# Final Project:

# Automatic Speech Recognition

Group 26

Members: 傅莉妮 林怡秀 許維也 陳昱喬

## Motivation & Introduction

We decided to work on speech recognition because our team members were discussing potential topics while riding the elevator to the upper floors. We noticed an interesting feature in Engineering Building 3: the elevator buttons for going up and down could be selected in a unique way. By swinging our hand downward, we could indicate pressing the button for going down, and waving our hand upward meant selecting the button for going up. As we waited for the elevator, we started pondering why the floor selection couldn't be done in a similar intuitive manner. This question piqued our curiosity, and we contemplated whether speech recognition could be utilized to choose the desired floor. We searched for relevant information online and discovered its feasibility, leading us to decide to implement this idea.

We believe this is a remarkable feature that can benefit not only individuals with limited mobility, particularly prior to the outbreak, but also contribute to reducing the risk of virus transmission by minimizing people's contact with public facilities.

### Related paper/works

This paper has some similarities with the speech recognition system we built. As far as feature extraction is concerned, in speech processing, we use MFCC and Fbank to extract feature representations of speech signals. In this paper, the authors use CNN to extract the spatial features of video frames, which are then combined with LSTM. Likewise, LSTMs are widely used in both fields. In speech recognition, LSTM is able to capture the context information in the speech signal, and in this paper, LSTM is used to deal with video classification and action prediction. Furthermore, the evaluation of classification accuracy is important in both domains.

However, there are some differences between the two fields of study. First, this paper focuses on video classification and action prediction, not speech recognition. Secondly, in terms of feature selection, speech recognition usually uses spectral features such as MFCC and Fbank, while the paper uses the spatial features of video frames extracted by CNN. Also, time scales are handled differently, with

different numbers of video frames studied in the paper, whereas speech recognition typically deals with continuous audio signals.

Besides, we listed some useful websites we've looked at while implementing in Reference.

(link: http://d-scholarship.pitt.edu/36069/1/Ramadan_Mona_PhD_dissertation.pdf)

## Brief Introduction to Sound

Sound is generated by the vibration of a medium, producing sound waves. These sound waves cause the molecules in the air to vibrate in a rhythmic manner, resulting in changes in air density and the formation of alternating regions of compression and rarefaction. Through a microphone and audio interface, we can convert these waveforms into digital signals, a process known as "sampling."

## Dataset

We use datasets from a challenge held by one of the biggest dataset websites Kaggle, TensorFlow Speech Recognition Challenge. There are many words in this dataset, such as numbers, directions, and so on. For this project, we use the data of numbers and "up", "down", which are those words we may use in an elevator.

(link: https://www.kaggle.com/competitions/tensorflow-speech-recognition-challenge/data)

---

# Baseline

We use MFCC and Fbank to preprocess the audio file.

## MFCC

MFCC (Mel Frequency Cepstral Coefficients) extracts features from speech signals, converts speech signals into spectral features, and extracts feature vectors representing speech. It works by first converting the speech signal into a spectrogram, and then converting the spectrogram into an MFCC feature vector. This process mainly includes the following steps:

1. Pre-emphasis: In order to remove the low-frequency part and high-frequency noise in the voice signal, the voice signal can be pre-emphasized.
2. Framing: divide the pre-emphasized voice signal into multiple time windows, usually the length of each time window is 20-30 milliseconds, and a time window is taken every 10 milliseconds.

3. Windowing: Perform Hamming Window (Hamming Window) processing on the speech signal in each time window to reduce the problem of spectrum leakage.

4. Fast Fourier Transform (FFT): Fast Fourier Transform is performed on the windowed speech signal to obtain the frequency spectrum of each time window.

5. Establish a Mel frequency filter bank: Divide a continuous spectrum interval into several equal-width Mel frequency intervals.

6. Take the logarithm: Take the logarithm of the energy in each Mel frequency interval to get the logarithm of the energy.

7. Discrete cosine transform (DCT): Perform discrete cosine transform on the logarithmic energy value to obtain the MFCC eigenvector of each time window.

## Fbank

Fbank extracts features from speech signals to represent the characteristics of different sounds in a compact form.

1. Pre-emphasis: First, pre-emphasize the original speech signal. Pre-emphasis is a high-pass filter that boosts high-frequency components and reduces energy in low-frequency components. This helps to balance the energy distribution in the spectrum and improve the effect of subsequent feature extraction.

2. Framing: Cut the pre-emphasized speech signal into short-term frames. Typically a window with a frame length of about 10-30 milliseconds is used, with an overlap between adjacent frames, for example 50% overlap. This preserves the temporal continuity of the speech signal to capture dynamic changes in speech features.

3. Fourier transform: Fourier transform is performed on each frame to convert the time domain signal into a frequency domain signal. This gets spectral information for each frame.

4. Filter composition: A set of filters is used to decompose the spectrum into sub-bands. These filters usually use a mel filter bank composed of triangular filters, and the frequency cutting of these filters is designed according to the characteristics of human ear perception. The filter response composed of the mel filter matches the perceptual characteristics of the human ear, which can better capture the important frequency components of the speech signal.

5. Energy calculation: Calculate energy or power for each sub-band. In general, the energy value in each sub-band is used here.

6. Logarithmic conversion: Usually logarithmic conversion is performed on energy or power, such as taking the logarithm, to enhance small energy differences and make it more in line with the human ear's perception of sound.

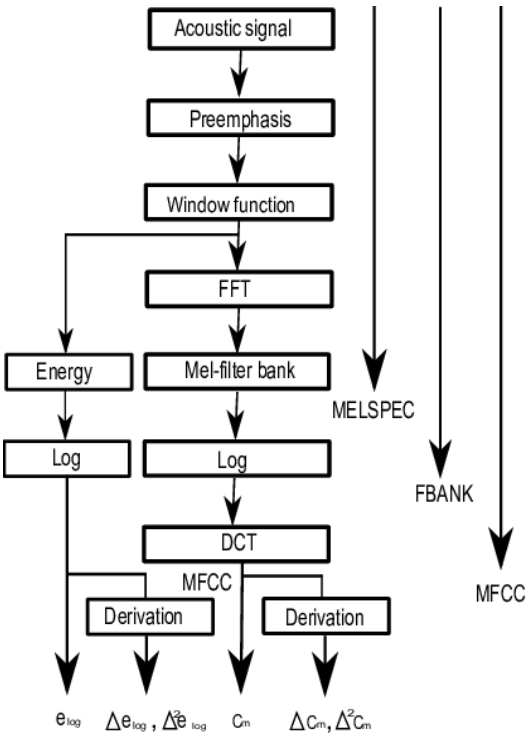
7. Feature vector extraction: The logarithmic energy value of each frame is used as a part of the feature vector to form the final Fbank feature representation. In practical applications, multiple filter combinations may be used to extract Fbank features in different frequency ranges, so as to obtain more comprehensive spectral information.

8. Applied to speech recognition: the extracted Fbank feature vector can be used as the input of the speech recognition system. These feature vectors may be used together with other features such as MFCCs for training speech models and for speech recognition tasks.

# Differences between MFCC and Fbank

1. Calculation method: MFCC features are obtained by performing Mel filter composition, logarithm, and discrete cosine transformation on the spectrum, while Fbank features are obtained only by Mel filter composition and logarithmic transformation.

2. Feature dimension: MFCC features usually contain more dimensions, in general, including energy terms, Mel-Frequency Cepstral Coefficients (Mel-Frequency Cepstral Coefficients) and time differential. Whereas the Fbank features only contain Mel-frequency energy terms.



# Use Them Together to Build a Speech Model

MFCC and Fbank features can be used together to build a speech model to improve the performance and *robustness of the model. A common approach is to concatenate MFCC and Fbank features to form a more comprehensive feature representation.

1. Calculate MFCC features for each speech frame, including energy terms and Mel frequency cepstral coefficients.
2. Calculate Fbank features for each speech frame, including the Mel frequency energy term.
3. Connect MFCC features and Fbank features to form longer feature vectors.
4. Use the connected feature vector as the input of the speech model, such as using a deep learning model (such as RNN, LSTM, GRU) for training and speech recognition.

*: Robustness refers to the resistance of a system or model to external changes, disturbances or noise. In speech recognition, robustness refers to the stability and reliability of the system for different environmental conditions, speech quality, noise interference and other factors.

## Benefits

There are many benefits to do this:

1. Fusion of multiple features: Provide more comprehensive spectrum information, including energy, frequency features and other dynamic features. This can enhance the model's ability to represent speech signals and help capture the features and changes of speech more accurately.
2. Increase robustness: MFCC and Fbank features take into account the characteristics of human ear perception in spectrum representation and filter design, and can provide better frequency resolution and sensitivity to important parts of speech signals. The combination of these two features can improve the speech recognition ability of the system under different environmental conditions.
3. Noise suppression: Since MFCC and Fbank features are based on the representation of spectral information, they can suppress the influence of noise and non-speech components, and improve the sensitivity of the speech recognition system to speech.

---

To implement automatic speech recognition, we use a model called Recurrent Neural Network (RNN) Long Short Term Memory (LSTM). In order to understand the core idea of LSTM, we will briefly introduce simple RNN first as background knowledge.

# Recurrent Neural Network

Recurrent Neural Network (RNN) is one of the artificial neural networks, which considers previous information of sequential data through feedback connections. For example, in this sentence: “NCTU is in a city called ___.” By the information before the blank, we can know that the blank should be filled in with “Hsinchu.” This is the basic concept of considering previous information. Time series is important in this model.

The algorithm can be simplified as the formula, modified by a simple sequential function: f(x) = Wx + b. After that, we add an activation function g(x) to make it the second formula.

$$h_t = W * h_{t-1} + U * x_t + b$$
$$y_t = g(V * h_t)$$

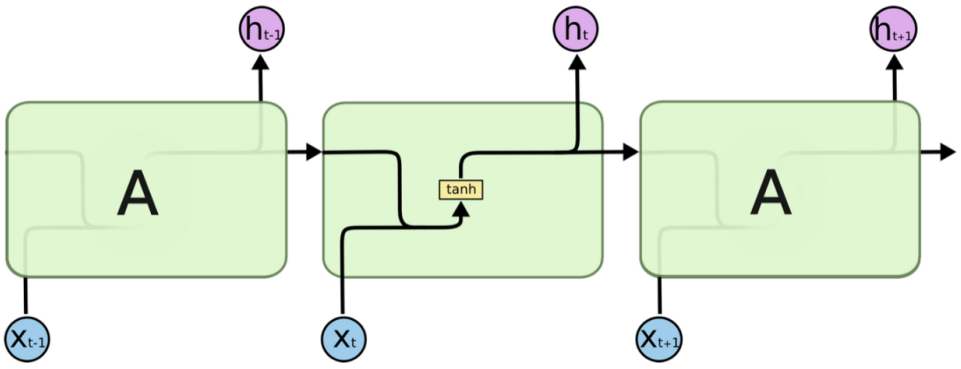
h : output

y : prediction result

W, U, V : weight matrix

b : bias (scalar)

g : activation function, mostly will use tanh




source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

# Long Short Term Memory

## Disadvantages of RNN

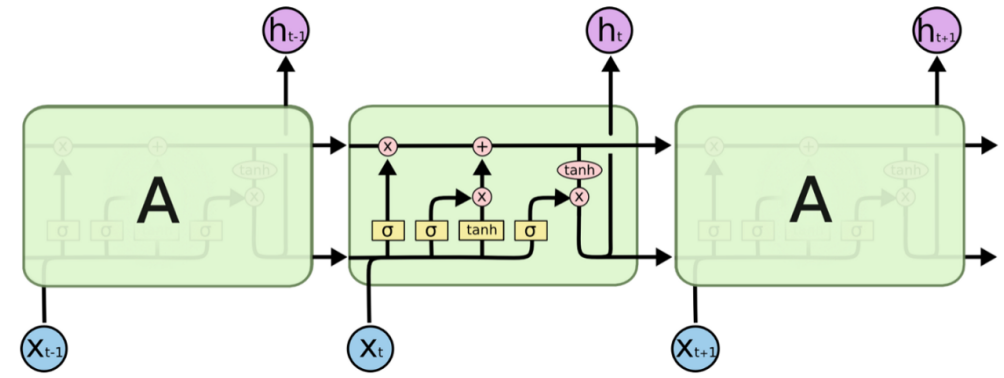
Although RNN is pretty much like our way to fill in a blank in a sentence, it still has some disadvantages. The information from distant sources will be “forgotten”, which is also called the vanishing gradient problem. Long Short Term Memory (LSTM) solved this problem by adding a forget gate, input gate, and output gate to decide the usage of memory.

We use the formula above to explain this. As W be updated for t times, it will become like below. W < 1. When t becomes larger, W^t will approach zero, which means that the information from a long time ago will have small weight, and that is not what we want.

To solve this problem, we add 3 gates to decide which memory to keep.

$$\begin{aligned} h_t &= (W^2 * h_{t-2} + W * U * h_{t-1} + W * b) + U * x_t + b \\ &= (W^t * h_0 + ...) + U * x_t + b \end{aligned}$$

## Core Concept of LSTM




source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Like RNN, LSTMs also have a similar structure, but the module inside it is different. There are four neural network layers in it. The horizontal line through the top of this diagram is the cell state, which is also the key point to LSTMs. LSTMs have gates, the yellow boxes pointing to red dots, to remove or add information to cell state. The sigmoid layers output between 1 and 0, which mean “keep all” and “block all.”

The first gate is called the forget gate, and it is made by a sigmoid layer. For example, the cell may include pronouns of a subject, and when we see a new subject, we want to forget the previous one. Next, we have an input gate to decide which value to update, and use another tanh layer to create a new candidate to update the information.

Finally, we decide what to output. The output will be a filter version based on the original cell state. We run a sigmoid layer first to decide which part to output. After that, we use tanh layer to push the value into -1 to 1, and multiply it by the results of the sigmoid one. For example, for a language model, it may output the information relevant to the number of some objects in case when it wants to output the verb next, so that we know which form of a verb should be used.

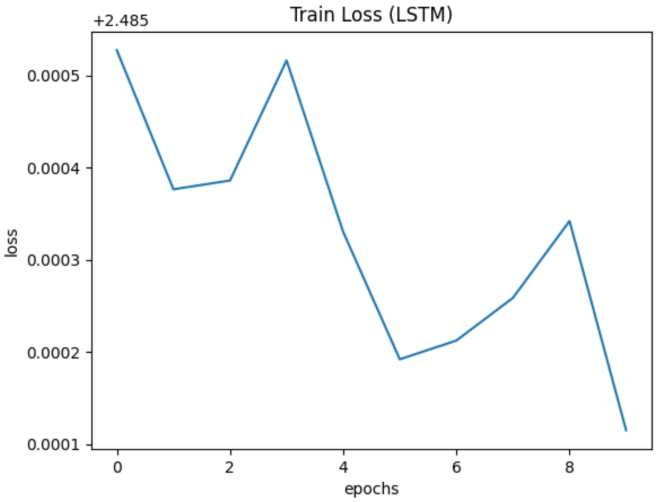
## Applications

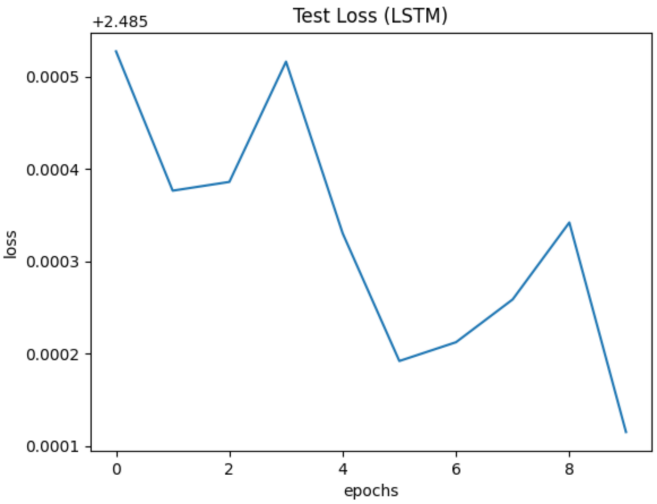
Besides audio recognition, this model is also useful in many fields, such as handwriting recognition, Natural Language Processing(NLP), robot control, music composition, and more. For instance, in NLP, it can be used for text generation, machine translation, and sentiment analysis. In music composition, it can be used to create new pieces of music that sound similar to existing pieces.

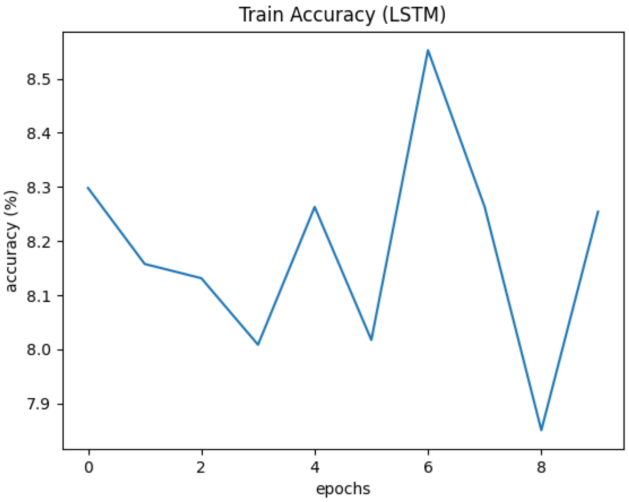
## Limitations

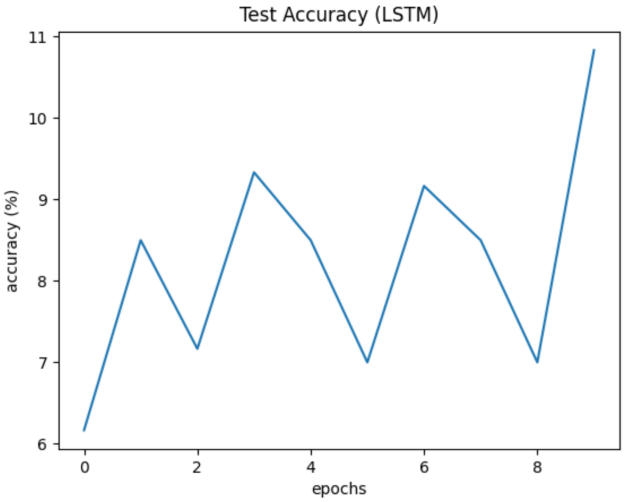
There are also limitations in RNN LSTM. One of them is that it requires a large amount of data and computing resources to train and implement. Additionally, the model may struggle with long sequences of data due to the vanishing gradient problem. Finally, the interpretability of the model may be difficult, making it hard to understand why certain decisions are being made.

# Implementation Results



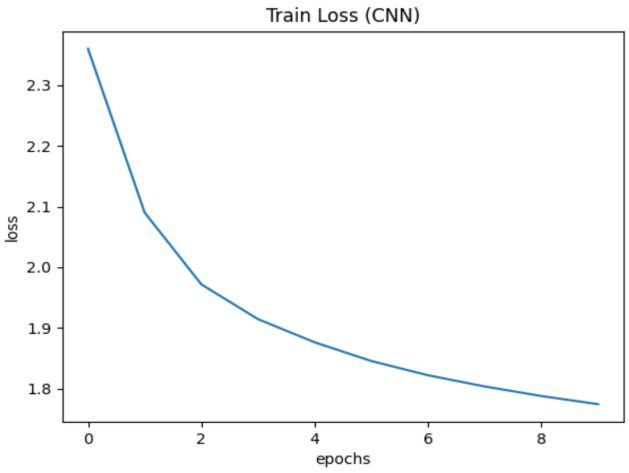



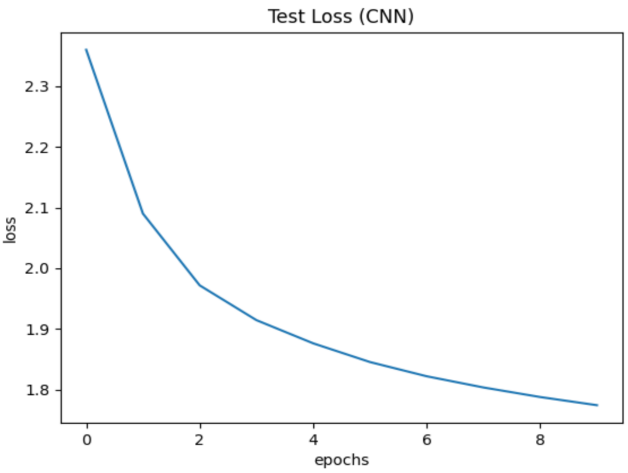



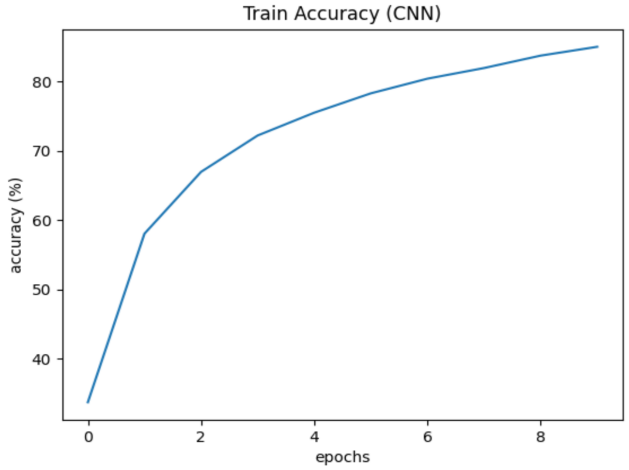

We're surprised to observe that the two trained models have no distinct difference in terms of the performance. The accuracy rate hovers around eight percent, while the loss is about 0.1. Besides, we observed the same result when applying the fBank feature.

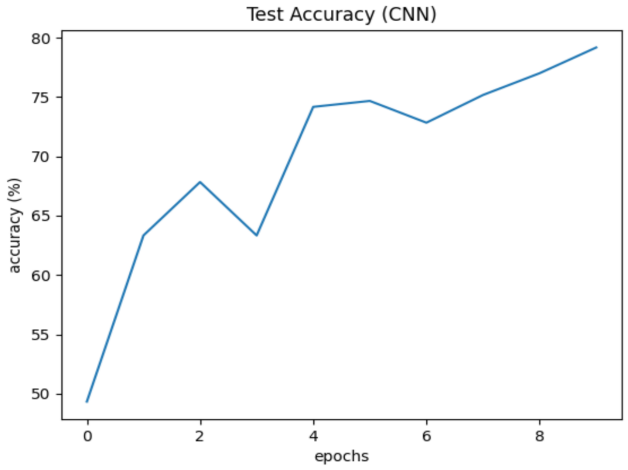
We have generated several graphs to visualize the performance across different time periods. In the LSTM model, the accuracy generally follows an upward trend, while the loss demonstrates a downward trend.. However, they suffer from the same issue, that is, instability.

The ability of the trained model is not that good as we expect, and we speculate that there might be an issue with the dataset leading to the poor performance. To verify our assumption, we applied the same dataset to a Convolution model as a control group, and it turns out yielding better results. From the charts below, it is obvious that both accuracy and loss are more stable going upward or downward.




Based on our observation, it appears that applying CNN yields better performance compared to LSTM. We list a few possible reasons:

1. the layers of LSTM models may require further adjustment.
2. LSTM may not be suitable for word classification. LSTM cells consider previous information, making them more suitable for data with significant time dependencies. However, our application aims at identifying independent instructions of numbers, so LSTM performs poorly in this case.

## Conclusion

In conclusion, we believe that LSTM does indeed work, but its performance falls short of our expectations. LSTM may not be suitable in this case.

# Links

## GitHub Repo

- https://github.com/YuChiao13579/Automatic_Speech_Recognition

## Dataset

- https://www.kaggle.com/competitions/tensorflow-speech-recognition-challenge/data

# Reference

## Implementation Method: Preprocessing


- https://www.researchgate.net/figure/Principial-block-scheme-of-MELPSEC-FBANK-and-MFCC-coefficients_fig1_286427067
- https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC
- https://www.researchgate.net/figure/Steps-to-obtain-FBANK-feature_fig2_341712278


## Implementation Method: Build Model


- http://d-scholarship.pitt.edu/36069/1/Ramadan_Mona_PhD_dissertation.pdf
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://github.com/mc6666/MyNeuralNetwork/tree/master/SpeechRecognition
- https://medium.com/ai-academy-taiwan/speech-recognition-using-neural-network-d1af6f482c9b
- https://github.com/lucko515/tesla-stocks-prediction/blob/master/tensorflow_lstm.ipynb