Chris Washer
DATS 6303 Final Project – Individual Report

1. Introduction:

Speech is an important part of communication among human beings and is a way to express thoughts, feelings, and moods. Beyond just information being transferred, emotion plays an important role in speech communication and emotional speech is an effective way of conveying a message. Subtle changes in lexical or grammatical emphasis can change the meaning of a statement, with changing the words (1). This fact can be best demonstrated by the fact that a voice call is more informative than a text message.

Speech Emotion Recognition (SER) is a field of study which focuses on inferring human emotion from speech signals. SER is emerging as an important topic with applications in healthcare, machine-human interaction, education, intelligent assistance and many others (2). Many questions of SER are classification problems and can be addressed with supervised machine learning methods.

This project focuses on SER applied to political campaign speeches. By detecting emotions such as anger, happiness, fear, and neutrality from voice alone, the goal is to gain deeper insights into the emotional tenor of political messaging. Understanding these emotional cues can help decode communication strategies and the underlying sentiments intended to influence public opinion.

For this project, our group created several models including a CNN, LSTM, pretrained transformer model, a fine-tuned audio spectrogram transformer model, and a multimodal model. The overall goal of the project was to compare these model types and their performance for SER tasks and then apply the best performing model to political speech audio files.

2. Individual Contribution:

To contribute to the group, I worked on implementing the Audio Spectrogram Transformer model and the multimodal model. This required preprocessing of the data, literature review to understand the models, and then implementing the code to fine tune the models.

Audio Spectrogram Transformer:

The Audio Spectrogram Transformer (AST) is a purely attention-based system that is applied directly to an audio spectrogram rather than the audio features (3). The AST takes an approach of transferring knowledge from the Vision Transformer model (4). The AST model takes audio input and converts it to a spectrogram image. The model interprets spectrogram images as a series of patches which contain information about time and frequency from the input. The attention mechanism of the model allows it to look at relationships between any parts of the audio, no matter the distance.

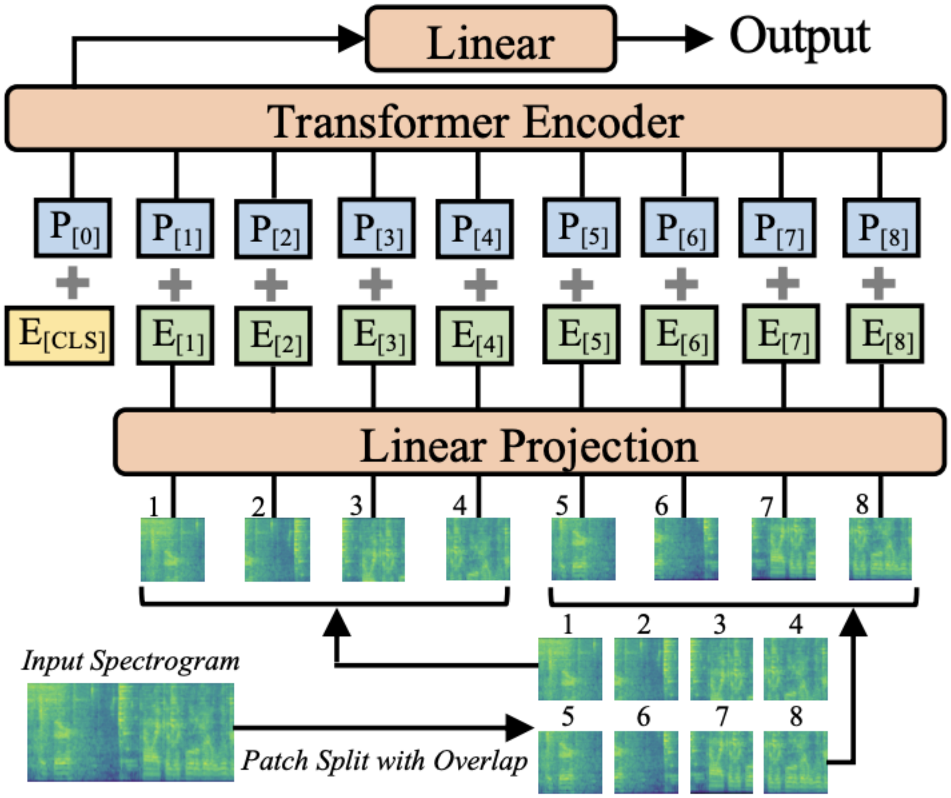


Figure 1: The AST model architecture. The model takes audio as an input in the form of a 2D spectrogram.

The AST architecture in Figure 1 from the initial publication demonstrates how the model uses a ViT-like architecture to analyze audio data. The AST model takes audio as an input in the form of a 2D spectrogram. The spectrogram is split into patches with overlap then linearly projected to a sequence of 1D patch embeddings, creating a single vector. Positional embeddings are added to the patch embeddings to keep order of the inputs. A classification token is added the embedding to accumulate information from all other tokens through the attention mechanisms.

After embedding, the model passes everything to the transformer encoder. The transformer consists of several layer of self-attention and feed forward networks, each layer allowing every patch to interact with every other patch and building more sophisticated representations of the audio input. After the transformer processes the input, the classification token contains a summary of the audio clip. The classification token's representation is passed to the final linear layer to get the classification output. (5)

In order to code the AST, I used the pretrained model from Huggingface and finetuned it on our dataset (6). Since I used a pretrained model, a lot of the code required was to properly set up the input data. Our dataset was raw wav files so I had to preprocess the data to extract audio spectrograms from the audio data, which can be seen in the AST.py script in the preprocessing functions. The script uses the pretrained Feature Extractor from AST.

This script also incorporates audio augmentation to improve the overall model and make it more generalizable. After the preprocess, the model was set to train to fine tune for the specific SER task of this project.

The final result of the model is an accuracy of 53.7% as shown in Figure 2.

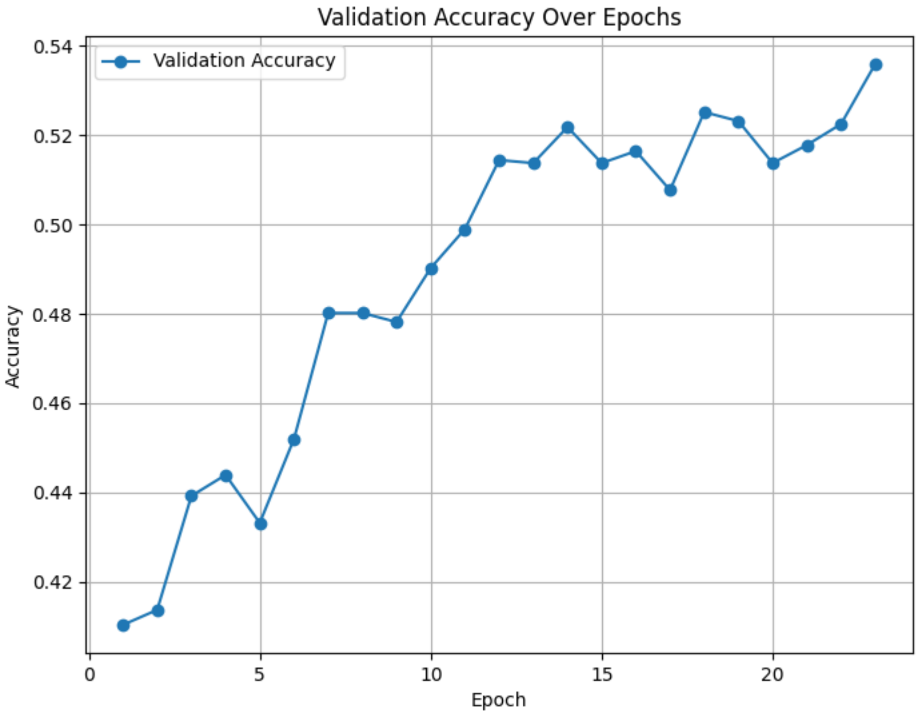


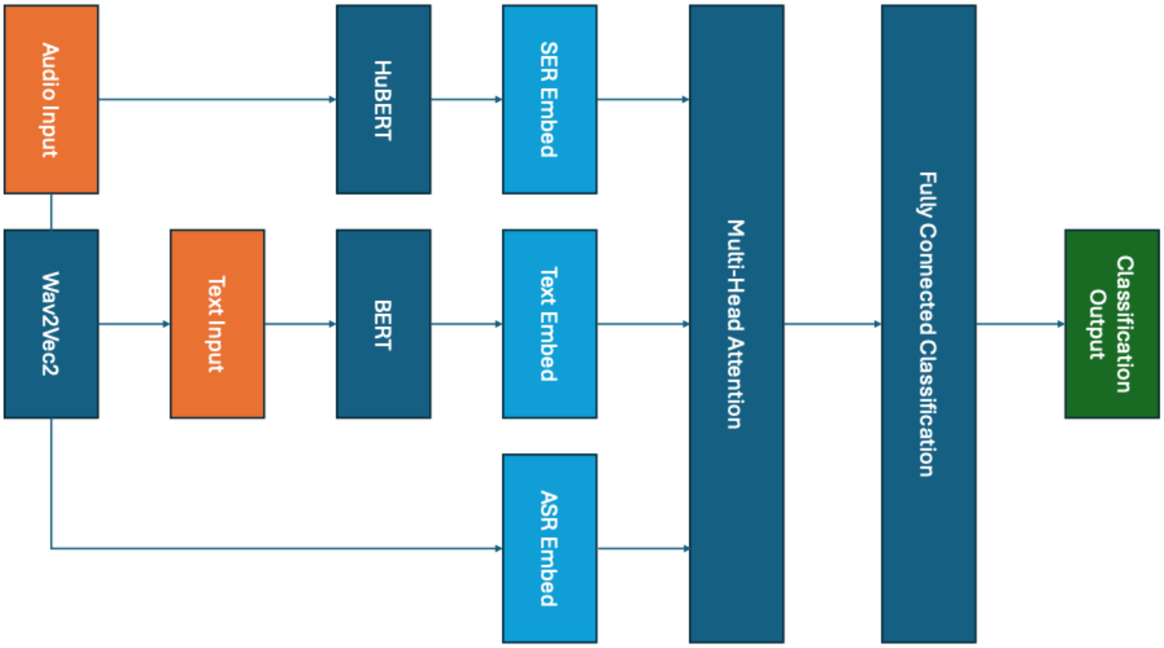
Figure 2 Results of AST model training over 25 epochs.

These results were an improvement on more traditional models such as the CNN and LSTM but was not as effective as other models incorporate in this project.

Multimodal Model:

After all of our modes were trained and evaluated on the dataset, I tried to incorporate Speech-to-Text and incorporate text classification to improve the overall model. This model combine our best performing model for SER, the HubertModel (8), added an element of Speech-to-Text using the pretrained Wav2Vec2 model (9), and a custom, fine-tune BERT model (10) for text classification.

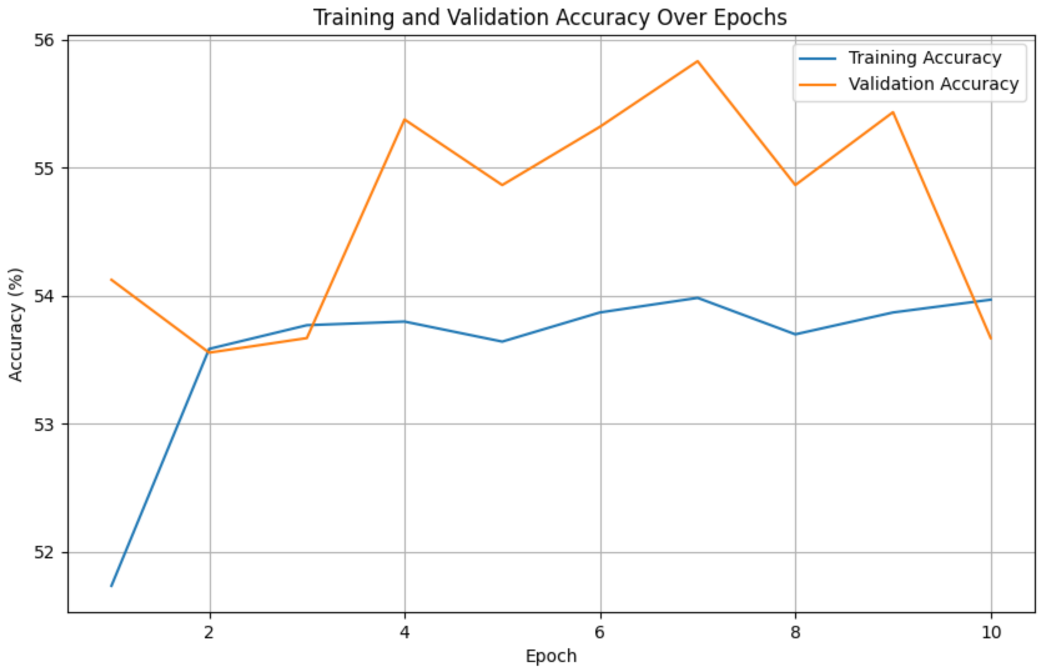
The multimodal model has the architecture shown in Figure 3.



This model incorporates several pretrained models to generate three modalities, SER, ASR, and text features. The embeddings are input to a multi-head attention layer. The multi-head attention layer aligns and integrates the three modalities and outputs a set of fused feature vectors that contains information from all modalities. The fully connected classifier layer produces the final output classification prediction.

In order to actually train this model, I needed to incorporate a new dataset. I used the MELD dataset (11). This dataset contains mp4 files from more than 1400 dialogues and 13000 utterances from the Friends TV series. The reason for this additional dataset was to provide a variety of dialogue from the CREMA-D data. This results in slight changes to the comparison from the other files.

Overall, the multimodal model achieved a total accuracy of 55.83%. This is a slight improvement from my AST model performance but the different datasets make it difficult to compare directly.



3. Conclusion

Overall, for this project I built two complex models to compare and improve SER on the CREMA-D dataset. The models both incorporated significant preprocessing. The ASD model generated spectrograms from the audio input. The multimodal model generated text transcripts and incorporated audio signal, SER, and text features as the input pretrained models.

SER is an important problem and these models provide an initial step to improve SER. These models can be applied to political data to help determine campaign approaches and understand how different candidates are communicating.

4. Code Usage:

For this project, approximately 65-70% of my code came from the internet. Much of this came directly from Huggingface or GitHub repository associated with the model publications. Significant portions of code were generate with ChatGPT (12) and Claude (13).

5. References

1. Pan, Z., Luo, Z., Yang, J., & Li, H. (2020). Multi-modal attention for speech emotion recognition. *arXiv preprint arXiv:2009.04107*.

2. George, S. M., & Ilyas, P. M. (2024). A review on speech emotion recognition: a survey, recent advances, challenges, and the influence of noise. *Neurocomputing*, *568*, 127015.
3. Gong, Y., Chung, Y. A., & Glass, J. (2021). Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.
4. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, July). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347-10357). PMLR.
5. https://sh-tsang.medium.com/review-ast-audio-spectrogram-transformer-a108a5775d2f
6. https://huggingface.co/docs/transformers/en/model_doc/audio-spectrogram-transformer
7. https://renumics.com/blog/how-to-fine-tune-the-audio-spectrogram-transformer
8. https://huggingface.co/docs/transformers/en/model_doc/hubert
9. https://huggingface.co/docs/transformers/en/model_doc/wav2vec2
10. https://huggingface.co/docs/transformers/en/model_doc/bert
11. https://affective-meld.github.io/
12. ChatGPT, OpenAI, 2024
13. Claude, Anthropic, 2024