,

Birzeit University

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Multimodal Emotion Recognition in English

Prepared by:

1190145 - Yazan Abdalmutee

1191740 - Taher Hasan

1190186 - Malek AboTouq

Supervised by: Dr. Abualseoud Hanani

A graduation project submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Engineering.

BIRZEIT

ENCS5300

July - 2024

Abstract

A person's emotional response to their surroundings is crucial for expressing what they are experiencing. Emotions like happiness, sadness, anger, and neutrality vary and pose a significant challenge for human-computer interaction. So, an effective emotion classifier is essential. In this project, we we faced the challenge of emotion classification using a multimodal emotion recognition system that incorporates both audio and text to classify emotions into four categories: happy, sad, angry, and neutral. Where in First we collected some datasets for emotions which are RAVDESS dataset for speech, ISEAR and GoEmotions datasets for text. After that we apply preprocessing techniques for both types of data sets like augmentation techniques for speech data, Also for text data we apply several techniques as stemming, tokenization to split the text into individual words . After prepare data we apply feature extraction for it which is the most important step in this project to get the related features for each type of data , and the first step is to extract speech features as there are techniques such as Frequency Melt Coefficients Cepstral (MFCC) and for extract text feature we apply TF-IDF to evaluate word importance. Then many methods have been applied for classification by applying traditional machine learning (ML) techniques such as Random Forest, XGBoost, and various deep learning (DL) models, including convolutional neural networks (CNN), long short-term memory (LSTM), GPT-2, and bidirectional encoder representations from transformers (BERT) were used for classification. After that, Late fusion technique were applied to combine different models to predict the output emotion. The final evaluation was conducted on a dataset gathered from YouTube, with 100 different voices, where the STT model used to converting speech to text using Whisper model. The accuracy of individual models on this dataset was as follows: GPT: 47%, XGBoost: 42%, BERT: 48%, CNN: 55%, LSTM: 49%, and Random Forest: 57% After applying late fusion, we experimented with 49 different combinations, resulting in accuracy ranges from 44% to 71%. The best combination, which achieved the highest accuracy of 71%,was BERT, XGB, CNN and RF,so we concluded that late fusion has increased the accuracy by good difference. Furthermore, a local online web application was implemented to visually demonstrate our work by allowing users to upload files and have ability to choose the model to classify the file to one of the four emotions.

Contents

List of Figures

List of Abbreviation

API	Application Programming Interface
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
DL	Deep Learning
FFT	Fast Fourier Transform
GPT	Generative Pre-trained Transformer
HCI	Human-Computer Interaction
IDF	Inverse Document Frequency
IEMOCAP	Interactive Emotional Dyadic Motion Capture
ISEAR	International Survey on Emotion Antecedents and Reactions
LSTM	Long Short-Term Memory
MER	Multimodal Emotion Recognition
MFCCs	Mel-frequency cepstral coefficients
ML	Machine Learning
NLP	Natural language processing
RAVDESS	Ryerson Audio-Visual Database of Emotional Speech and Song
RF	Random forest
RNN	Recurrent Neural Network
STT	Speech-to-Text
TF	Term Frequency
WER	Word Error Rate
XGB	eXtreme Gradient Boosting

1 Introduction

1.1 Motivation and Overview

Thinking in the speech and the emotions inside it gives a feel that is the speech is the vital thing in communication unlike quick messages with emojies,on the other side machines have difficulties to understand the emotions and other details like gender and personality.

The special thing that the tunes add like inotations and layers of information which is context absent in written exchanges. which is special information on childrens when navigating the complexities in social interactions, so if machines dont handle that problem to help childrens they may have complexity and problems that can cause problems in interactions with others and how to recognize the emotions in the context [1].

To cover this problem and help who need to recognize the emotions on context, its important to develop machines with cabability to know the right feelings in expresions of anger, happy, sad, or neutral. So that will be helpful not only for adults but also for childrens who desperately need help in emotion recognition ,so this kind of systems is critical for improve human computer interactions(HCI).

The enoremous progress in technology gives the machines the cababilities to collect detailed information from sources like microphones, cameras, sound. all collection of this resources increase Human Computer Interaction(HCI) by integrating machines into daily life and enhance interactions that are helps machines to understand the human emotions and how to recognize it [2].

Having all emotional cuse in the speech give machines the cababilites to analyse the emotions can make a revolution in human coumputer interactions(HCI) ,imagine theres a device celebrate with you when your happy and ampathizes when you sad, so these devices can be used in different fields in real life like education, health care, personal assistant and other applications.

1.2 Problem Statement

In shade of huge diversity of voice ,current emotion recognition system struggle in emotion detection and that cause descent in his effectivness in its applications like virtual assistant and customer service and even in helping the childrens to know thier emotions in context of speech. In our project we aim to develop a machine learning model to classify and analyise the emtions in speech and address the chalenge of voice diversity that limit the cabapility of emotion recognition systems to detect the emotions in speech. our system designed for different emotios like happiness, sadness, anger, and neutral. So after doing this the emotion detection system enhace human computer interactions. The applications for the system exists in every field of daily usage aproximatelly as in virtual assistants like Amazon Alexa,Google Assistant, Apple Siri , and in health monitoring to track patient's emotion and this give doctors apility to identifying early signs of mental health issues,and in educational systems like give adapt learning experience based on emotional state of students, and in music and movie recommendations based on user's emotional state,and even in security surviellance to identify stress or anxity in voices potentially prevnting security incidents before they escalate.In summary, all these applications enhance the human computer interactions and this lead to very accurate emotional recognition systems.

1.3 Project Objectives

- Gather a new dataset from different sources, such as YouTube, for testing the model after fusion.

- Apply different preprocessing techniques for both speech and text.

- Apply speech feature extraction techniques like MFCC, Chroma and Mel spectrogram.

- Apply text feature extraction techniques such as, Term Frequency-Inverse Document Frequency (TF-IDF), and word embeddings (e.g. BERT, GPT).

- Implement emotion classification using traditional ML (e.g., Random Forest) and DL models (e.g., CNN, LSTM).

- Apply late fusion approaches for integrating speech and text data.

- Comparing the accuracy of individual models with the accuracy after fusion.

- Build a website as an application of MER systems to detect emotions in speech, and to let the users interact with emotions recognition systems.

2 Related Work

In this chapter, we revised some realated studies in filed of emotional recognition, where some of these studies focused on emotions of users based on signal processing and machine learning techniques ,where other studies focused on deep learning based techniques.we revised three types of these works first the studies which focused on emotion recognition from text such as comments and tweets on social media, second is emotion recognition from speech, and last type of studies which focused on emotion recognition from both of speech and text.

2.1 Related Work to Emotion Recognition from Text

Seal et al. [3] employed a keyword-centric approach for emotion detection, focusing on phrasal verbs and utilizing the ISEAR data set. After preprocessing, they applied the keyword-based method, revealing phrasal verbs not initially associated with emotions. To do this, they constructed a custom database, categorizing recognized phrasal verbs and emotion-related keywords. Despite achieving a commendable 65% accuracy, limitations persisted, including an inadequate emotion keyword list and insufficient consideration for word semantics.

M. Hasan et al. [4] employed a supervised machine learning method along with an emotion dictionary in their proposed model to recognize emotions from text. Their approach involved two sequential tasks: an offline phase and an online phase for emotion classification. Utilizing emotion-labeled text from Twitter and other classifiers, an offline model was developed with a training dataset prepared through data preprocessing. After that, the online approach classified real-time streaming tweets using the model developed in the offline phase. Particularly, their model demonstrated an impressive 90% overall accuracy.

Bharti et al. [5] present a carefully developed hybrid model for text-based emotion recognition that effectively combines machine learning (ML) and deep learning (DL). The model's prowess is demonstrated on different datasets—ISEAR, WASSA, and Emotion-Stimulus— encompassing various text types, including normal sentences, tweets, and dialogs. Leveraging Support Vector Machine (SVM) within the ML realm, the model achieves an accuracy of 78.97%. Within the DL components, Convolutional Neural Network (CNN) and Bidirectional Gated Recurrent Unit (Bi-GRU) are deployed, with Bi-GRU attaining the highest accuracy at

79.46%. This advanced fusion of SVM, CNN, and Bi-GRU underscores a nuanced understanding of both ML and DL techniques, leading to overall accuracy of 80.11%.

2.2 Related Work to Emotion Recognition from Speech

Yang Li and Yunxin Zhao [6] aimed to identify the emotional status of individual speakers through speech features. They collected an emotional speech corpus from 5 speakers expressing 6 different emotions. Short-term features, such as formants, formant bandwidths, pitch, log energy, and normalized autocorrelation coefficient, were extracted using a shifting short-time window. Long-term features, including mean and standard deviation of pitch and log energy, were derived from the entire utterance. Principal Component Analysis (PCA) was applied for dimension reduction and feature selection. The authors employed three classification methods: vector quantization (VQ), artificial neural networks (ANN), and Gaussian mixture density (GMD) models. The best recognition performance of 62% accuracy was achieved by using the GMD method with both short-term and long-term features. The study revealed that both feature types contributed to emotion classification, with GMD providing the optimal results.

Lin et al. [7] employed Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) to classify emotions, specifically tackling issues related to information loss. In the early 21st century, Hidden Markov Models were utilized for emotion classification, but they were limited by their capability to handle low-dimensional data. Demonstrated the application of HMMs in predicting gender-independent emotions, reaching an impressive accuracy rate of 88.9% on a Danish emotional speech dataset.

Kandali et al. [8] utilized a method based on a Gaussian mixture model (GMM) classifier and Mel-frequency cepstral coefficients (MFCC) as features for emotion recognition from Assamese speeches. The experiments were conducted in Jorhat, Assam, India, with data collected from 27 speakers (14 Male and 13 Female) who performed acted speeches of emotionally biased sentences. Two cases were considered: (i) text-independent but speaker-dependent and (ii) text-independent and speaker-independent. The authors reported success scores using different configurations of GMM parameters and MFCC features. In experiment (i), the highest mean classification scores ranged from 66.9% to 76.5%. In experiment (ii), the speaker-independent scenario, the average mean success scores were generally lower than those in (i). The authors concluded that the surprise emotion was the most challenging to distinguish from other emotions.

Shen et al. [9] explored three emotional states (happy, sad, neutral) using various features including energy, pitch, linear predictive spectrum coding (LPCC), mel-frequency spectrum coefficients (MFCC), and mel-energy spectrum dynamic coefficients (MEDC). They employed a Support Vector Machine (SVM) classifier trained on a German Corpus (Berlin Database of Emotional Speech) and a self-built Chinese emotional database. The results showed that the combination of MFCC+MEDC+Energy achieved the highest accuracy rates on both the Chinese emotional database (91.3%) and the Berlin emotional database (95.1%). The study shows the importance of changing features for different languages and corpora. The authors also highlighted the potential applications of SER in areas such as in-car systems, call centers, and e-learning for timely detection of emotions and improving teaching quality.

Aljuhani et al. [10] developed an emotion recognition system for the Saudi dialect by creating a dataset from YouTube videos with labeled emotions (anger, happiness, sadness, neutral). They applied spectral features, including MFCC and mel spectrogram, and utilized classifiers such as SVM, MLP, and KNN. Results showed SVM achieved the highest accuracy (77.14%), especially in recognizing anger and neutral emotions. Challenges included the limited dataset size, but the study provides valuable insights into Arabic speech emotion recognition, emphasizing the scarcity of such datasets suggesting future work on dataset extension, and exploring deep learning models for improved emotion classification.

Pandey et al. [11] explore architectures such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) using standard speech representations like mel spectrogram, magnitude spectrogram, and Mel-Frequency Cepstral Coefficients (MFCCs) on two datasets, EMO-DB and IEMOCAP. They conduct experiments to determine the effectiveness of different feature-architecture combinations for speech-emotion recognition. The study highlights the significance of deep learning in surpassing traditional methods, such as handcrafted features and classifiers like GMM, HMM, and SVM, especially in capturing complex phenomena like emotional states. The authors present experimental results, including confusion matrices, and conclude that a CNN+LSTM architecture with log-mel spectrograms performs well on the tested datasets. They emphasize the importance of exploring different features and deep learning architectures for effective speech-emotion recognition. These accuracy values using CNN+BLSTM architecture with MFCC features achieved the highest accuracy on the EMO-DB dataset (82.35%), while the CNN+BLSTM architecture with Mel-Spectrogram features performed best on the IEMOCAP dataset (50.05%).

Zhao et al. [12] demonstrated the effectiveness of the designed networks, with the 2D CNN LSTM network outperforming traditional methods such as Deep Belief Network (DBN) and CNN. The 2D CNN LSTM network achieved high recognition accuracies, notably 95.33% and 95.89% on the Berlin EmoDB in speaker-dependent and speaker-independent experiments, respectively. While the experiments indicated effective recognition of emotional states, Zhao et al also acknowledged challenges, such as the ”black box” nature of deep networks. Efforts were made to address issues like overfitting through regularization, batch normalization, cross-validation, and early stopping. Despite improvements, the study suggests areas for further research, including explaining the detailed workings of the networks and exploring new architectures or optimization algorithms for enhanced performance in speech emotion recognition.

2.3 Related Work to Emotion Recognition from Both Text & Speech

Ma et al. [13] proposed a method to boost speech emotion recognition (SER) with the state-of-the-art speech pre-trained model (PTM), data2vec, text generation technique, GPT-4, and speech synthesis technique, Azure TTS. They investigated the representation ability of different speech self-supervised pre-trained models and found that data2vec has a good representation ability on the SER task. They employed a powerful large language model (LLM), GPT-4, and emotional text-to-speech (TTS) model, Azure TTS, to generate emotionally congruent text and speech. They carefully designed the text prompt and dataset construction to obtain the synthetic emotional speech data with high quality. They studied different ways of data augmentation to promote the SER task with synthetic speech, including random mixing, adversarial training, transfer learning, and curriculum learning. Experiments and ablation studies on the IEMOCAP dataset demonstrate the effectiveness of their method, compared with other data augmentation methods, and data augmentation with other synthetic data.

Yoon et al. [14] proposed a novel deep dual recurrent encoder model that utilizes text data and audio signals simultaneously to obtain a better understanding of speech data. The model encodes the information from audio and text sequences using dual recurrent neural networks (RNNs) and then combines the information from these sources to predict the emotion class. This architecture analyzes speech data from the signal level to the language level, and it thus utilizes the information within the data more comprehensively than models that focus on audio features. The proposed model outperforms previous state-of-the-art methods in assigning data to one of four emotion categories (i.e., angry, happy, sad, and neutral) when the model is applied to the IEMOCAP dataset, as reflected by accuracies ranging from 68.8% to 71.8%.

Sailunaz et al. [15] conducted a survey on emotion detection from text and speech. The authors reviewed research efforts analyzing emotions based on text and speech, including emotion models, emotion datasets, emotion detection techniques, their features, limitations, and some possible future directions. They focused on reviewing research efforts analyzing emotions based on text and speech and investigated different feature sets that have been used in existing methodologies. The authors summarized basic achievements in the field and highlighted possible extensions for better outcomes. The survey covers existing emotion detection research efforts and provides a comprehensive overview of the field.

Mikel deVelasco et al. [16] proposed a method for automatic emotion detection from a speech by using both acoustic and textual information 1. The authors extracted a set of audio from a TV show where different guests discussed topics of current interest. The selected audios were transcribed and annotated in terms of emotional status using a crowdsourcing platform. A 3-dimensional model was used to define a specific emotional status to pick up the nuances in what the speaker is expressing instead of being restricted to a predefined set of discrete categories. Different sets of acoustic parameters were considered to obtain the input vectors for a neural network. To represent each sequence of words, a model based on word embeddings was used. Different deep learning architectures were tested providing promising results, although having a corpus of a limited size.

3 Methodology

Figure 3.1 illustrates the model's architecture, featuring two distinct pathways for processing a given speech signal. The first involves directly extracting audio features for speech encoding, while the second employs an ASR system to generate text, which is then converted to word-based embeddings (high-dimensional vectors representing words/sentences) for text encoding. Therefore, the entire model consists of a speech encoder, a text encoder, and a multimodal fusion, And we will use several models for classification to compare them. Detailed explanations for each component are provided in this section.

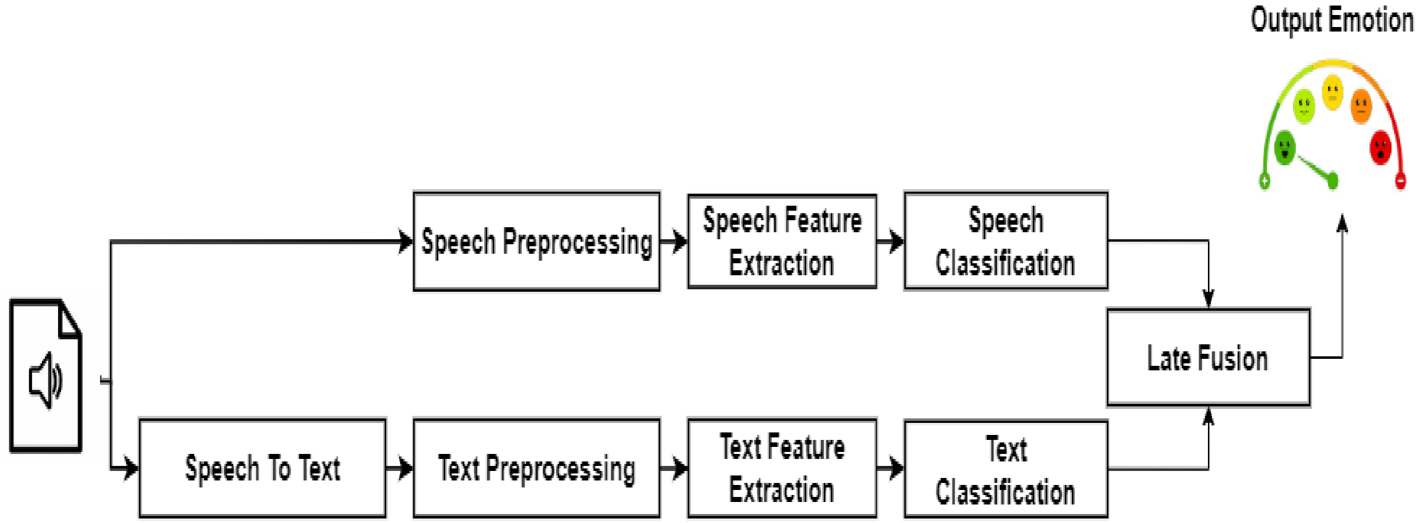


Figure 3.1: Emotion recognition block diagram

3.1 Dataset

The dataset is very important part of any machine learning project, were it used as a lesson to make the machine practice to know how it should work and deal with similar data.

So for our project we decide to use different datasets for both of speech and text, we use one dataset for speech features and other two datasets for text features,also we gather a new dataset for testing , here the details about each type of datastes.

3.1.1 Speech Dataset

RAVDESS dataset

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)[17] is a valuable resource for emotion recognition research. This dataset created by 24 professional actors (12 male, 12 female) who vocalize two lexically-matched statements in a neutral North American accent, expressing eight emotions: calm, happy, sad, angry, fearful, surprise, disgust, and neutral, where we just use four of these emotions in our project which are happy, sad, angry, neutral,with total of 672 files for used emotions. The dataset includes audio recordings, facilitating multimodal emotion recognition research. With 7,356 files, including 1,440 audio files (speech and song) , this dataset considered is a rich resource for training and evaluating emotion recognition models.

Figure 3.2 shows the distribution of the emotions we used.

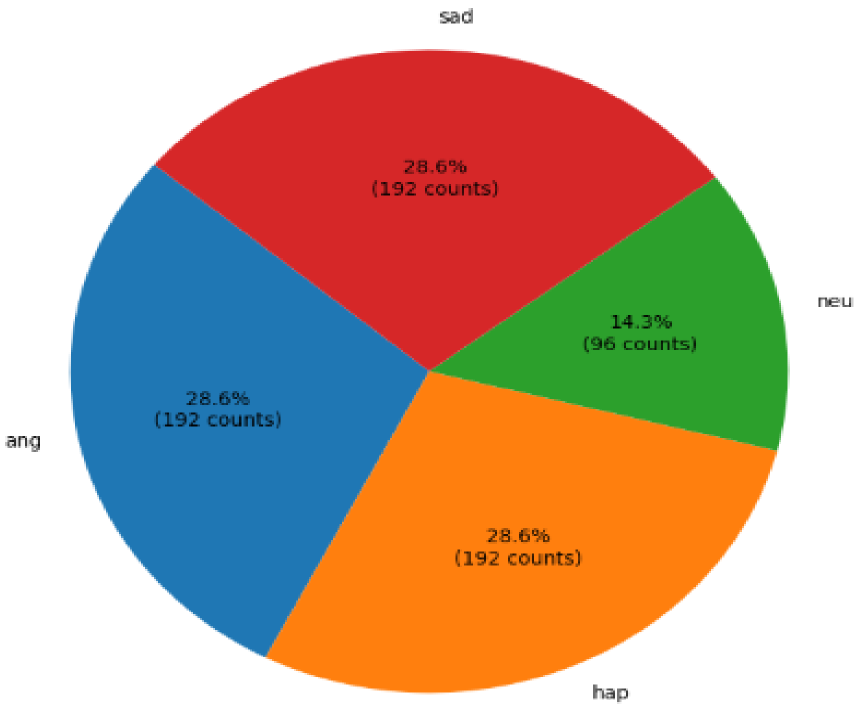


Figure 3.2: The label distribution of RAVDESS dataset.

3.1.2 Text Datasets

ISEAR dataset

The International Survey on Emotion Antecedents and Reactions (ISEAR)[18] dataset provides a psychological perspective on emotions. Compiled from self-reports of emotional experiences by of individuals from diverse cultural backgrounds, it covers seven emotions: joy, fear, anger, sadness, disgust, shame, and guilt, we will use 4 of them; Anger, Fear, Joy and sadness, with a total of 4365 files for used emotions. This survey-based dataset contains detailed descriptions of situations that elicited each emotion, making it a rich resource for studies on emotional experiences.

Figure 3.3 shows the distribution of the emotions we used from ISEAR.

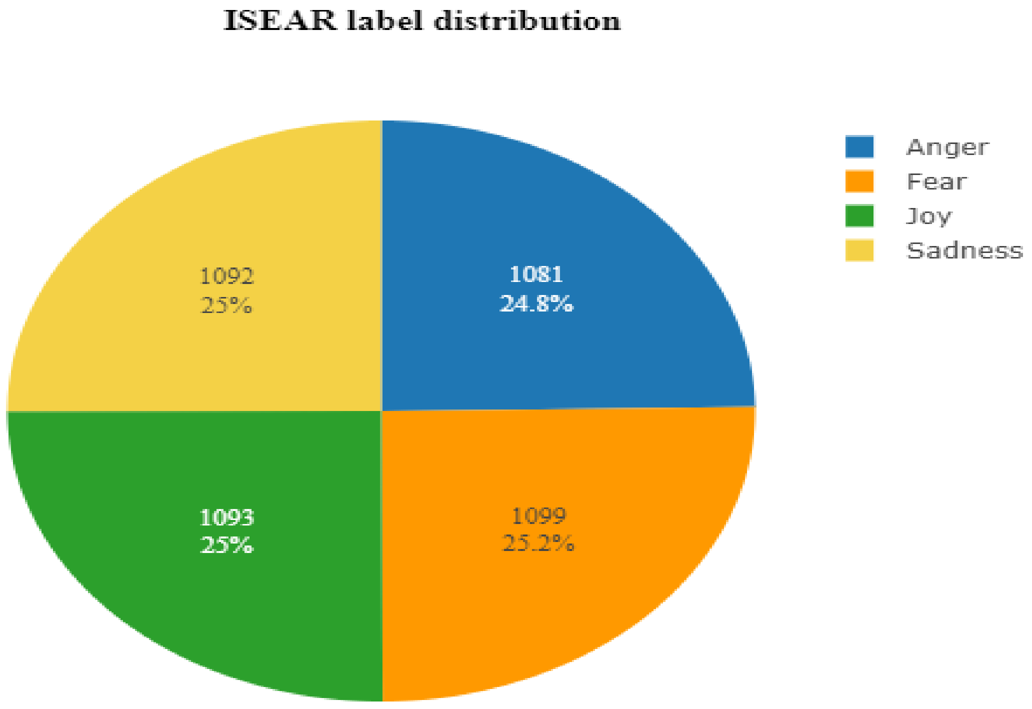


Figure 3.3: The label distribution of ISEAR dataset.

GoEmotions dataset

GoEmotions[19], developed by Google, is a dataset designed for fine-grained emotion classification. It includes 58,000 carefully curated Reddit comments, annotated for 27 emotion categories or neutral categories. we will use 4 of them; Angry, Happy, Sad, and Neutral, with a total of 4365 files for used emotions. The annotations, performed by human raters with high inter-annotator agreement, cover a wide range of topics and contexts from social media comments, providing a real-world usage of language. This dataset is ideal for enhancing sentiment analysis models and developing emotionally responsive AI systems.

Figure 3.4 shows the distribution of the emotions we used from GoEmotions.

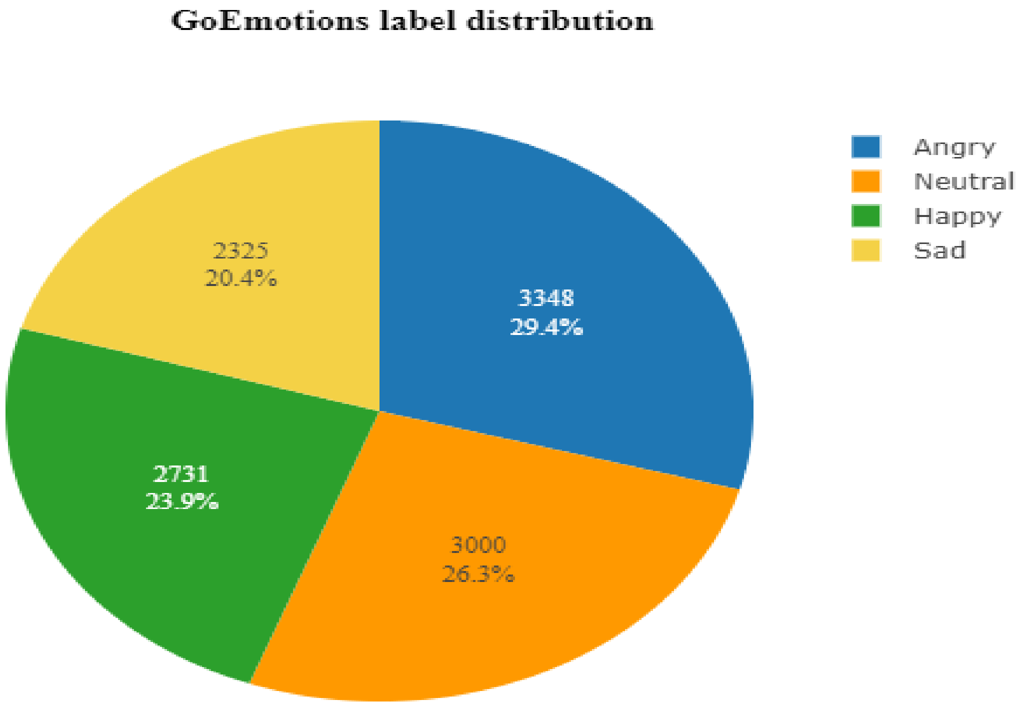


Figure 3.4: The label distribution of GoEmotions dataset.

3.1.3 Testing Dataset

To test our proposed system, we decided to collect a testing dataset containing 100 audio files from YouTube which used for testing the fusion model. This decision was made due to the limited availability and poor quality of open-source datasets.
Figure 3.5 shows the distribution of the testing dataset.

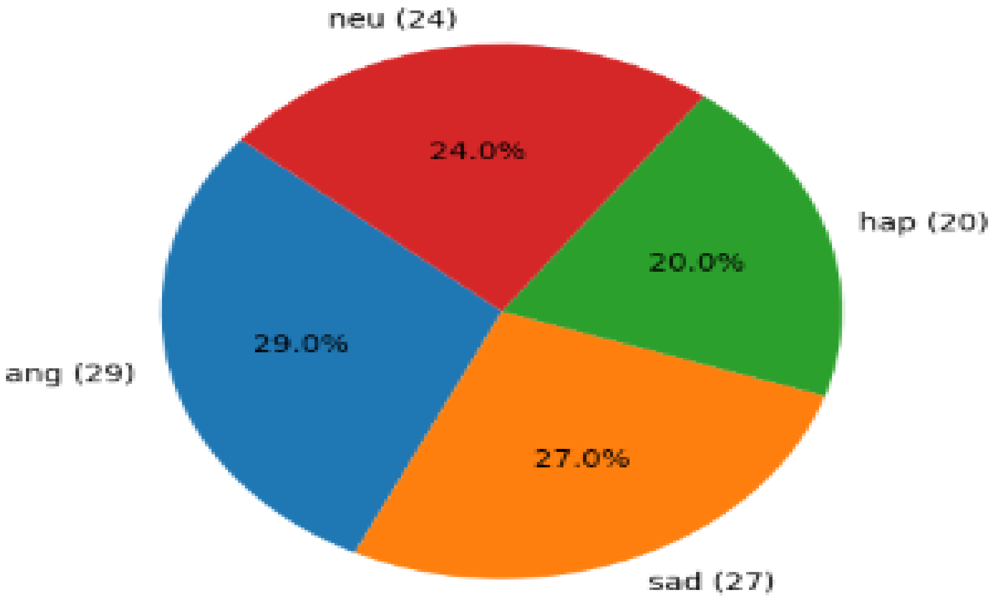


Figure 3.5: The label distribution of MER dataset

3.2 Data Preprocessing

Effective data preprocessing is a crucial step in ensuring the success of any machine learning project, as it prepares the raw data for model training and helps improve the model's performance. For our Multimodal Emotion Recognition (MER) project, we apply preprocessing techniques for audio and text data to ensure each modality is optimized for analysis.

3.2.1 Audio Data Preprocessing

For audio data, we apply several preprocessing techniques, including augmentation techniques to enhance the diversity and robustness of the training dataset, and audio trimming for the entire dataset.

1. Augmentation Techniques:

- **Shift Waveform Data:** This technique involves shifting the audio signals slightly to the left or right, which helps the model become invariant to small temporal variations.

  The figure 3.6 shows the affect of shifting at the waveform.

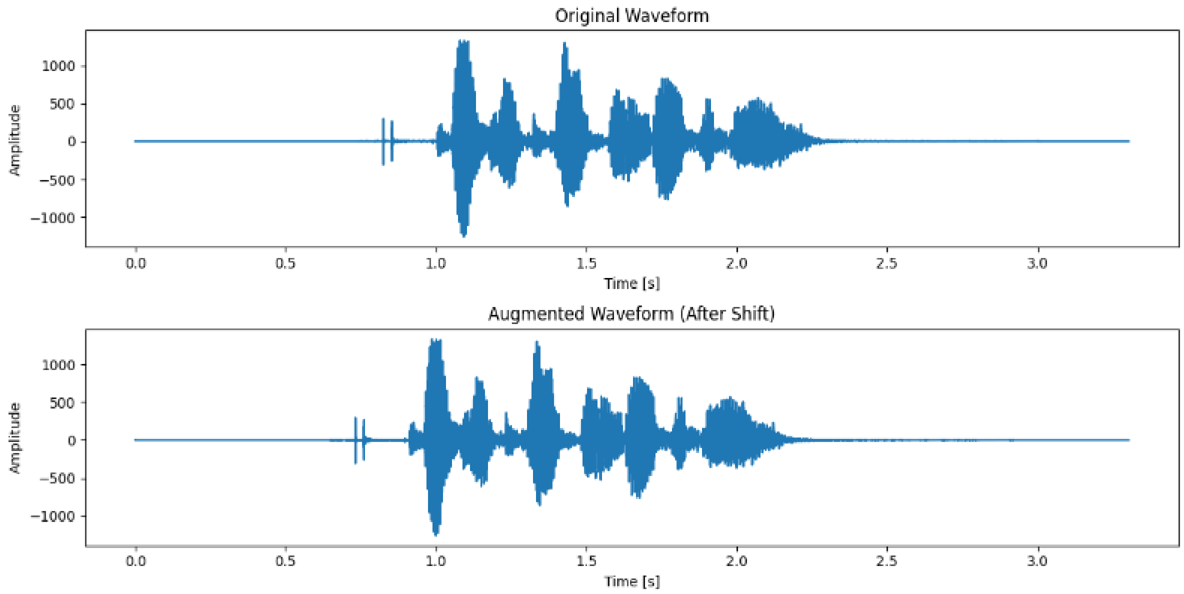


Figure 3.6: Waveform before and after shifting

- **Add Random Noise:** Introducing random noise into the audio signals helps the model learn to focus on the essential features of the audio, making it more robust to real-world noisy conditions.

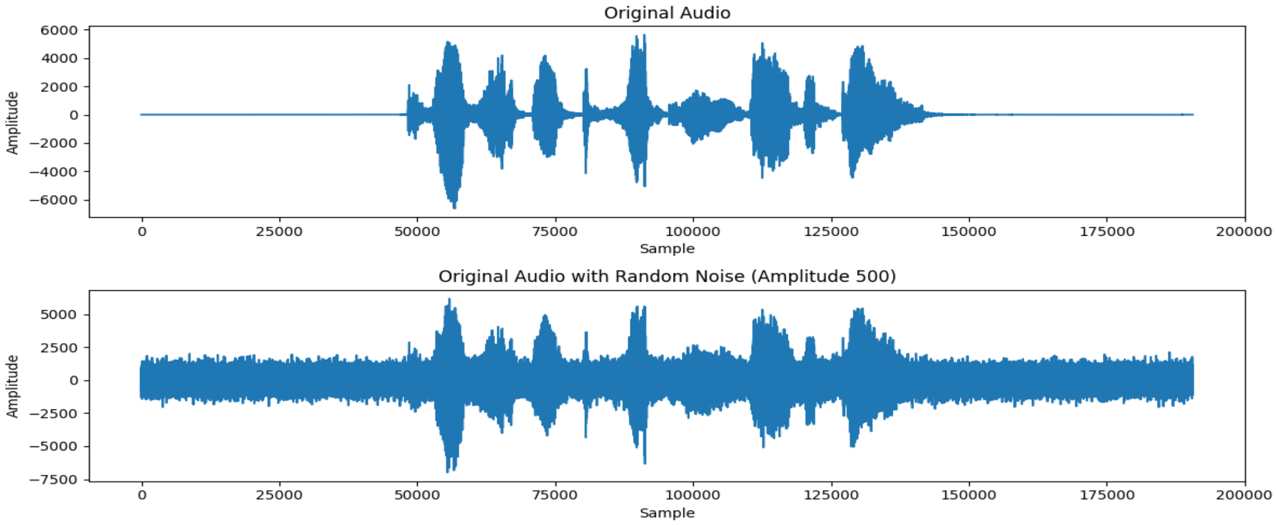


Figure 3.7: Waveform after adding the noise

2. Audio Trimming : Trimming the audio clips to remove silence from the start and end ensures uniformity in the dataset, which simplifies the model training process.

Figures 3.8 illustrate the effects of trimming audio from silence on the audio ,

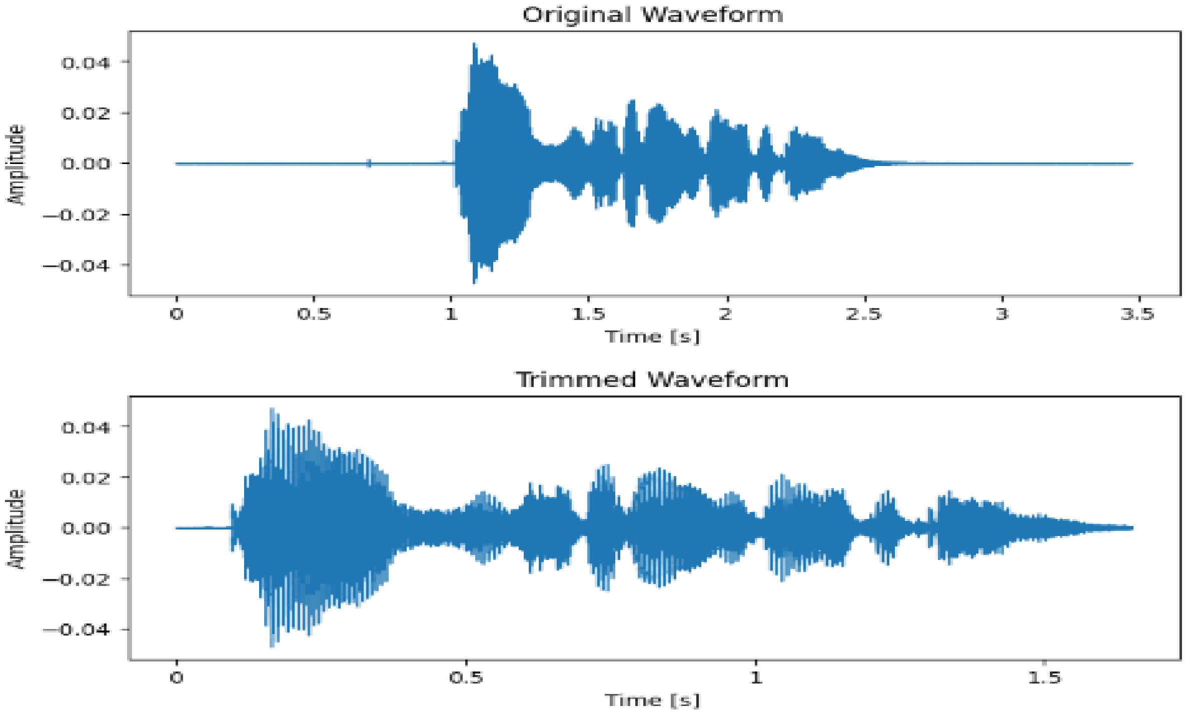


Figure 3.8: Waveform before and after the trimming

3.2.2 Text Data Preprocessing

For text classfication, Effective text data preprocessing is essential to ensure the quality and performance of the model. Text preprocessing involves various techniques to clean and prepare the raw text data for analysis, making it more suitable for model training. By applying these techniques, we can enhance the accuracy of our emotion recognition system.

Preprocessing Steps:

- **Removing Stop Words:** Stop words are common words that do not add significant meaning to the content, such as ”and,” ”the,” and ”is.” Removing them helps focus on the more meaningful words in the text.

Example for remove stop words:

Orginal text: ”This is an example for stop word removal”

Preprocessed text: ”This example stop word removal”

- **Stemming:** Applying stemming reduces words to their root forms. This technique ensures that variations of a word, such as ”running,” ”runner,” and ”ran,” are treated as a single entity, reducing redundancy and improving the model’s efficiency.

  Example for stemming:

  Original text: ”The runners were running in the race.”

  Preprocessed text: ”The runner were run in the race.”

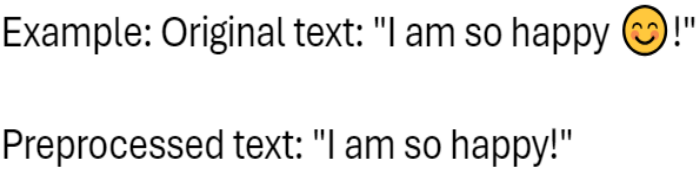
- **Converting Numbers to Text:** Converting numbers into their corresponding text format, such as changing ”1” to ”one,” helps to improve the model’s understanding.

  Example: Original text: ”She has 2 cats and 1 dog.”

  Preprocessed text: ”She has two cats and one dog.”

- **Removing Emojis:** Emojis, while they can convey emotions, are often inconsistent and can add noise to the data. Removing emojis ensures that the text remains clean and focused.

Figure 3.9 shows the affect of removing emoji at the text.



Example: Original text: "I am so happy 😊!"

Preprocessed text: "I am so happy!"

Figure 3.9: Removing emoji effect

- **Removing Links:** Removing links present in the text is crucial as links often contain irrelevant information that can distract from the primary content.

  Example:

  Original text: ”Check out our website at http://Linkexample.com for more info.”

  Preprocessed text: ”Check out our website for more info.”

- **Removing Punctuation:** Removing punctuation marks is important as they can introduce variability and noise into the data. Eliminating punctuation simplifies the text.

  Example:

  Original text: ”Hi there! How are you doing today?”

  Preprocessed text: ”Hi there How are you doing today”

- **Tokenization:** Tokenizing the text by breaking it down into individual words or tokens facilitates easier processing and analysis by the model. Tokenization converts the text into a structured format that the model can work with.

  Example:

  Original text: ”Let’s meet at the café at 5pm.”

  Preprocessed text: [”Let”, ”us”, ”meet”, ”at”, ”the”, ”café”, ”at”, ”5”, ”pm”]

3.3 Speech-to-Text (STT)

Speech-to-Text (STT) technology, also known as Automatic Speech Recognition (ASR), converts spoken language into written text. It leverages machine learning and natural language processing (NLP) techniques to recognize and transcribe audio data. STT models are crucial in various applications, including voice assistants, transcription services, language translation, and voice-controlled interfaces. Key features of STT models include accuracy, speed, language support, and additional functionalities such as real-time transcription and speaker identification.

When choosing an STT model, it’s essential to compare key factors such as accuracy, speed, features, and language support.

3.3.1 Comparison of STT Models[1]

- **OpenAI Whisper:** Achieves a median Word Error Rate (WER) of 8.06% and transcribes an hour of audio in 10-30 minutes, with various model sizes for flexibility.
- **Google Speech-to-Text:** WER ranges from 16.51% to 20.63%, supports over 125 languages, leveraging advanced AI capabilities.
- **Amazon Transcribe:** WER between 18.42% and 22%, features include real-time transcription, speaker identification, noise reduction, and sentiment analysis.

3.3.2 Why We Choose Whisper?

OpenAI Whisper offers exceptional accuracy and speed, wide language support, and easy integration despite fewer audio intelligence features.

3.3.3 Whisper Details:

OpenAI Whisper is a neural network model trained on 680,000 hours of multilingual audio, supporting 98 languages. It offers various model sizes, custom vocabulary support, and is available as both an open-source model and an API. With a median WER of 8.06%, Whisper is highly accurate and suitable for our project to get fast and accurate transcription.

3.4 Feature Extraction

Feature extraction in Multimodal Emotion Recognition (MER) systems involves extract meaningful representations from both speech and text data to effectively capture emotional cues. For speech data, techniques like Mel-Frequency Cepstral Coefficients (MFCC), Chroma features, and Mel spectrograms are employed to extract critical aspects of the audio signal. For text data, methods such as Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings (e.g., BERT, GPT). These extracted features from speech and text are crucial for building MER models, enabling accurate emotion classification through traditional machine learning and deep learning approaches.

[1] https://www.gladia.io/blog/openai-whisper-vs-google-speech-to-text-vs-amazon-transcribe

3.4.1 Speech Feature Extraction:

For any recognition system depends on the effect of feature extraction, the accuracy and performance have a very important effect on the results. Key acoustic properties, such as Mel Frequency Cepstral Coefcients (MFCCs)[20] ,chroma, mel spectrogram, serve as key elements for analyzing the nuances and accuracy of a speech signal.Chroma features, which represent the energy distribution of musical notes across the chromatic scale, are valuable for tasks involving audio analysis and classification. They capture the tonal content and harmonic structure of audio signals, providing important information for distinguishing between different segments. On the other hand, mel spectrograms, which utilize the Mel scale to represent frequency content, offer a detailed view of the spectral characteristics of an audio signal.On the other hand, MFCCs, well-known for their ability to capture the spectral characteristics of an audio signal, are useful in recognizing acoustic characteristics and are widely used in voice recognition and speaker identification systems. By extracting and analyzing these features we can significantly enhance the system's ability to interpret, classify and understand complex audio signals, leading to improvements in accuracy and performance across a wide range of applications and even sentiment analysis in speech.

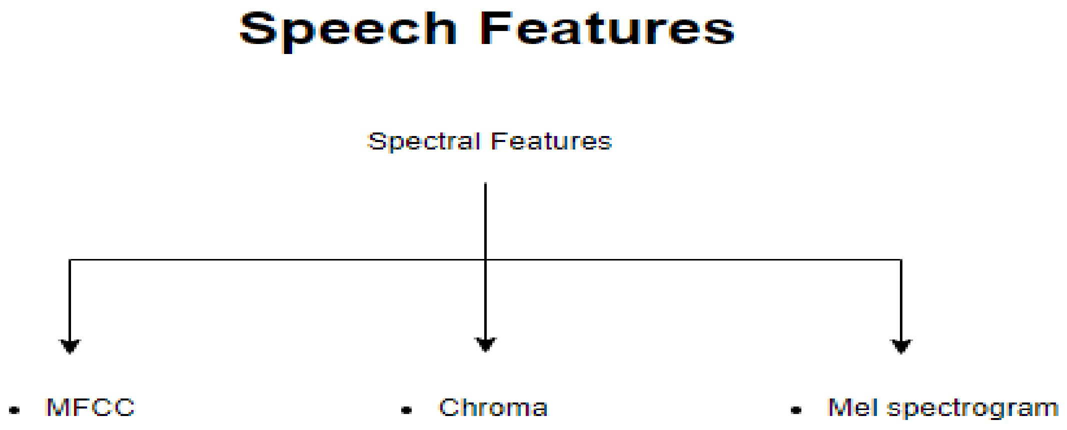


Figure 3.10: Categorization of speech features

Spectral Features:

Mel Frequency Cepstral Coefcients(MFCCs):

Mfcc is a feature extraction method from audio files,it designed to be like human auditory perception, and it has steps simulate how human ear processes sound.for example the Mel scale transformation used in computing MFCCs is based on the observation that the human ear's sensitivity to changes in pitch is nonlinear, with more sensitivity at lower frequencies compared to higher frequencies.This nonlinearity is captured in the Mel scale, which is used to map frequencies to a scale that better reflects human perception.so in general its a powerful technique to emulate characteristics of human auditory, that make it very suitable for various machine learning tasks, such as speech recognition and music analysis.

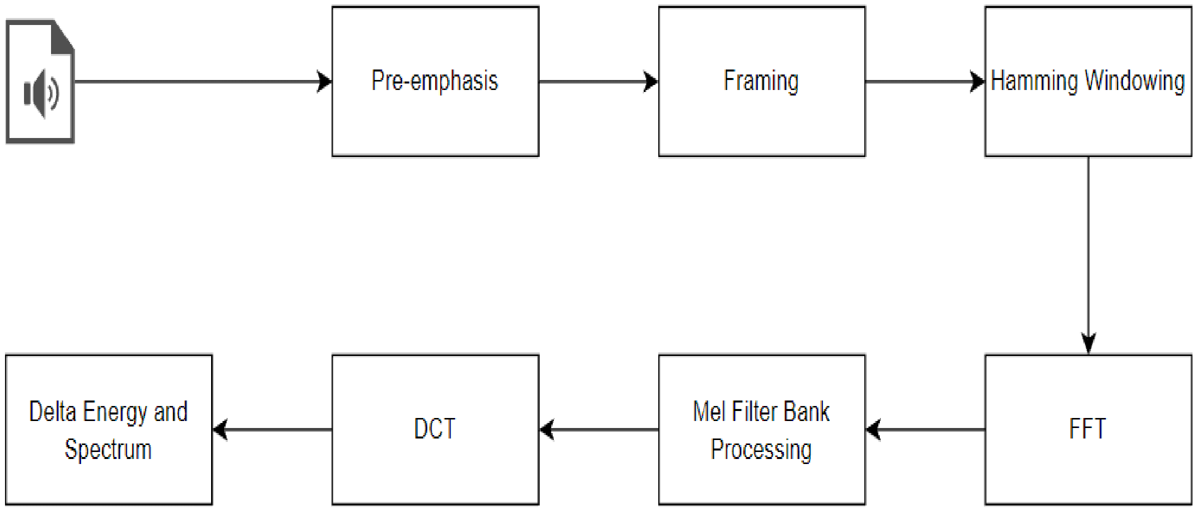


Figure 3.11: Mfccs block diagram

As shown in figure 3.11, Mel Frequency Cepstral Coefficients (MFCCs) represent the short-term power spectrum of a sound signal. They are derived by applying a linear cosine transform to overlapping frames of the signal. The frames undergo a Fast Fourier Transform (FFT), and the resulting spectrum is then transformed using a non-linear Mel-frequency scale. This process involves breaking down the signal into overlapping frames, applying FFT, and converting to the Mel-scale. The final step involves generating cepstral coefficients, providing a representation of the sound signal's spectral characteristics.

Chroma: The chroma feature is a representation of musical audio, designed to capture both melodic and harmonic content while accommodating variations in tempo and tuning. By aligning audio against a time base defined by detected beats, variations in tempo are normalized, ensuring consistency across different renditions of the same piece. Chroma features focus on the distribution of energy across the 12 semitones of the musical scale, providing insights into both the melody and harmonic accompaniment. Techniques like phase-derivative analysis within FFT bins enhance frequency resolution, while adjustments to frequency mapping mitigate tuning discrepancies. Overall, the chroma feature offers a comprehensive yet compact representation of musical audio, facilitating tasks such as analysis, classification.

Mel spectrogram: Mel spectrogram is a visual representation of the frequency content of an audio signal over time, with a particular focus on how humans perceive sound. It's derived from the traditional spectrogram, which displays the intensity of different frequencies in an audio signal over time.

The "mel" in mel spectrogram refers to the Mel scale, which is a perceptual scale of pitches that approximates the human ear's response to different frequencies. Human hearing is more sensitive to changes in pitch at lower frequencies than at higher frequencies. The Mel scale is designed to reflect this non-linear relationship.

So, instead of representing frequency in Hertz directly, the mel spectrogram converts frequencies into the Mel scale. This conversion is done using a mathematical formula known as the Mel scale transformation, which essentially warps the frequency axis such that it corresponds more closely to how humans perceive pitch.

In summary, the mel spectrogram provides a more accurate representation of the frequency content of an audio signal as perceived by humans, making it useful in various applications such as speech recognition and sound processing.

3.4.2 Text Feature Extraction:

When it comes to capturing the meaning of words within a document, the Term Frequency-Inverse Document Frequency (TF-IDF) method stands out. TF-IDF is a statistical measure that evaluates the importance of a word in a document relative to a collection of documents, known as a corpus. It combines two components: Term Frequency (TF), which measures how often a term (word) appears in a document, and Inverse Document Frequency (IDF), which measures the rarity of a term across the entire corpus. Rare terms that appear in few documents receive higher IDF scores[21].

In addition to traditional methods like TF-IDF, modern deep learning models have transformed text feature extraction. BERT, or Bidirectional Encoder Representations from Transformers, is one such model. Unlike traditional models, BERT considers the context of each word in both directions, significantly improving its ability to understand the meaning of words within sentences[22].

Another advanced model is GPT, or Generative Pre-trained Transformer. GPT is designed for generating human-like text and succeeds in various natural language processing tasks. It uses a transformer architecture to predict the next word in a sentence, based on the context provided by the preceding words. This capability allows GPT to generate coherent and contextually relevant text, making it a powerful tool for language understanding and generation[23].

3.5 Fusion

Feature fusion is a pivotal step in Multimodal Emotion Recognition (MER), where information from different modalities, such as speech and text, is integrated to enhance overall performance. There are two types of fusion:

Early Fusion: Combines features at the input level through concatenation, enabling early correlations and simplicity.

Late Fusion: Combines predictions from independently processed modalities, offering flexibility but assuming modality independence.

The figure 3.12 shows the flowchart for early and late fusion

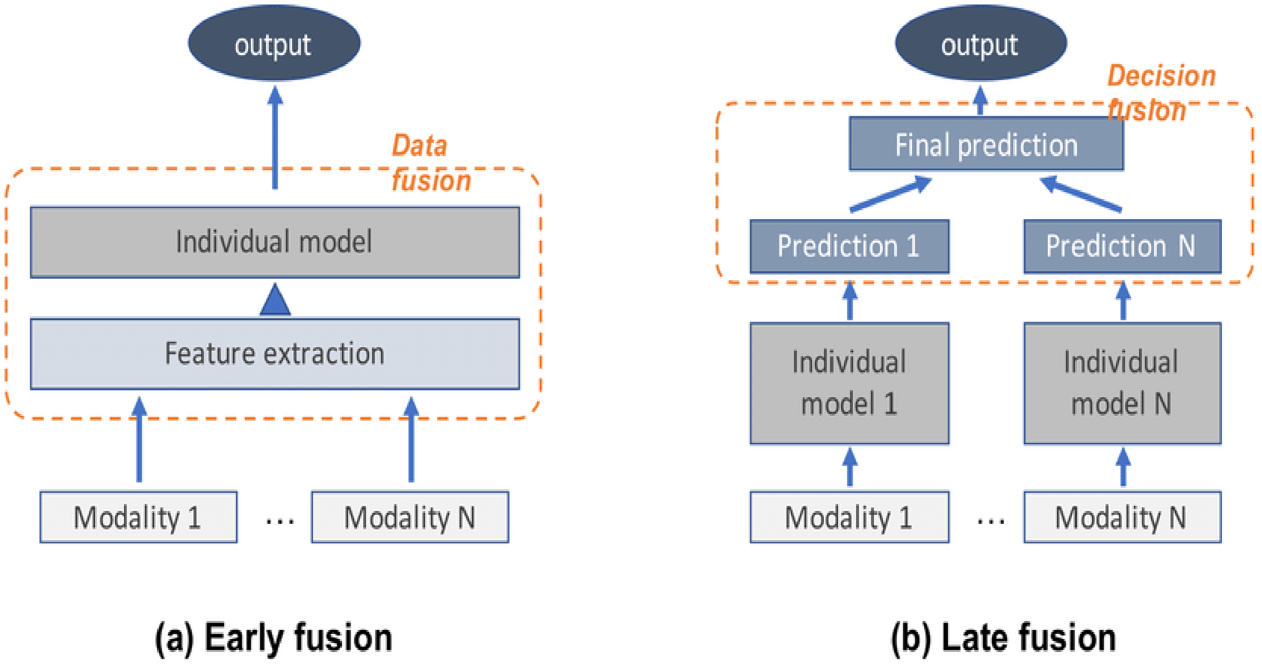


Figure 3.12: Early and late fusion flowchart [24]

In our project, we have focused on using late fusion because it allows each modality to be processed independently, maximizing the strengths of each before combining their outputs for a more robust and flexible final prediction. Additionally, the reason to use late fusion is the limiting size of the multimodal dataset we have gathered.

3.6 Emotion Classification

In the domain of emotion categorization, our approach is informed by a thorough review of existing literature and research papers, which has led us to select two prominent methodologies. Following an extensive analysis of various techniques, we have opted for a combination of traditional machine learning approaches, specifically the Random Forest for precise emotion classification. Additionally, we have integrated deep learning techniques for both speech and text, such as CNN, BERT, etc. This selection is based on the comprehensive evaluation of studies and papers, which also lead us to compare our results rigorously with findings from other relevant studies in the field.

3.6.1 Traditional Machine Learning Techniques

Random Forest: Random Forest is an ensemble learning system that blends multiple decision trees to generate predictions. Each tree in the forest is trained on a subset of the data, and the collective predictions of all individual trees contribute to the final prediction. The architecture of Random Forest, as depicted in Fig 3.13, is renowned for its ability to resist overfitting and handle large-scale, multidimensional data.

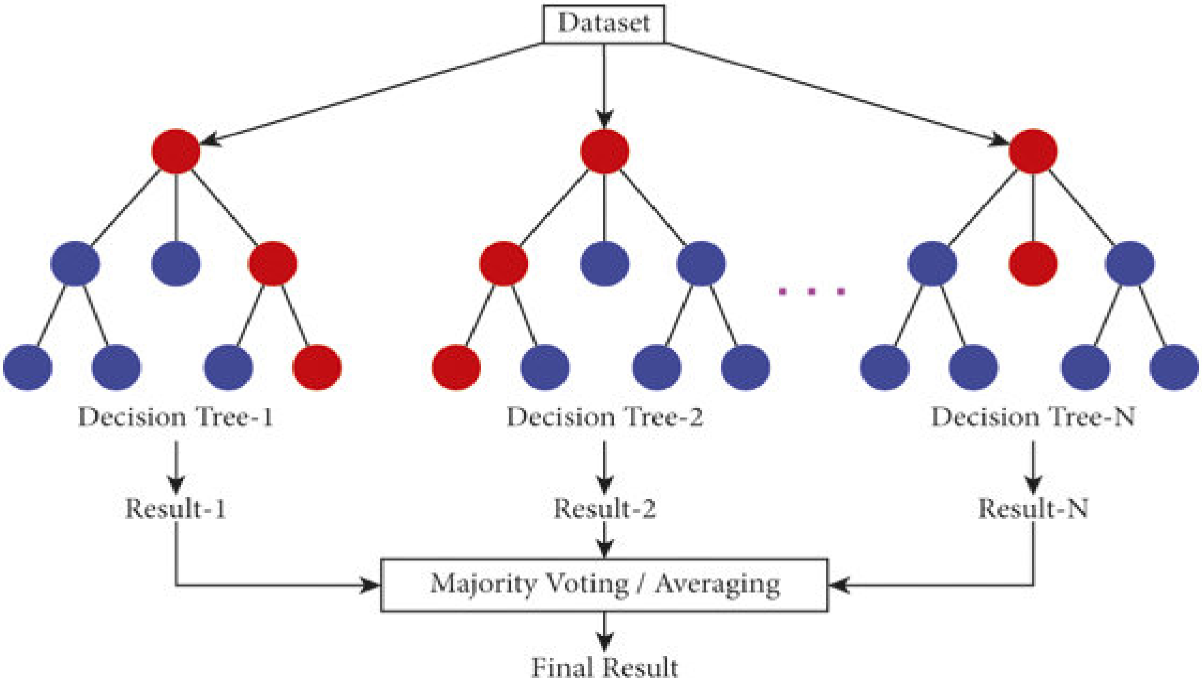


Figure 3.13: Schematic diagram of the random forest [25]

XGBoost: XGBoost, an abbreviation for eXtreme Gradient Boosting, stands out as a robust machine learning algorithm commonly employed for classification tasks. It falls under the ensemble learning category, leveraging the strengths of decision trees to enhance predictive accuracy. Through iterative boosting of weak learners, XGBoost minimizes errors, thereby improving the overall performance of the model.In Figure 3.14, there is an illustrated representation of a boosting algorithm.

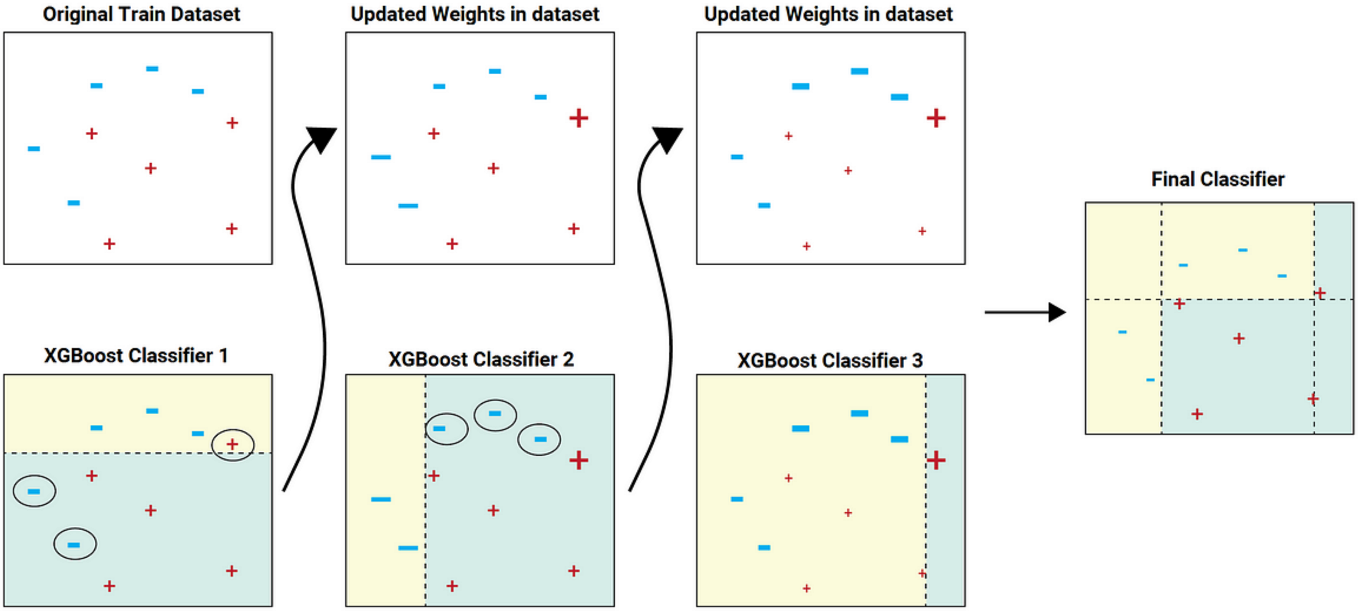


Figure 3.14: Schematic diagram of the boosting algorithm [26]

3.6.2 Deep Learning Techniques

For deep learning, many models have been used for both speech and text. For speech, CNN and LSTM were used, while for text, the pre-trained models GPT and BERT were used.

Long Short-Term Memory (LSTM) Networks:

LSTM networks succeed in gathering and keeping information throughout their operation, employing feedback mechanisms.The LSTM structure, explained in references like figure 3.15, involves activating the same cell repetitively while modifying its internal state. Key components of LSTM cells include the cell state, the forget gate determining information omission, and the input gate deciding what should advance to the next activation.This capability is crucial in speech classification, where temporal dependencies and context play a significant role in accurately identifying speech patterns and emotions.

.

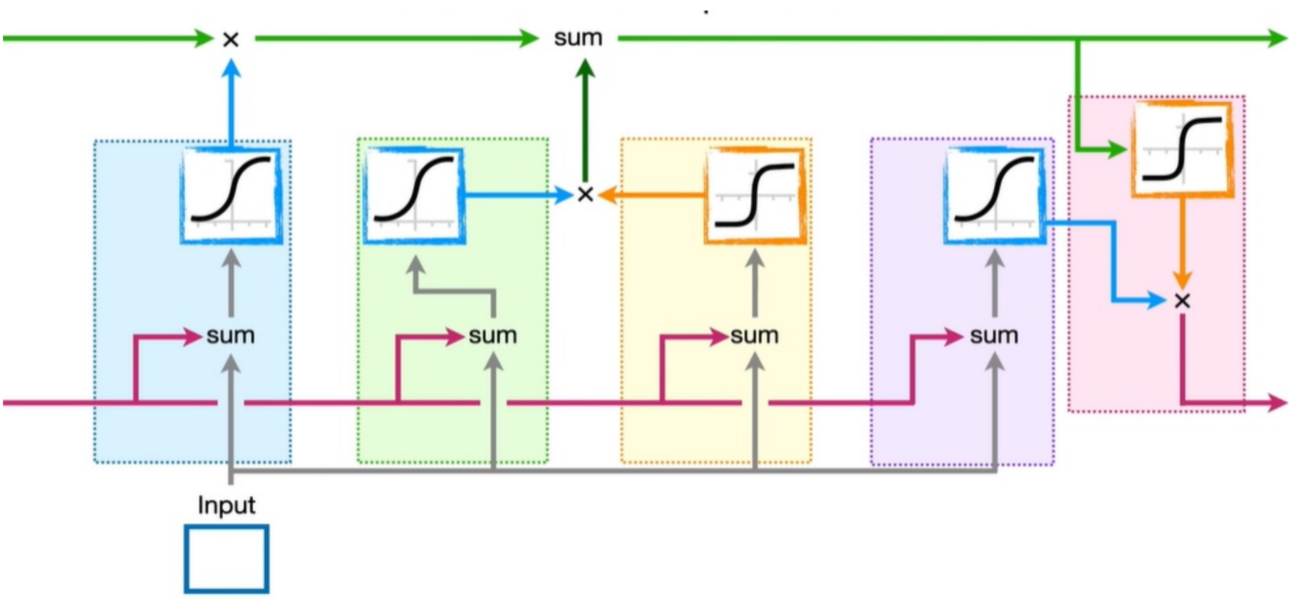


Figure 3.15: LSTM network [27]

Convolutional Neural Networks (CNNs):

Convolutional Neural Networks (CNNs) have become a powerful tool for a wide range of classification tasks as audio classification. Originally recognized for their exceptional capabilities in computer vision, CNNs have proven to be highly versatile. By effectively capturing spatial and temporal patterns, CNNs offer a robust framework for accurate and efficient decision-making across various datasets and modalities.And there are 2 types of CNN's which are 1D CNNs which used for audio classification. and 2D CNNs which used for image and video processing, where data is represented in a two-dimensional grid.In our project we focus on **1D CNNs** because its effective for audio classification tasks due to their ability to process input as a one-dimensional sequence. In audio classification, these networks can capture local patterns and features within audio signals, discerning crucial acoustic nuances necessary for accurate classification.

The figure 3.16 show the architecture of CNN network

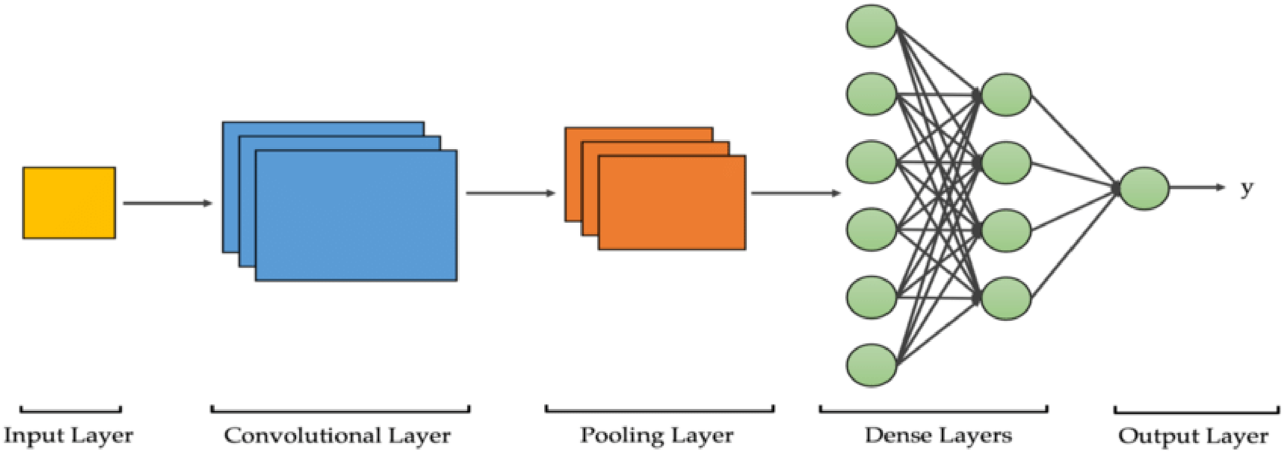


Figure 3.16: CNN network [28]

And here the description of each layer of cnn network

Input Sequence: The audio signal is represented as a sequence of data points, forming the input sequence for the CNN.

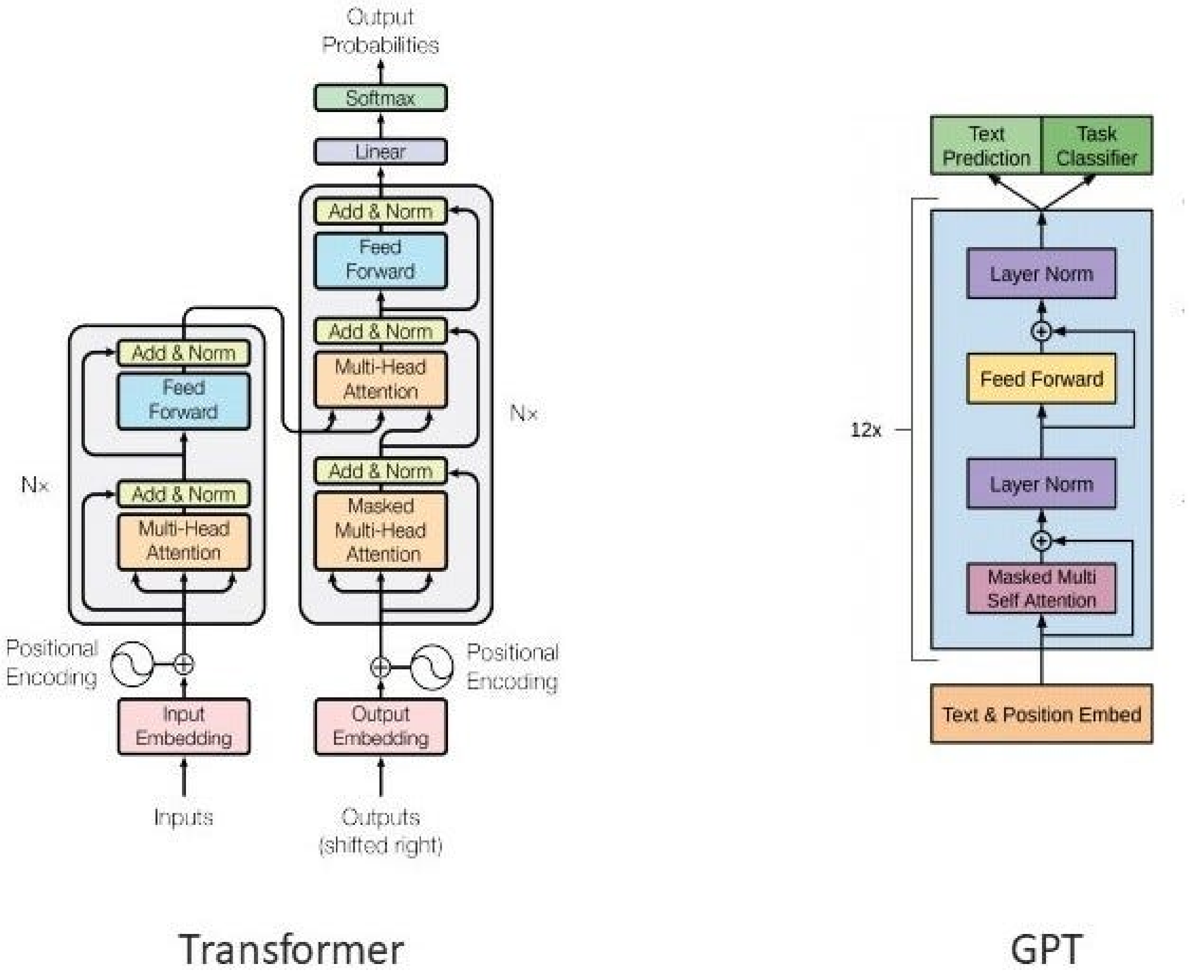
Convolutional Layers: These layers detect local patterns within the audio signals, capturing significant features.

Pooling layer: Reduces the dimensions of feature maps, lowering computational load and helping to prevent overfitting.

Dense Layers: The final dense layers are crucial for classification accuracy, integrating the extracted features to make precise predictions.

GPT-2: GPT-2 which Developed by OpenAI is a powerful transformer-based language model that is pre-trained on a large corpus of text which around 8 million high-quality webpages. The main goal of GPT-2 is to predict the next word based on the context provided by all previous words in a given text [29]. The GPT-2 model can be trained with an additional custom dataset using a method called transfer learning to produce more relevant text [30].

This model can be adapted for classification tasks, including emotion detection, through a process called fine-tuning. This process involves training the pre-trained GPT-2 model on an additional dataset related to the specific task. During fine-tuning, the model learns to recognize emotional content of text based on the features it identified in the training data. This process leverages the extensive language understanding capabilities of GPT-2, allowing it to achieve high accuracy in detecting emotions in text [31]. In emotion detection, the model is trained on a labeled dataset where each sample has a text with the label of its specific emotion, and by setting the model's parameters and optimizing its performance on the specific task, GPT-2 can effectively distinguish between different emotional states such as happiness, sadness, anger, and neutrality.



Transformer

GPT

Figure 3.17: Transformer model architecture [32]

GPT-2 uses an unmodified Transformer decoder, except that it lacks the encoder attention part. As shown in the diagrams 3.17. The GPT2 is built using transformer decoder blocks.

BERT (BERT): BERT is built on transformers to learn the contextual relationships between words in a text. A basic transformer consists of encoders that read the text input to understand the language and context, and decoders to generate predictions for the task. However, BERT only utilizes the transformer's encoder because its objective is to create a language representation model. BERT takes a sequence of tokens as input, and its transformer encoder processes the entire sequence simultaneously. This enables BERT to understand a word's context from all directions, not limited to just left or right.

BERT learns the language through two training techniques at the same time; Masks Language Modeling (MLM) and Next Sentence Prediction (NSP).

1. MLM (Masked Language Modeling): This technique enables BERT to grasp the context

of a sentence by randomly masking approximately 15% of the words (tokens) in the sentence. BERT then predicts these masked words based on the context provided by the surrounding unmasked words in the sequence. To predict the masked words, the following steps are executed:

- In figure 3.18, a classification layer is placed on the top of the encoder's output. The resulting vectors are then multiplied by the embedding matrix to convert them into vocabulary vectors.

- SoftMax is a mathematical function that transforms a vector of numbers into a vector of probabilities it used to calculate the probability of each word in the vocabulary vector. Which using the following equation:

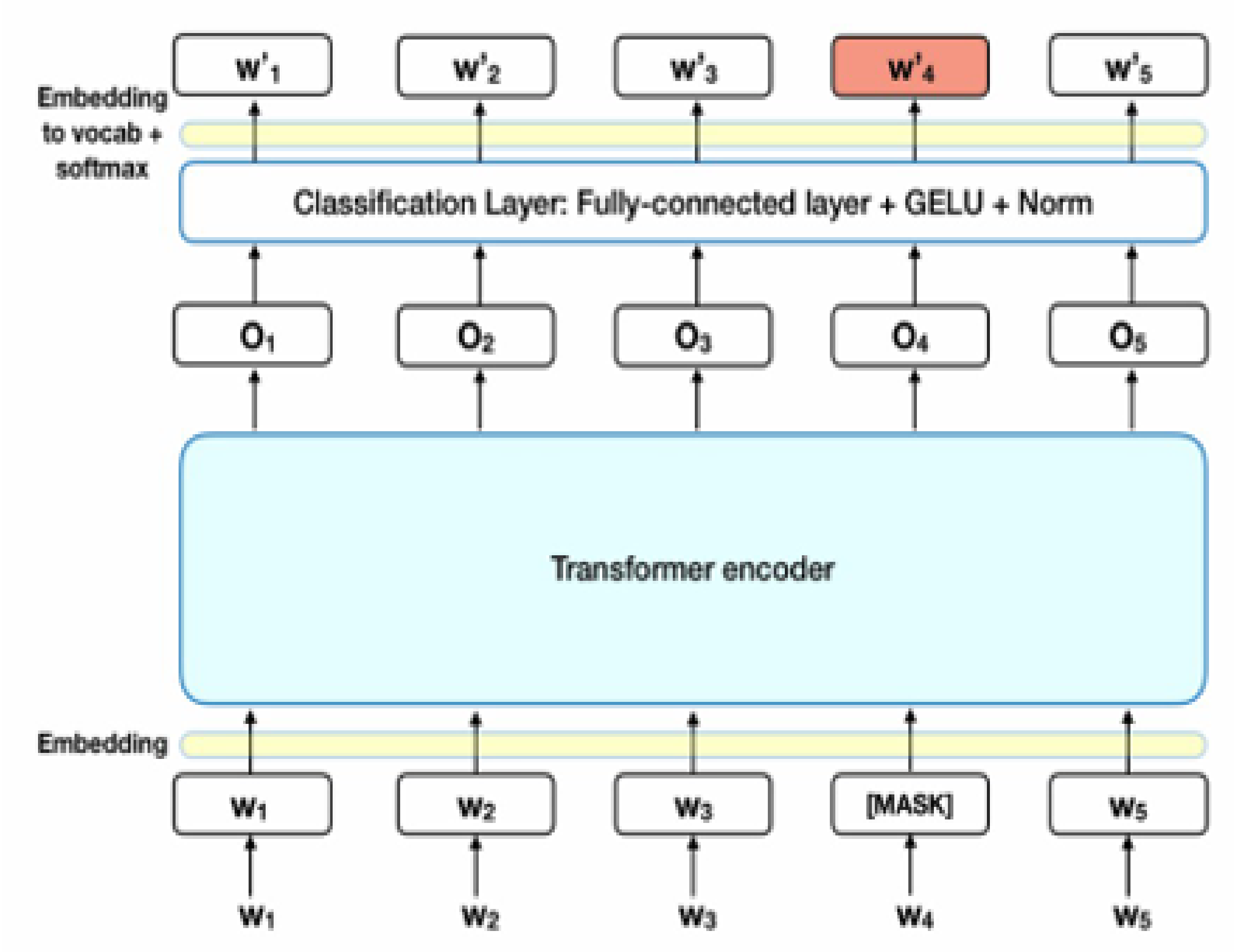

W'1
W'2
W'3
W'4
W'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
W1
W2
W3
[MASK]
W5
W1
W2
W3
W4
W5

Figure 3.18: BERT model: Input and output of masked word(s). [33]

2. NSP: Helps BERT in understanding the relationship between sentences by assessing whether the second sentence follows logically from the first. To differentiate between the two sentences, BERT's input embedding has to include the following three embeddings before start-

ing the training:

• Token embeddings: involve adding two tokens to the input word tokens. A [CLS] token is appended to the beginning of the first sentence, and a [SEP] token is appended to the end of each sentence.

• Segment embeddings: involve adding a sentence embedding to each input token to differentiate between the sentences (e.g., distinguishing sentence A from sentence B).

• Positional embeddings: this embedding is added to each token in order to know the token's position in the sentence.

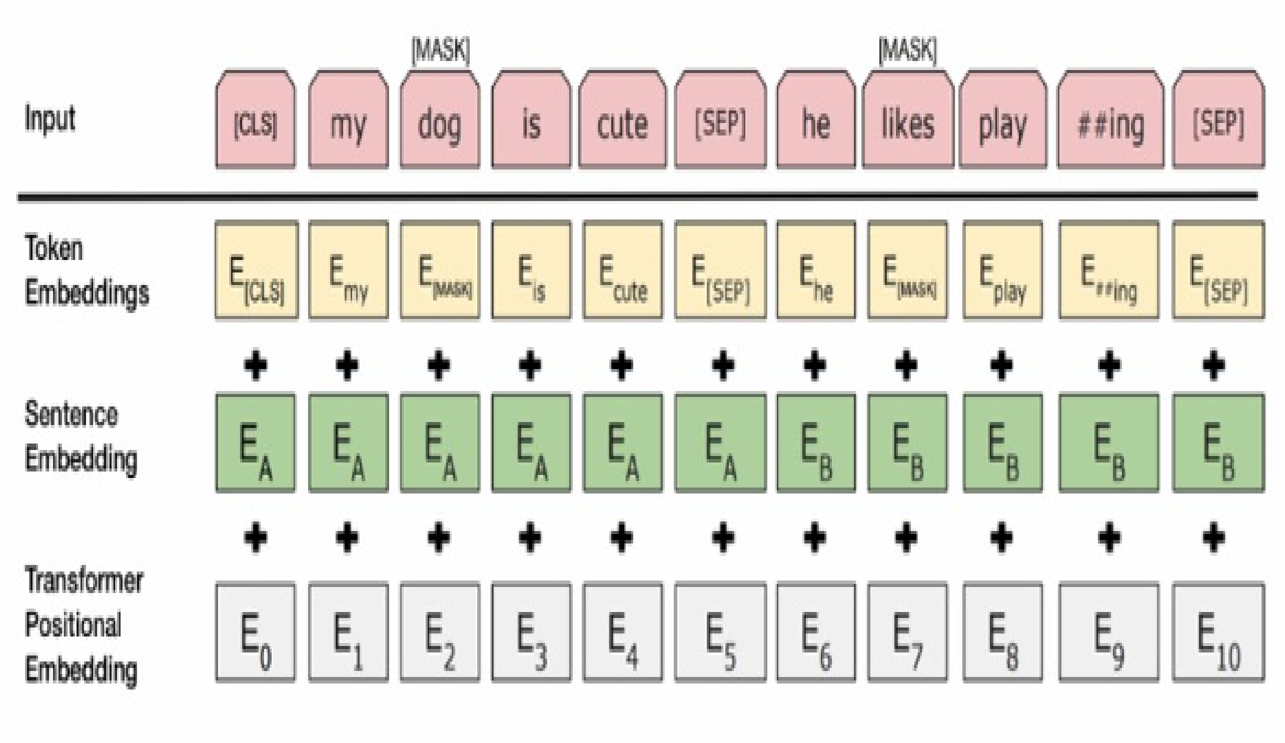


Figure 3.19: BERT input representation. [34]

4 Experiments and Results

In this section, we present the audio and text emotion classification using multiple classifiers, and showing the fusion of these models. The results are expressed in terms of accuracy, recall, precision, and F1-score.

4.1 Speech Emotion Classification

In this subsection, the RAVDESS dataset was utilized, split into training and testing sets with a ratio of 75:25. After the split, preprocessing techniques were applied. For the training dataset, data augmentation techniques were employed, while both of them the training and testing datasets were trimmed silence.

1. Data augmentation:

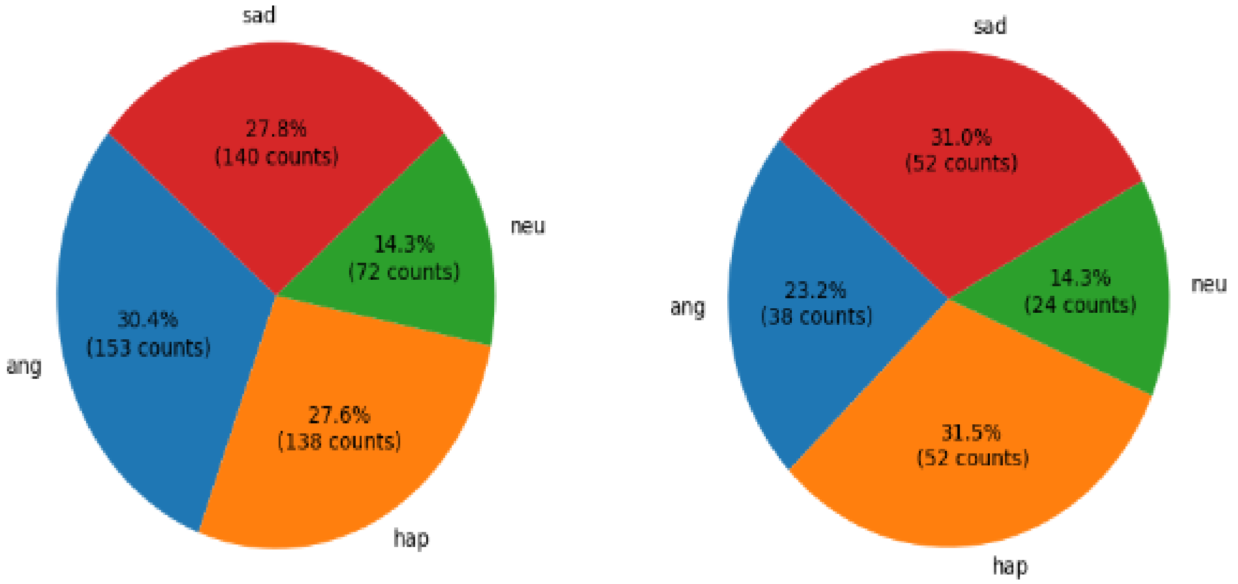
   (1) Adding a random Gaussian noise with a factor of 0.001.

   (2) shifting the waveform of a WAV file to the right by a random amount.

2. Audio trimming:

   (1) Removing silence from the audio using a threshold of -65 dB.

Figure 4.1 shows the distribution of labels for the training and testing datasets after splitting and preprocessing.



(a) Training data

(b) Testing data

Figure 4.1: Label distributions comparison between the training and testing data

After applying feature extraction techniques to each voice sample, including chroma, mel spectrogram, and MFCC, the resulting features were shaped into an array with dimensions (180, 1). The labels were encoded as follows: 0 for anger, 1 for happiness, 2 for neutral, and 3 for sadness. After that , the following classifiers were applied to the extracted features:

4.1.1 CNN Classifier

The CNN model has used consists of 4 layers, and it's hyperparameters were fine-tuned using grid search with the configurations :

```
param_grid = {
    ’neurons’: [64, 128, 256],
    ’conv_size’: [2, 3, 5],
    ’dropout_rate’: [0.1, 0.5]
}
```

After performing grid search, it was found that the optimal parameters for the CNN model are {convSize : 5, dropout_rate: 0.1, neurons : 256}.

Figure 4.2 shows the final model architecture for the CNN model.

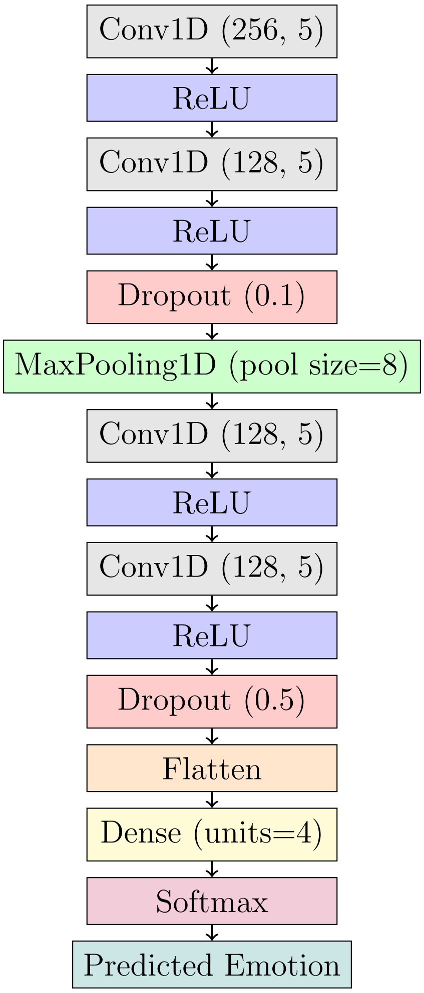


Figure 4.2: Cnn model architecture

After training and testing the model with 50 epochs, it achieved a very good results, as illustrated in Table 4.1. Additionally, Table 4.2 presents the corresponding confusion matrix.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Angry	0.8780	0.9231	0.9000	39
Happy	0.8039	0.7736	0.7885	53
Neutral	0.8636	0.7917	0.8261	24
Sad	0.7407	0.7692	0.7547	52
Accuracy	0.8095			
Macro avg	0.8215	0.8144	0.8173	168

Table 4.1: CNN Classification Report

Actual \ Predicted	Angry	Happy	Neutral	Sad	All
Angry	36	1	0	2	39
Happy	3	41	0	9	53
Neutral	1	1	19	3	24
Sad	1	8	3	40	52
All	41	51	22	54	168

Table 4.2: Confusion matrix of CNN

4.1.2 LSTM Classifier

The LSTM model was built using TensorFlow's Keras API. It consists of an LSTM layer with 128 units to process the sequential data, followed by a Dropout layer to prevent overfitting. The model flattens the output and uses a Dense layer with a softmax activation function to classify the features into four emotional states: anger, happiness, neutral, and sadness.

After training and testing the model with 50 epochs, it achieved a good results, as illustrated in Table 4.3. Additionally, Table 4.4 presents the corresponding confusion matrix.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Angry	0.8537	0.8974	0.8750	39
Happy	0.8667	0.7358	0.7959	53
Neutral	0.6923	0.7500	0.7200	24
Sad	0.7143	0.7692	0.7407	52
Accuracy	0.7857			
Macro avg	0.7818	0.7881	0.7829	168

Table 4.3: LSTM Classification Report

Predicted Actual	Angry	Happy	Neutral	Sad	All
Angry	35	1	0	3	39
Happy	2	39	3	9	53
Neutral	1	1	18	4	24
Sad	3	4	5	40	52
All	41	45	26	56	504

Table 4.4: Confusion matrix of LSTM

4.1.3 Random Forest Classifier

The random forest model was utilized for classification with hyperparameters set to number of estimators=300 and random state=42. After evaluating the model using the dataset, it achieved moderate results, as shown in Table 4.5. Additionally, Table 4.6 presents the corresponding confusion matrix.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Angry	0.9091	0.7692	0.8333	39
Happy	0.6613	0.7736	0.7130	53
Neutral	0.7895	0.6250	0.6977	24
Sad	0.7037	0.7308	0.7170	52
Accuracy	0.7381			
Macro avg	0.7659	0.7246	0.7403	168

Table 4.5: Random Forest Classification Report

Predicted Actual	Angry	Happy	Neutral	Sad	All
Angry	30	7	0	2	39
Happy	3	41	0	9	53
Neutral	0	4	15	5	24
Sad	0	10	4	38	52
All	33	62	19	54	168

Table 4.6: Confusion matrix of RF

4.2 Text Emotion Classification

In this section, we explore the performance of different models—BERT, GPT, and XGBoost on text emotion classification tasks. The evaluation includes analyzing the models' accuracy, precision, recall, and F1-score to determine their effectiveness in recognizing and classifying different emotions.

4.2.1 BERT Classifier

The bert model were used is a "bert-base-uncased" which is a pretrained model on English language using a masked language modeling (MLM) objective [35], the training process was obtained with a batch size of 64. Early stopping was employed, and the best result was achieved during epoch 2, where the training loss was 34.29%, and the validation loss was 47.59% , the training accuracy reached 91.80%, and the validation accuracy was 82.63%.

The results obtained are shown in the table [4.8], and The obtained confusion matrix is shown in figure [4.7] which gave an accuracy rate of 85%. By extracting the results of the precision of recognition for each emotion, neutrality gave the highest accuracy rate of 89%, followed by happiness, nurture, sadness, and anger, respectively.

Predicted Actual	Angry	Happy	Neutral	Sad	All
Angry	590	1	61	2	654
Happy	0	524	131	0	655
Neutral	3	2	559	91	655
Sad	3	2	54	595	654
All	596	529	805	688	2618

Table 4.7: BERT confusion matrix.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Angry	0.81	0.91	0.86	654
Happy	0.89	0.92	0.90	655
Neutral	0.86	0.70	0.78	655
Sad	0.84	0.86	0.85	654
Accuracy	0.85			
Macro avg	0.85	0.85	0.85	2281

Table 4.8: BERT Classification Report

4.2.2 GPT Classifier

The GPT model were used is gpt2, the training process was obtained with a batch size of 64. Early stopping was employed, and the best result was achieved during epoch 18, where the training loss was 36.5%, and the validation loss was 53.2% , the training accuracy reached 86.7%, and the validation accuracy was 80.8%.

The results obtained are shown in the table [4.10], and The obtained confusion matrix is shown in figure [4.7] which gave an accuracy rate of 82%. By extracting the results of the precision of recognition for each emotion, neutrality gave the highest accuracy rate of 86%, followed by anger, nurture, happiness, and sadness respectively.

Predicted Actual	Angry	Happy	Neutral	Sad	All
Angry	550	26	40	91	707
Happy	14	476	10	38	538
Neutral	40	27	379	78	524
Sad	45	24	24	419	512
All	649	553	453	626	2281

Table 4.9: GPT-2 confusion matrix.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Angry	0.86	0.78	0.82	707
Happy	0.82	0.91	0.86	538
Neutral	0.86	0.85	0.86	524
Sad	0.77	0.78	0.78	512
Accuracy	0.82			
Macro avg	0.83	0.83	0.83	2281

Table 4.10: GPT-2 Classification Report

4.2.3 XGBoost Classifier

The results are shown in Table [4.12], and the confusion matrix is displayed in Figure [4.11]. The model achieved an accuracy rate of 77%. Among the emotions, neutrality had the highest precision with an accuracy rate of 93%, followed by happiness, sadness, anger, and nurture respectively, nurture had the lowest accuracy rate among the emotions, indicating the model's poorest performance in this category

Predicted Actual	Angry	Happy	Neutral	Sad	All
Angry	467	11	184	14	676
Happy	28	430	93	6	557
Neutral	40	10	522	15	587
Sad	25	13	94	329	52
All	33	62	19	54	461

Table 4.11: XGBoost confusion matrix

Class	**Precision**	**Recall**	**F1-score**	**Support**
Angry	0.83	0.69	0.76	676
Happy	0.93	0.77	0.84	557
Neutral	0.58	0.89	0.71	587
Sad	0.90	0.71	0.80	461
Accuracy	0.77			
Macro avg	0.81	0.77	0.78	2281

Table 4.12: XGBoost Classification Report

4.3 Model Fusion

In this subsection, we have applied Late fusion technique by using various combinations of speech and text models. Each combination will include at least one model from the speech category and one from the text category. so in this case we will have a total of 49 different combinations. These combinations will be applied to the MER dataset that we have gathered.

Before we start the fusion process, we evaluated each model independently on the MER dataset. This allows us to compare the accuracy of each model model against the accuracy achieved through model fusion.

Table 4.13 shows accuracy of each model independentn at the MER dataset.

Model Name	Accuracy
GPT (Text)	0.47
XGB (Text)	0.42
BERT (Text)	0.48
CNN (Speech)	0.55
LSTM (Speech)	0.49
RF (Speech)	0.57

Table 4.13: Models accuracy at MER dataset before fusion

As shown in Figure 4.13, the accuracies appear low for several reasons. For the speech models, they were trained on acted data but evaluated on real data, which is significantly different. For the text models, there are cases where the speaker's emotion does not align with their spoken words; for example, a speaker might sound angry but their words do not looks anger.

In the case of fusion, The output of each fusion will depend on the majority vote of the predicted emotions. In the case of tie, the majority output from the speech model will be used to determine the final emotion because the speech models give a higher accuracy that text when we applied it alone at MER dataset as shown in table 4.13.

After evaluating each model independently on the MER dataset, we applied the 49 combinations of fusion models. The combination [BERT, XGP, CNN, RF] achieved the highest accuracy of 0.71, while the combination [XGP, LSTM] resulted in the lowest accuracy of 0.44.

Figure 4.3 displays a bar chart illustrating the number of voices predicted by different numbers of models:

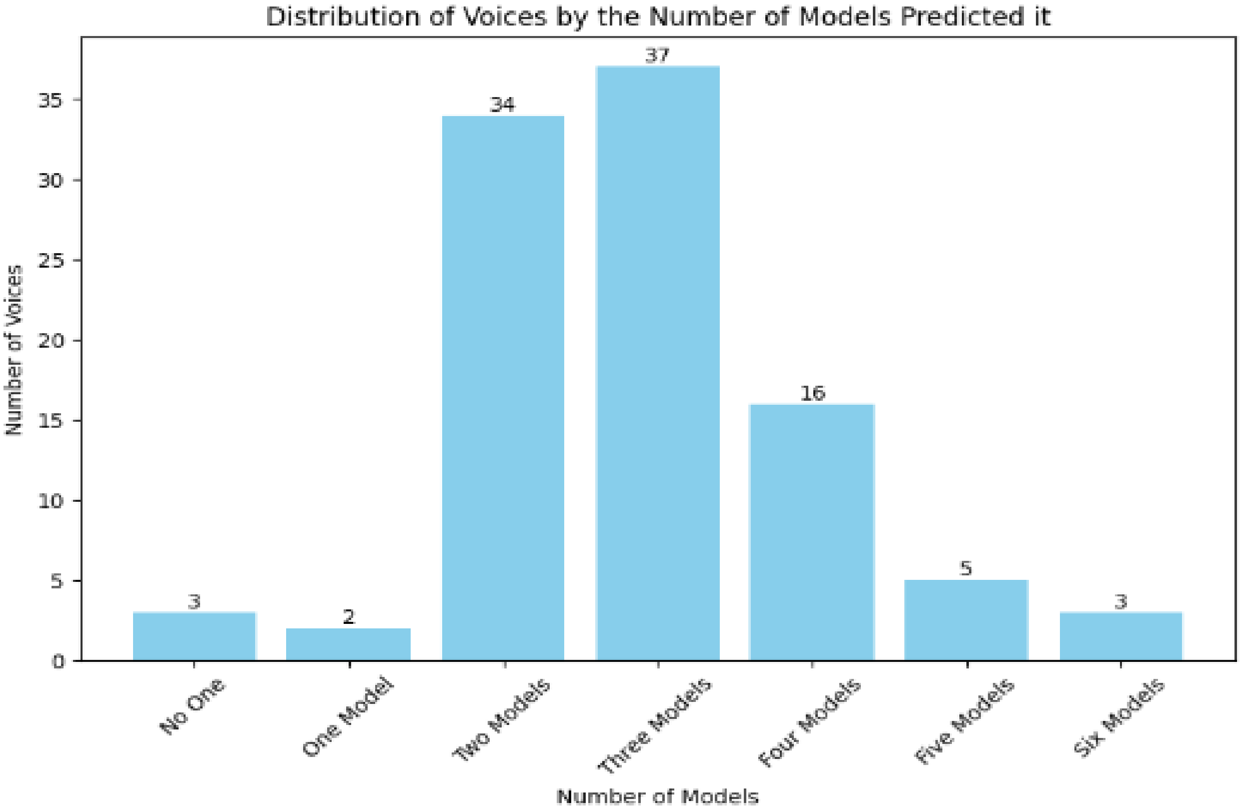


Figure 4.3: Number of voices predicted by different numbers of models

In Figure 4.3, we observe that there are three instances where the voice was not predicted by any model, two instances where it was predicted by one model, and so on.

Figure 4.4 displays a bar chart illustrating the top 15 combinations based on their accuracy.

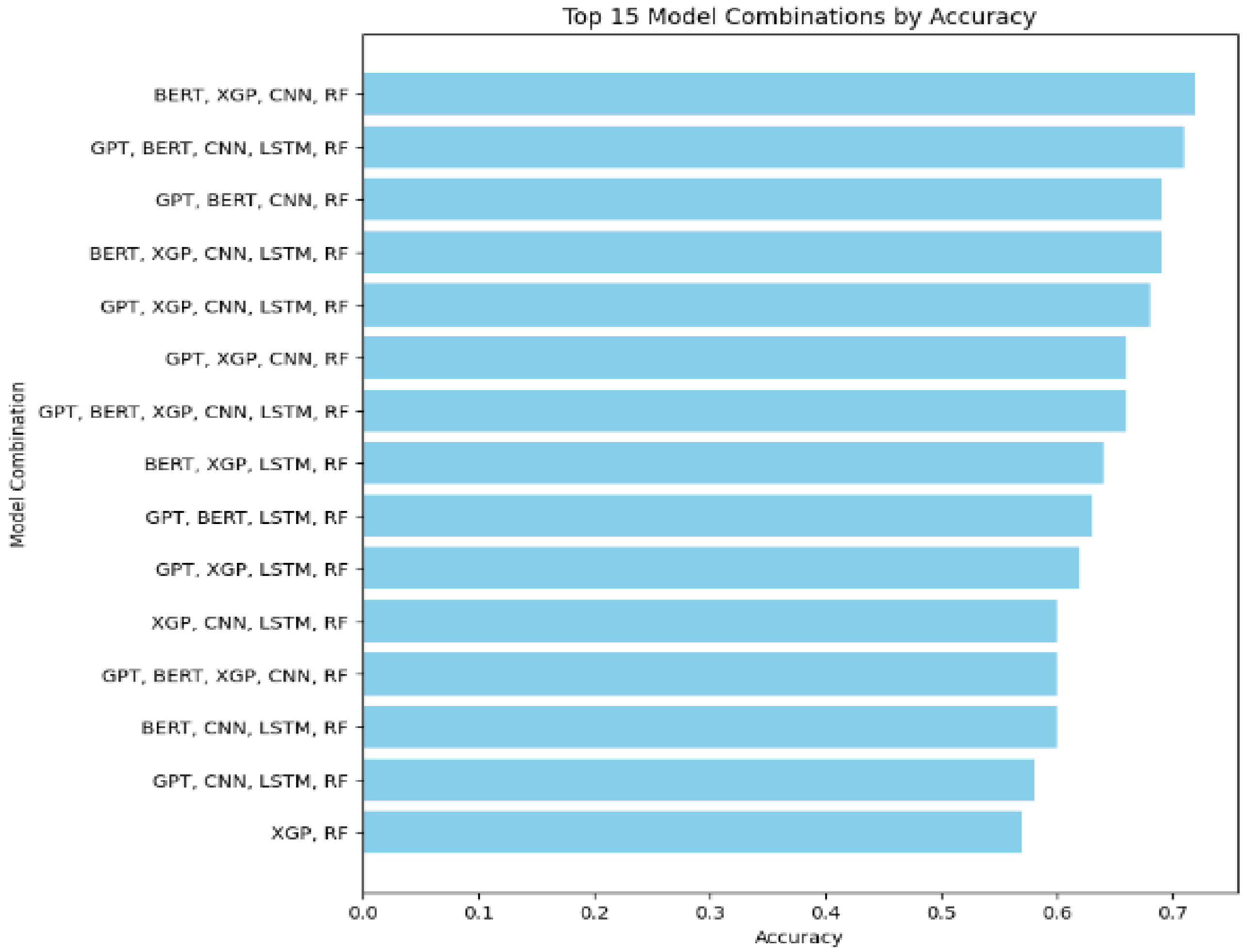


Figure 4.4: Top 15 model combinations by accuracy

In Figure 4.4, it is clear that fusion models improve accuracy compared to individual models across many combinations. This is because several reasons: speech and text provide different information, where the fusion offers a more complete understanding of the emotion. Also, in cases where one of speech or text alone is unclear, the other modality can help clarify the emotional state.

5 WebApplication Development

To help users try our system easily, we created a web app. It can classify any English voice into target classes. Currently, it runs only on our laptops since we couldn't find a web server to host it. The next sections explain the technologies we used for the app's backend and frontend, along with its features and design.

5.1 Backend

We used Flask, a Python web framework, to build the backend of our project demo website and interact with the frontend. When a user uploads a WAV file, Flask receives the request from the frontend, processes it using the appropriate models for converting speech to text, and then predicts using all six models, sending back an array of results for display. Flask is valued for its simplicity and ease of use, enabling quick development and deployment of web applications with minimal setup. It also offers numerous features and extensions that help developers create robust, scalable web apps capable of handling high traffic. Flask's flexibility and power made it the ideal choice for our project demo's backend.

5.2 Frontend

HTML, CSS, JavaScript, and Bootstrap were used to create the look and design of our website. These technologies handle how things appear on the screen, but they don't talk directly to the backend. Instead, they work together with the backend using AJAX requests. When a user submits a WAV file, JavaScript sends this data to the backend through a specific URL. Flask, our backend system, then processes this data using the models to analyze the emotions. The results come back to the frontend as JSON data, which JavaScript uses to update the website and show the user the text analysis findings. This mix of frontend and backend technologies, providing a smooth experience for users.

5.3 Website Functionality and User Interface

5.3.1 Main Page View

The following screen Figure 5.1, is the main page that will appear to the user once he/she opens the website.

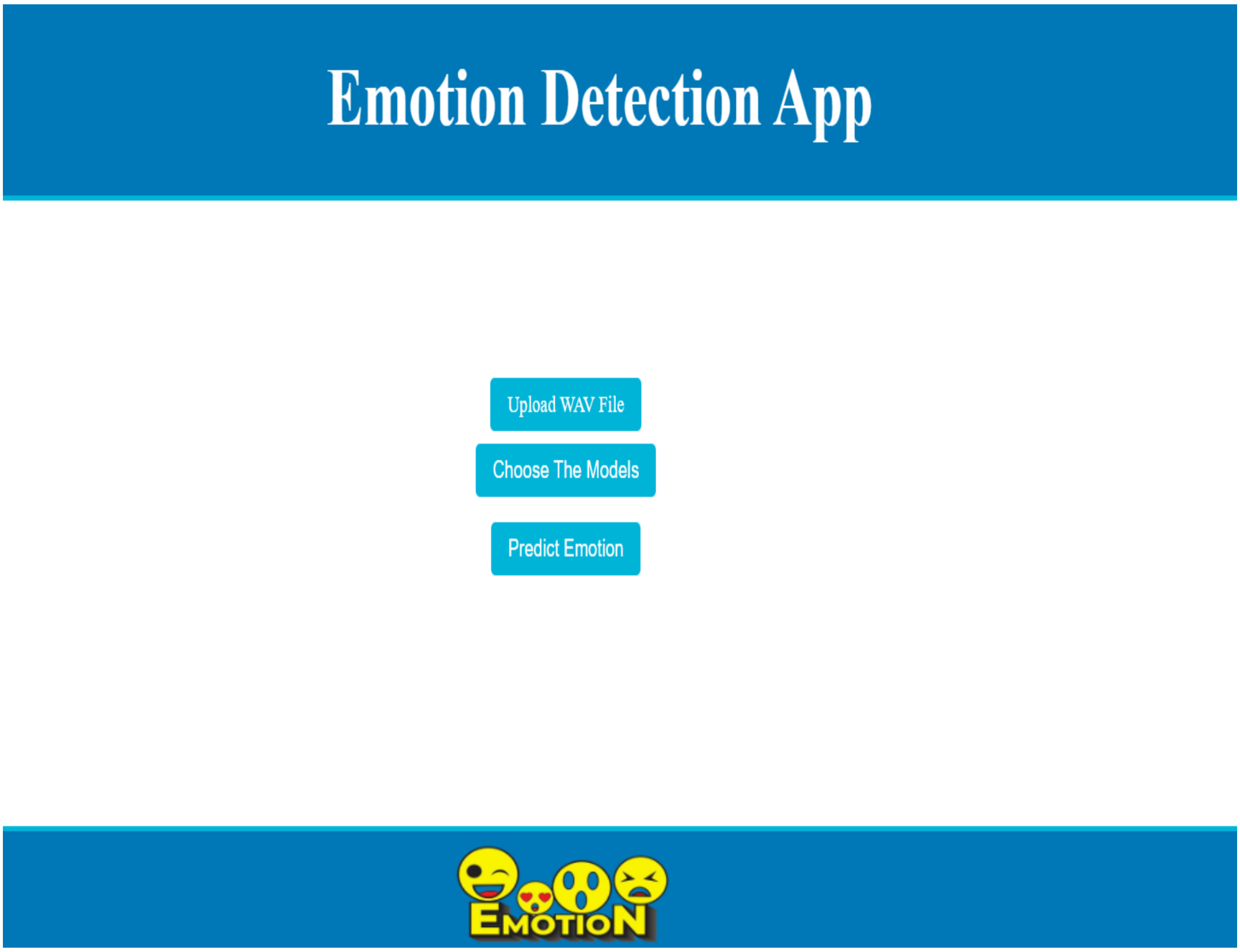
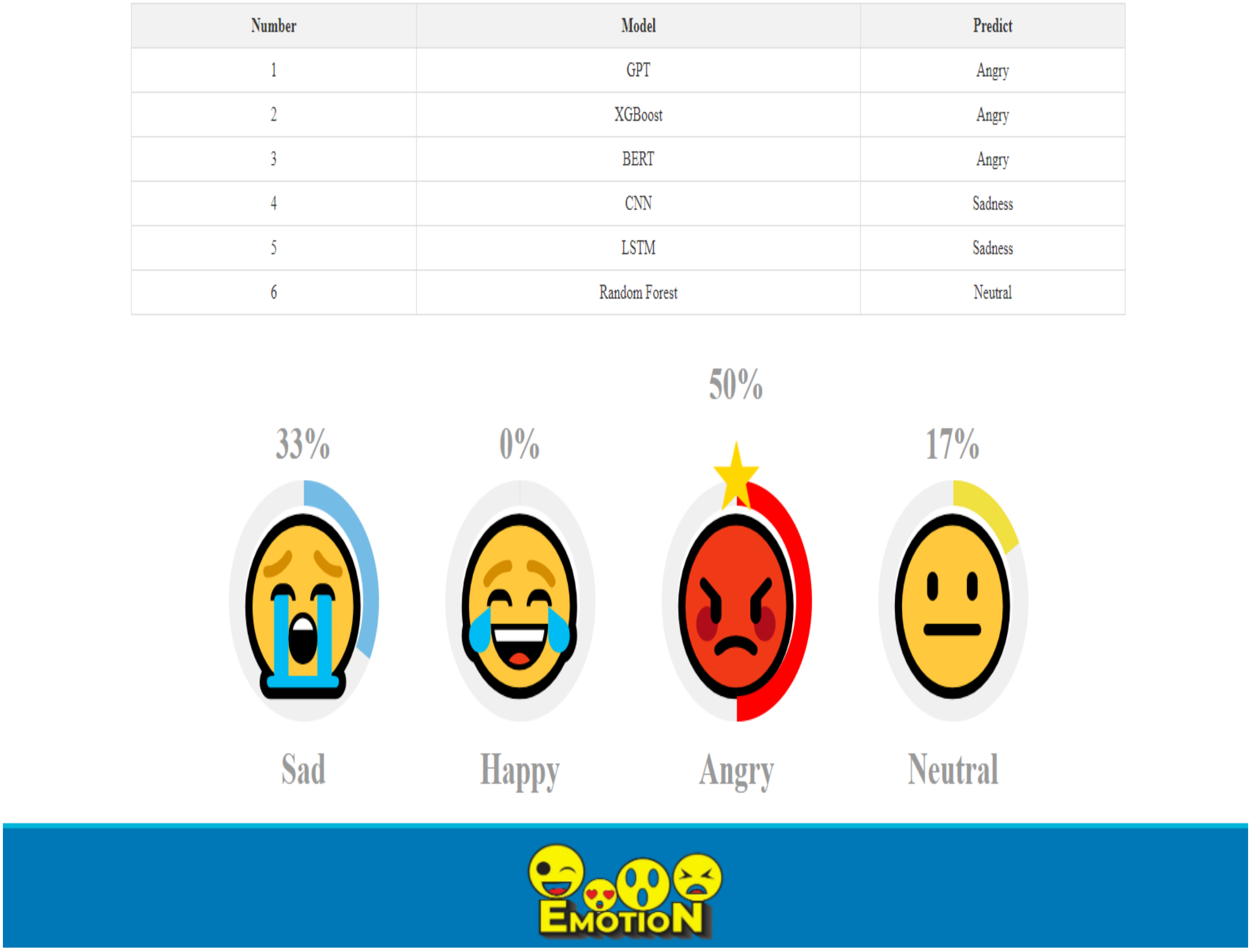


Figure 5.1: main page

The main page of the project demo website will have a simple and user-friendly interface for speech emotion classification. Users can upload a WAV file through an HTML form and select a model from 'Choose the Models' button. After uploading their file and choosing a model, they click a 'Predict Emotion' button to start the classification process. The website will then display the predicted label and probability scores for the input speech in a clear, easy-to-understand format, covering four labels: happy, sad, angry, and neutral.

5.3.2 Results View

The result page shown in Figure 5.2 appears to the user once he/she press the 'run' button.



Number	Model	Predict
1	GPT	Angry
2	XGBoost	Angry
3	BERT	Angry
4	CNN	Sadness
5	LSTM	Sadness
6	Random Forest	Neutral

Figure 5.2: Results View

The first part is a table that display the predicted classes from each selected model. Below the table, a chart will show the final percentages for each class represented as emojis, with a circle drawn around each emoji corresponding to the percentage (e.g., 50% shown as a half circle).

6 Conclusion and Future Work

6.1 Conclusion

The purpose of this project is to improve emotion recognition by using a multimodal approach that integrates speech and text using multiple machine learning and deep learning techniques. The speech models is trained on the RAVDESS dataset, while the text models is trained on the ISEAR and GoEmotions datasets. Multiple features were extracted for the speech data (including mel spectrogram, chroma, and MFCC), and various text feature extraction techniques were used (such as TF-IDF and word embeddings like BERT and GPT). The evaluation of the models was conducted using several machine learning and deep learning techniques. For speech, the models used include Random Forest , Long Short-Term Memory , and Convolutional Neural Networks . For text, the models used include XGBoost , GPT-2, and BERT. After that, Late fusion technique were applied to combine different models to predict the output emotion, which was one of the following: happy, angry, sad, or neutral. The final evaluation of the system was conducted on a dataset gathered from YouTube, which contains 100 different voices. The accuracy of individual models on this dataset was as follows: GPT: 0.47, XGBoost: 0.42, BERT: 0.48, CNN: 0.55, LSTM: 0.49, and Random Forest: 0.57 After applying late fusion, we experimented with 49 different combinations, resulting in accuracy ranges from 0.44 to 0.71. The best combination, which achieved the highest accuracy of 0.71, was BERT, XGB, CNN and RF. The worst combination, which resulted in the lowest accuracy of 0.44, was XGB and LSTM]. Out of the experiments conducted in this project, we concluded that late fusion has increased the accuracy by good a difference.

6.2 Future Work Plan

Based on our results and the progress made in our project, there are several future tasks to do:

- **Expand the Emotion Categories:** Instead of using only four emotions (happy, sad, neutral, angry), we can including additional emotions such as disgust, fear, and surprise. This addition will make our model more comprehensive and clear for different instances of voices.

- **Build a Larger Real-World Dataset for Multimodal Emotion Recognition:** To improve the performance of our model, its better to create a larger dataset for the MER . This dataset should be used not only for testing but also for training the model instead of using acted data for training.

- **Support Real-Time Emotion Recognition:** Developing our model to support real-time emotion recognition would expand its applications by making it suitable for live interactions and responsive systems.

- **Incorporate Facial Expression Recognition:** Adding another model for facial expression recognition will provide a more understanding of emotions. By integrating this with our existing system, we can achieve more accurate and emotion detection.

These improvements will enhance the performance and applicability of our emotion recognition system.

References

[1] Nikita Chitre, Namrata Bhorade, Pradnya Topale, Jyoti Ramteke, and C. R. Gajbhiye. "Speech Emotion Recognition to assist Autistic Children". *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC).* 2022, pp. 983–990. DOI: 10.1109/ICAAIC53929.2022.9792663.

[2] Choubeila Maaoui, Alain Pruski, and Faiza Abdat. "Emotion Recognition for hHman-Machine Communication". *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2008, pp. 1210–1215. DOI: 10.1109/IROS.2008.4650870.

[3] Dibyendu Seal, Uttam K Roy, and Rohini Basak. "Sentence-level emotion detection from text based on semantic rules". *Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018.* Springer. 2020, pp. 423–430.

[4] Maryam Hasan, Elke Rundensteiner, and Emmanuel Agu. "Automatic emotion detection in text streams by analyzing twitter data". *International Journal of Data Science and Analytics* 7 (2019), pp. 35–51.

[5] Santosh Kumar Bharti, S Varadhaganapathy, Rajeev Kumar Gupta, Prashant Kumar Shukla, Mohamed Bouye, Simon Karanja Hingaa, and Amena Mahmoud. "Text-Based Emotion Recognition Using Deep Learning Approach". *Computational Intelligence and Neuroscience* 2022 (2022).

[6] Yang Li and Yunxin Zhao. "Recognizing emotions in speech using short-term and long-term features". *Fifth international conference on spoken language processing.* 1998.

[7] Yi-Lin Lin and Gang Wei. "Speech emotion recognition based on HMM and SVM". *2005 international conference on machine learning and cybernetics.* Vol. 8. IEEE. 2005, pp. 4898–4901.

[8] Aditya Bihar Kandali, Aurobinda Routray, and Tapan Kumar Basu. "Emotion recognition from Assamese speeches using MFCC features and GMM classifier". *TENCON 2008-2008 IEEE region 10 conference.* IEEE. 2008, pp. 1–5.

[9] Peipei Shen, Zhou Changjun, and Xiong Chen. "Automatic speech emotion recognition using support vector machine". *Proceedings of 2011 international conference on electronic & mechanical engineering and information technology.* Vol. 2. IEEE. 2011, pp. 621–625.

[10] Reem Hamed Aljuhani, Areej Alshutayri, and Shahd Alahdal. “Arabic speech emotion recognition from saudi dialect corpus”. *IEEE Access* 9 (2021), pp. 127081–127085.

[11] Sandeep Kumar Pandey, Hanumant Singh Shekhawat, and SR Mahadeva Prasanna. “Deep learning techniques for speech emotion recognition: A review”. *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*. IEEE. 2019, pp. 1–6.

[12] Sandeep Kumar Pandey, Hanumant Singh Shekhawat, and SR Mahadeva Prasanna. “Deep learning techniques for speech emotion recognition: A review”. *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*. IEEE. 2019, pp. 1–6.

[13] Ziyang Ma, Wen Wu, Zhisheng Zheng, Yiwei Guo, Qian Chen, Shiliang Zhang, and Xie Chen. “Leveraging speech ptm, text llm, and emotional tts for speech emotion recognition”. *arXiv preprint arXiv:2309.10294* (2023).

[14] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. “Multimodal speech emotion recognition using audio and text”. *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 112–118.

[15] Kashfia Sailunaz, Manmeet Dhaliwal, Jon Rokne, and Reda Alhajj. “Emotion detection from text and speech: a survey”. *Social Network Analysis and Mining* 8 (2018), pp. 1–26.

[16] Mikel de Velasco, Raquel Justo, Josu Antón, Mikel Carrilero, and M Inés Torres. “Emotion Detection from Speech and Text.” *IberSPEECH*. 2018, pp. 68–71.

[17] Kaggle. *RAVDESS Emotional Speech Audio*. `https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio`. Accessed: 2024-07-04. 2018.

[18] Kaggle. *ISEAR Dataset*. `https://www.kaggle.com/datasets/faisalsanto007/isear-dataset`. Accessed: 2024-07-04. 2024.

[19] Kaggle. *GoEmotions*. `https://www.kaggle.com/datasets/debarshichanda/goemotions`. Accessed: 2024-07-04. 2024.

[20] KJ Patil, PH Zope, and SR Suralkar. “Emotion detection from speech using Mfcc & GMM”. *Int. J. Eng. Res. Technol.(IJERT)* 1.9 (2012).

[21] Qing Liu, Jing Wang, Dehai Zhang, Yun Yang, and NaiYao Wang. “Text features extraction based on TF-IDF associating semantic”. *2018 IEEE 4th international conference on computer and communications (ICCC)*. IEEE. 2018, pp. 2338–2343.

[22] Luiz Gomes, Ricardo da Silva Torres, and Mario Lúcio Côrtes. "BERT-and TF-IDF-based feature extraction for long-lived bug prediction in FLOSS: a comparative study". *Information and Software Technology* 160 (2023), p. 107217.

[23] Christoph Alt, Marc Hübner, and Leonhard Hennig. "Fine-tuning pre-trained transformer language models to distantly supervised relation extraction". *arXiv preprint arXiv:1906.08646* (2019).

[24] Gyeongho Kim, Jaegyeong Choi, Minjoo Ku, Hyewon Cho, and Sunghoon Lim. "A Multimodal Deep Learning-Based Fault Detection Model for a Plastic Injection Molding Process". *IEEE Access* PP (Sept. 2021), pp. 132455–132467. DOI: 10.1109/ACCESS.2021.3115665.

[25] Muhammad Yaseen Khan, Abdul Qayoom, Muhammad Suffian Nizami, Muhammad Shoaib Siddiqui, Shaukat Wasi, and Syed Muhammad Khaliq-ur-Rahman Raazi. "Automated prediction of Good Dictionary EXamples (GDEX): a comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques". *Complexity* 2021 (2021), pp. 1–18.

[26] The Data Beast. *Interview Questions for XG Boost.* https://medium.com/@thedatabeast/interview-questions-for-xg-boost-2d4c7b7f4bbf. Accessed: 2024-07-05. 2024.

[27] *Example of LSTM network call for text classification.* Available online: https://medium.com/@abdullah.khurram_44006/understanding-lstms-9034fcdd4d09. Accessed: 2023-2-6.

[28] *Example of cnnNetwork network call for text classification.* Available online: https://medium.com/swlh/an-overview-on-convolutional-neural-networks-ea48e76fb186. Accessed: 2023-2-8.

[29] *Better language models and their implications.* Available online: https://openai.com/index/better-language-models. Accessed: 2023-12-18.

[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. "Language models are unsupervised multitask learners". *OpenAI blog* 1.8 (2019), p. 9.

[31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[32] *Better language models and their implications.* Available online: `https://medium.com/nerd-for-tech/gpt3-and-chat-gpt-detailed-architecture-study-deep-nlp-horse-db3af9de8a5d`. Accessed: 2023-12-18.

[33] Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. "A survey of word embeddings for clinical text". *Journal of Biomedical Informatics* 100 (2019). Articles initially published in Journal of Biomedical Informatics: X 1-4, 2019, p. 100057. ISSN: 1532-0464. DOI: `https://doi.org/10.1016/j.yjbinx.2019.100057`. URL: `https://www.sciencedirect.com/science/article/pii/S2590177X19300563`.

[34] Abdurahmman Alzahrani, Eyad Babkier, Faisal Yanbaawi, Firas Yanbaawi, and Hassan Alhuzali. "Investigating Persuasion Techniques in Arabic: An Empirical Study Leveraging Large Language Models". *arXiv preprint arXiv:2405.12884* (2024).

[35] Hugging Face. *google-bert/bert-base-uncased.* Accessed: 2024-07-02. 2024. URL: `https://huggingface.co/google-bert/bert-base-uncased`.