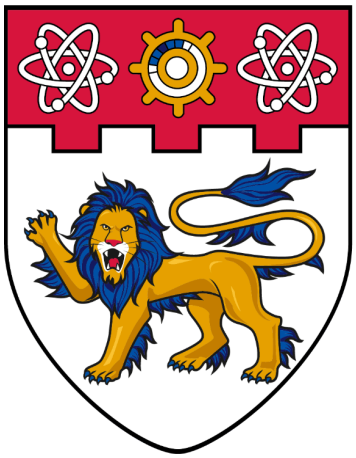


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC4001 - Neural Networks and Deep Learning

Group Project Report

Project Idea A&B: Speech and Text Emotion Recognition

Cholakov Kristiyan Kamenov (U2123543B)

Mishra Pradyumn (U2123912E)

Denzyl David Peh (U2122190F)

Table of Contents

1. Introduction

In the rapidly evolving worlds of Artificial Intelligence and Virtual/Augmented Reality, emotion recognition through speech stands out as a significant area of interest. Understanding and translating emotion can enhance the communication between individuals in a software environment. The choice to focus on this topic was inspired by **Topic A** and **Topic B** and the common gap between human emotions and AI's understanding of them. Most prior workings in the field focus on either speech or text emotion recognition, often treating these as disconnected areas.

In this report, we aim to present our solution for speech emotion recognition, integrated in combination with text emotion recognition. We are introducing a Machine Learning model combining a CNN for text emotion recognition with MLP for speech emotion recognition. Our decision to introduce an additional text emotion detection on top of the speech one was motivated by the numerous cases in which humans' real feelings are expressed as a combination or complete divergence of both. For example, often, LLM models struggle to detect the emotion of the user as they rely mainly on text. On the other hand, purely speech emotion detection via audio features extraction limits the model to understand only the way of speaking, such models are not able to understand the meaning behind the speech.

Our model was designed to interfere with the features that can be extracted from an audio file and the speech text from it. The aim of the proposed solution was to not use any context-related features or speaker characteristics.

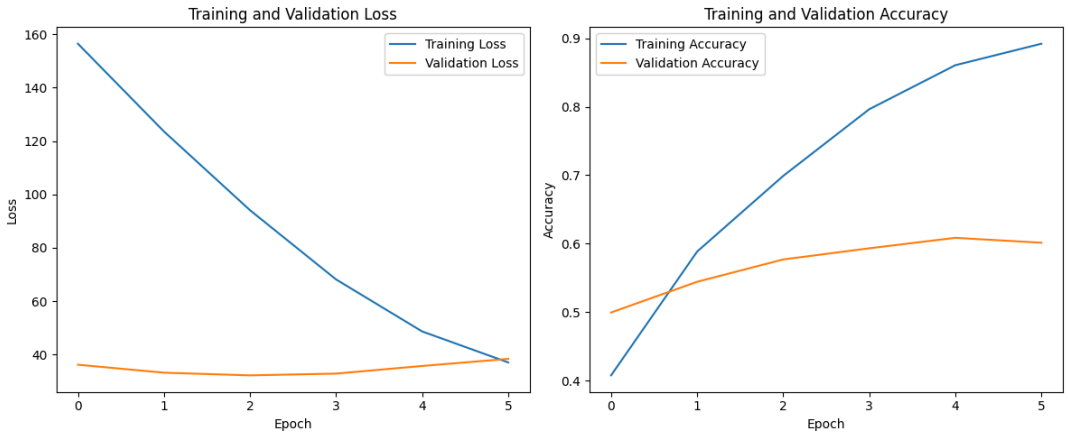
2. Dataset Selection

After researching the available datasets for speech emotion recognition, we decided to use the Interactive Emotional Dyadic Motion Capture [1] database developed at the SAIL Lab at the University of Southern California. The dataset is constructed of 10039 audio files in the "wav" format. Using this IEMOCAP for our project was motivated by the diverse distribution of the records in it. The audio files present both distinct audio features (screams, arguing, laughing, etc.) and semantic meaning. We consider this dataset suitable for our topic because both the audio and text feature models can be trained on it, meaning that there is enough, which is meaningful to the models. In comparison, other datasets we experimented with, like EmoDB [2] and RAVDESS [3], consist of a small number of sentences with different intonations. Such datasets are not applicable to our project as they do not provide sufficient features for our text model.

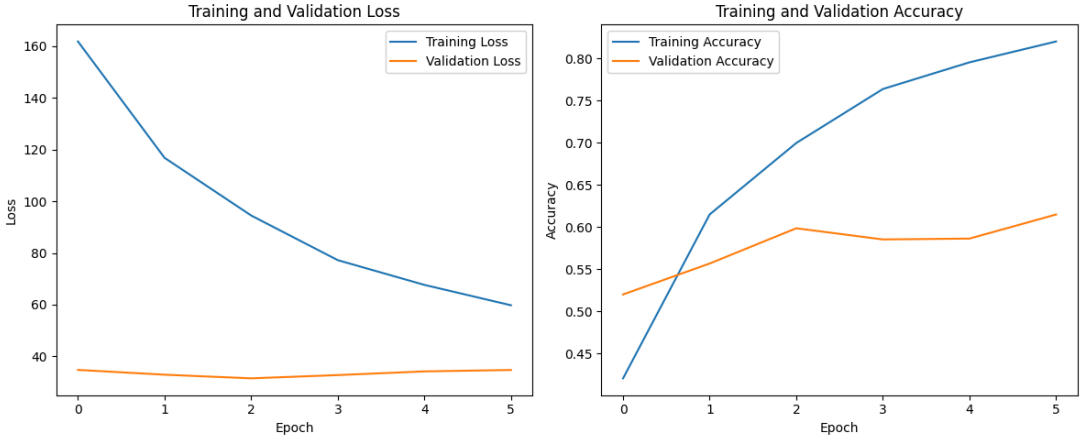
3. Model Selection

3.1 Text Emotion Recognition

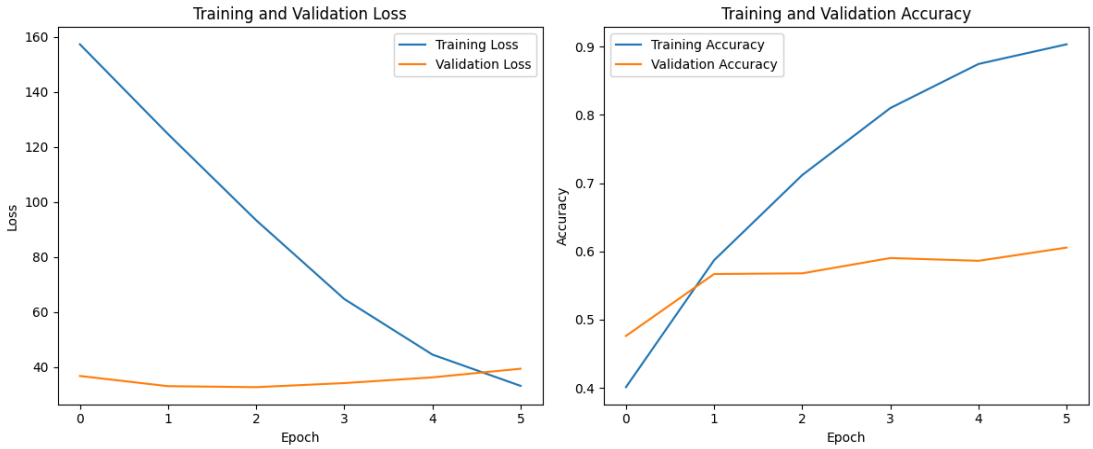
We decided to use a Convolutional neural network (CNN) to analyze the text data. While experimenting with other models like GRU, Bi-GRU and LSTM, we found that CNN achieved better results, although the accuracy for all models were roughly similar. This is also supported by the findings of the paper Text-Based Emotion Recognition Using Deep Learning Approach [4]. The architecture of the CNN model is discussed thoroughly under Section 6. The comparison between the three models can be seen in the graphs below (Graphs 1-3).



Graph 1. Loss & Accuracy evolution of the GRU model



Graph 2. Loss & Accuracy evolution of the CNN model



Graph 3. Loss & Accuracy evolution of the Bi-GRU model

3.2 Audio Features Speech Emotion Recognition

For speech emotion recognition based on audio features, we chose to implement an MLP. The connectivity pattern of MLPs allows complex non-linear mappings, which is useful for such high dimensional data. As mentioned in a study on neural network models on speech recognition models [5], MLPs outperformed both CNN and SVM in terms of accuracy. Their model was trained on the RAVDESS dataset, which is very similar to IEMOCAP in terms of size, utterances and structure. With these in mind, we decided to use a MLP for the audio features speech emotion recognition.

4. Audio Speech Text Extraction

For the purpose of extracting the text from the speech audio files, we integrated OpenAI's Whisper [6]. This choice was motivated by Whisper's [6] flexible and intuitive functionality, the model's python package integration allows seamless integration in our architecture. Furthermore, Whisper [6] has shown some significant improvements over other candidates like wav2vec 2.0 [7]. As presented in the experiments of Whisper, although both models achieve similar performance of WER=2.7 on the LibriSpeech Clean Dataset [8], Whisper shows a significant improvement on other sets with more noise in the audio files. Moreover, OpenAI's model has a language recognition functionality as well, which was highly beneficial during our experiments on various foreign language datasets. In order to make sure the model was performing optimally, we tested it on the LibriSpeech Other Dataset, we chose this dataset instead of the clean version because it consists of audio files with more noise that may present difficulties. We got a WER (%) of 15.3 for the "base" Whisper model, and 7.5 for the "large" one.

5. Data Processing

5.1 Text Processing

Speech text is crucial for emotion recognition accuracy. The transcribed text from the Whisper model must be processed before it can be used in the deep learning model. Tokenization converts each text in a sentence to a sequence of integers. These integers serve as features for the deep learning model. For our use case, we didn't process the Whisper output before passing it on to the tokenizer. For a standard NLP pipeline, steps such as lower casing and punctuation removal would be performed on the text before tokenization. For emotion recognition, punctuation marks may be crucial in correctly classifying the speech to its emotion. A sentence may be said in different tones to convey different emotions, the only way this can be reflected in text is through capitalization and punctuation marks.

The Whisper speech to text output was passed to a tokenizer to create a dictionary of the vocabulary of the text, mapping each text in the vocabulary to an integer and converting each sentence to a list of these integers. The Keras tokenizer from the Keras.preprocessing library was used for this task. Each output sentence from the tokenizer was padded to standardize the input length for the model.

5.2 Audio Feature Extraction

In order to perform deep learning on audio data, the data must be first processed to obtain audio features. Raw audio data typically has high dimensionality, which makes deep learning computationally expensive and makes it more challenging for the model to learn the patterns and features from the data. Audio feature extraction combats this by condensing the data into more meaningful descriptions of the data, in addition to reducing the dimensionality of the data. This not only reduces the computational overhead of

training, but also allows the model to learn faster and more effectively from these more informative features.

For audio feature extraction, we used 2 libraries: Librosa [9] and openSMILE (standalone python package) [10]. Both Librosa and openSMILE are commonly used for audio feature extraction. Librosa is more suited to music and audio analysis, while openSMILE extracts low-level features and functionals better suited to speech and music analysis.

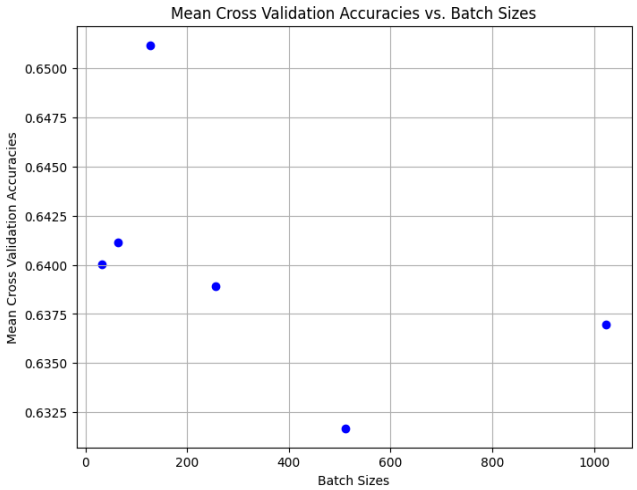
We used Librosa for the following:

- Mel-Frequency Cepstral Coefficients (MFCCs) – this provides information about the spectral characteristics of the signal, sometimes described as 'timbre'. This type of information is known to be useful for tasks like emotion recognition.
- Chroma Features – this is the distribution of pitch classes. Information on pitch and intonation could help with our task as well.
- Mel Spectrogram – this is a representation of the short-time Fourier transform of the audio signal, or in other words how loud different pitches are over time. This is useful for our task as well.

OpenSMILE offers a few different configuration sets for feature extraction, with the option of Low Level Descriptors (LLDs) or Functionals (computed from LLDs). We used the ComParE_2016 Functionals configuration to extract our audio features, for a total of 6373 additional features. This high dimensional feature representation captures many different aspects of the audio data, which we might have overlooked with Librosa.

5.3 Data Loading

To ensure that the model converges and that there isn't any numerical instability caused by large differences in features, we scale the data to normalize all features before passing it to the data loader. To ensure that the model does not memorize the order of the data, the data is shuffled and processed in batches. A batch size of 128 was used after testing the mean validation accuracy of the model over various batch sizes as shown in Graph 4.

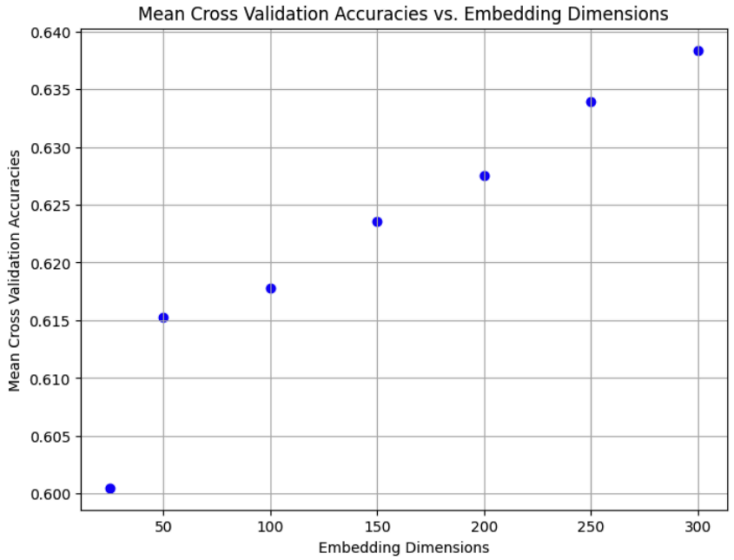


Graph 4. Mean Cross Validation Accuracies vs Batch Sizes

6. Convolutional Neural Network (CNN)

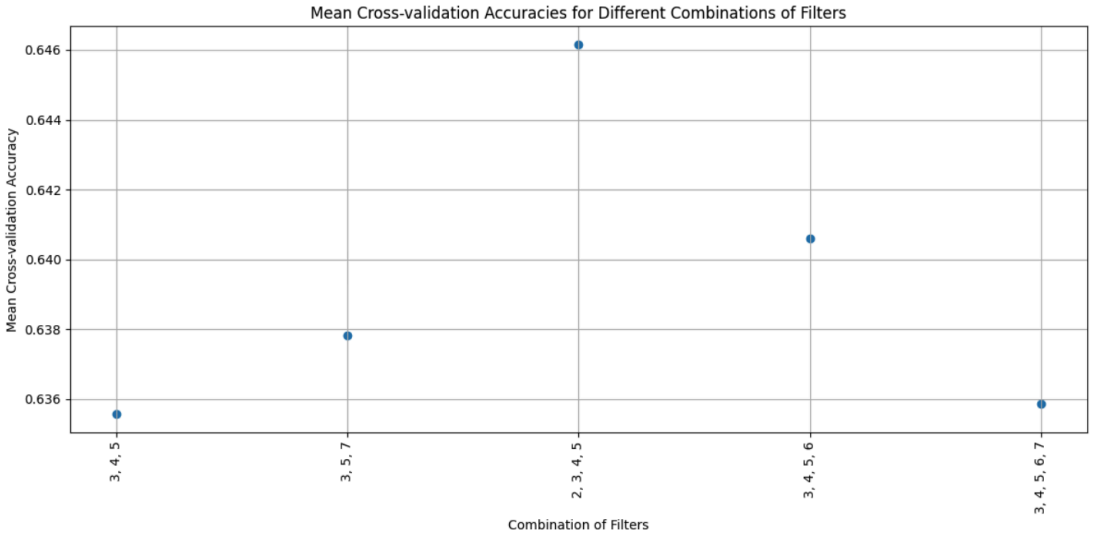
For classification using only text features, we used a CNN model with an Adam optimizer, using cross entropy loss as our loss function. We decided on the model architecture below:

- Embedding Layer: The input of the model is a set of tokenized integers that are passed to the embedding layer. The role of the embedding layer is to map each token to a dense vector representation of dimensions. The number of dimensions (embedding_dimensions) was tuned at 300, as seen from the graph below. It learns to represent words in a continuous vector space where similar words are closer together.

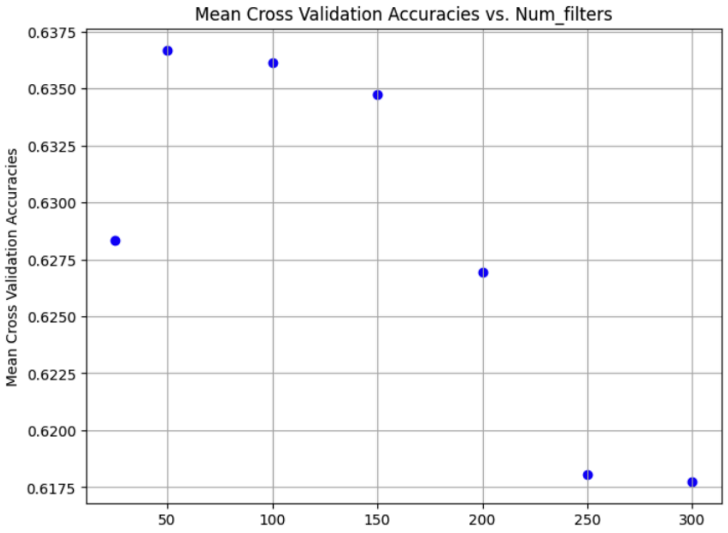


Graph 5. Mean Cross Validation Accuracies vs. Embedding Dimensions

- 4 Convolution layers: After tuning the “filter_sizes” hyperparameter, as seen from Graph 6., we decided to use 4 convolution layers with filter sizes - 2,3,4,5 and 6. For each filter size, a 2D convolution operation is applied with a kernel size of (num_filters, embedding_dim). This operation convolves a sliding window of the filter size over the embedded input tokens, producing a feature map. Filter_size was tuned at 50 as seen from the Graph 7.



Graph 6. Mean Cross Validation Accuracies vs.Combination of Filters



Graph 7. Mean Cross Validation Accuracies vs. Num_filters

- Activation function: We use the ReLU activation function to introduce non-linearity into the model. It is very computationally efficient, since it simply returns the input if it is positive, or 0 otherwise. It has been shown to perform well empirically as well.

- Max Pooling: Max pooling reduces the dimensionality of the feature maps while retaining the most important information. It extracts the maximum value within a sliding window and discards the rest.

- Concatenation: The output of max pooling operations from all filter sizes is concatenated along the channel dimension, resulting in a single tensor representing the extracted features from different filter sizes.

- Dropout: We use a dropout layer with p=0.5 between each layer in the model. This helps to reduce overfitting or over-reliance on certain neurons, making the model more robust. In addition to this, dropout reduces the number of computations done for every forward pass during training due to the reduced number of neurons active in each pass.

- Output layer: We passed the features from the dropout layer into a fully connected layer with 5 outputs. We then pass these outputs through a softmax layer to get the prediction probabilities of each class.

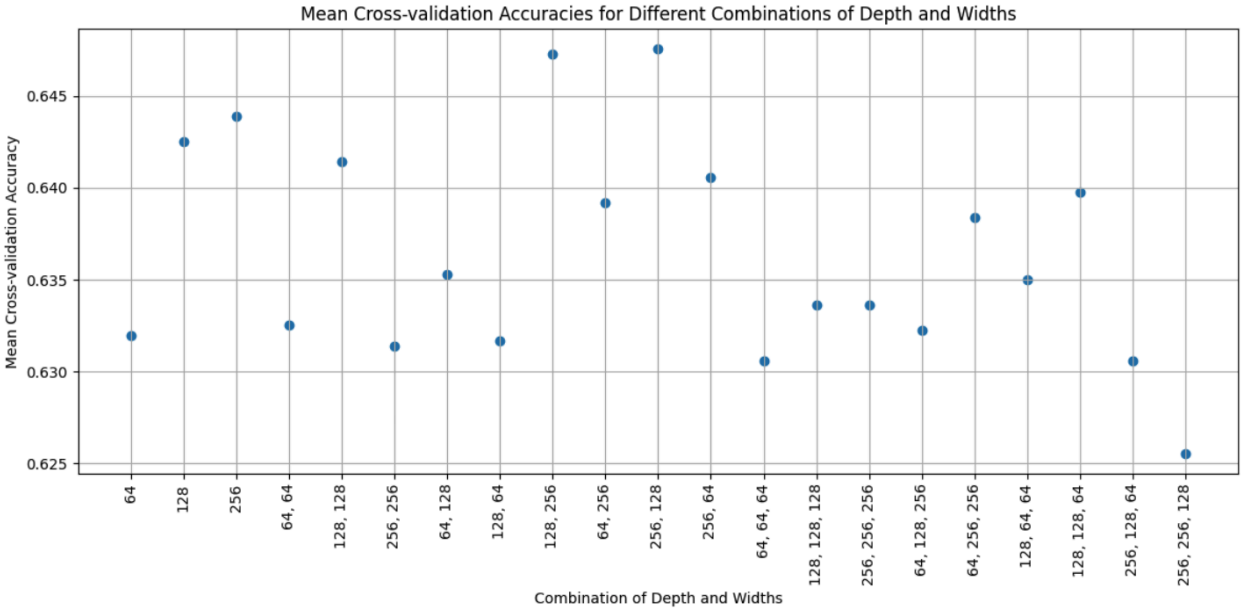
- Optimizer: Adam optimizer. This is a popular optimization algorithm, known for its adaptive learning rates and using momentum. This optimizer is also known for its empirical efficiency, as models that use this optimizer have been found to converge to lower losses with fewer epochs compared to other optimizers.

- Loss function: Cross Entropy Loss. This is a commonly used loss function for classification problems because it computes loss based on probabilities. It has clearly defined gradients, which helps gradient-based optimization algorithms like Adam work more efficiently.

7. Multi-Layer Perceptron (MLP)

For classification using only audio features, we used an MLP model with an Adam optimizer, using cross entropy loss as our loss function. We decided on the model architecture below:

- Hidden layers: 2 layers of 256 and 128 neurons. This was decided using a 5-fold cross validation algorithm varying the hidden layer architecture from 1 to 3 layers, with each layer having 64, 128 or 256 neurons each, as seen from Graph 8.



Graph 8. Mean Cross Validation Accuracies for Different combinations of Layers

- Activation function: We use the ReLU activation function to introduce non-linearity into the model. It is very computationally efficient, since it simply returns the input if it is positive, or 0 otherwise. It has been shown to perform well empirically as well.

- Dropout: We use a dropout layer with p=0.2 between each layer in the model. This helps to reduce overfitting or over-reliance on certain neurons, making the model more robust. In addition to this, dropout reduces the number of computations done for every forward pass during training due to the reduced number of neurons active in each pass.

- Output layer: We passed the features from the hidden layers into a fully connected layer with 5 outputs. We then pass these outputs through a softmax layer to get the prediction probabilities of each class.

- Optimizer: Adam optimizer. This is a popular optimization algorithm, known for its adaptive learning rates and using momentum. This optimizer is also known for its empirical efficiency, as models that use this optimizer have been found to converge to lower losses with fewer epochs compared to other optimizers.

- Loss function: Cross Entropy Loss. This is a commonly used loss function for classification problems because it computes loss based on probabilities. It has clearly defined gradients, which helps gradient-based optimization algorithms like Adam work more efficiently.

8. Fusion Model

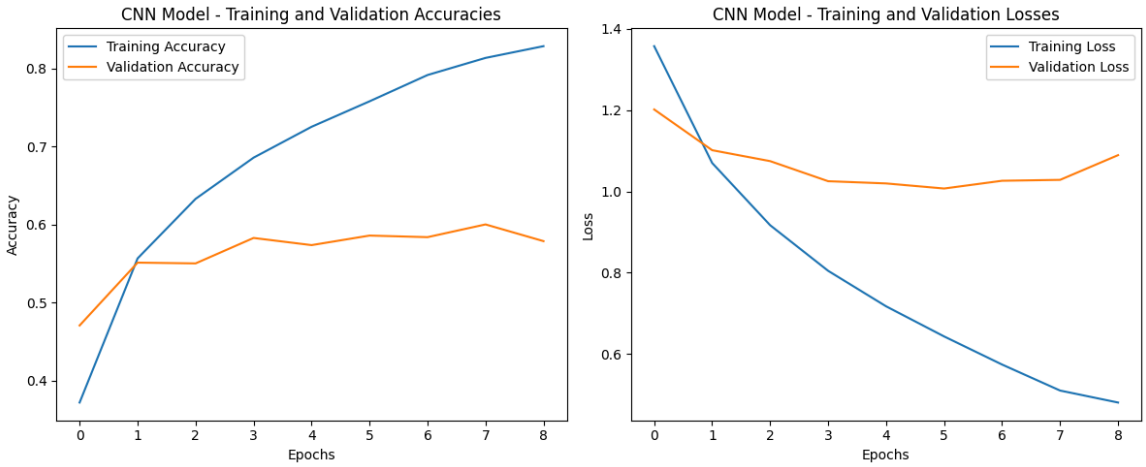
In order to combine the predictions based on audio features and textual features, we decided to develop another model that takes the 2 sets of output predictions and outputs a final prediction. We implemented this with a single fully connected layer and a softmax layer. After both models are trained and have their optimal weights saved, we load the weights into the models for inference when training the fusion model. The 2 models are set to ‘eval’ mode, so that their weights will not be updated during the backpropagation step.

The idea behind this is that the predictions from the CNN model and the MLP model may work better for some classes and worse for others. This fusion model learns appropriate weights to prioritize audio or textual features depending on the outputs of the models. With this model, we see improvements in accuracy compared to the two standalone models.

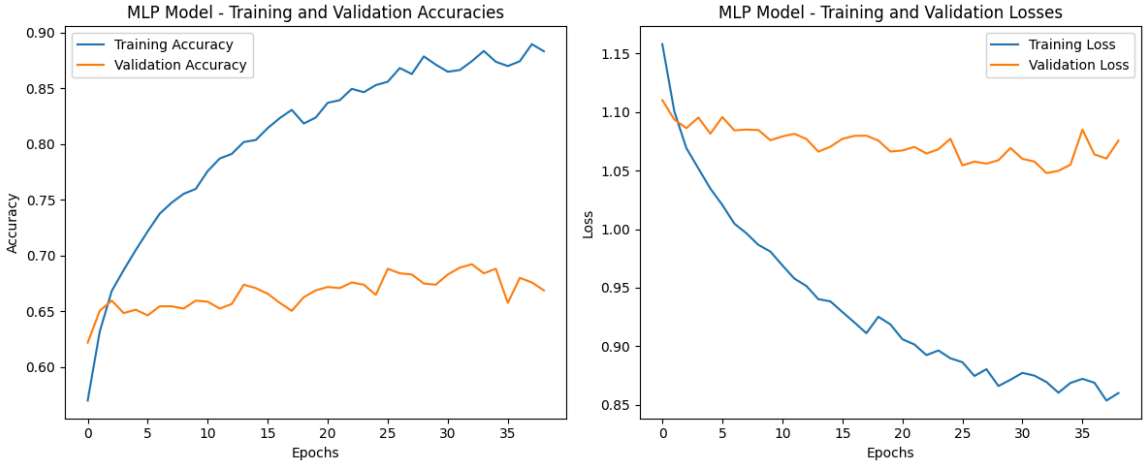
9. Results

We trained the model for a maximum of 100 epochs, using an early stopping algorithm with patience=3 and delta=0.1. Model accuracy was calculated by dividing the number of correct predictions by the total number of data points used in the train or test set. Whenever the validation loss improves, the model weights are saved, ensuring that we keep the optimal weights for later use.

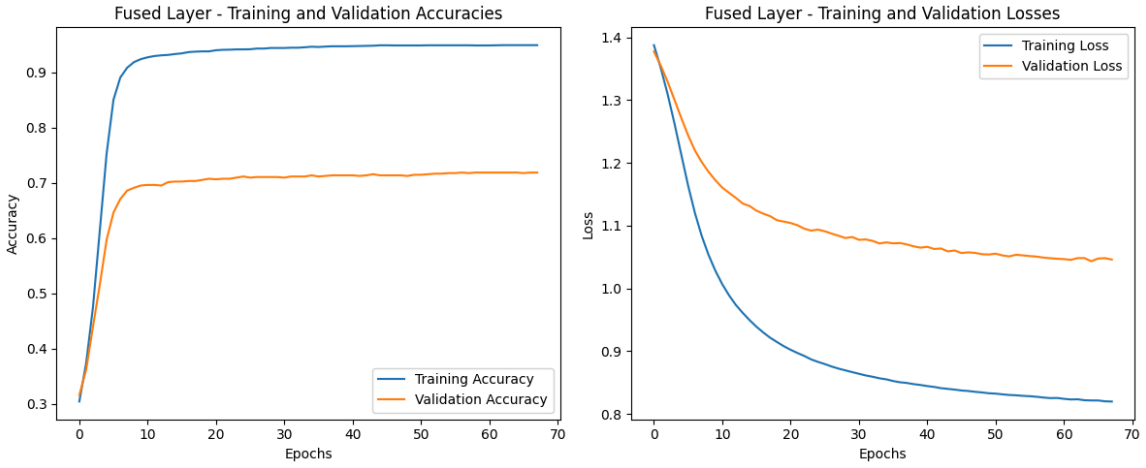
The following graphs show the model losses and accuracy during training for the 3 models. Note that the highest accuracy reached is 71.87% for the fused model.



Graph 9. CNN Model Accuracies and loss



Graph 10. MLP Model Accuracies and loss



Graph 11. Fused Model Accuracies and loss

Model	Validation Accuracy
CNN (Text)	0.6004077
MLP (Audio)	0.6890927
Fused (Text + Audio)	0.7186544

Table 1. Validation Accuracy of the Models

As seen from the results in Section 9, the fused model is able to classify the emotions better than either the text or audio individually at 71.87% compared to 60.04% for CNN and 68.91% for MLP. This shows that both the textual and audio features are needed to accurately classify the emotion for speech.

These models were also trained using text available in the IEMOCAP dataset. As the text obtained from the Whisper isn't 100% accurate, and also does not include some very important features such as punctuations that depict emotion, the accuracy for this model is lower by roughly 5%.

As seen from the graphs above, both the CNN and MLP have relatively low accuracy throughout, and don't increase much over epochs. This means that the model isn't able to learn the relationship between the variables very accurately. We believe that this may be because of the relatively small dataset, with roughly 4500 data points over 4 emotions. These data points may not be enough to accurately predict the relationships between the features and emotions.

Our model accuracy is quite high compared to other models. In a paper [11] on RNNs for emotion detection, that used partial context to infer emotion, achieved a maximum accuracy of 63.4%. Although their model is trained to predict 2 extra classes, the relatively high accuracy of our model shows that it is comparable to other models that use context to infer emotion.

10. Conclusion

In this project, we explored speech emotion recognition using text analysis on a pre-trained speech to text model and fusing it with an audio features analysis model. We aimed to accurately predict emotions by combining a Convolutional Neural Network for textual analysis and a Multi Layered Perceptron Model for audio feature analysis.

Our experiments on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) database revealed several important insights. Firstly, we observed that combining both textual and audio features significantly improved emotion classification accuracy compared to individual modalities. The fusion model achieved a validation accuracy of 71.87%, outperforming both the CNN (60.04%) and MLP (68.91%) models. This shows the importance of using both textual and audio features to infer emotional characteristics from speech. While text-based models rely on linguistic cues, such as words and phrases, audio-based models capture variations in tone, pitch, and intensity. By integrating these modalities, our fusion model learned to effectively combine the strengths of both approaches, leading to improved performance.

However, a few challenges remain, such as dealing with inaccuracies in speech-to-text conversion and the limited size of the dataset. The accuracy of the models may be further enhanced by refining the preprocessing steps and augmenting the dataset with additional examples.

11. Bibliography

[1] Interactive Emotional Dyadic Motion Capture (IEMOCAP). University of Southern California. https://sail.usc.edu/iemocap/. Accessed 15 Apr. 2024.

[2] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of German emotional speech," in Proc. 9th European Conf. Speech Communication and Technology, 2005, pp. 1517-1520. doi: 10.21437/Interspeech.2005-446.

[3] S. Livingstone and F. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," PLOS ONE, vol. 13, e0196391, 2018. doi: 10.1371/journal.pone.0196391.

[4] Bharti, Santosh Kumar, et al. “Text-Based Emotion Recognition Using Deep Learning Approach.” Computational Intelligence and Neuroscience, edited by Vijay Kumar, vol. 2022, Hindawi Limited, Aug. 2022, pp. 1–8. Crossref, doi:10.1155/2022/2645381.

[5] A. P. Singh, I. Singhal, K. Kaushik, and N. Sharma, “Speech emotion recognition using MLP classifier,” SSRN Electronic Journal, 2023. doi:10.2139/ssrn.4637856

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision.” arXiv, Dec. 06, 2022. Accessed: Apr. 15, 2024. [Online]. Available: http://arxiv.org/abs/2212.04356

[7] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.” arXiv, Oct. 22, 2020. Accessed: Apr. 15, 2024. [Online]. Available: http://arxiv.org/abs/2006.11477

[8] A. Meister, M. Novikov, N. Karpov, E. Bakhturina, V. Lavrukhin, and B. Ginsburg, “LibriSpeech-PC: Benchmark for Evaluation of Punctuation and Capitalization Capabilities of end-to-end ASR Models.” arXiv, Oct. 04, 2023. Accessed: Apr. 15, 2024. [Online]. Available: http://arxiv.org/abs/2310.02943

[9] B. McFee *et al.*, “Librosa: Audio and Music Signal Analysis in python,” *Proceedings of the 14th Python in Science Conference*, 2015. doi:10.25080/majora-7b98e3ed-003

[10] F. Eyben, M. Wöllmer, and B. Schuller, “openSMILE -- The Munich Versatile and Fast Open-Source Audio Feature Extractor,” *Proceedings of the 18th ACM international conference on Multimedia*, Oct. 2010. doi:10.1145/1873951.1874246

[11] N. Majumder *et al.*, “Dialoguernn: An attentive RNN for emotion detection in conversations,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6818–6825, Jul. 2019. doi:10.1609/aaai.v33i01.33016818

12. Links

Training Data Link:

https://entuedu-my.sharepoint.com/:u:/g/personal/kristiya001_e_ntu_edu_sg/EfZGKzUHznJHmHPrR8G96sABZUSOtPDlKB7eUCAxgENLWA?e=Vc8bmP

IEMOCAP Database:

https://www.kaggle.com/datasets/dejolilandry/iemocapfullrelease

Whisper Evaluation Data:

https://www.openslr.org/12