal ← An empty array
    rows ← The length of rows of spectrogram_data
    cols ← The length of columns of spectrogram_data
    for i ← 0 to cols do
        temp_s ← An empty array
        for i ← 0 to rows do
            Append signal[j][i] and index j into temp_s
        sorted_temp_s ← A sorted array of temp_s
        new_signal ← Append frequencies[sorted_temp_s[n]]
    return new_signal
```

Additionally, the three features mentioned are also combined and trained simultaneously in order to explore the possibilities of archiving a higher accuracy with the existing data set. For example, combining Spectrogram with MFCC together then feed it into our neural network.

5.5 Classification Model Training

For the implementation of the training process, we used Keras library to build a classification model from the data set that are prepared by applying the proposed method of pre-process and features extraction. The convolutional neural network architecture is as shown below.

Table 5.6: ConvNet architecture for all CNN

Layer	Filter Size	No. of Filter	Activation	Dropout	Shape
conv2d	3×3	32	ReLU	-	62×46
conv2d	3×3	64	ReLU	-	60×44
maxpooling2d	2×2	64	-	-	30×22
conv2d	3×3	128	ReLU	-	28×20
maxpooling2d	2×2	128	-	-	14×10
dropout	-	128	-	25	14×10
flatten	-	-	-	-	14×10
dense	-	128	-	-	14×10

dropout	-	128	-	25	14×10
dense	-	7	-	-	14×10

Table 5.7: Parameter Table of Conv2d function

Parameter	Type	Description
filters	int	Dimensionality of the output space.
kernel_size	tuple	Specifying the height and width of the 2D convolution
activation	string	Which activation function to use.

Table 5.8: Parameter Table of MaxPooling2D function

Parameter	Type	Description
pool_size	tuple	Factors by which to downscale to.

Table 5.9: Parameter Table of Dense function

Unit	Type	Description
filters	int	Dimensionality of the output space.
activation	string	Which activation function to use.

Table 5.10: Parameter Table of Droupout function

Parameter	Type	Description
rate	float	Fraction of the input units to drop.

5.6 Assessment

5.6.1 Dynamic Time Warping

For the implementation of Dynamic Time Warping, we decided to use dtw module to calculate a normalized distance of two voice signals which is the result of the accumulated cost matrix and the shortest path found. The ranging domain of DTW from the library is between 100 - 400 meaning the lower the more similar. The result is then converted to the scale of 0 - 100 using the equation shown below.

$$final\_dtw = \frac{min(400 - a, 300)}{3}$$

5.6.2 Period of Phonetic Parts

For the implementation of period of phonetic parts, the sample and template signals are segmented using adaptive thresholding and then stored in arrays. If the lengths of both arrays are the same, then find absolute sum of the differences of time.

Algorithm 10 Period of Phonetic Parts

procedure PoP($word\_length, template\_sig, sig$)
    $total \leftarrow 0$
    $new\_template\_sig \leftarrow$ An array of segmented signals after iterative segmentation
    $new\_sig \leftarrow$ An array of segmented signals after iterative segmentation
    $len\_template\_sig \leftarrow$ The length of new_template_sig
    $len\_sig \leftarrow$ The length of new_sig
    if len_template equals to len_sig **then**
        for $i \leftarrow 0$ to $len\_template$ **do**
            $diff \leftarrow$ The length of new_template_sig[i] - new_sig[i]
            $total \leftarrow$ total + diff / 44100
        $total \leftarrow 100 - min(total, 1) * 100$
    else
        $total \leftarrow 100$
    return $total$

5.6.3 Fuzzy Logic System

For the implementation of fuzzy logic system, we used scikit-fuzzy library to build a fuzzy membership functions as well as using centroid estimation method provided in the library in defuzzication process [20].

Chapter 6
Experimentation

6.1 Pre-processing

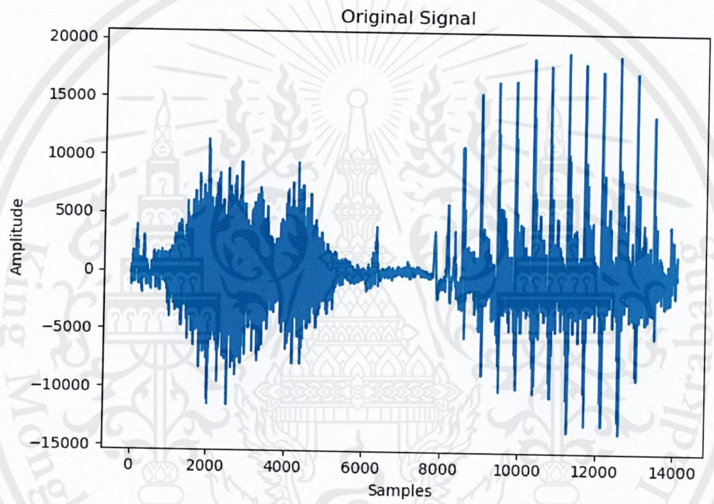


Figure 6.1: Original audio signal in time domain.

In this experiment, the Figure 6.1 above shows a spoken word "snack" pronounced by a native English speaker.

6.1.1 Zero mean

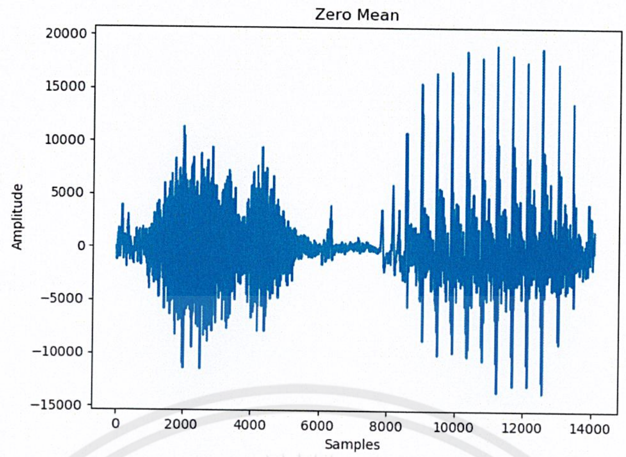


Figure 6.2: Zero Mean.

The mean of the signal data after this process is approximately 0.

6.1.2 Normalization

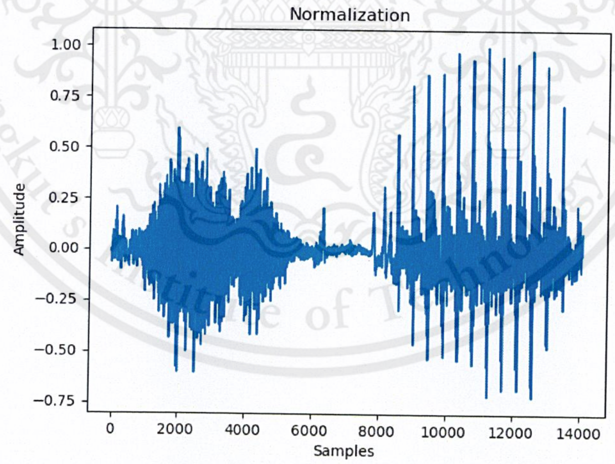


Figure 6.3: Normalization.

The scale of the signal data after this process is between 1.0 and -1.0

6.1.3 Fixed-Length

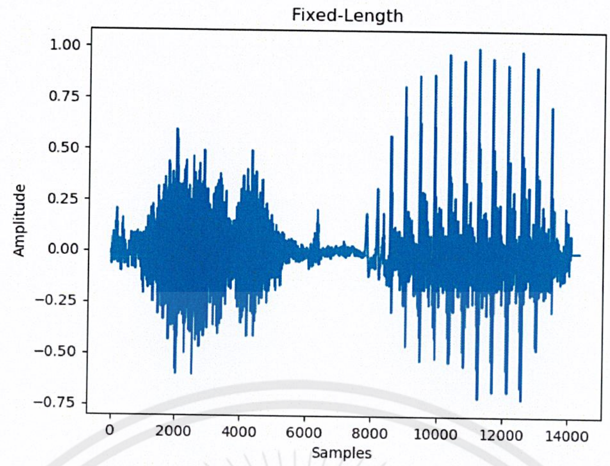


Figure 6.4: Fixed Length of the normalized signal.

The signal data length after this process is zero padding based on 2^n.

6.1.4 Power

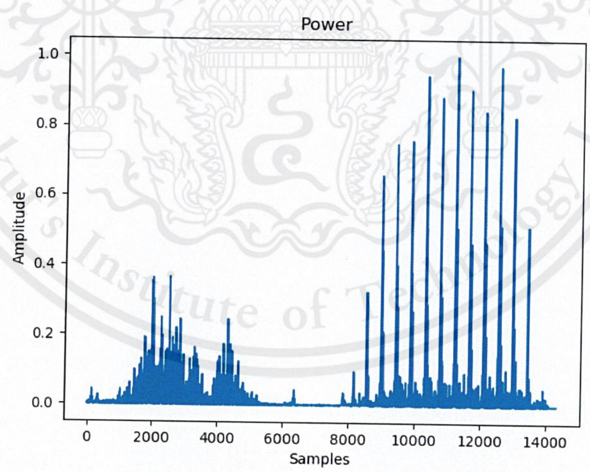


Figure 6.5: Powered signal.

In this process, the signal data is powered by 2.

6.1.5 Windowing-based Standard Deviation

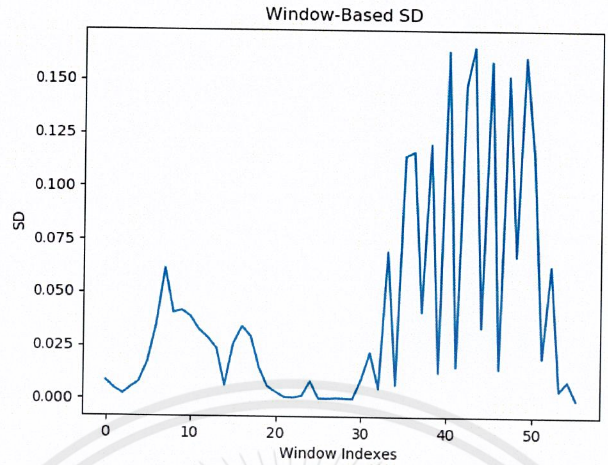


Figure 6.6: Windowing-based standard deviation result.

In this process, the graph shows SD of each 256 samples.

6.1.6 Speech Segmentation

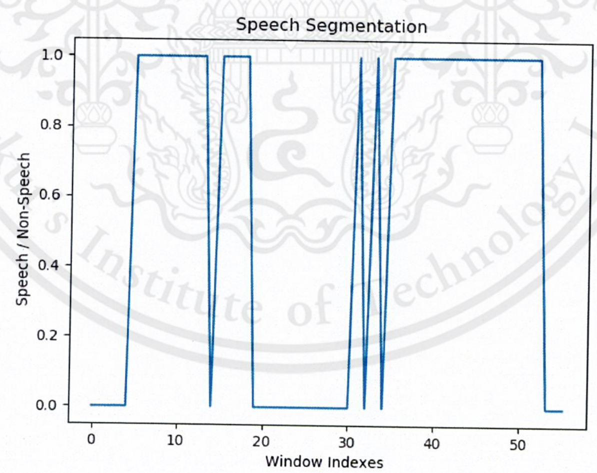


Figure 6.7: Speech Segmentation.

In this process, all noise data is set to 0.

6.1.7 Removing Error

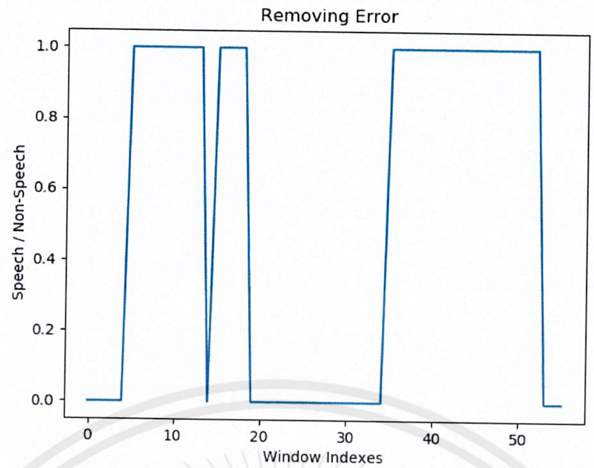


Figure 6.8: Removing Error.

All the errors from the previous segmentation are eliminated in this process.

6.1.8 Speech Reconstruction

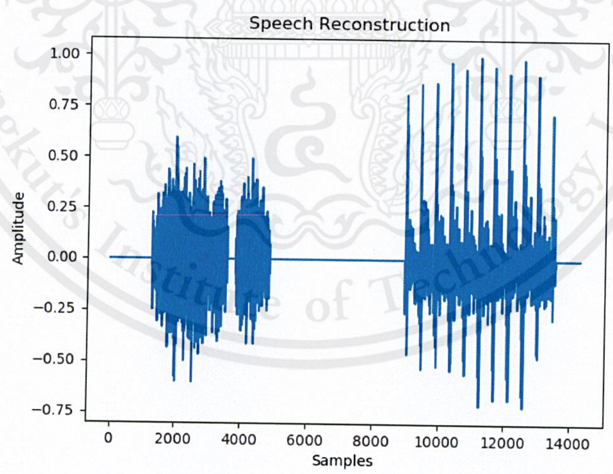


Figure 6.9: Speech Reconstruction.

The signal data is reconstructed with only detected speech parts.

6.1.9 Pre-Processed Signal

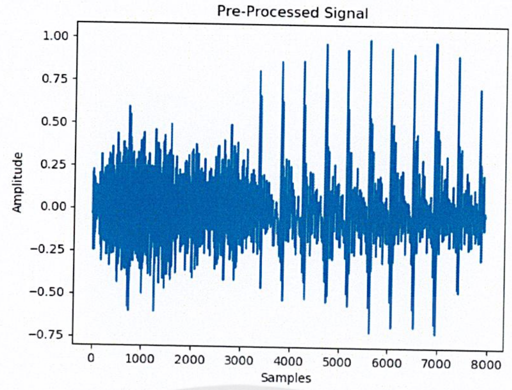


Figure 6.10: Pre-Processed Signal.

In this process, only segmented speech parts are combined together to make a new signal data.

6.2 Feature Extraction

6.2.1 Spectrogram

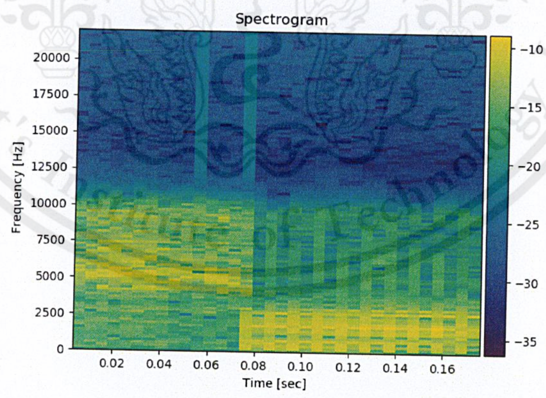


Figure 6.11: Spectrogram.

The Figure reffig:spec, shows the spectrogram of an example word "snack" which will be used in a data set for model training using deep learning approach. All data will have this representation of a signal as an image and vector format.

6.2.2 Mel-frequency cepstral coefficients (MFCCs)

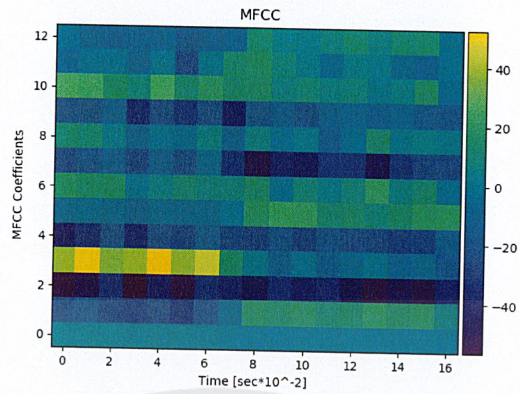


Figure 6.12: MFCC.

The Figure 6.12, shows the MFCC of an example word "snack" which will be used in a data set for model training using deep learning approach. All data will have this representation of a signal as an image and vector format.

6.2.3 Formant Frequency

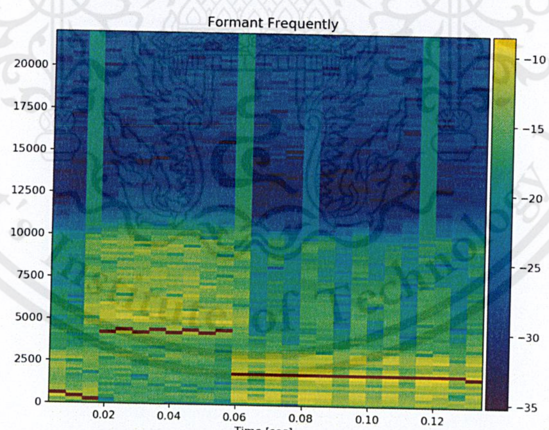


Figure 6.13: Formant Frequency.

The Figure 6.13, shows the Formant Frequency of an example word "snack" which will be used in a data set for model training using deep learning approach. All data will have this representation of a signal as an image and vector format.

6.3 Classification Model Training

6.3.1 Training MFCC Data

As we learned that MFCC can yield a high precision when aiming for classifying accent[12]. We want to explore how our set of data and the pre-processing method would affect the result of classifying data into 8 classes.

Table 6.1: Training result for MFCC with different sizes

	MFCC 28×28		MFCC 64×48		MFCC 128×48	
Iter.	Test acc.	Time(sec)	Test acc.	Time(sec)	Test acc.	Time(sec)
1	40.48	26	45.37	123	45.21	552
5	57.73	130	60.16	612	60.43	2734
10	64.21	261	65.54	1231	63.94	5467
15	67.67	391	69.45	1830	66.40	8251
20	70.29	522	70.72	2421	67.05	11024
30	71.89	781	73.27	3636	67.51	16591

The experiment are done with multiple sizes of image data with the threshold value of 0.005 on pre-processing. And the results obtained are mediocre where the precision reach stagnant at around 0.7 and never going above 0.72 with the highest precision sit at 0.719. Where after adding the the eighth class (Thai accent) the result accuracy does not have any significant change that could be perceived.

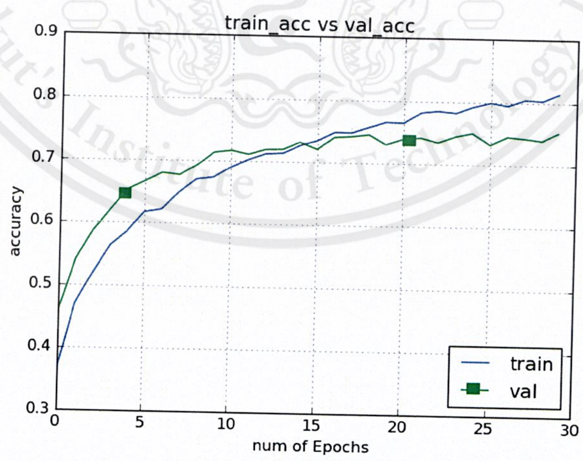


Figure 6.14: Training result for MFCC with 0.005 threshold

The Figure 6.14, shows the average accuracy of all parameters while training the classification model using MFCC data set with threshold value of 0.005 with 30 epoch.

After passing the 15 epochs, the data start over-fitting and training accuracy continue growing while the testing accuracy stop growing and became stagnant.

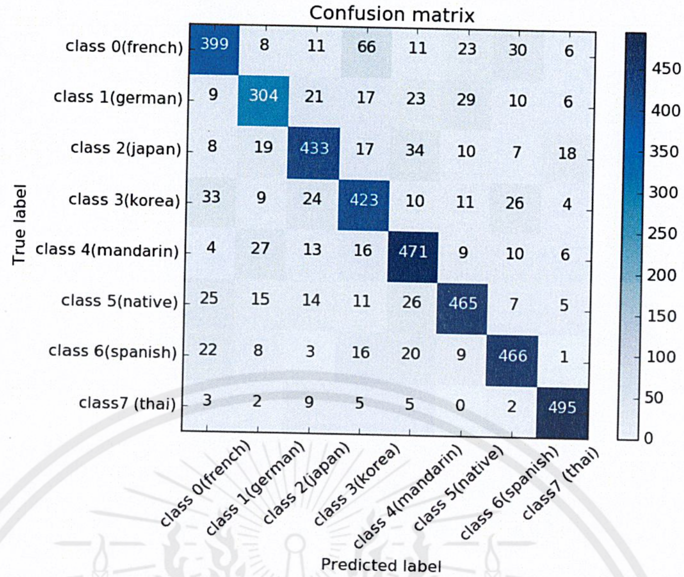


Figure 6.15: Confusion matrix for MFCC with 0.005 threshold.

The figure 6.15, shows the result prediction map to all classes from the classification model trained using MFCC. Where the proportion of test data set is the same between all accent most has the frequency of correctly predicting the accent going above 300 of all test data. The reasoning behind this is due to the differences in the amount of data where the data of German speaker are less that the other accent by roughly 20 percent.

6.3.2 Training Spectrogram Data

As for the spectrogram the experiment was conducted with the same multiple value of parameters as MFCC and with the same threshold value when pre-processing the data.

Table 6.2: Training result for spectrogram with different sizes

	Spec 28×28		Spec 64×48		Spec 128×48	
Iter.	Test acc.	Time(sec)	Test acc.	Time(sec)	Test acc.	Time(sec)
1	34.21	26	38.51	122	38.29	552
5	63.48	130	71.62	615	68.43	2734
10	70.45	260	76.19	1231	77.13	5467
15	75.11	390	78.35	1846	80.84	8251
20	76.83	520	79.97	2459	81.94	11024
30	78.78	780	81.16	3690	82.86	16591

In Table 6.3, the highest precision that the model could archived is at 0.829 while the lowest precision between all parameters derived using MFCC data as a base was still at 0.7878 which are clearly higher than the result obtained from MFCCs.

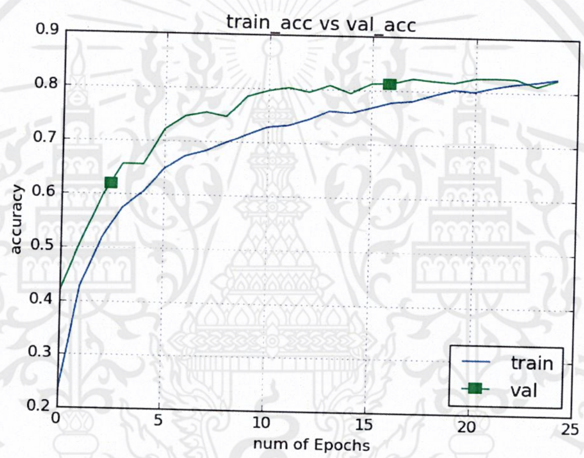


Figure 6.16: Training result for spectrogram with 0.005 threshold

The Figure 6.16, shows the average accuracy of all parameters while training the classification model using spectrogram data set with threshold value of 0.005 with 30 epoch. After passing the 24th or so the accuracy became stagnant.

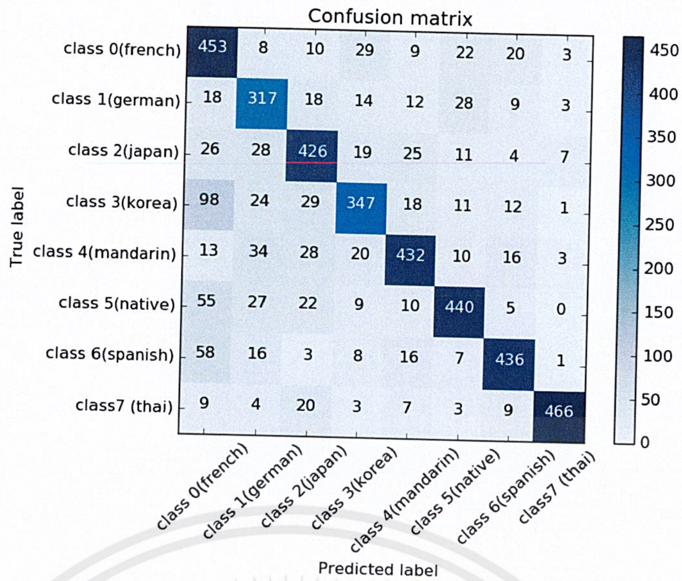


Figure 6.17: Confusion matrix for spectrogram with 0.005 threshold

The Figure 6.17, shows the result prediction map to all classes from the classification model trained using spectrogram. The pattern are the same where all classes have a very high prediction rate only left with German accent.

6.3.3 Training Multiple-Format Data

After completing with all types of data there is a suggestion that with the format of image data there might be possible to archived better precision rate with the same proposed method by combining data together to increase the amount of information the neural network could learn.

Table 6.3: Training result for spectrogram with different sizes

	SpecMFCC 128×48	
Iter.	Testing acc.	Time(sec)
1	57.53	256
5	79.96	1280
10	84.27	2560
15	87.44	3840
20	87.37	5120
30	87.59	7680

The result of training multiple-format of data together are quite surprising where the accuracy escalate to 0.88 as the highest precision when training MFCC and spectrogram together.

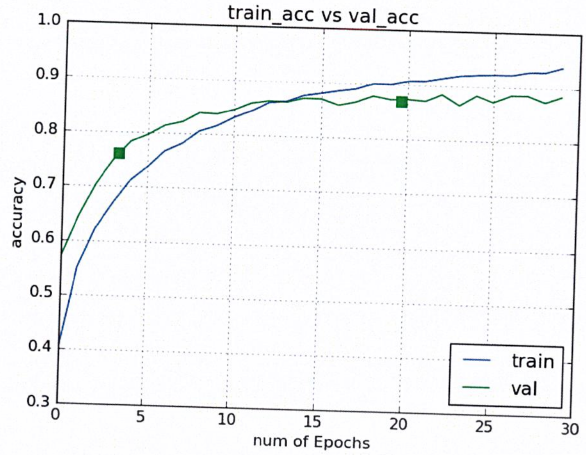


Figure 6.18: Training result for spectrogram and MFCC with 0.005 threshold

The figure 6.18, shows the accuracy while training the classification model using spectrogram and MFCC data set with threshold value of 0.005 with 30 epoch. The prediction accuracy stop increasing upon reaching the 12-13 epoch.

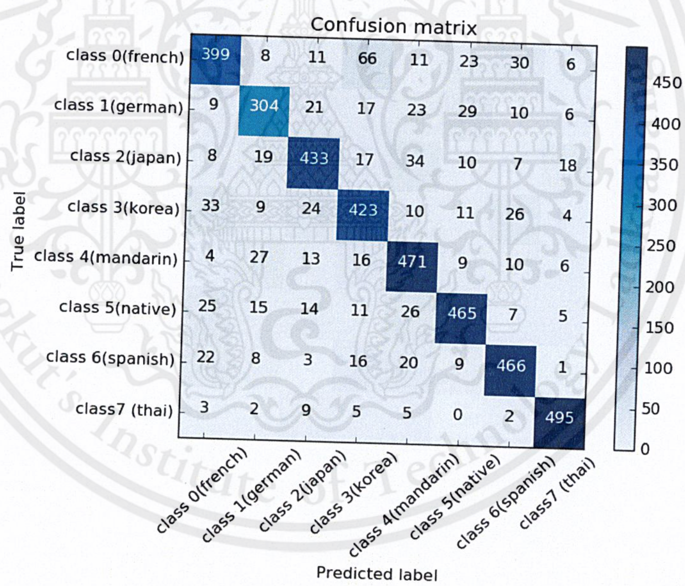


Figure 6.19: Confusion matrix for spectrogram and MFCC with 0.005 threshold

The Figure 6.19, shows the result prediction map to all classes from the classification model trained using spectrogram. Even with the significant increase in accuracy the pattern stayed the same where the German accent staying at the bottom with the test data that consist of each accent with the same amount of data.

6.4 Assessment System

Table 6.4: Assessment table

Template	Sample	Fuzzy score
bread-1.wav	bread-2.wav bread-3.wav bread-4.wav	60.21 78.36 77.52
		mean = 72.03
desk-1.wav	desk-2.wav desk3.wav desk4.wav	64.92 78.73 85.13
		mean = 76.26
flag-1.wav	flag-2.wav flag-3.wav flag-4.wav	54.6 80.53 79.65
		mean = 78.51
football-1.wav	football-2.wav football-3.wav football-4.wav	32.02 77.41 78.23
		mean = 62.55
icecream-1.wav	icecream-2.wav icecream-3.wav icecream-4.wav	9.72 16.28 79.54
		mean = 35.18
meat-1.wav	meat-2.wav meat-3.wav meat-4.wav	51.46 78.1 79.57
		mean = 69.71
pancake-1.wav	pancake-2.wav pancake-3.wav pancake-4.wav	48.46 81.28 81.74
		mean = 70.49
school-1.wav	school-2.wav school-3.wav school-4.wav	29.84 37.2 78.32

		mean = 48.45
shirt-1.wav	shirt-2.wav shirt-3.wav shirt-4.wav	70.95 78.51 79.59
		mean = 76.35
under-1.wav	under-2.wav under-3.wav under-4.wav	51.02 77.16 24.76
		mean = 50.98

The Table 6.4, shows the result of fuzzy scores of selected words by comparing native audio files by using the first one as template and the rest as samples to find the fuzzy scores. The average score of all words is 64.501. The result behaves poorly in some cases, since period of phonetic parts (PoP) could not perform in the process of iterative thresholding to give a suitable threshold correctly. The poor behavior also happens because in some cases the audio signals are recorded with different conditions which results in the appearance of noise in the background. On the other hand, the samples with acceptable levels of noise can perform with the score of at least 70.

Chapter 7
Conclusion

This chapter will discuss the facts discovered during the development process, and how the procedure could be improved to yield a better result. Which can be capitalized on to expanding the base knowledge field.

7.1 Pre-processing and Classification

As it turned out, the pre-processing approach and initialization of the audio signal has drastically improve the prediction with deep learning. It the research of Deep Learning Approach to Accent Classification [12] paper shows that the capability of deep neural network architectures is enough to classify both native and non-native English speakers with the approach reaching 0.9 accuracy when predicting between couple of accents but in later work when more classes are introduced the accuracy drop down below 0.8 with the same proposed method.

Where in this proposed method using Hoosier native language speakers data with pre-processing procedure differentiate 8 classes testing with not only MFCC representation but also spectrogram and formant frequency. As the results has shown that with this proposed method working with the representation of spectrogram can yield higher precision in predicting an accent and can be further improve by mixing both representation together. Still with the result shown in the experiments, it is still not enough to conclude that when working with audio type signal in deep neural network approach this proposed pre-processing methods would be the most efficient but it is clear that even with data that come from a strict environment it could still contain information that could be considered noises or an irrelevant information. A suitable pre-processing method performing with a suitable set of data is required in order to archive a higher precision in a prediction problem in this field.

When working with deep learning approach, there are multiple aspects that could be captured and learned from an audio signal of human language. For this experiment we stick with the based line in our problem description of how communication breakdown with various factors focuses on classification of accent from different native languages.

Convolutional Neural Network

The convolutional neural network performed as expected yield a good result with precision going as high as 0.8538. This is likely because CNN is known to perform well on image classification tasks and in this context. The extraction of MFCCs from the raw audio to form an image data that is fed to CNN and as such it proved to effectively reduced the accent classification task from an audio one to an image one, thus resulting better performance.

Precision on classifying accent still varies between the representation of data and its parameters. Spectrogram proved to be more effective than MFCC when dealing with neural network as to be expected of how a spectrogram contain more detail information that could better represent a signal while with formant frequency the representation seems to be unsuitable with CNN. Even when testing with multiple parameters the precision of a model trained with MFCCs sits at 0.74 where with spectrogram the lowest archived was still at 0.79 (Only considering after fine tuning parameter to a near saturation point) which is consider plenty better.

This result obtained might be heavily related to the data set used in this experiments and with other data set spectrogram could out perform the MFCC by a larger margin, with the possibility of the vice-versa case unlikely to occur if stick to CNN. In conclusion spectrogram should be a better solution when working with deep learning approach due to the information it represent.

Mixed Data

As seen from the experiment results, training mixed type of representation surprisingly yield a result that out perform spectrogram image data of size 128x48 which have the highest precision in predicting an accent by quite a margin. By training the neural network model with a representation of MFCC and spectrogram in one single item the predictive model reach the accuracy of 0.8759 when predicting an accent of the 8 different classes.

The result archived suggest that when working with audio representation in deep learning context a combination of representation could notably be a solution for obtaining high value precision with classification problem and possibly other type of problem description. From this it could be theorized that deep neural network can work with multiple combination of audio signal representation and capture some characteristic of the signal that come from those combined representation that even human might not be able to perceive in low level. Still it need to be mentioned that not all combination will have a positive affect when training a neural network and some representation have a limitation that will not be able to used with other representation. This type of approach needs to be carefully monitor as it could very likely result in an overfitting model or the data pattern or characteristic the neural network learned and grasped might take the wrong turn and interpret the problem description in the wrong way.

7.2 Assessment System

The assessment turns to be one of the indicators for a speaker to be able to understand how close their pronunciation compared to the native speaker is. Fuzzy logic System is integrated into this system due to the fact that it is widely used in speech recognition system. Furthermore, evaluating or assessing the pronunciation is not an easy task to do since it is ambiguous. Fuzzy comes to our mind to solve the ambiguity by transforming the features to fuzzy sets. Dynamic time warping is another algorithm to extract individual feature from a signal with various speed and returns a result which is the optimal path of similarity. At first, we tried to integrate sum of absolute differences (SAD) to be one of the crisp inputs for fuzzy logic system. Unfortunately, SAD is not consistent enough to find its characteristic which affects the fuzzy logic system to accept only two crisp inputs which are DTW and PoP.

With the results shown in the experiments, the result is acceptable if the voice recorded does not contain too much noise in the level that iterative segmentation based on threshold can segment a signal correctly.

7.3 Proposed System Development

A result system is developed in order to showcase the progress so far using the mobile application to input audio signal and sent it to the server to be process. The back-end of the server can monitor the output of each procedure both on the pre-processing and features extracting process.

7.3.1 Server Development

The server side of the proposed system is responsible for handling input audio sent from mobile application then pre-process that signal before classifying it using the model and assess it. Aside from actual processing the server side also serve as a monitor to visualize the process being done to obtain the result before sending that result back to the mobile application.

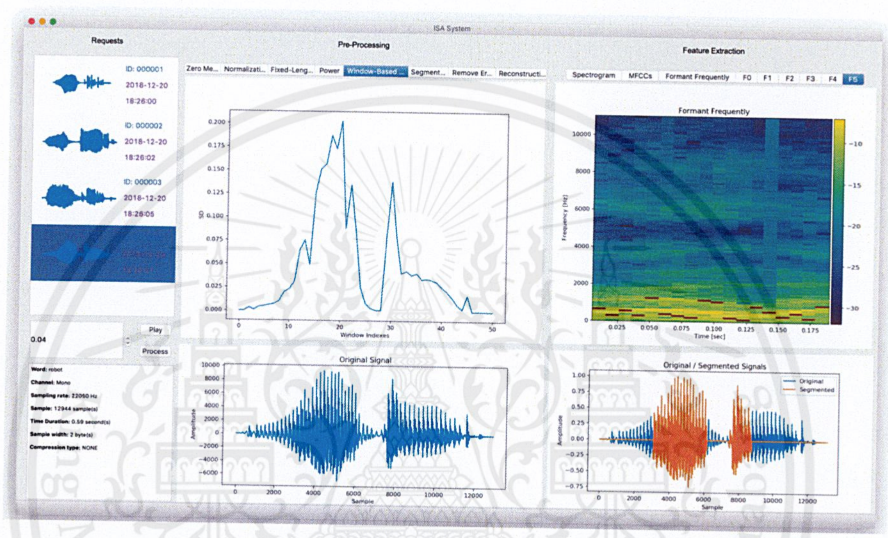


Figure 7.1: Server Monitoring

7.3.2 Mobile Application Development

In order to implement an interface on a mobile platform to showcase the process of the system from recording audio to displaying result sent from server we select React Native framework which is a hybrid mobile development framework that need lesser time for development and works both on Android and IOS. Below are some of the user interface of the mobile application with Figure 7.2 as a main page that interact with user by pressing the microphone button to record audio then it will be automatically sent to server. Figure 7.3 showing the available word that could be assess using the proposed method.

Figure 7.2: Main Page

Figure 7.3: Words Selection

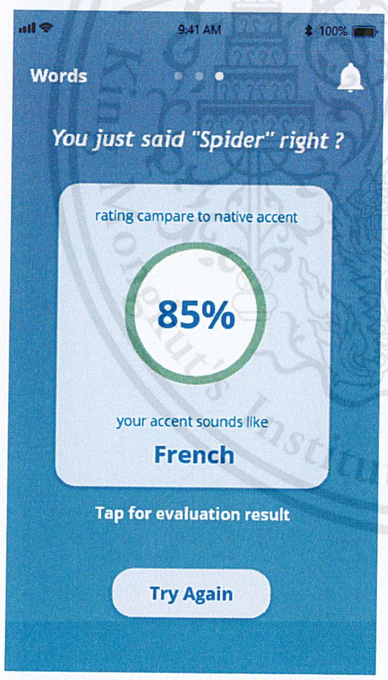


Figure 7.4: Processed Result

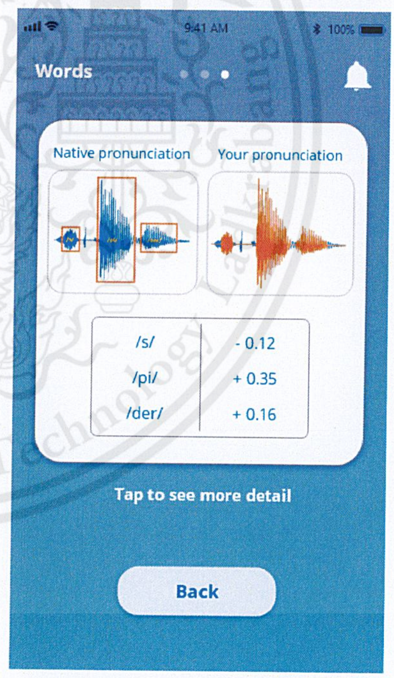


Figure 7.5: Detail Result

After the record has been sent it will be process in the server to evaluate which accent the input audio is closest to and also the assessment result compute by comparing to a template native pronunciation of that particular word. As seen in Figure , as an result page that got response back from the server and in Figure showing some feedback detail of the assessment part.

7.4 User Testing

After the system is finished we decided to test our application with real users and tried to see the result of both classification and assessment. There were 10 Thai subjects, 5 men and 5 women. Each subjects speaks 10 words with 5 trial on each word.

7.4.1 Classification Result

In Figure 7.6, shows the result of the 10 subject that tried speaking 10 words with 5 trial each word. The result tend heavily on Thai classes because out of the 10 subject, they were all Thais. There were a total of 500 times of word spoke and around 210 times were actually considered to be Thai.

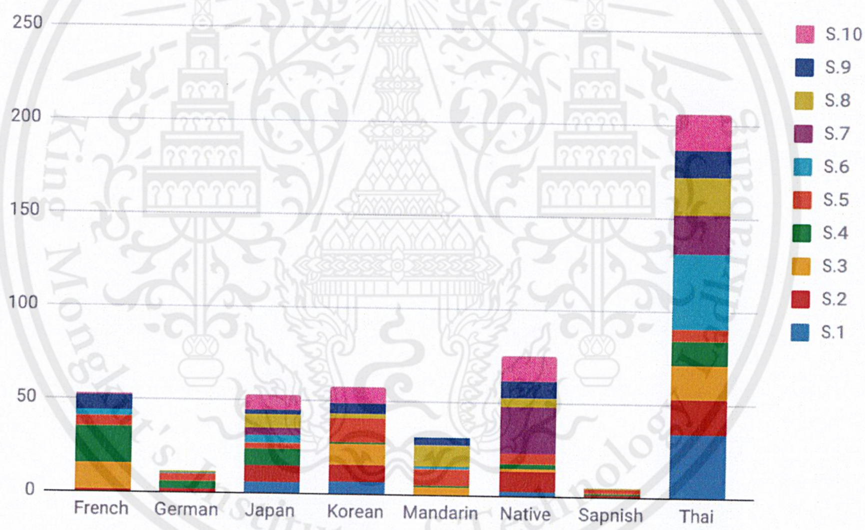


Figure 7.6: Classification result on real subjects

7.4.2 Assessment Result

In Figure 7.7, shows the result of the 10 subject that tried speaking 10 words with 5 trial each word. The range of score started from 55 to 80. Compare to native score which range from 80 to 100. Also the standard deviation can be seen on the error bar. However, the subjects were all studied in international program, which can be a cause of achieving high score.

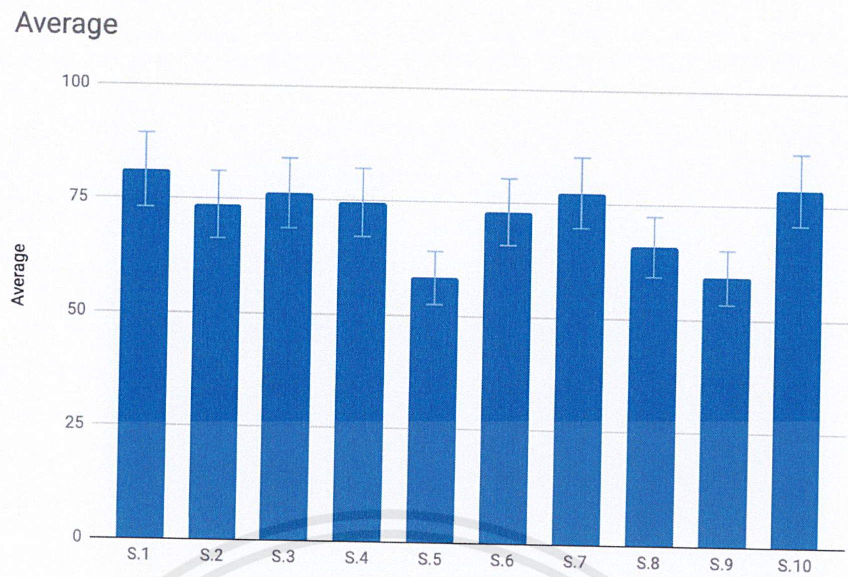


Figure 7.7: Assessment result on real subjects

Bibliography

[1] J. Jensen, "The Biggest Problem When Non-Native Speakers Use English", 2018 [Online]. Available: http://ltprofessionals.com/biggest-problem-non-native-speakers-use-english-phonemes [Accessed: 08-November-2018].

[2] C. Huang, T. Chen, E. Chang, "Accent Issues in Large Vocabulary Continuous Speech Recognition", International Journal of Speech Technology 7, 2004, pp. 141-153.

[3] R. Upadhyay, S. Lui, "Foreign English Accent Classification Using Deep Belief Networks", IEEE 12th International Conference on Semantic Computing, 2018, pp. 290-293.

[4] M. Russell , M. Najafian, "Modelling Accents for Automatic Speech Recognition", 23rd European Signal Processing Conference (EUSIPCO), 2015, pp. 1568.

[5] A. Ensslin, T. Goorimoorthee, S. Carleton, V. Bulitko, S. Hernandez, "Deep Learning for Speech Accent Detection in Videogames", 2017, Association for the Advancement of Artificial Intelligence, pp. 69-74.

[6] Sejdic E., Djurovic I., Jiang J, "Time-frequency feature representation using energy concentration: An overview of recent advances", Digital Signal Processing, 2009,pp.153-183.

[7] S. Kasahara, N. Minematsu, H. Shen, D. Saito and K. Hirose, "Structure-based prediction of English pronunciation distances and its analytical investigation", 4th IEEE International Conference on Information Science and Technology, 2014, pp. 332-336.

[8] J. Lee, C. H. Lee, D. Kim and B. Kang, "Smartphone-Assisted Pronunciation Learning Technique for Ambient Intelligence", IEEE Access vol. 5, 2017, pp. 312-325.

[9] M. Nicolao, Amy V. Beeston, T. Hain, "Automatic Assessment of English Learner Pronunciation Using Discriminative Classifiers", Speech and Hearing Research Group IEEE Access, 2015, pp.326-331.

[10] B. Unhalekajit, N. Chanamool, "Computer Software for Phonetic Analyzing of Thai Language using speech processing Techniques", Engineering Department of Electrical Engineering Burapha, 2008, pp. 1-86.

[11] U. Grenander, "The Nyquist frequency is that frequency whose period is two sampling intervals", Probability and Statistics: The Harald Cramer Volume, 1959.

[12] L. Sheng, M. Edmund, "Deep Learning Approach to Accent Classification", Machine Learning Stanford, 2017, pp. 1-5.

[13] MathWorks, "Feature Learning, Layers, and Classification", 2017. [Online]. Available: http://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html [Accessed: 09-November-2018].

[14] D. Karunakaran, "Deep learning series 1: Intro to deep learning", 2018. [Online]. Available: https://medium.com/intro-to-artificial-intelligence/deep-learning-series-1-intro-to-deep-learning-abb1780ee20 [Accessed: 08-November-2018]

[15] George Reese "Distributed Application Architecture" (PDF). Sun Microsystem. Archived from the original (PDF) on 6 April 2011. Retrieved 2009-06-16.

[16] A. Lee, Y. Zhang, and J. Glass, "Mispronunciation detection via Dynamic Time Warping on Deep Belief Network-based posteriorgrams," ICASSP, 2013, pp. 8227-8231.

[17] SciPy, "scipy.signal.spectrogram", 2018. [Online]. Available: https://docs.sci py.org /doc/scipy/reference/generated/scipy.signal.spectrogram.html [Accessed: 08-February-2019]

[18] J. Lyons, "python speech features's", 2018. [Online]. Available: https://python-speech-features.readthedocs.io/en/latest/ [Accessed: 11-March-2019]

[19] W. Campbell, D. Reynolds, J. Campbell, K. Brady, "Estimating and Evaluating Confidence for Forensic Speaker Recognition" in ICASSP 2005, Philadelphia, PA, 2005, pp. 717-720.

[20] Scikit-Fuzzy, "scitkit-fuzzy", 2018. [Online]. Available: https://pythonhosted.org/scikit-fuzzy/overview.html [Accessed: 11-March-2019]