Campaign Speech Emotion Recognition
Group 7 Individual Final Report

Meng-Fei Hung

1. Introduction.

This project aims to develop a Speech Emotion Recognition (SER) model to analyze and interpret the emotional tone of campaign speeches delivered by U.S. presidential candidates. By examining these emotional dynamics, the project seeks to provide insights into the overarching strategies employed during their campaigns.

Various models were explored, with the best-performing one selected as the final SER model. As part of my individual contributions, I conducted extensive testing to evaluate the effectiveness of an LSTM model for SER using the CREMA dataset. Additionally, I implemented Streamlit to enrich the project by offering interactive visualizations and presenting the model's results. These contributions will be detailed further in this report.

2. Background Information

In recent years, LSTM networks have gained significant attention for applications involving time series events, with speech emotion recognition being a prominent example. Various techniques are utilized for building LSTM models, such as MFCC and autoencoders. For instance, Leelavathi (2021) and Hadhami (2020) demonstrate the use of MFCC for extracting emotional states from speech signals. Additionally, Hadhami (2020) and Abbaschian (2021) highlight that employing autoencoder-based dimensionality reduction enhances identification accuracy. These techniques will be implemented in this project to develop robust models.

3. Methodology

1) Data augmentation:
   Data augmentation is a technique that uses existing data to create new data samples to train machine learning (ML) models. It's used to increase the size and variability of a dataset to improve the model's accuracy.
   2) Feature Extraction(MFCC):
      MFCC is a feature extraction technique widely used in speech and audio

processing. MFCCs are used to represent the spectral characteristics of sound. Experiment with different numbers of MFCC features, such as 20,30,40.

3) Autoencoders:

Since the target labels are categorical, encoders are used to process these categorical target labels to make them compatible with machine learning models.

4) Model Architecture

Due to challenges in improving the performance of the LSTM model during the experiment, a hybrid approach was adopted, integrating CNN with LSTM for enhanced performance. The model architecture will include CNN layers and LSTM layers.

5) Hyperparameter Tuning

- Batch Size: Experiment with larger or smaller batch sizes to balance training stability and speed. Start from 32 or 64.
- Epochs: Train for more epochs.
- Number of layers:

The model is constructed following the described methodology, and the construction process is illustrated in the accompanying figure.
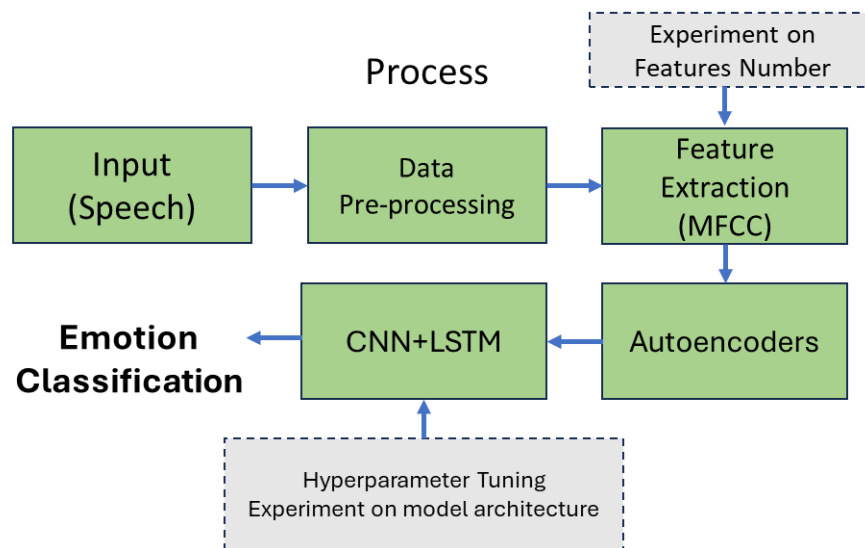

Figure 1 Process of Training Model

4. Results

1) LSTM-CNN model:

The data augmentation experiment revealed that applying augmentation reduced the model's performance compared to training without it. As a result, data augmentation was excluded when building the LSTM-CNN model.

- Model architecture :

  The model begins with two convolutional layers to capture local patterns and short-term temporal features in the data, such as those found in audio signals. Next, LSTM layers are employed to learn temporal relationships and long-term dependencies in the sequential data. Finally, a fully connected layer translates the extracted features into the final predictions.

- Hyperparameter Tuning:

  The experiment results of CNN layers are close, a model with 64 and 128 layers slightly outperforms the others. Increasing LSTM layers does not improve the performance of the model, thus the layer remains two.

Table 1 Hyperparameter Tuning: CNN Layers

CNN		LSTM		Batch Size	MFCC	Loss		Accuracy	
Conv1	Conv2	Number of Layers	Hidden Size			Train	Validation	Train	Validation
32	128	2	256	32	30	0.0409	0.0450	0.4694	0.4292
64	128	2	256	32	30	0.0396	0.0452	0.4893	0.4480
64	128	2	128	32	30	0.0433	0.4423	0.0447	0.4391
64	256	2	256	32	30	0.0356	0.5508	0.0473	0.4220

Table 2 Hyperparameter Tuning: LSTM Layers

CNN		LSTM		Batch Size	MFCC	Loss		Accuracy	
Conv1	Conv2	Number of Layers	Hidden Size			Train	Validation	Train	Validation
64	128	2	256	32	30	0.0396	0.0452	0.4893	0.4480
64	128	3	256	32	30	0.0400	0.0453	0.4780	0.4328
64	128	4	256	32	30	0.0432	0.0446	0.4362	0.4427

(conv1): Conv1d(1, 64, kernel_size=(3,), stride=(1,), padding=(1,))

(conv2): Conv1d(64, 128, kernel_size=(3,), stride=(1,), padding=(1,))

(pool): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)

(dropout): Dropout(p=0.5, inplace=False)

(lstm): LSTM(128, 256, num_layers=2, batch_first=True, dropout=0.5, bidirectional=True)

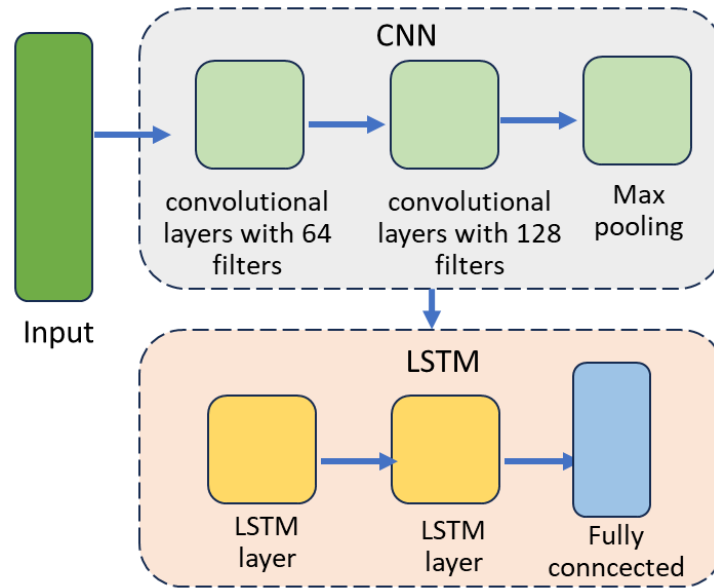(fc): Linear(in_features=512, out_features=6, bias=True)

Figure 2 LSTM-CNN Model Architecture

The validation accuracy of the LSTM-CNN model ranges from 40% to 45%, achieving a maximum validation accuracy of 44.9. The hybrid approach demonstrates a performance improvement of 1–2% compared to the standalone LSTM model.
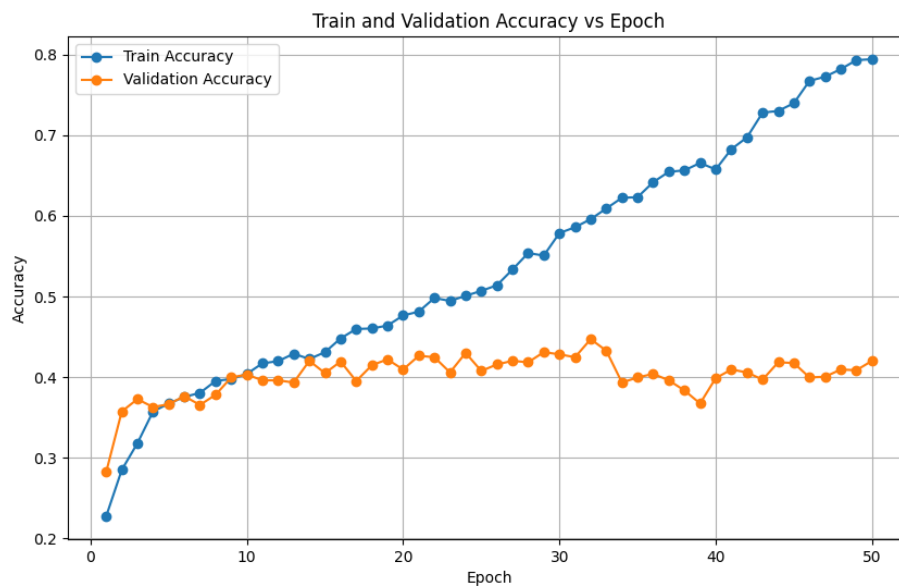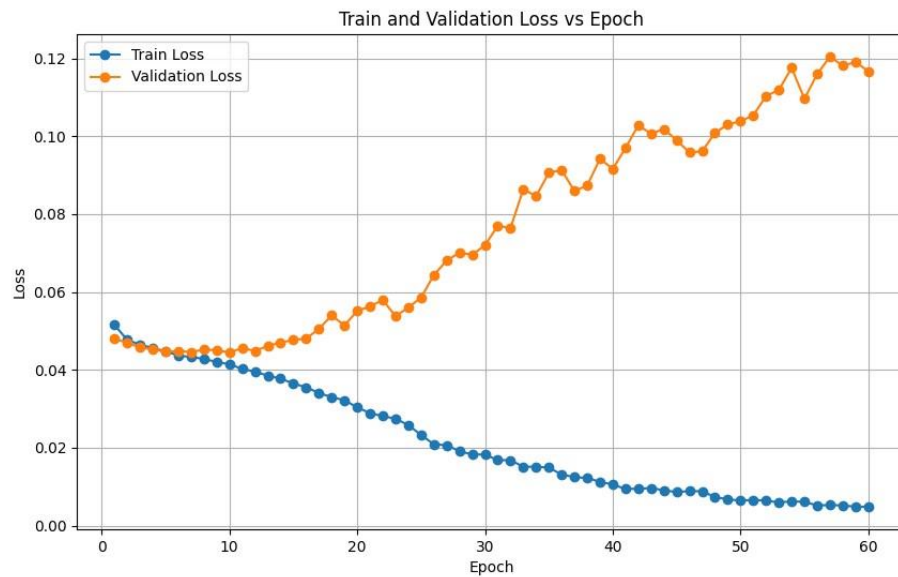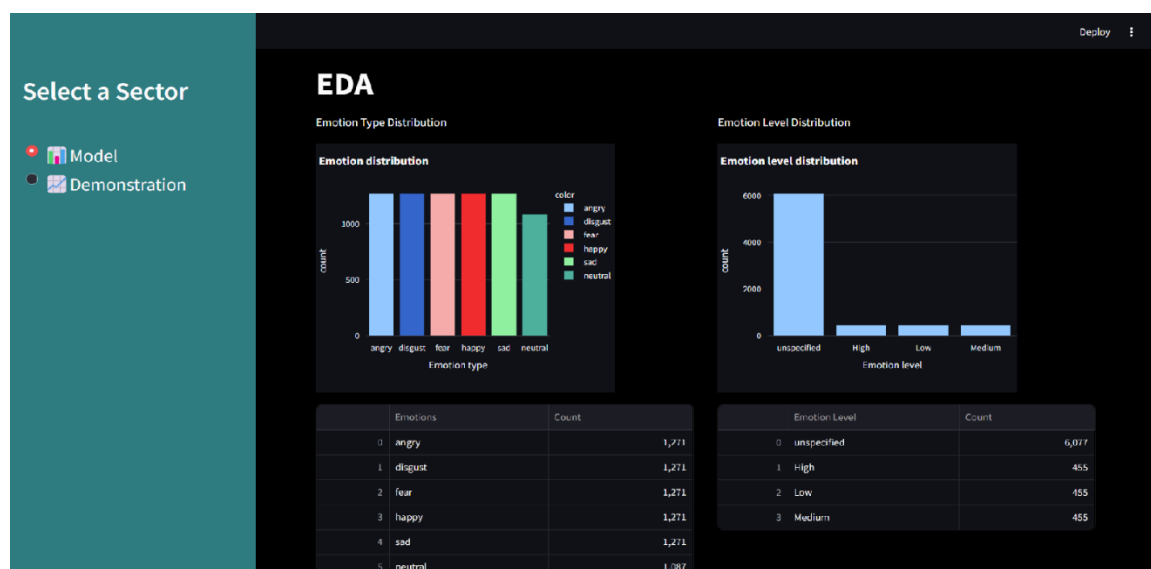

Figure 3 Accuracies of Train and Validation

Figure 4 Loss of Train and Validation

2) Streamlit:

I use Streamlit to create a dashboard showcasing our project research results and demonstrating speech recognition using our best models. The dashboard is divided into two sections: **Model** and **Demonstration**.

In the **Model** section, I present:

- Datasets, exploratory data analysis (EDA)
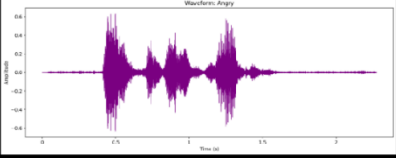
- sound and waveform display
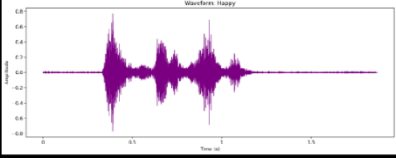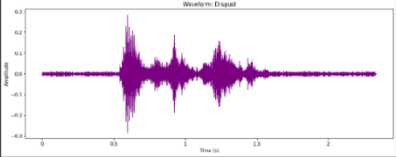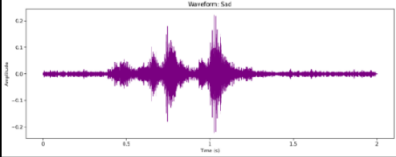
- Architecture and performance of five models.
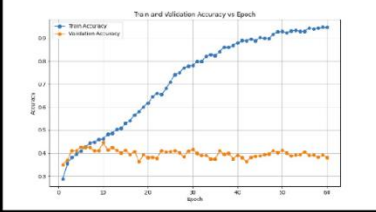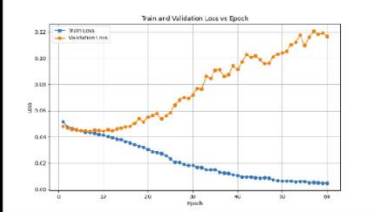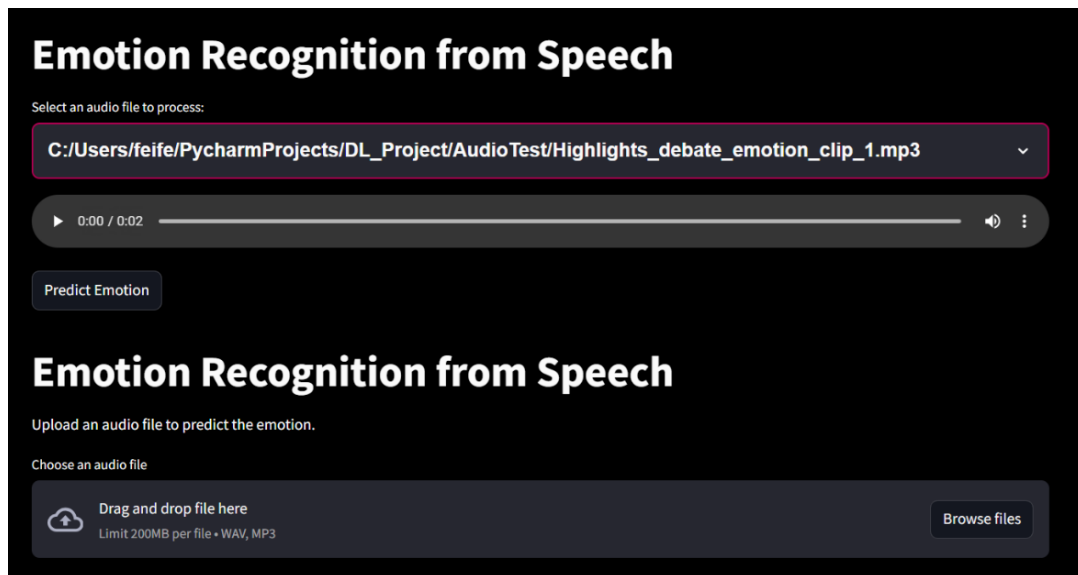

In the **Demonstration** section, we showcase the best-performing model by providing several preloaded audio files as examples. Users can also upload their own selected files to test the audio emotion recognition.

5. Summary and conclusions

The hybrid LSTM-CNN model demonstrates only a modest improvement of 1–2% compared to the standalone LSTM model, achieving validation accuracy between 40% and 45%. This suggests that the model's capability for speech emotion recognition is limited. One possible reason for the suboptimal performance could be the dataset's short audio clips, each lasting only two seconds, which may not provide enough data to effectively capture meaningful patterns for the LSTM component. In the future, incorporating audio datasets with longer durations or combining such datasets with the current ones could enhance the model's performance when using this hybrid approach.

6. The percentage of the code that is from the internet is 5%.

7. References.

1) Abbaschian, B. J., Sierra-Sosa, D., & Elmaghraby, A. (2021). Deep learning techniques for speech emotion recognition, from databases to models. *Sensors*, *21*(4), 1249. https://pmc.ncbi.nlm.nih.gov/articles/PMC7916477/
2) Aouani, Hadhami, and Yassine Ben Ayed. "Speech emotion recognition with deep learning." *Procedia Computer Science* 176 (2020): 251-260. https://www.sciencedirect.com/science/article/pii/S1877050920318512
3) Leelavathi, R., & ARUNA, V. (2021). Speech emotion recognition using LSTM. *International Research Journal of Engineering and Technology*.