



Sound signal identification

อาจารย์ที่ปรึกษาโครงงาน

รศ.ดร. ลลิตา จินตราวัน

อจ.ดร. ณัฐพล ดำรงค์พลาสีทธิ์

สมาชิกโครงงาน

นาย พิชากร หวังเจริญวงศ์ 6130364121

นาย รวิ เชิดชูสกุลชัย 6130462021

การวิจัยนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต

รายวิชา 2103302 Engineering measurement

จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2564

Abstract

Humans can recognize the speaker's voice easily, but for computers, it is hard to distinguish the human voice. The voice must be processed to make the machine understand before implementing it in the world application. The study aims to explore and test whether there is a way to create a system so that the computer can also recognize the speaker's sound. To achieve the research objectives, seven people's audios were collected from two sources and were processed to be 1-second audio. In addition, audio must be ensured that it contains 1 second of speech, not noise. Thus, we took out some of the audio by verifying the power of the signal. After data was collected, each sound audio was extracted to get its identity features. In this project, there were 4 types of feature extraction which were spectrogram, Mel-spectrogram, MFCC, and MFCC Delta & MFCC Delta-Delta. Then the whole dataset was split into three datasets to implement in the machine learning model. Four models were established for each feature extraction. The model extracted features or patterns to identify the audio's owner on its own. Finally, we compared the results of all four models. It was found that the MFCC Delta-Delta model gave the best performance among all models. However, for categorical problems like this project, we cannot consider only accuracy performance. Therefore, we made a confusion matrix to see whose voice that model predicted incorrectly. In conclusion, the results showed that the deep learning approach could be used to extract voice patterns and determine the speaker's voice using the unstructured audio representation derived from the audio signal. Moreover, within line of expectation, concatenation of MFCC, MFCC Delta, and MFCC Delta-Delta tends to be the best feature among all features.

Background & Introduction

By human nature, humans can easily recognize the speaker's voice suddenly after the person speaks if they are acquainted with that person. The human brain extracts a variety of information from natural speech, allowing the listener to understand the message as well as distinguish who is speaking. In this project, the author aims to explore and test whether there is a way to create a system so that the computer can also recognize the speaker's sound.

As the use of machine learning and deep learning in the field of audio analysis is fast expanding, automatic speech recognition, digital signal processing, audio classification, tagging, and production are only a few examples. Virtual assistants like Alexa, Siri, and Google Home are built largely on models that can do artificial cognition based on auditory data. The author decided to process with a machine learning approach in this experiment. To train any ML model, we need first to extract useful features from an audio signal. Audio feature extraction is a necessary step in audio signal processing, which is a subfield of signal processing. It deals with the processing or manipulation of audio signals. The feature used in the traditional machine learning approach considered almost all of the features from both time and frequency domains as inputs of the model. Some of the example features include Root Mean Square (RMS) Energy, Zero-Crossing Rate (ZCR), Spectral Centroid, Band Energy Ratio, and Spectral Bandwidth. However, these features have to be handpicked based on their effect on the model's performance. Thus, the deep learning approach has become the preferred approach lately. This approach considers unstructured audio representations such as spectrograms or MFCCs and is able to extract patterns on its own. In our experiment, we extracted a variety of unstructured audio to feed into neural network architectures in order to see what the best features for a speaker identification model are.

There are three sections to the project's scope. To begin, our deep learning model can recognize the speaker's voice from a group of persons who are only present in the dataset. The number of speakers in this experiment, which will be utilized as a training dataset, is therefore limited to 2-7 people. Finally, to eliminate noise and unneeded dataset confusion, each speaker must talk in a regular tone.

Process

In this project, we tried to implement machine learning to identify the voice owner. However, machine learning cannot work well with just a bare sound signal. Thus, the sound signal must be preprocessed before entering the machine learning paradigm. The author used four approaches to extract sound features which were 1. Spectrogram 2. Mel-Scale Spectrogram 3. MFCC (Mel frequency cepstral coefficients) 4. MFCC Delta & MFCC delta delta. Note that each feature extraction was developed from the previous methods. These sound feature extraction methods were recommended by domain experts in this field. As the sound was processed in the figure formation, the type of deep learning called convolutional neural network was used in this project to find patterns from unstructured data.

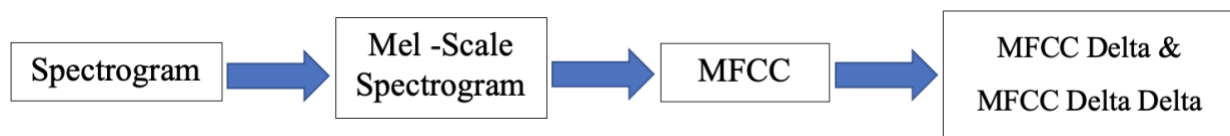


Fig. 1 order of feature extraction

1. Data acquisition

The first step was data acquisition. In this project, seven different peoples' voices were collected into datasets. Each dataset contains 1500 speeches in 1 second. 5 datasets were collected from Kaggle, and the rest 2 (our professors' voices) were collected manually from previously recorded lectures.

The first source of datasets is from Kaggle. The voices' owner in this source is the famous politicians, including Benjamin Netanyahu, Jens Stoltenberg, Julia Gillard, Margaret Thatcher, and Nelson Mandela, as shown in fig. 2.

Data

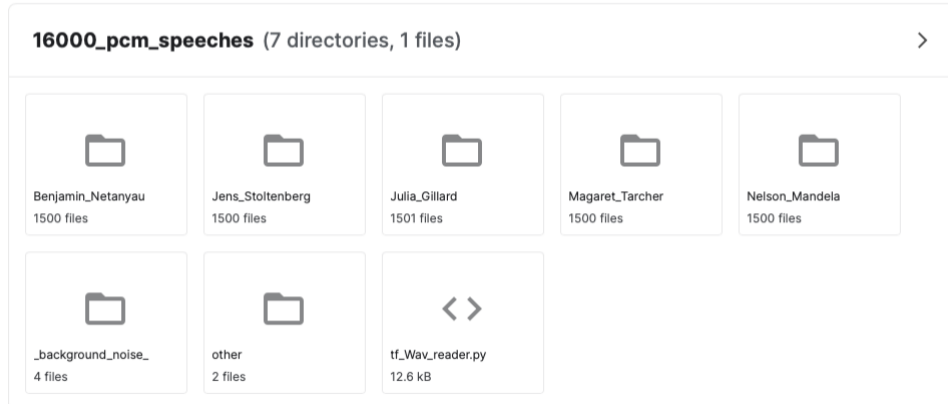


Fig. 2 Kaggle datasets

The second source of datasets is our two professors Aj. Natapol and Aj. Thitima, as shown in fig. 3 and 4. We gathered the voice recorded from YouTube and Zoom for 4 clips. Each professor's clip was 10 minutes long.

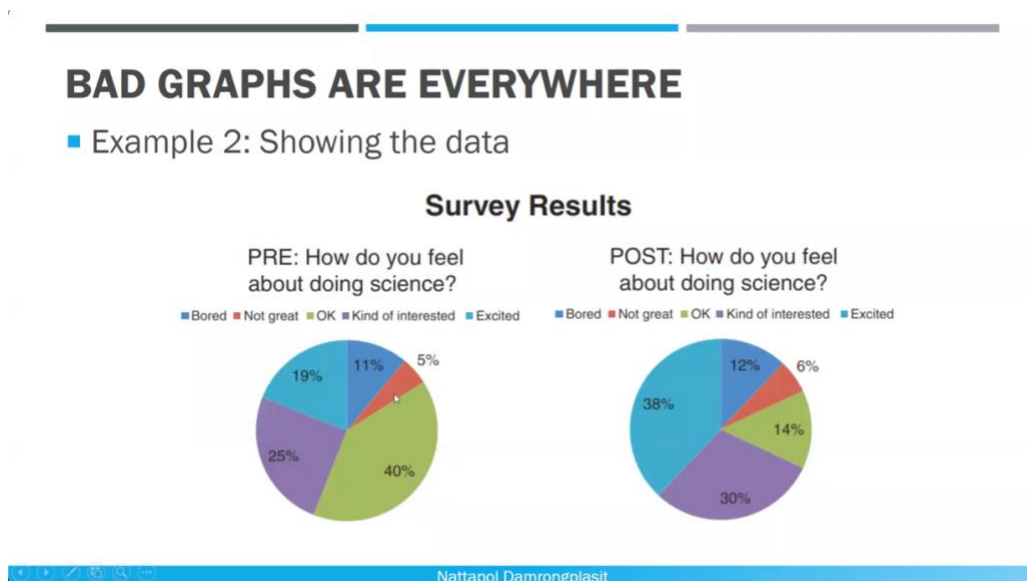


Fig. 3 Example of Aj. Natapol recorded lecture

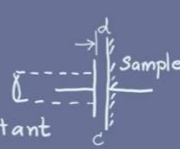

Capacitive type

Output voltage is related to the capacitance depending on a separation of the conductive plates.

$$C = f(K, A, d)$$

$K \sim$ dielectric constant

- Very high precision (nm or better) and small range
 - Medium response time (bandwidth)
 - Non-contact and alignment is critical
 - Need charge amplifier

See further: [https://www.researchgate.net/publication/318000000/figure/fig/1/figure-fig1/1518811111111/1518811111111.png](#)

Fig. 4 Example of Aj. Thitima recorded lecture

Because the duration of each voice in Kaggle is 1 second long, we need to process our professors' voices to be the same length as the one in Kaggle, as shown in figure TT. So, we split 10 minutes clip into 1-second clips shown in fig 5.

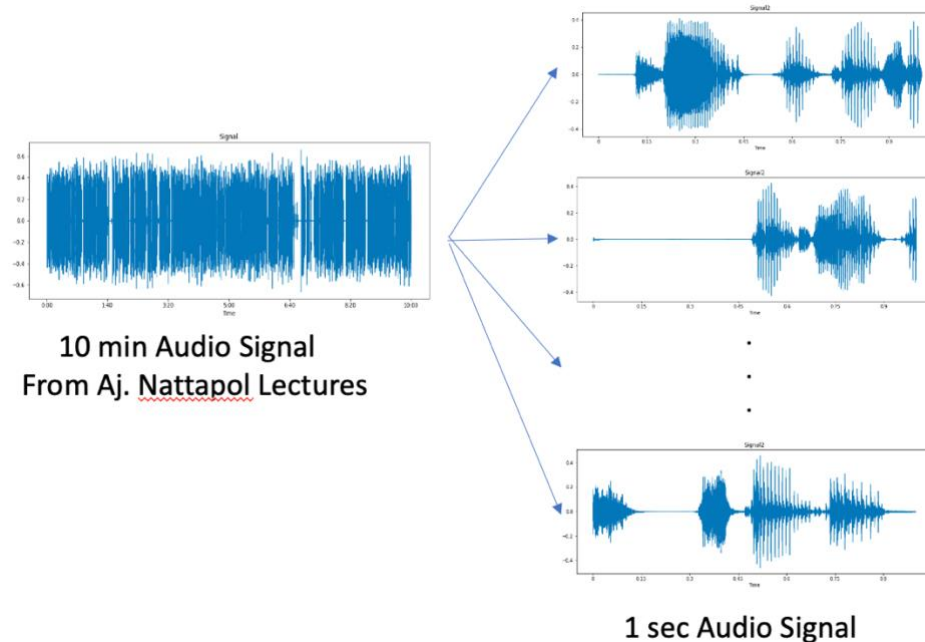


Fig. 5 Splitting 10 minutes voice clips into 1 second

The problems then came after because some of the 1-second clips contained empty sound or noise, which is not what we wanted. Consequently, we decided to take those clips out by computing the signal energy shown in equation 1. Every clip that had the energy of the signal less than 50% of the median energy of the whole clip will be taken off.

$$P_x = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{n=N-1} |x(n)|^2 \quad - \quad (1)$$

In this step, we acquired datasets to process in the next step. Next will be features extraction of the sound as mentioned above that machine learning cannot work with the normal sound signal.

2. Feature Extraction

This is the most important step in this project. This process will extract features from sound signals to implement in the machine learning module. Experts in this field suggested the most used feature extraction in the sound preprocessed field is MFCC combined with the MFCC delta and MFCC Delta Delta. However, to get the MFCC feature, we need to start from the spectrogram and then the Mel-scale spectrogram. So, we decided to use four feature extraction to experiment on which is the best to identify the human voice.

2.1 Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. The process of the spectrogram started by slicing the audio wave into small windows of time. The next step was to apply the Fourier transform to reduce the memory of the data. We choose to use a fast Fourier transform to resample the signal. The last step was to flip the signal and combine all the windows. The result was the spectrum over time, as shown in fig 6.

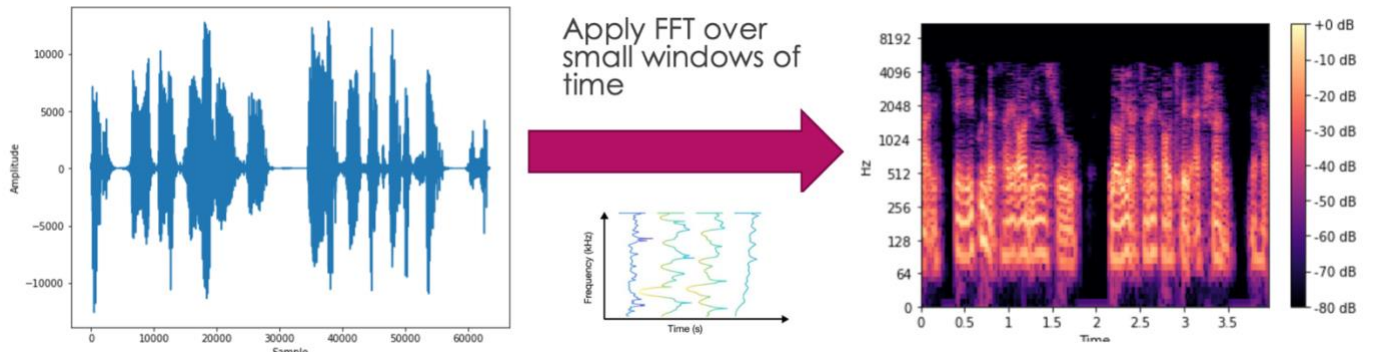


Fig. 6 Converting audio signal to spectrogram

2.2 Mel-Scale Spectrogram

This is also the spectrogram but on a different scale which is called Mel-scale. The reason why we need to convert from the hertz scale into Mel-scale is that humans perceive frequency logarithmically shown in fig 7. For example, if the frequency is low, starting from 65 Hz to 262 Hz, humans can hear the difference obviously, while if the frequency is higher from 1568 to 1760 Hz, humans will not know the difference much. Even though the difference between both examples is the same but humans do not perceive the same.

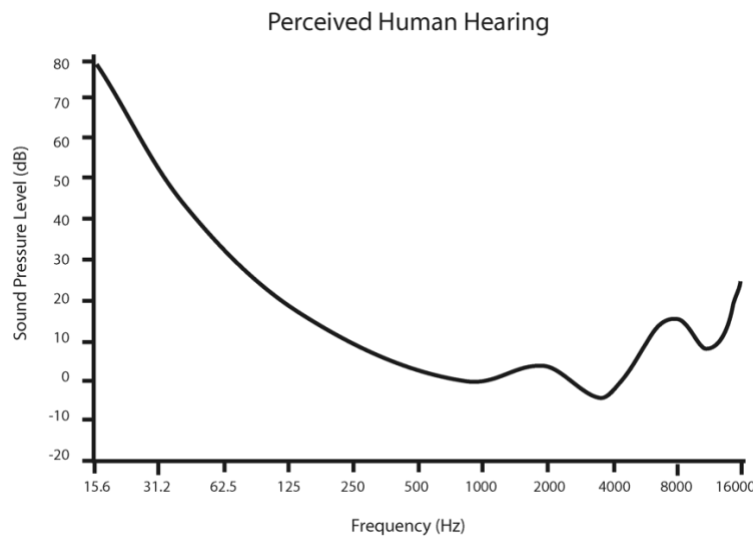


Fig. 7 human hearing

The idea of the Mel-scale is to make the sound in a range that is close to human perception. For example, unlike in hertz frequency, 65 – 262 Mel-scale and 1568 – 1760 Mel scale, humans can detect the difference the same. To convert into Mel-scale, we need to apply non-linear transformation as shown in equation (2). Fig 8 shows the converting value between hertz and mel-scale.

$$f = 700(10^{\frac{m}{2595}} - 1) \quad - \quad (2)$$

$$M = 2595 \log(1 + \frac{f}{700}) \quad - \quad (3)$$

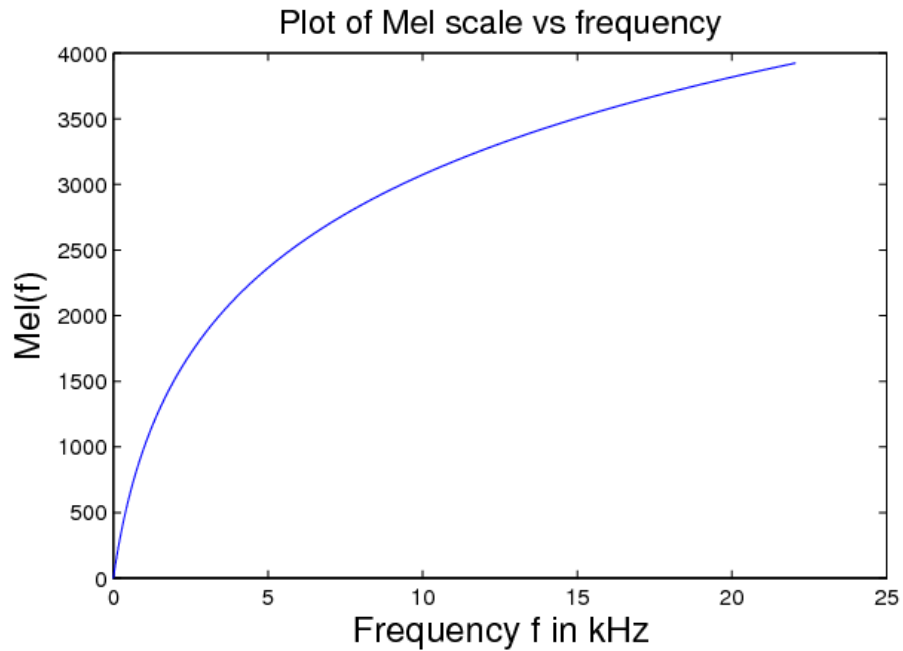


Fig. 8 plot between Mel-scale and frequency scale

There were three steps to convert the scale. The first step was to choose Mel band or the number of bins. Instead of converting every value into the Mel-scale, we partition the Hz scale into bins and transform each bin into a corresponding bin in the Mel Scale, using overlapping triangular filters. Note that the number of bins depends on the task. The next step was to construct a Mel filter bank or overlapping triangular filters, as an example shown in fig 9. [5]

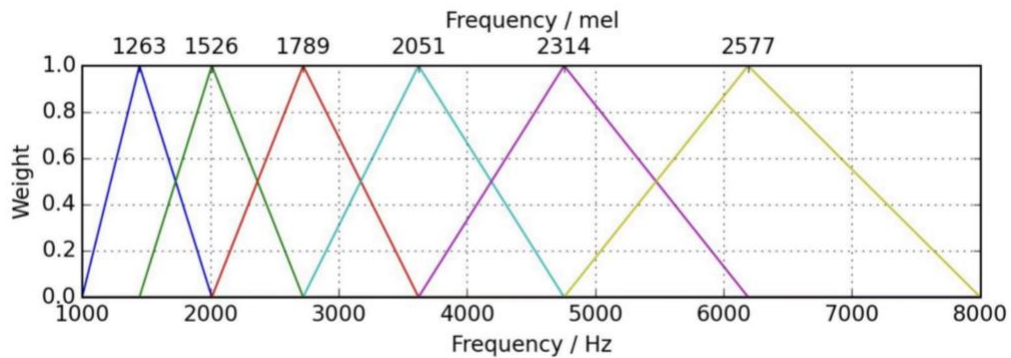


Fig. 9 example of mel filter bank of 6 mel band

The last step was to apply the Mel filter bank to the normal spectrogram, as shown in fig 10. We chose to use 10 Mel bands as we found they fit our model very well.

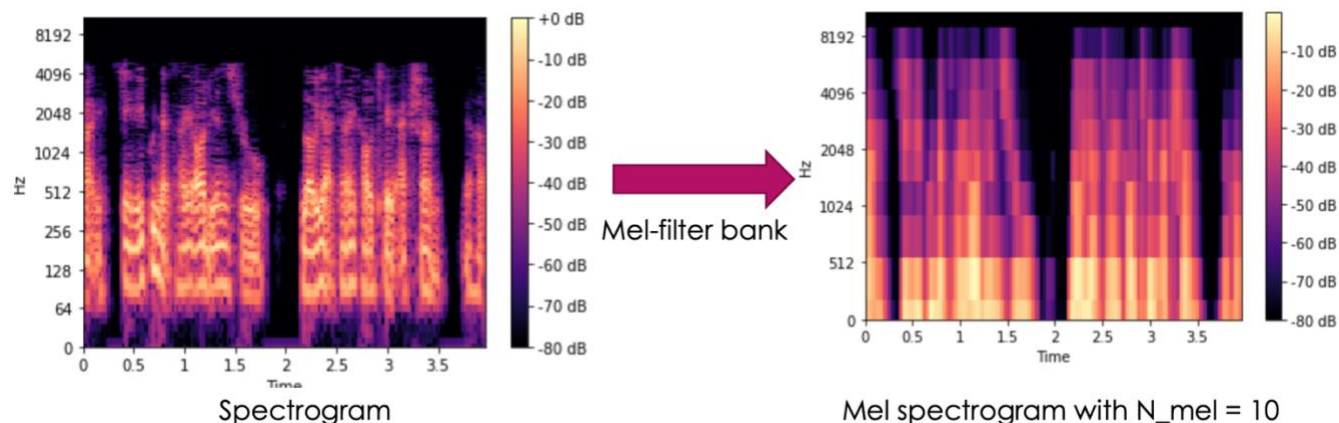


Fig. 10 Mel-spectrogram after applying Mel filter bank

2.3 MFCC

MFCC or Mel frequency cepstral coefficient is the most popular speech preprocess technique. It was found that MFCC could represent sound characteristics such as timbre very well. MFCC is built on top of the Mel-scale spectrogram by applying discrete cosine transform [4], as shown in fig 11. The result was the coefficient of MFCC over time.

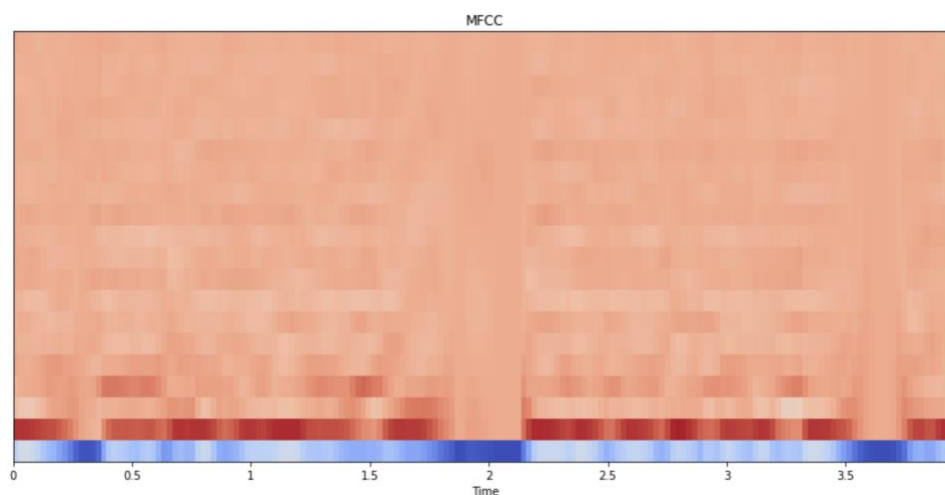


Fig. 11 MFCC from audio

2.4 MFCC delta and MFCC delta delta

To represent more characteristics of the sound signal. We added two more features which were MFCC delta and MFCC delta-delta. By adding this, it represented the derivative of MFCC change over time in the audio file. The input of this feature in the machine learning model would be three figures concatenated together as one input, as shown in fig 12.

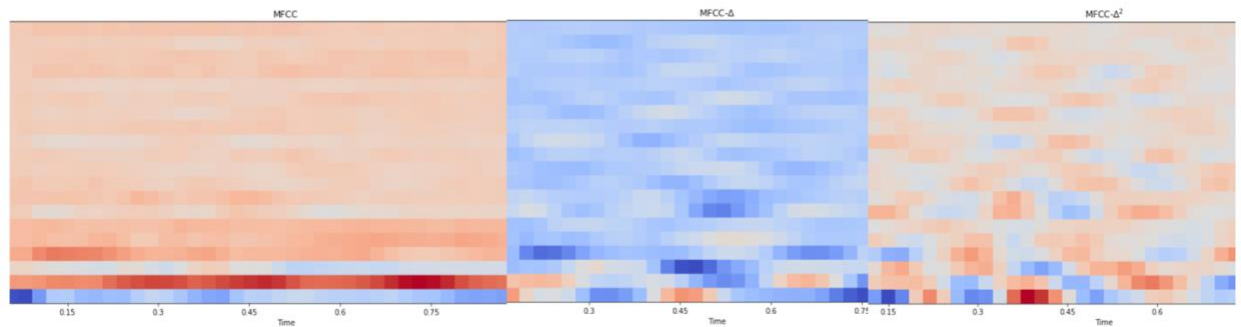


Fig. 12 Concatenate of MFCC (left), MFCC delta and MFCC delta delta

3. Machine learning

Before building deep learning, we must split our data into 3 sets shown in fig 13. The first dataset is called the Training dataset. The training dataset (70 % of the whole data) is used to compute an algorithm in deep learning. This dataset holds the most data because we want our deep learning to see many data as we can. The next dataset is called validation datasets (15 % of the whole data). The main purpose of this dataset is to find the performance of the trained model from the seen data to avoid problems with machine learning called overfitting and underfitting. The last dataset is called the test dataset (15 % of the whole data). This last dataset will never be used to train the model. This dataset is used only to report the result.

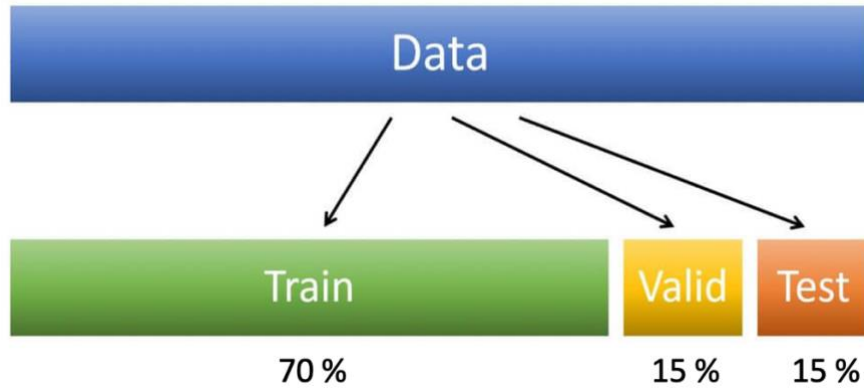


Fig. 13 splitting of the whole dataset

Sound signal after extracted features is more likely to be unstructured data which means it is difficult for a human to detect the pattern. To deal with the unstructured data type, we used deep learning to find such a pattern or trend to classify human audio. As the input data are images, we used a convolution neural network to deal with the images shown in fig 14.

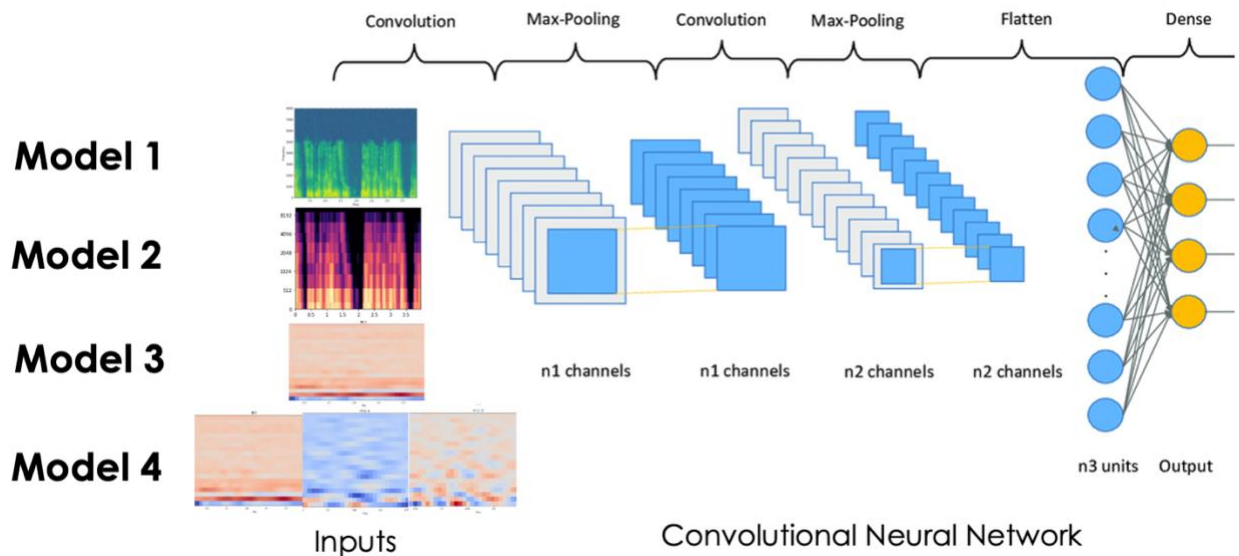


Fig. 14 Convolution neural network architecture

We built four models for each feature extraction. The last layer is the output layer, with seven nodes for each voice. Each node in the last layer represents the probability of guessing for each voice owner. Finally, the model will give a result by the node with the most probability.

Result & Discussion

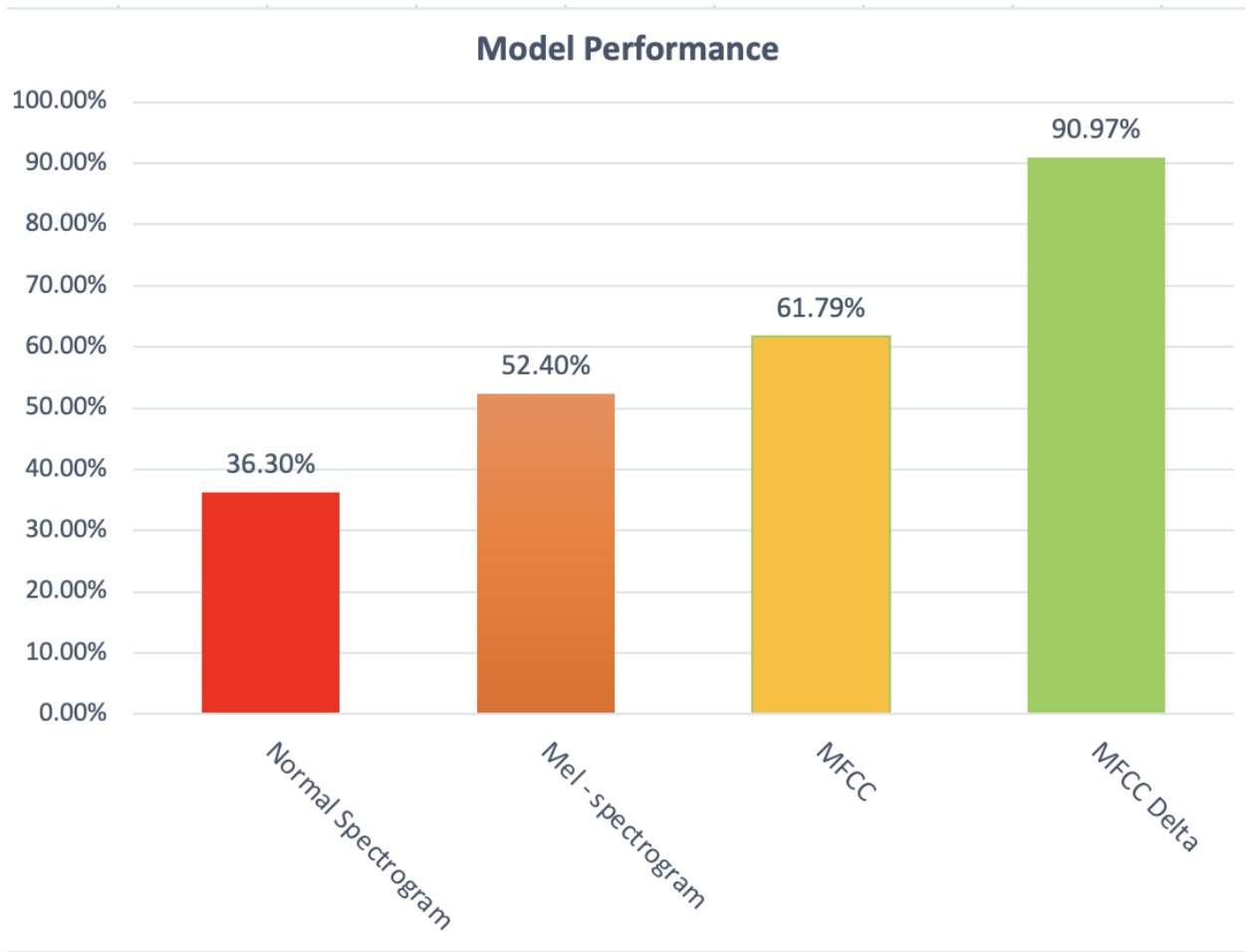


Fig. 15 Performance of each model

Fig 15 illustrates the relationship between model performance and the extracting feature used to feed into the neural network architecture. In this experiment, we fix the data preprocessing process and hyperparameters of the architecture to be the same for all cases. The accuracy of each model was tested by our test dataset, which we split before we started training the model. It was found that the model with normal spectrogram features has the lowest accuracy of only 36.3%, whereas the highest accuracy model of 90.97% was found in the one with MFCC and MFCC-Delta features extracted. Furthermore, as the transformation from regular spectrogram to MFCC-Delta progressed, the features derived from these

steps, such as Mel-Spectrogram and MFCC, improved model performance continually.

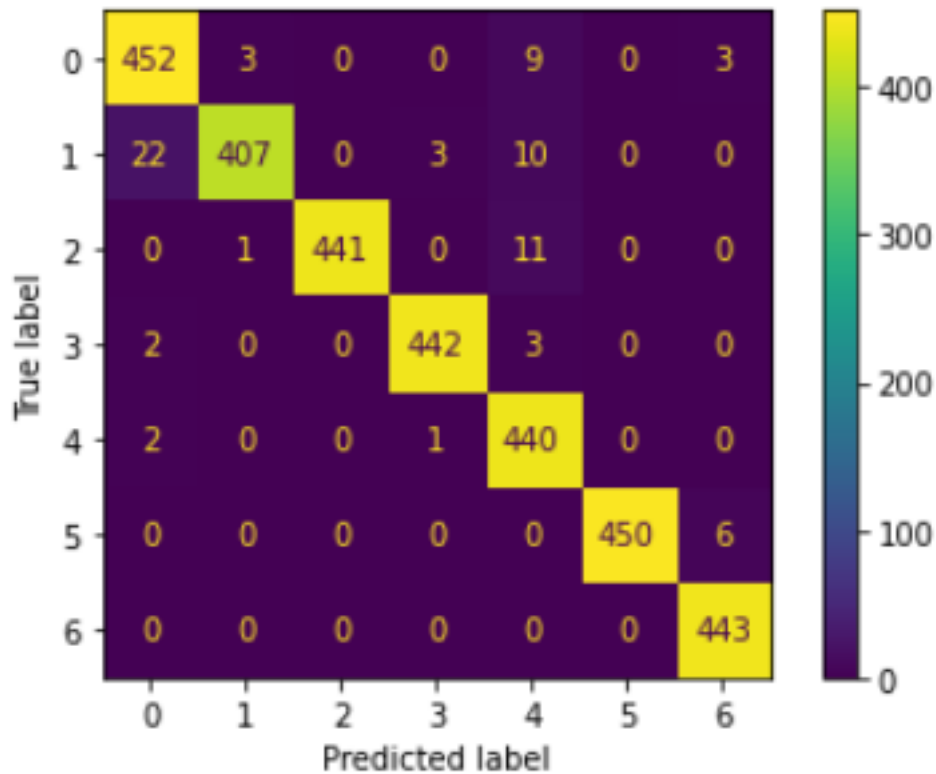


Fig 16 Confusion matrix of the model's output

Fig 16 shows the confusion matrix of the MFCC Delta model. According to the table, the accuracy of the model was about 97% because the author had already tuned some hyperparameters to get a better result. It was found that label number 1, which was Julia Gillard in this case, had the most false prediction. Additionally, label 4 was the label with the most false-positive result. The author believes that the similar speech characteristics of some people, as well as the noise in the dataset, are responsible for the model's incorrect prediction. To improve accuracy in the future, we'll need to expand the training dataset and discover a technique to create the noise-tolerant model, such as by performing data augmentation before training. Furthermore, as the number of people we want to forecast grows, we'll need a lot larger dataset because the likelihood of several persons having very similar features in the extracted feature grows.

Conclusion

This study aims to examine the method to perform a speaker identification system and determine the best feature to be extracted from the audio signal. The results showed that the deep learning approach could be used to extract voice patterns and determine the speaker's voice using the unstructured audio representation derived from the audio signal. By varying the unstructured audio data to feed into the neural network architecture, it was found that concatenation of MFCC, MFCC-Delta, and MFCC-Delta-Delta was the best feature to be extracted and used in training. It could be concluded from the present finding that because of its accuracy, the speaker identification model trained from the concatenation of MFCC, MFCC-Delta, and MFCC-Delta-Delta has a strong potential to be used in various industrial applications in the future.

Reference

- [1] <https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd>
- [2] <https://devopedia.org/audio-feature-extraction>
- [3] <https://desh2608.github.io/2019-07-26-delta-feats/>
- [4] https://en.wikipedia.org/wiki/Discrete_cosine_transform
- [5] https://www.researchgate.net/publication/259195300_A_Mel_filter_and_kepstrum_based_algorithm_for_noise_suppression_in_cochlear_implants/figures?lo=1&utm_source=google&utm_medium=organic