Speech Emotion Recognition

Rosalind Margaret Paulson
IIT Jodhpur
Karwar, Jodhpur, India
m23csa526@iitj.ac.in

Torsha Chatterjee
IIT Jodhpur
Karwar, Jodhpur, India
m23csa536@iitj.ac.in

Abstract

This paper explores the application of deep learning methods to speech emotion recognition (SER) using the CREMA-D dataset. The primary objective is to classify audio recordings into six basic emotional categories - angry, disgust, fear, happy, neutral, and sad - based on acoustic features extracted from the speech signal. Mel-Frequency Cepstral Coefficients (MFCCs) are employed to represent each audio sample in a form suitable for machine learning. A Convolutional Neural Network (CNN) is trained to learn discriminative patterns from the MFCCs and perform multi-class classification. The study demonstrates that MFCC-based CNNs are effective in recognizing emotional states in speech, providing a foundation for future development of emotion-aware audio systems.

1. Introduction

Speech emotion recognition (SER) is a critical area within affective computing and human-computer interaction, aiming to identify the emotional state of a speaker based on vocal cues. With growing applications in virtual assistants, healthcare, customer service, and adaptive learning systems, the ability to accurately interpret emotions from speech signals is both relevant and impactful.

This study focuses on the development of a speech emotion classification model using the CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset), which comprises audio recordings of actors portraying a range of emotions. The dataset includes six discrete emotion labels: Angry, Disgust, Fear, Happy, Neutral, and Sad.

The objectives of this work are threefold:

1. To preprocess the dataset and organize it into a machine learning-friendly format.

2. To extract relevant audio features, with an emphasis on Mel-Frequency Cepstral Coefficients (MFCCs), which capture the timbral texture of speech.

3. Design and train a deep learning model—specifically, a Convolutional Neural Network (CNN) — to classify emotional states based on extracted features.

The methodology involves systematic data loading, feature engineering, label encoding, and model training using supervised learning. Model performance is evaluated using standard metrics such as classification accuracy and cross-entropy loss. The results demonstrate the potential of MFCC-based CNN architectures in achieving reliable emotion classification from raw audio signals.

This work contributes to ongoing research in emotion-aware technologies and provides a reproducible pipeline for future investigations into multimodal or cross-lingual emotion recognition tasks.

2. Methodology

2.1. Dataset and Preprocessing

The CREMA-D dataset, a publicly available resource consisting of 7442 audio clips labeled by emotion, is used in this study. Emotion labels are parsed from the filenames, and all audio files are standardized using a uniform sampling rate.

2.2. Feature Extraction

Each audio file is processed using the Librosa library to extract three key audio features: Zero-Crossing Rate (ZCR), Root Mean Square Energy (RMSE), and Mel-Frequency Cepstral Coefficients (MFCCs). Together, these features provide a compact representation of the temporal, energy, and spectral characteristics of the audio signal. The extracted features are concatenated into a single feature vector for each audio file, which serves as input to the downstream machine learning models.

Purpose: To transform raw.wav audio files into fixed-size numerical feature vectors that capture essential speech characteristics for classification or regression tasks.

librosa.load(): The audio waveform is loaded from the file and resampled to a consistent sampling rate (default: 22,050 Hz).

librosa.feature.`zero_crossing_rate()`: Computes the rate of sign changes in the waveform, which correlates with the signal's frequency content and noisiness.

librosa.feature.rms(): Measures short-time signal energy, which reflects loudness and is helpful in identifying emphasis or emotion in speech.

librosa.feature.mfcc(): Computes Mel-Frequency Cepstral Coefficients, which summarize the short-term spectral shape of the signal. In this implementation, the MFCCs are flattened into a 1D vector to fit the traditional machine learning models.

Feature Combination: The output of ZCR, RMSE, and MFCCs is horizontally stacked (np.hstack) to form a single combined feature vector.

Flattening: MFCC features are optionally flattened using np.ravel() to ensure a fixed-length 1D representation, suitable for feeding directly into classifiers.

2.3. Data Augmentation

To improve model generalization and robustness, audio data augmentation techniques are applied before extracting features. Each .wav file is loaded using librosa, and several variants of the original signal are generated using common augmentation methods such as adding noise, pitch shifting, and time shifting. For each augmented version, relevant audio features are extracted and stacked together to form an extended feature set for the original audio sample. The following augmentation methods are implemented:

- Noise Injection: Adds Gaussian noise to the signal with amplitude proportional to the maximum signal amplitude.
  Purpose: Simulates background noise conditions.
- Pitch Shifting: Shifts the pitch of the audio up or down without changing its duration. Default value of pitch factor is 0.7 (librosa.effects.`pitch_shift`)
  Purpose: Introduces pitch variation to improve robustness.
- Combined Noise and Pitch Shifting

2.4. Label Extraction and Encoding

Emotion labels identified in the dataset are:

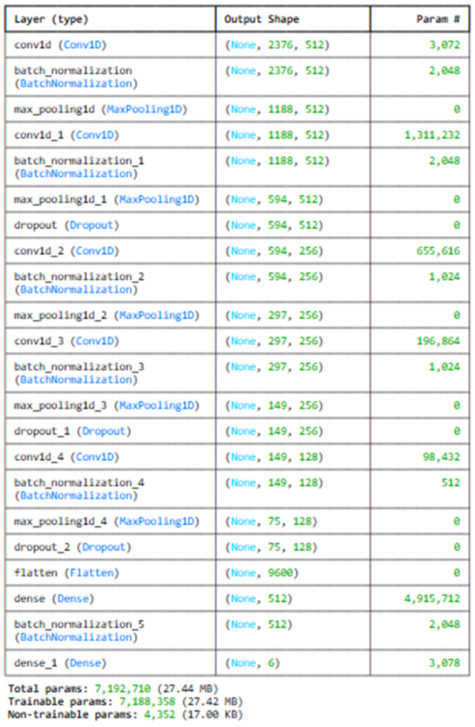
'ANG' = Angry
'DIS' = Disgust
'FEA' = Fear
'HAP' = Happy
'NEU' = Neutral
'SAD' = Sad

Emotion labels are encoded using one-hot vectors to facilitate multi-class classification. Label encoding is performed using LabelEncoder from sklearn, and labels are converted to categorical format using TensorFlow utilities.

2.5. Model Architecture

A Convolutional Neural Network (CNN) is designed with two convolutional blocks, each followed by max pooling and dropout layers to prevent overfitting. The flattened output is passed through a dense hidden layer before final classification via a softmax output layer. The model is compiled with the Adam optimizer and categorical cross-entropy loss function. Key layers:

- Conv2D: Applies convolution filters to extract local features from the MFCCs.
- MaxPooling2D: Reduces dimensionality and overfitting by taking the maximum in a region.
- Dropout: Randomly drops neurons during training to prevent overfitting.
- Flatten: Converts 2D features into 1D vector for fully connected layers.
- Dense: Fully connected layers to perform classification.
- Final layer uses softmax to output probabilities for each emotion.



Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2376, 512)	3,072
batch_normalization (BatchNormalization)	(None, 2376, 512)	2,048
max_pooling1d (MaxPooling1D)	(None, 1188, 512)	0
conv1d_1 (Conv1D)	(None, 1188, 512)	1,311,232
batch_normalization_1 (BatchNormalization)	(None, 1188, 512)	2,048
max_pooling1d_1 (MaxPooling1D)	(None, 594, 512)	0
dropout (Dropout)	(None, 594, 512)	0
conv1d_2 (Conv1D)	(None, 594, 256)	655,616
batch_normalization_2 (BatchNormalization)	(None, 594, 256)	1,024
max_pooling1d_2 (MaxPooling1D)	(None, 297, 256)	0
conv1d_3 (Conv1D)	(None, 297, 256)	196,864
batch_normalization_3 (BatchNormalization)	(None, 297, 256)	1,024
max_pooling1d_3 (MaxPooling1D)	(None, 149, 256)	0
dropout_1 (Dropout)	(None, 149, 256)	0
conv1d_4 (Conv1D)	(None, 149, 128)	98,432
batch_normalization_4 (BatchNormalization)	(None, 149, 128)	512
max_pooling1d_4 (MaxPooling1D)	(None, 75, 128)	0
dropout_2 (Dropout)	(None, 75, 128)	0
flatten (Flatten)	(None, 9600)	0
dense (Dense)	(None, 512)	4,915,712
batch_normalization_5 (BatchNormalization)	(None, 512)	2,048
dense_1 (Dense)	(None, 6)	3,078

```
Total params: 7,192,710 (27.44 MB)
Trainable params: 7,188,358 (27.42 MB)
Non-trainable params: 4,352 (17.00 KB)
```

Figure 1. CNN Model Architecture

2.6. Training and Evaluation

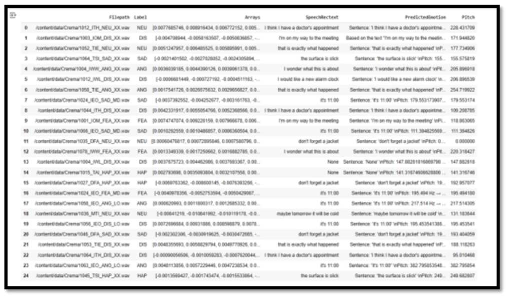
The dataset is split into training and testing subsets (80/20 split). The CNN is trained for 30 epochs with a batch size of 32. Model performance is evaluated on the test set using accuracy and loss metrics. Learning curves are plotted to analyse training dynamics and generalization performance.

3. Analysis

3.1. LLM-based Emotion Prediction

An alternative pipeline was developed to predict emotions from speech by leveraging both prosodic and textual information. The audio was first processed using an automatic speech recognition (ASR) system to obtain the spoken text, and corresponding pitch values (in Hz) were extracted. These two modalities were then passed together as a structured prompt to OpenAI's GPT-4o model via the Chat Completion API for prompt-based emotion inference. The process involved:

Pipeline: Speech → Pitch Extraction + Speech-to-Text → GPT-4o (via prompt-based emotion inference)



	Filepath	Label	Arrays	SpeechRectext	PredictedEmotion	Pitch
0	/content/data/Crema/1012_ITH_NEU_XX.wav	NEU	[0.0077685746, 0.008916434, 0.006772152, 0.005...	I think I have a doctor's appointment	Sentence: 'I think I have a doctor's appointme...	228.431709
1	/content/data/Crema/1003_IOM_DIS_XX.wav	DIS	[-0.004708944, -0.0058163507, -0.0050836857, -...	I'm on my way to the meeting	Based on the text "I'm on my way to the meetin...	171.944820
2	/content/data/Crema/1052_TIE_NEU_XX.wav	NEU	[0.0051247957, 0.006485525, 0.005895991, 0.005...	that is exactly what happened	Sentence: 'that is exactly what happened' \nP...	177.734906
3	/content/data/Crema/1064_TSI_SAD_XX.wav	SAD	[-0.0021401502, -0.0027028052, -0.0024305894, ...	the surface is slick	Sentence: 'the surface is slick' \nPitch: 155....	155.575819
4	/content/data/Crema/1004_IWW_ANG_XX.wav	ANG	[0.0036039185, 0.0044390126, 0.0039061378, 0.0...	I wonder what this is about	Sentence: 'I wonder what this is about' \nPit...	205.896918
5	/content/data/Crema/1012_IWL_DIS_XX.wav	DIS	[-0.0006681449, -0.000727192, -0.0004511163, -...	I would like a new alarm clock	Sentence: 'I would like a new alarm clock' \n...	206.896539
6	/content/data/Crema/1058_TIE_ANG_XX.wav	ANG	[0.0017541726, 0.0026575632, 0.0029656827, 0.0...	that is exactly what happened	Sentence: 'that is exactly what happened' \nP...	254.719922
7	/content/data/Crema/1024_IEO_SAD_MD.wav	SAD	[-0.0037392552, -0.004252677, -0.003161763, -0...	it's 11:00	Sentence: 'it's 11:00' \nPitch: 179.553173907...	179.553174
8	/content/data/Crema/1044_ITH_DIS_XX.wav	DIS	[0.0042331917, 0.0055054766, 0.0052368566, 0.0...	I think I have a doctor's appointment	Sentence: 'I think I have a doctor's appointme...	109.208785
9	/content/data/Crema/1001_IOM_FEA_XX.wav	FEA	[0.0074747074, 0.009228159, 0.007966678, 0.006...	I'm on my way to the meeting	Sentence: 'I'm on my way to the meeting' \nPi...	118.863065
10	/content/data/Crema/1066_IEO_SAD_MD.wav	SAD	[0.0010292559, 0.0010486857, 0.0006360504, 0.0...	it's 11:00	Sentence: 'it's 11:00' \nPitch: 111.394825569...	111.394826
11	/content/data/Crema/1035_DFA_NEU_XX.wav	NEU	[0.00060476817, 0.00072895846, 0.0007580796, 0...	don't forget a jacket	Sentence: 'don't forget a jacket' \nPitch: 0....	0.000000
12	/content/data/Crema/1078_IWW_FEA_XX.wav	FEA	[0.0013349339, 0.0017250662, 0.0016882785, 0.0...	I wonder what this is about	Sentence: 'I wonder what this is about' \nPit...	220.318427
13	/content/data/Crema/1004_IWL_DIS_XX.wav	DIS	[0.0037675723, 0.004462086, 0.0037693367, 0.00...	None	Sentence: 'None' \nPitch: 147.88281816869798 ...	147.882818
14	/content/data/Crema/1015_TAI_HAP_XX.wav	HAP	[0.002793698, 0.0035093804, 0.0032107558, 0.00...	None	Sentence: 'None' \nPitch: 141.31674608628806 ...	141.316746
15	/content/data/Crema/1027_DFA_HAP_XX.wav	HAP	[-0.0069763362, -0.008600145, -0.0076393266, -...	don't forget a jacket	Sentence: 'don't forget a jacket' \nPitch: 19...	192.957077
16	/content/data/Crema/1024_IEO_FEA_MD.wav	FEA	[-0.0040978356, -0.0052753594, -0.0050429087, ...	it's 11:00	Sentence: 'it's 11:00' \nPitch: 195.494 Hz → ...	195.494180
17	/content/data/Crema/1058_IEO_ANG_LO.wav	ANG	[0.000820993, 0.0011800317, 0.0012685332, 0.00...	it's 11:00	Sentence: 'it's 11:00' \nPitch: 217.514 Hz → ...	217.514305
18	/content/data/Crema/1036_MTI_NEU_XX.wav	NEU	[-0.00841219, -0.010841992, -0.010119178, -0.0...	maybe tomorrow it will be cold	Sentence: 'maybe tomorrow it will be cold' \n...	131.183644
19	/content/data/Crema/1056_IEO_DIS_LO.wav	DIS	[0.0072696684, 0.00931886, 0.008598879, 0.0078...	it's 11:00	Sentence: 'it's 11:00' \nPitch: 195.453541386...	195.453541
20	/content/data/Crema/1046_DFA_SAD_XX.wav	SAD	[-0.002302306, -0.0030919625, -0.0030472665, -...	don't forget a jacket	Sentence: 'don't forget a jacket' \nPitch: 19...	193.404059
21	/content/data/Crema/1053_TIE_DIS_XX.wav	DIS	[0.0048355693, 0.0058829794, 0.0049770926, 0.0...	that is exactly what happened	Sentence: 'that is exactly what happened' \nP...	188.118263
22	/content/data/Crema/1064_ITH_DIS_XX.wav	DIS	[-0.00090056506, -0.0010059283, -0.0007620044,...	I think I have a doctor's appointment	Sentence: 'I think I have a doctor's appointme...	95.010468
23	/content/data/Crema/1063_IEO_ANG_LO.wav	ANG	[0.0048113856, 0.0057229446, 0.0047238534, 0.0...	it's 11:00	Sentence: "it's 11:00" \nPitch: 382.795853548...	382.795854
24	/content/data/Crema/1045_TSI_HAP_XX.wav	HAP	[-0.0013569427, -0.001743474, -0.0015533864, -...	the surface is slick	Sentence: 'the surface is slick' \nPitch: 249...	249.682607

Figure 2. LLM Sample Output

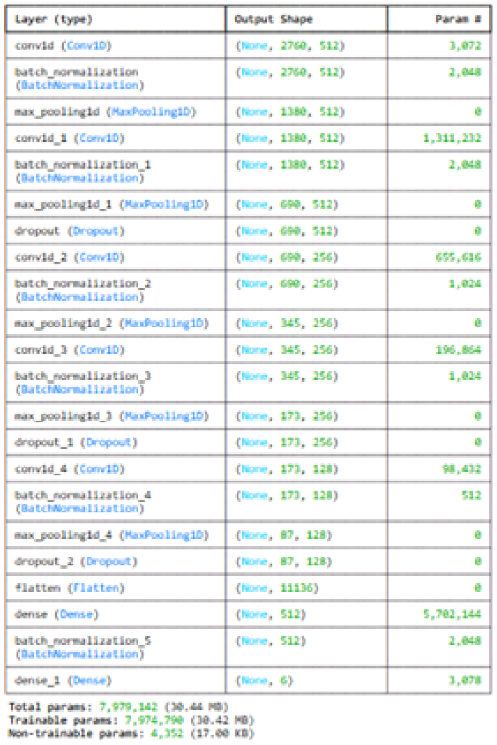
The goal was to utilize the language model's contextual understanding to classify emotions from the semantic content of the spoken utterance. However, the results were suboptimal, with the majority of predictions defaulting to "neutral", indicating limited sensitivity to emotional cues in the transcribed text alone—especially in cases where emotion is conveyed more through tone and prosody than lexical content.

This outcome suggests that text-based inference alone may not be sufficient for accurate emotion recognition, particularly when the emotional expression is subtle or embedded in vocal tone rather than word choice. It highlights the importance of integrating acoustic features (e.g., pitch, MFCCs) alongside textual analysis for a more robust emotion classification system.

3.2. CNN-based Model

1. Audio-only CNN: A convolutional neural network was trained on audio features (e.g., MFCCs) extracted from the raw speech signals (Fig. 1).
2. CNN + Text Embedding: To incorporate textual information, text embeddings were combined with audio features (Fig. 3):
   - Speech was first transcribed using an ASR model
   - Text embeddings were generated using sentence-transformers (all-MiniLM-L6-v2), as the OpenAI text-embedding-ada-002 model (1536 dimensions) proved too resource-intensive
   - Audio features and text embeddings were concatenated horizontally and fed into the model

Pipeline: CNN audio features+ text embedding are stacked horizontally. Data augmentations were added vertically (i.e., treated as additional samples) during training and testing.



Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2760, 512)	3,072
batch_normalization (BatchNormalization)	(None, 2760, 512)	2,048
max_pooling1d (MaxPooling1D)	(None, 1380, 512)	0
conv1d_1 (Conv1D)	(None, 1380, 512)	1,311,232
batch_normalization_1 (BatchNormalization)	(None, 1380, 512)	2,048
max_pooling1d_1 (MaxPooling1D)	(None, 690, 512)	0
dropout (Dropout)	(None, 690, 512)	0
conv1d_2 (Conv1D)	(None, 690, 256)	655,616
batch_normalization_2 (BatchNormalization)	(None, 690, 256)	1,024
max_pooling1d_2 (MaxPooling1D)	(None, 345, 256)	0
conv1d_3 (Conv1D)	(None, 345, 256)	196,864
batch_normalization_3 (BatchNormalization)	(None, 345, 256)	1,024
max_pooling1d_3 (MaxPooling1D)	(None, 173, 256)	0
dropout_1 (Dropout)	(None, 173, 256)	0
conv1d_4 (Conv1D)	(None, 173, 128)	98,432
batch_normalization_4 (BatchNormalization)	(None, 173, 128)	512
max_pooling1d_4 (MaxPooling1D)	(None, 87, 128)	0
dropout_2 (Dropout)	(None, 87, 128)	0
flatten (Flatten)	(None, 11136)	0
dense (Dense)	(None, 512)	5,702,144
batch_normalization_5 (BatchNormalization)	(None, 512)	2,048
dense_1 (Dense)	(None, 6)	3,078

```
Total params: 7,979,142 (30.44 MB)
Trainable params: 7,974,790 (30.42 MB)
Non-trainable params: 4,352 (17.00 KB)
```

Figure 3. CNN + Text Embedding Model Architecture

4. Results

The CNN model was trained using extracted audio features such as MFCCs, pitch, ZCR, and RMSE. After training, the model achieved the following results on test dataset:
Final Training Accuracy: 91.73%
Final Training Loss: 0.3464
Test Accuracy: 92.19%

The CNN+Text Embedding model was trained using extracted audio features such as MFCCs, pitch, ZCR, RMSE and text embeddings. After training, the model achieved the following results on test dataset:
Final Training Accuracy: 91.92%
Final Training Loss: 0.3874
Test Accuracy: 92.47%

These results demonstrate that the models is able to generalize well to unseen data and effectively learn discriminative patterns from the extracted acoustic features. The rela-

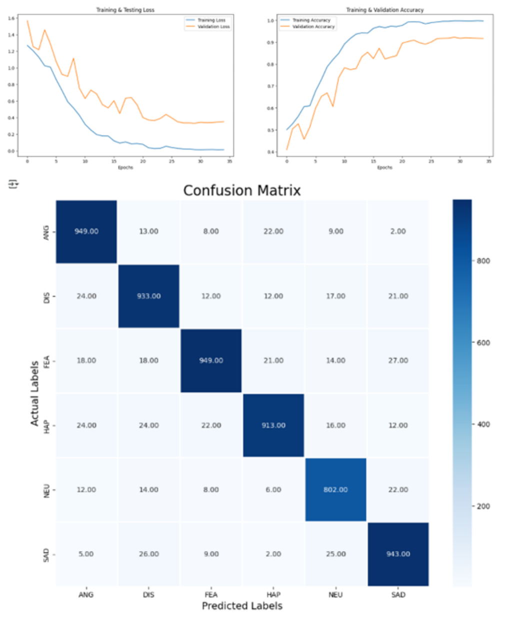


Figure 4. CNN Loss and Accuracy curve

Class	Precision	Recall	F1-Score	Support
ANG (Anger)	0.92	0.95	0.93	1003
DIS (Disgust)	0.91	0.92	0.91	1019
FEA (Fear)	0.94	0.91	0.92	1047
HAP (Happiness)	0.94	0.90	0.92	1011
NEU (Neutral)	0.91	0.93	0.92	864
SAD (Sadness)	0.92	0.93	0.93	1010
Accuracy			**0.92**	5954
Macro Avg	0.92	0.92	0.92	5954
Weighted Avg	0.92	0.92	0.92	5954

Table 1. Classification report showing precision, recall, F1-score, and support for each emotion class for CNN model

tively low loss and high test accuracy indicate both stability during training and strong performance on real-world audio samples.

5. Future Scope

The field of Speech Emotion Recognition (SER) is rapidly evolving, and the current project lays a solid foundation for advanced research and real-world deployment. Potential directions for future work include:

1. Real-time Emotion Recognition
    - Integration with Microphones/Devices: Deploy the model on edge devices (e.g., smartphones, voice assistants) to classify emotions in real-time.
    - Latency Optimization: Optimize model inference time for low-latency applications like virtual therapy, smart classrooms, or gaming.
2. Multi-Modal Emotion Fusion
    - Combine facial expressions, body posture, or text sentiment along with audio and pitch for enhanced emotion detection.
    - Use multi-stream models (e.g., audio + video + text)

Class	Precision	Recall	F1-Score	Support
ANG (Anger)	0.92	0.93	0.93	1005
DIS (Disgust)	0.93	0.91	0.92	1025
FEA (Fear)	0.93	0.92	0.92	1010
HAP (Happiness)	0.91	0.91	0.91	984
NEU (Neutral)	0.93	0.93	0.93	891
SAD (Sadness)	0.93	0.95	0.94	1039
Accuracy			**0.92**	5954
Macro Avg	0.92	0.92	0.92	5954
Weighted Avg	0.92	0.92	0.92	5954

Table 2. Classification report showing precision, recall, F1-score, and support for each emotion class for CNN+Text Embedding model.

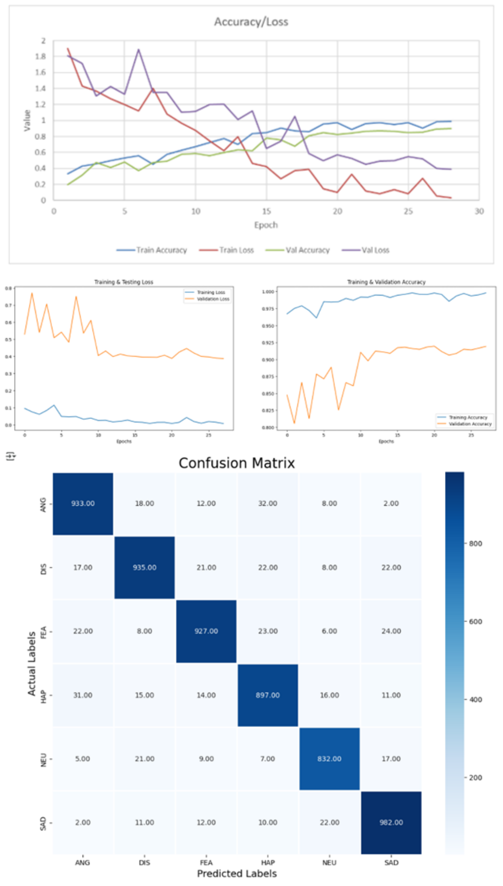


Figure 5. CNN+Text Embedding Loss and Accuracy curve

with attention mechanisms or transformers.

3. Speaker and Language Independence
   - Extend the dataset to include speakers of diverse demographics and various languages to make the system more generalizable and inclusive.
   - Go beyond classification into six emotions and predict emotion intensity (e.g., mild anger vs. extreme rage) using regression models or ordinal classification.
4. Continual and Self-Supervised Learning
   - Enable the model to adapt continuously by learning from new users with minimal retraining.
   - Explore self-supervised audio representation models like Wav2Vec 2.0, HuBERT, or audio transformers.
5. Enhanced Language Model Use
   - Fine-tune LLMs specifically for emotion understanding from speech transcripts.
   - Use LLMs not only for prediction but also for explanation (”why this emotion?”) for interpretable AI.
6. Deployment and APIs
   - Wrap the trained model into an API for use in web/mobile applications.
   - Integrate with platforms like Dialogflow, Alexa, or Zoom for emotion-aware conversation agents.

6. Conclusion

This study presents a practical and effective pipeline for speech emotion recognition using MFCC features and convolutional neural networks. Through systematic preprocessing and model design, the system is able to classify emotional states from speech recordings with promising accuracy. The results support the viability of MFCC-based deep learning approaches in emotion-aware applications. Future work may incorporate additional features such as prosodic cues, apply data augmentation strategies, or expand the framework to support real-time, multilingual, or multimodal emotion recognition systems.

Github Link: https://github.com/rosalindmpaulson/Sppech-Understanding-Course-Project/tree/main