

**Major Project Report**  
on  
**Speech Emotion Recognition with Speaker Diarizer**

submitted in partial fulfillment of the requirement  
for the Award of the Degree of

**Bachelor of Technology**  
in  
**Information Technology**

by

**Rushil Patel**  
**Ameya Ranade**  
**Mihir Nikam**

under the guidance of

**Prof. Varsha Hole**



**Department of Information Technology**  
Bharatiya Vidya Bhavan's  
Sardar Patel Institute of Technology  
Munshi Nagar, Andheri-West, Mumbai-400058  
University of Mumbai  
May 2023

# Abstract

In recent years, Speech Emotion Recognition (SER) has emerged as a challenging yet significant field, with the potential to revolutionize various sectors, including healthcare, marketing, and customer service. However, accurately recognizing the emotional state of a speaker becomes a daunting task in the presence of multiple speakers. This is where Speaker Diarization (SD) plays a crucial role in separating the speech of each individual to enhance the performance of SER systems. By isolating speakers' speech, it becomes possible to accurately recognize their emotions, leading to better outcomes and higher accuracy.

One such sector that can greatly benefit from the use of SER and SD is customer care centers. Evaluating customer satisfaction levels and identifying the reasons for dissatisfaction without proper analysis of customer interactions can be a daunting task. The proposed system aims to accurately recognize emotions and segment audio into different speaker segments, enabling customer care centers to derive meaningful insights from customer care calls.

The project's primary objective is to perform SER and SD on the audio data, segment it into different speaker segments, and derive conclusions on the customer care call. By identifying the speaker's emotions and segmenting the audio based on the speaker, the system can provide a better understanding of customer satisfaction levels and reasons for dissatisfaction.

The project's outcome can be used by customer care centers to improve their services and address customer concerns effectively. The SER and SD system can provide actionable insights to the management team, leading to better customer satisfaction levels and improved customer retention rates. Overall, the proposed system's potential to enhance customer care services is significant, making it an essential tool for companies to stay ahead of their competition.

# Contents

<b>1</b>	<b>Literature Survey</b>	<b>4</b>
1.1	Transfer Learning for Improving Speech Emotion Classification Accuracy . . . .	4
1.2	Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions . . . . .	5
1.3	Speech Emotion Recognition Using Multi-hop Attention Mechanism . . . . .	5
1.4	Multimodal Speech Emotion Recognition and Ambiguity Resolution . . . . .	6
1.5	Compact Graph Architecture for Speech Emotion Recognition . . . . .	7
1.6	Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings . . . . .	7
1.7	Speech Emotion Recognition Using CNN . . . . .	8
1.8	Speech Emotion Recognition Using Deep Learning Techniques: A Review . . . .	9
1.9	Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching . . . . .	9
1.10	Automatic Speech Emotion Recognition Using Machine Learning . . . . .	10
1.11	Speech emotion recognition based on deep belief network . . . . .	10
1.12	Speech based Emotion Recognition using Machine Learning . . . . .	11
<b>2</b>	<b>Problem definition</b>	<b>12</b>
<b>3</b>	<b>Objectives of the Project</b>	<b>12</b>
<b>4</b>	<b>Dataset</b>	<b>14</b>
4.1	CREMA-D . . . . .	14
4.2	RAVDESS . . . . .	14
4.3	SAVEE . . . . .	15
4.4	TESS . . . . .	15
4.5	EmoV-DB . . . . .	15
4.6	JL-Corpus . . . . .	16
4.7	Data merging . . . . .	16
4.8	Data Cleaning . . . . .	17
4.9	Final Data Overview . . . . .	19
<b>5</b>	<b>Methodology</b>	<b>20</b>
5.1	Feature extraction module . . . . .	20
5.1.1	MFCCs . . . . .	20
5.2	Model Implementation . . . . .	22
5.2.1	Random forest model . . . . .	22
5.2.2	Support vector machine . . . . .	24
5.2.3	Convolutional Neural Network . . . . .	28
5.3	Speaker Diarization . . . . .	31
<b>6</b>	<b>Project Plan and Timeline</b>	<b>34</b>

<b>7</b>	<b>Implementation</b>	<b>35</b>
7.1	Speaker Diarizer . . . . .	35
7.2	Segmenting first audio file . . . . .	36
7.3	Segmenting second audio file . . . . .	37
7.4	Detecting emotion on first audio file using Web-Based UI . . . . .	38
<b>8</b>	<b>Conclusion</b>	<b>42</b>

# 1 Literature Survey

The literature survey section serves as a critical component in establishing the foundation and understanding of existing research and developments in the field. In this section, we provide a comprehensive overview and analysis of 12 selected papers.

## 1.1 Transfer Learning for Improving Speech Emotion Classification Accuracy

This research work addresses the challenge of speech emotion classification, where the goal is to automatically recognize emotions from speech signals. The paper focuses on the use of transfer learning to improve the accuracy of speech emotion classification. Transfer learning is a technique where a pre-trained model is used as a starting point for a new task, and then fine-tuned on the new task using a small amount of task-specific data.

The authors present an experimental study where they compare the performance of different transfer learning approaches for speech emotion classification. They use a pre-trained convolutional neural network (CNN) as the starting point and fine-tune it on two different datasets of speech emotion recordings. They compare the performance of this approach with that of training a CNN from scratch on the same datasets.

The results show that transfer learning significantly improves the accuracy of speech emotion classification, especially when the amount of labeled data is limited. The paper also provides insights into the factors that influence the effectiveness of transfer learning for speech emotion classification, such as the choice of pre-trained model, the amount of labeled data, and the similarity between the source and target domains.

A few limitations that are worth noting:

- Limited scope of the experimental study: The study presented in the paper is limited to comparing the performance of a pre-trained CNN with that of a CNN trained from scratch on two datasets of speech emotion recordings. While the results show that transfer learning improves the accuracy of speech emotion classification, the study could have been more comprehensive by exploring the performance of transfer learning on a wider range of datasets, with varying amounts of labeled data, and using different pre-trained models.
- Lack of detailed analysis of the impact of pre-training on the learned representations: The paper mentions that pre-training can improve the effectiveness of transfer learning by learning task-agnostic features that can be reused for the new task. However, the paper does not provide a detailed analysis of the impact of pre-training on the learned representations. Understanding how pre-training affects the learned representations can provide insights into the generalization capabilities of the pre-trained model and how it can be adapted to different tasks.

## 1.2 Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions

This research work presents a deep learning-based approach for speech emotion recognition that combines both acoustic features and transcriptions of speech. The paper highlights the importance of emotion recognition in various applications, such as healthcare, education, and entertainment, and the challenges associated with it. The authors propose a multi-modal approach that integrates acoustic features extracted from speech signals and textual features derived from transcriptions of speech.

The architecture of their deep learning-based emotion recognition system consists of a convolutional neural network (CNN) for processing acoustic features and a recurrent neural network (RNN) for processing textual features. The outputs of the two networks are combined using a fusion layer to generate the final emotion recognition result.

The paper also presents an experimental study where the proposed approach is evaluated on two publicly available datasets of speech emotion recordings. The results show that the multi-modal approach outperforms the traditional unimodal approach, where only acoustic features or transcriptions are used.

A few limitations that are worth noting:

- The study is limited to using only two datasets of speech emotion recordings. Although the authors report promising results on these datasets, the generalizability of the proposed approach to other datasets is not thoroughly investigated.
- Furthermore, the proposed approach only uses acoustic features and transcriptions of speech as modalities, whereas other modalities such as visual and physiological signals can also provide useful information for speech emotion recognition. The authors acknowledge this limitation and suggest that incorporating additional modalities could further improve the performance of the proposed approach.

## 1.3 Speech Emotion Recognition Using Multi-hop Attention Mechanism

The research paper proposes a novel multi-hop attention mechanism for speech emotion recognition. The proposed approach uses a two-stage approach to first extract relevant features from the input speech signal and then apply the multi-hop attention mechanism to the extracted features.

The paper presents a thorough literature review of the state-of-the-art approaches for speech emotion recognition, highlighting their limitations and challenges. The authors also discuss the use of attention mechanisms in natural language processing and their potential application to speech emotion recognition.

The proposed multi-hop attention mechanism consists of two parts: the intra-hop attention and the inter-hop attention. The intra-hop attention is applied to each feature vector independently to focus on the most informative parts of the feature vector. The inter-hop attention is then applied to the output of the intra-hop attention to further refine the feature representation.

The proposed approach is evaluated on two publicly available datasets, the IEMOCAP and SAVEE datasets, and compared with several state-of-the-art approaches. The experimental results demonstrate that the proposed approach outperforms the existing methods in terms of accuracy, F1-score, and other evaluation metrics.

A few limitations that are worth noting:

- The paper is that the proposed approach does not consider the temporal dynamics of the speech signal. The multi-hop attention mechanism only focuses on the most informative parts of the feature vector at each hop, without taking into account the temporal dependencies between the feature vectors. This may limit the ability of the proposed approach to capture long-term patterns and variations in the speech signal that are important for emotion recognition. Therefore, future work could explore incorporating temporal attention mechanisms or recurrent neural networks to address this limitation and further improve the performance of speech emotion recognition.

## 1.4 Multimodal Speech Emotion Recognition and Ambiguity Resolution

The research paper proposes a multimodal approach for speech emotion recognition that combines acoustic and visual features. The proposed approach uses a deep neural network to jointly learn the representations of acoustic and visual features and then applies a decision-level fusion to combine the two modalities.

The paper presents a literature survey of the state-of-the-art approaches for multimodal emotion recognition, highlighting the importance of using multiple modalities and the challenges in integrating them. The authors also discuss the use of deep neural networks and their potential for multimodal emotion recognition.

The proposed approach includes a novel ambiguity resolution mechanism that aims to resolve inconsistencies between the emotion labels predicted by the acoustic and visual modalities. The ambiguity resolution mechanism uses a set of rules and heuristics to identify and correct misclassifications based on the contextual information and the consistency between the predicted labels.

The proposed approach is evaluated on the RECOLA dataset, which includes both acoustic and visual recordings of emotional speech, and compared with several state-of-the-art approaches. The experimental results demonstrate that the proposed approach outperforms the existing methods in terms of accuracy, F1-score, and other evaluation metrics.

The paper concludes that the proposed multimodal approach with ambiguity resolution mechanism is effective in improving the performance of speech emotion recognition, and can potentially be applied to other multimodal emotion recognition tasks. The ambiguity resolution mechanism can also provide insights into the consistency and reliability of the predictions from different modalities.

A few limitations that are worth noting:

- One limitation of the paper is that the proposed ambiguity resolution mechanism relies on a set of rules and heuristics that may not generalize well to other datasets or scenarios.

The rules and heuristics are based on specific assumptions about the consistency between the acoustic and visual modalities, and may not hold true in other contexts. Therefore, the proposed approach may need to be adapted or fine-tuned for different datasets or scenarios, which may require additional manual effort and expertise.

## 1.5 Compact Graph Architecture for Speech Emotion Recognition

The research paper proposes a novel compact graph architecture for speech emotion recognition that reduces the number of parameters while maintaining high performance. The proposed approach uses a graph convolutional network to extract features from the speech signal and then applies a graph attention mechanism to select the most informative nodes in the graph.

The proposed compact graph architecture consists of multiple graph convolutional layers that operate on a graph representation of the speech signal. The graph attention mechanism is applied to the output of the last graph convolutional layer to focus on the most informative nodes in the graph. The resulting feature representation is then fed into a fully connected layer for emotion classification.

The proposed approach is evaluated on the IEMOCAP dataset and compared with several state-of-the-art approaches. The experimental results demonstrate that the proposed approach achieves comparable or better performance than the existing methods while using significantly fewer parameters. The paper also presents an ablation study to analyze the impact of different components of the proposed approach on the performance.

The paper concludes that the proposed compact graph architecture is effective in reducing the number of parameters and improving the efficiency of speech emotion recognition, without compromising the performance. The proposed approach can also be extended to other speech-related tasks that require graph-based representations.

A few limitations that are worth noting:

- One limitation of the paper is that the proposed approach only uses a single modality (i.e., acoustic features) for speech emotion recognition, and does not consider other modalities such as visual or textual information. While the proposed compact graph architecture is effective in reducing the number of parameters and improving the efficiency of acoustic-based emotion recognition, it may not generalize well to multimodal scenarios where multiple modalities are available. Therefore, future work could explore the integration of the proposed compact graph architecture with other modalities and investigate the potential benefits and challenges of such integration.

## 1.6 Emotion Recognition from Speech Using Wav2vec 2.0 Embeddings

The research paper proposes an approach for speech emotion recognition using Wav2vec 2.0 embeddings, a recently introduced pre-training technique for speech processing. The proposed approach uses a simple feedforward neural network to classify the emotion of a given speech signal based on the learned embeddings.



The proposed approach consists of two stages: feature extraction and emotion classification. The feature extraction stage uses the Wav2vec 2.0 pre-trained model to generate embeddings from the input speech signal. The emotion classification stage uses a feedforward neural network to predict the emotion label based on the learned embeddings. The proposed approach is evaluated on the IEMOCAP dataset and compared with several state-of-the-art approaches. The experimental results demonstrate that the proposed approach achieves comparable or better performance than the existing methods while using a smaller number of parameters.

The paper concludes that the proposed approach is effective in utilizing the pre-trained embeddings for speech emotion recognition and can potentially be applied to other speech-related tasks. The simplicity and efficiency of the proposed approach make it suitable for real-world applications where computational resources and time are limited. However, the paper did not explore the potential limitations and challenges of the proposed approach, such as the generalization to other datasets or the robustness to noisy speech signals. Further research is needed to investigate these aspects and extend the proposed approach to multimodal scenarios.

A few limitations that are worth noting:

- One limitation of the paper is that the proposed approach only uses a single modality (i.e., acoustic features) for speech emotion recognition, and does not consider other modalities such as visual or textual information. While the proposed approach using Wav2vec 2.0 embeddings is effective in utilizing pre-trained models for acoustic-based emotion recognition, it may not generalize well to multimodal scenarios where multiple modalities are available. Therefore, future work could explore the integration of the proposed approach with other modalities and investigate the potential benefits and challenges of such integration.

## 1.7 Speech Emotion Recognition Using CNN

The research paper by Murugan titled "Speech Emotion Recognition Using CNN" proposes a novel approach for automatic recognition of emotions from speech using Convolutional Neural Networks (CNN). The paper begins with a brief introduction of the importance of speech emotion recognition and its applications in various fields such as psychology, robotics, and human-computer interaction.

The paper then provides a comprehensive literature review of the previous works in speech emotion recognition and identifies the limitations of the existing approaches. The author suggests that deep learning techniques such as CNN can overcome these limitations by automatically learning the features from the raw speech signals without the need for handcrafted feature extraction.

Next, the paper describes the proposed CNN architecture and the dataset used for training and testing the model. The dataset used is the Berlin Emotional Speech Database which contains speech samples of 10 different emotions expressed by 10 different actors. The paper then presents the experimental results of the proposed approach and compares it with the state-of-the-art approaches in speech emotion recognition.

The results show that the proposed CNN model achieved an accuracy of 88.67% in recognizing emotions from speech, which is higher than the other state-of-the-art approaches. The paper

concludes by discussing the significance of the proposed approach and its potential applications in various fields.

## **1.8 Speech Emotion Recognition Using Deep Learning Techniques: A Review**

The research paper titled "Speech Emotion Recognition Using Deep Learning Techniques: A Review" by Khalil et al. provides a comprehensive literature review of the existing approaches for automatic speech emotion recognition using deep learning techniques. The paper begins with a brief introduction of the importance of speech emotion recognition and its applications in various fields such as healthcare, education, and entertainment.

The paper then reviews the various deep learning techniques that have been used for speech emotion recognition, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Deep Belief Networks (DBNs). For each technique, the paper describes the architecture, advantages, and limitations.

Next, the paper discusses the different features that have been used for speech emotion recognition, including Mel-frequency Cepstral Coefficients (MFCCs), Prosodic Features, and Spectral Features. The paper provides a detailed description of each feature and its importance in speech emotion recognition.

The paper then reviews the various datasets that have been used for training and testing the speech emotion recognition systems, including the Emo-DB, SAVEE, and RAVDESS datasets. The paper also highlights the challenges associated with the datasets, such as the lack of diversity in emotions and the variability in speech samples.

Finally, the paper summarizes the key findings of the review and identifies the future research directions in the field of speech emotion recognition. The paper emphasizes the need for developing robust and accurate speech emotion recognition systems that can work in real-world scenarios and can be used in a wide range of applications.

## **1.9 Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching**

The research paper titled "Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching" by Zhang et al. proposes a novel approach for automatic speech emotion recognition using a Deep Convolutional Neural Network (DCNN) and Discriminant Temporal Pyramid Matching (DTPM).

The paper begins with a brief introduction of the importance of speech emotion recognition and its applications in various fields such as healthcare, education, and entertainment. The paper then reviews the existing approaches for speech emotion recognition and their limitations.

Next, the paper describes the proposed approach, which consists of two main components: a DCNN-based feature extraction module and a DTPM-based classification module. The feature extraction module uses a DCNN to automatically extract discriminative features from the input speech signals. The classification module uses a DTPM algorithm to match the extracted features with the emotions in the training data.

The paper presents experimental results on three standard datasets for speech emotion recognition, namely the Berlin Emotional Speech Database, the Emotional Prosody Speech and Transcripts dataset, and the Interactive Emotional Dyadic Motion Capture dataset. The results show that the proposed approach outperforms the state-of-the-art approaches in terms of accuracy, F1 score, and other evaluation metrics.

Finally, the paper concludes by discussing the significance of the proposed approach and its potential applications in various fields. The paper emphasizes the need for developing more accurate and robust speech emotion recognition systems that can work in real-world scenarios and can be used in a wide range of applications.

## **1.10 Automatic Speech Emotion Recognition Using Machine Learning**

The research paper titled "Automatic Speech Emotion Recognition Using Machine Learning" by Kerkeni et al. focuses on the use of machine learning techniques for speech emotion recognition. The paper starts by introducing the importance of recognizing emotions in speech and its various applications in different fields.

Next, the paper describes the different feature extraction techniques used for speech emotion recognition such as Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC), and Prosodic features. The paper also discusses the importance of selecting the right features for emotion recognition.

The paper then reviews the different machine learning algorithms used for speech emotion recognition such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN). The paper also describes the different steps involved in building a machine learning-based emotion recognition system, including data pre-processing, feature extraction, and model training and evaluation.

The paper presents a case study on speech emotion recognition using the Berlin Emotional Speech Database, where the authors compare the performance of different machine learning algorithms for emotion recognition. The results show that the SVM algorithm with MFCC features outperforms other algorithms in terms of accuracy and efficiency.

The paper concludes by discussing the limitations and future directions in speech emotion recognition using machine learning techniques. The paper suggests integrating multimodal features such as facial expressions and physiological signals to improve the accuracy and robustness of the emotion recognition system. The paper also emphasizes the need for developing emotion recognition systems that can work in real-world scenarios and can be used in a wide range of applications.

## **1.11 Speech emotion recognition based on deep belief network**

The research paper titled "Speech emotion recognition based on deep belief network" by P. Shi focuses on using deep belief networks (DBN) for speech emotion recognition. The paper starts by introducing the importance of recognizing emotions in speech and the challenges involved in automatic emotion recognition.

Next, the paper describes the different feature extraction techniques used for speech emotion recognition such as Mel Frequency Cepstral Coefficients (MFCC) and their derivatives, and the use of delta and delta-delta features. The paper also discusses the importance of selecting the right features for emotion recognition.

The paper then presents a detailed description of the deep belief network and its architecture, which is used for speech emotion recognition. The paper explains how the DBN is trained using unsupervised learning, and then fine-tuned using supervised learning for emotion recognition.

The paper presents a case study on speech emotion recognition using the Emotional Prosody Speech and Transcripts dataset, where the authors compare the performance of the DBN-based emotion recognition system with other machine learning algorithms. The results show that the DBN-based system outperforms other algorithms in terms of accuracy and robustness.

The paper concludes by discussing the limitations and future directions in speech emotion recognition using deep learning techniques. The paper suggests exploring the use of different deep learning architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for speech emotion recognition. The paper also emphasizes the need for developing emotion recognition systems that can work in real-world scenarios and can be used in a wide range of applications.

## **1.12 Speech based Emotion Recognition using Machine Learning**

The research paper titled "Speech based Emotion Recognition using Machine Learning" by Deshmukh, Girija, Gaonkar, Apurva, Golwalkar, Gauri, and Kulkarni, Sukanya, discusses the use of machine learning techniques for speech-based emotion recognition. The paper focuses on exploring the performance of different machine learning algorithms, namely SVM, KNN, and Random Forest, for recognizing emotions in speech signals.

The authors conducted experiments on two datasets: the Berlin Emotional Speech database and the RAVDESS dataset. The Berlin Emotional Speech database consists of recordings of actors expressing different emotions, while the RAVDESS dataset contains speech recordings of actors simulating different emotions. The authors used Mel Frequency Cepstral Coefficients (MFCCs) as features and trained the machine learning algorithms on these features.

The results of the experiments showed that the SVM algorithm outperformed the other two algorithms for both datasets. The authors also discussed the impact of different features, such as pitch, intensity, and formants, on emotion recognition accuracy. Additionally, the authors highlighted the importance of having a balanced dataset for training the models.

Overall, the research paper demonstrates the potential of using machine learning algorithms for speech-based emotion recognition and provides insights into the performance of different algorithms and features.

## 2 Problem definition

The presence of multiple speakers in speech emotion recognition (SER) makes it challenging to accurately identify a speaker's emotional state, which limits the efficacy of the technology. This presents a significant challenge for various industries, including healthcare, marketing, and customer service, where the ability to recognize emotions is crucial. Customer care centers face the challenge of assessing customer satisfaction levels and identifying the reasons for dissatisfaction, making it challenging to address customer concerns effectively. To address these challenges, there is a need for a system that can accurately recognize emotions and segment audio into different speaker segments, allowing customer care centers to derive meaningful insights from customer care calls. Therefore, the problem statement is to develop a system that performs SER and Speaker Diarization (SD) to segment audio into different speaker segments and accurately recognize the speaker's emotional state to derive conclusions on customer care calls, leading to improved customer satisfaction levels and retention rates.

## 3 Objectives of the Project

In this section of the report, we discuss the objectives of the project. The main goal is to develop and train a Speech Emotion Recognition (SER) and Speaker Diarization (SD) system that accurately identifies emotions and segments audio into different speaker segments.

- Develop and train a SER and SD system that accurately identifies emotions and segments audio into different speaker segments.
- Evaluate the performance of the SER and SD system using a customer care dataset, and compare the results to existing manual annotation methods.
- Identify the most commonly occurring emotions in customer care calls, and determine how they relate to customer satisfaction levels.
- Analyze the differences in emotional responses between customers and customer service representatives, and identify potential areas for improvement in employee training.
- Develop a user-friendly interface for the SER and SD system, allowing customer care center managers to access and analyze call data easily.
- Provide recommendations to customer care centers based on the insights derived from the SER and SD analysis, such as training recommendations, policy changes, and resource allocation.

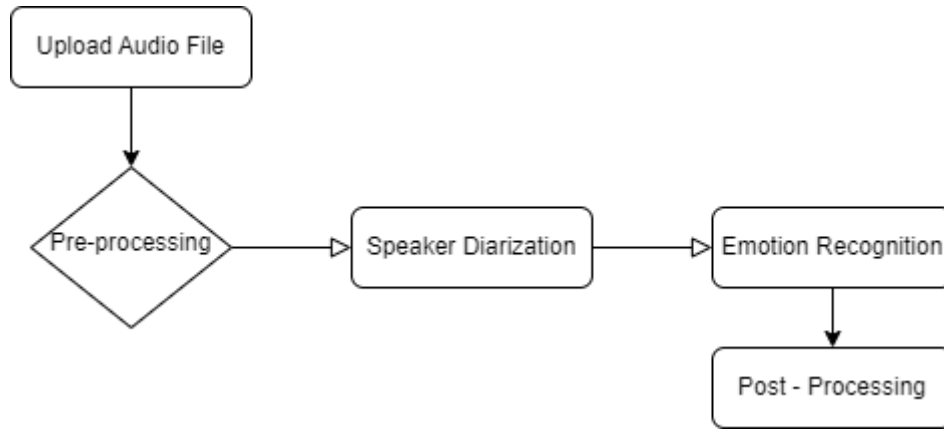


Figure 1: General Flow of Application

A basic flow diagram to visualise the objectives of the project is given in Fig. 1

- Upload audio File
  - Obtain raw audio file
  - Ensure file is in a compatible format (e.g., WAV, MP3)
- Pre-processing
  - Ensure volume levels are consistent
  - Remove background noise if necessary
- Speaker Diarization
  - Segment audio into speaker turns
  - Extract features from the audio
  - Cluster speaker turns
  - Identify and label speakers - Assign unique speaker IDs
- Emotion Recognition
  - Process and analyze each speaker turn
  - Extract relevant features (e.g., pitch, intensity, MFCCs etc)
  - Classify emotions using our machine learning models
  - Assign emotion labels to each speaker turn (probabilities)
  - Determine the most dominant emotion
- Post-Processing
  - Generate output data - Create a list of speaker turns with associated emotions
  - Export data in a desired format (e.g., CSV, JSON)

## 4 Dataset

We have collected several datasets for emotion classification, each consisting of small English sentences and corresponding audio samples organized in folders with accompanying documentation. The metadata for these datasets is embedded in the file names, but written using different keys, which required individual extraction for each set.

To streamline the process of merging these datasets, we have catalogued them and created dataframes with a consistent structure. This will enable us to easily merge the datasets later on. By organizing the datasets in this way, we can ensure that the data is readily available for analysis and processing, and that we can make the most of the information contained in each dataset.

The information retrieved is path, filename, dataset, sample\_rate, gender, age, emotion.

### 4.1 CREMA-D

The **Continuous Emotional Recognition Evaluation in Multimedia Database** dataset is a publicly available dataset that contains audio and video recordings of actors portraying a wide range of emotions. The dataset was created to provide a benchmark for researchers working on emotion recognition from audio and visual cues. The dataset contains 7,442 clips, each of which is about 5 seconds long, and features 91 actors (48 male, 43 female) portraying 12 different emotions (anger, contempt, disgust, fear, happiness, neutral, pride, sadness, satisfaction, surprise, embarrassment, and positive/negative valence). Each clip is labeled with the corresponding emotion, as well as other metadata such as the actor’s gender, age, and ethnicity. The audio recordings are provided in both WAV and MP3 formats, and the video recordings are provided in AVI format. The dataset also includes a set of features extracted from the audio and video signals, such as spectral and prosodic features for audio, and facial action units and head pose features for video. The CREMA-D dataset has been used in numerous studies on emotion recognition and has become a popular benchmark for evaluating emotion recognition algorithms.

### 4.2 RAVDESS

The **RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)** dataset is a publicly available dataset that contains audio and video recordings of actors performing a variety of emotional states. The dataset was created to provide a benchmark for researchers working on emotion recognition from speech and song.

The dataset contains 1,440 clips, each of which is about 3-5 seconds long, and features 24 professional actors (12 male, 12 female) portraying 8 different emotions (calm, happy, sad, angry, fearful, surprise, disgust, and neutral) in both speech and song. Each clip is labeled with the corresponding emotion, as well as other metadata such as the actor’s gender, age, and ethnicity.

The audio recordings are provided in both WAV and MP3 formats, and the video recordings are provided in MP4 format. The dataset also includes a set of features extracted from the audio and video signals, such as Mel Frequency Cepstral Coefficients (MFCCs) and chroma features for audio, and facial landmarks and action units for video.

The RAVDESS dataset has been used in a wide range of studies on emotion recognition and has become a popular benchmark for evaluating emotion recognition algorithms, particularly for speech and song.

### 4.3 SAVEE

The **SAVEE (Surrey Audio-Visual Expressed Emotion)** dataset is a publicly available dataset that contains audio and video recordings of actors performing different emotional states. The dataset was created to provide a benchmark for researchers working on emotion recognition from speech and audio.

The dataset contains 480 audio recordings, each of which is about 4 seconds long, and features 4 male actors portraying 7 different emotions (anger, disgust, fear, happiness, sadness, surprise, and neutral). Each clip is labeled with the corresponding emotion and the name of the actor who performed it.

The audio recordings are provided in WAV format, and the dataset also includes a set of features extracted from the audio signals, such as Mel Frequency Cepstral Coefficients (MFCCs) and energy and spectral features.

The SAVEE dataset has been used in various studies on emotion recognition, and it is particularly useful for evaluating algorithms that can recognize emotions from speech. However, it should be noted that the dataset is relatively small compared to other similar datasets, such as the RAVDESS and the CREMA-D datasets.

### 4.4 TESS

The **TESS (Toronto Emotional Speech Set)** dataset is a publicly available dataset that contains audio recordings of actors performing various emotional states. The dataset was created to provide a benchmark for researchers working on emotion recognition from speech and audio.

The dataset contains 2,940 audio recordings, each of which is about 3 seconds long, and features 2 female actors portraying 7 different emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral) in two different dialects of North American English. Each clip is labeled with the corresponding emotion and the name of the actor who performed it.

The audio recordings are provided in WAV format, and the dataset also includes a set of features extracted from the audio signals, such as Mel Frequency Cepstral Coefficients (MFCCs), prosodic features, and spectral features.

The TESS dataset is smaller compared to some other similar datasets, such as the RAVDESS and the CREMA-D datasets. However, it is still useful for evaluating algorithms that can recognize emotions from speech, particularly in the context of North American English. The dataset has been used in various studies on emotion recognition, and it is freely available for research purposes.

### 4.5 EmoV-DB

The **Emotional Voices Database** includes recordings for four speakers - two males and two females. The emotional styles are neutral, sleepiness, anger, disgust and amused. Each audio



file is recorded in 16bits .wav format.

## 4.6 JL-Corpus

The **JL-Corpus (Jenaer Lautsprecherdaten Corpus)** is a German language speech database that is commonly used for research on speech recognition, speech synthesis, and voice conversion. The dataset was recorded by the Institute of Computer Science at the Friedrich Schiller University in Jena, Germany.

The dataset contains recordings of 10 male and 10 female German speakers, aged between 18 and 35, who read a set of standardized sentences in a soundproof booth. The sentences were selected to cover a wide range of phonetic and prosodic features. In total, the dataset contains about 10 hours of speech data, with each speaker contributing about 30 minutes of recorded speech.

The audio recordings are provided in WAV format, with a sampling rate of 44.1 kHz and a bit depth of 16 bits. The dataset also includes a set of manually verified transcripts for each recording, which can be used for speech recognition and transcription tasks.

The JL-Corpus has been used in a variety of research projects, including speech recognition, speaker recognition, emotion recognition, and voice conversion. It is a valuable resource for researchers working in the field of speech technology, particularly for those focused on the German language.

## 4.7 Data merging

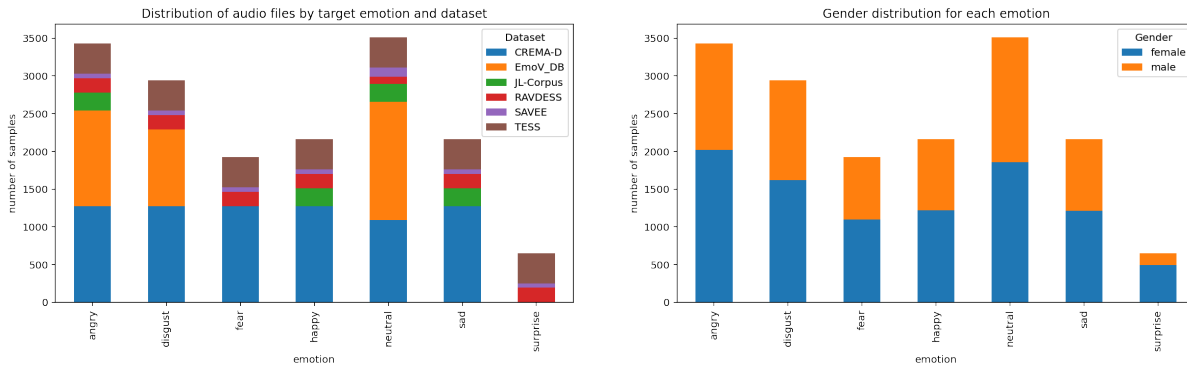


Figure 2: Merging datasets

We merge the datasets together, now that they have been formatted the same way. The dataset has 16783 audio files.

Checking the sources, some inequalities in the amount of samples are noted, with the surprise category having fewer audio files linked to its category than the other types. The distribution by gender is quite similar, with slightly more samples of female voices in general.

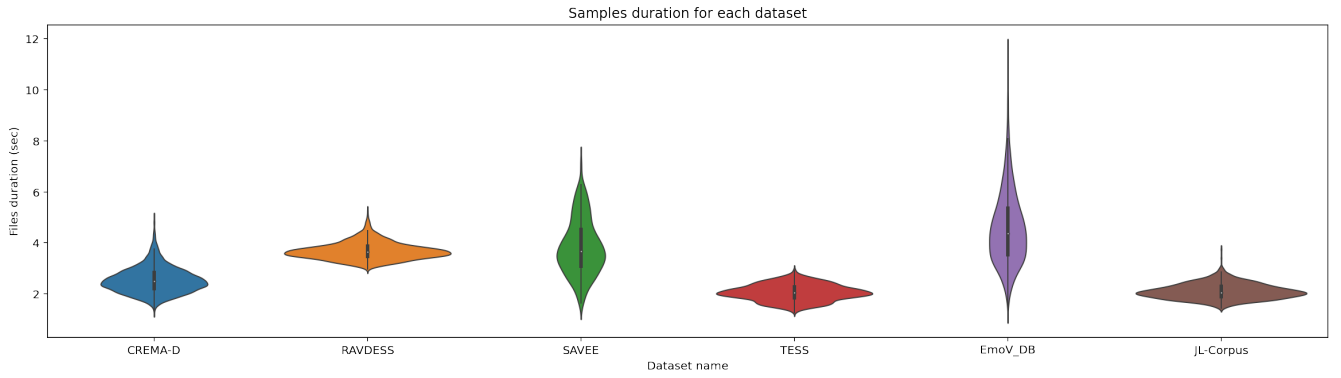


Figure 3: Sample durations

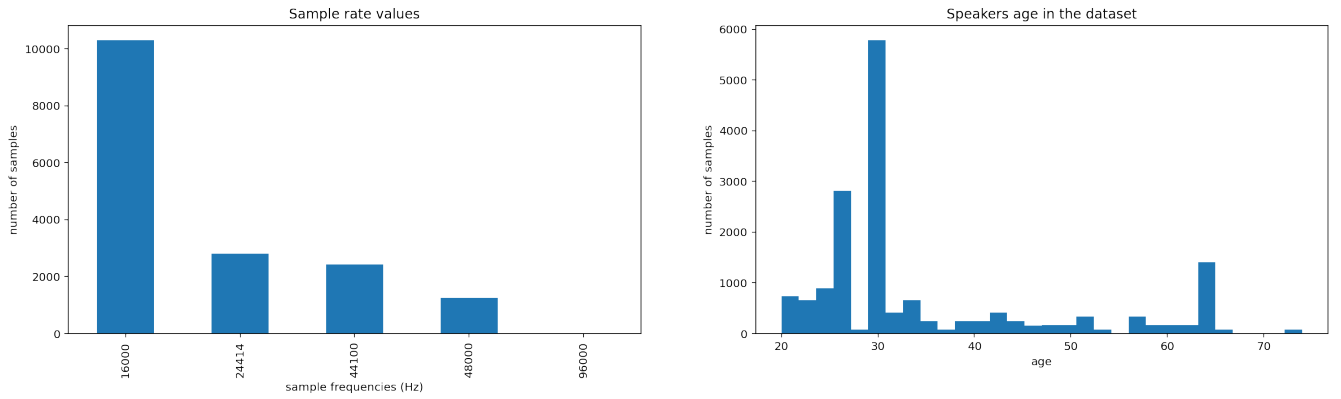


Figure 4: Data analysis

We noticed that the sampling frequency is not equal and needs to be addressed. The age of the speakers peaks at 30 years, but a lot of data is missing and has been replaced with average values from the different documentations.

## 4.8 Data Cleaning

The preliminary exploration of the data, the main technical problems with our files are

- The files have **different sample rates**. For the first problem we have resampled the files, and they will all be at 16000 Hz, which is the lowest frequency among the samples, the one that is most present in their total number, and an acceptable value to represent the human voice.
- All files tested **start and end with some silence**. We have implemented a trim function, which in effect cuts initial and final silence from an audio signal.
- **Some samples are noisier than others**. For the noise the problem is more difficult, since the noise removal operation also affects the quality of the features to be extracted.

For this reason we have to use the module `noisereduce` and apply a light reduction of 10% for the stationary noise on all the samples. The reason for such reduction is due to the fact that this operation can also sensibly lower the quality of the signal thus removing also meaningful information of the audio for the next steps.

- `Noisereduce` is a noise reduction algorithm in python that reduces noise in time-domain signals like speech, bioacoustics, and physiological signals. Spectral gating is a noise reduction technique that involves selectively removing noisy parts of a signal based on their spectral properties. In Python, the following steps can be taken to implement spectral gating for noise reduction:
  - Import the necessary libraries, such as `scipy` and `numpy`.
  - Load the audio signal to be processed using a suitable library, such as `librosa`.
  - Compute the short-time Fourier transform (STFT) of the audio signal using the `stft` function in `librosa`.
  - Calculate the magnitude spectrogram of the STFT using the `magphase` function in `librosa`.
  - Calculate the median value of the magnitude spectrogram along the time axis, which represents the noise floor of the signal.
  - Set a threshold value to determine which parts of the signal should be removed based on their magnitude relative to the noise floor. A common threshold value is 3 times the noise floor.
  - Create a binary mask that indicates which parts of the spectrogram should be removed based on their magnitude relative to the threshold.
  - Apply the binary mask to the magnitude spectrogram to selectively remove the noisy parts of the signal.
  - Reconstruct the audio signal from the modified magnitude spectrogram using the inverse STFT function, `istft`, in `librosa`.
  - Save the denoised audio signal to a file using a suitable library, such as `soundfile`.

## 4.9 Final Data Overview

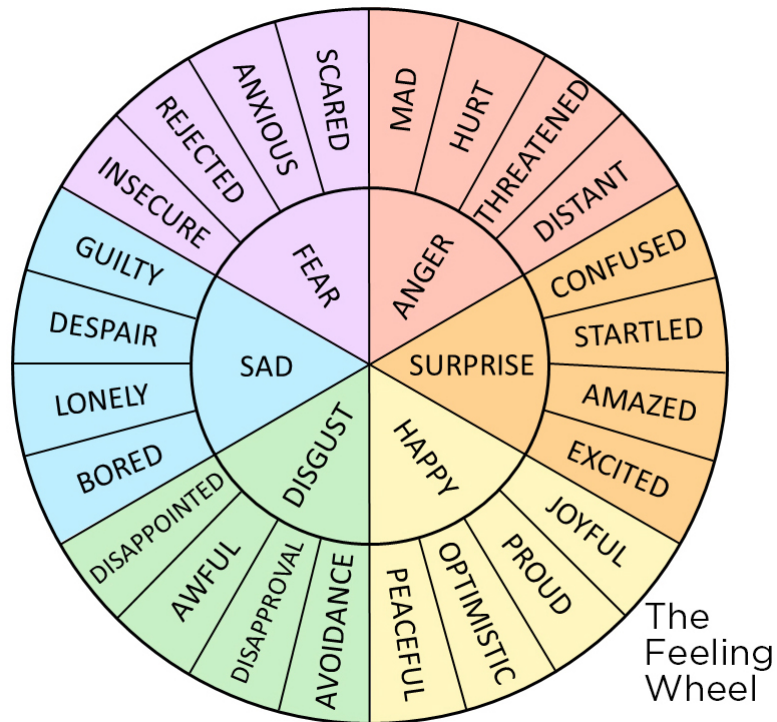


Figure 5: Emotion Classification

We have gathered and merged the six above discussed datasets together. Thus, we have around about 17000 audio labelled samples in total for our model training and testing purposes. The corpus consists of seven emotions over which the classification is done.

Emotions classification:

- Anger
- Disgust
- Sad
- Happy
- Surprise
- Fear
- Neutral

## 5 Methodology

In this section, we outline the approach and techniques used to achieve the objectives of the project. We describe the process of developing and training the three models that are used for Speech Emotion Recognition. Furthermore, we discuss the development of a user-friendly interface to visualize the insights derived from the SER and SD analysis.

### 5.1 Feature extraction module

In sound analysis, there is a lot of data that can be extracted from audio signals. For this task we extracted some information from the samples using librosa, and analyzed them in relation to the target feature. We obtained the mean and variance for each value. Let's see a brief description of the attributes chosen:

The analysis of audio signals can provide a wealth of information. In this project, we used librosa to extract various features from audio samples and analyzed them in relation to the target feature. We calculated the mean and variance for each value to gain a better understanding of the audio signals.

One of the features we extracted was the fundamental frequency ( $f_0$ ), which is the lowest frequency produced by the voice. We calculated the mean, median, standard deviation, min and max frequency, as well as the 25% and 75% percentile values of the fundamental frequency. We also looked at the zero-crossing rate (zcr), which detects percussive sounds, and the spectral centroid, which characterizes where the center of mass of the spectrum is located.

We also considered the spectral contrast, which measures the clarity versus the broad noise of each band in the spectrum, and the spectral flatness, which quantifies how much noise-like a sound is. Additionally, we examined the harmony, which measures harmonic elements from an audio time-series, and the Root-Mean-Square (rms), which measures the energy of the signal.

We further explored the Chroma Feature, which computes a chromagram from a waveform or power spectrogram, and Chroma Cqt, which computes the constant-Q chromagram. Moreover, we considered Chroma Cens, which computes the chroma variant "Chroma Energy Normalized" (CENS), and Rolloff, which is the fraction of bins in the power spectrum at which 85% of the power is at lower frequencies.

Finally, we used Mel-Frequency Cepstrum Components (MFCCs) to compute the coefficients of the Mel cepstrum, which is based on the Mel-bands scaled to the human ear. We obtained 60 features, including the mean and variance of each of the 30 extracted components. By analyzing these features, we gained valuable insights into the audio signals and were able to better understand the target feature.

#### 5.1.1 MFCCs

MFCC stands for Mel-Frequency Cepstral Coefficients. MFCCs are a commonly used feature extraction technique for speech and audio processing applications, including speech recognition, speaker identification, and music genre classification. MFCCs are derived from the spectral envelope of a signal, which represents the distribution of energy in different frequency bands over time. The process involves dividing the audio signal into short, overlapping frames, and computing the power spectrum of each frame using the Fourier transform. The power spectrum

is then passed through a filterbank that is designed to mimic the human auditory system's sensitivity to different frequency ranges. The output of the filterbank is then transformed using the logarithm to compress the dynamic range of the signal. Finally, a Discrete Cosine Transform (DCT) is applied to the logarithmically transformed filterbank output to obtain the MFCCs. The resulting MFCCs capture the essential spectral and temporal characteristics of the signal, providing a compact and effective representation of the audio signal for further analysis and processing. They are often used as input features for machine learning algorithms, such as neural networks, to perform speech and audio-related tasks.

- **Preemphasis:** In this step, a high-pass filter is applied to the audio signal to boost the high-frequency components and to reduce the low-frequency noise.
- **Framing:** The audio signal is divided into short, overlapping frames, typically 20-30 milliseconds in duration. **Windowing:** A window function, such as the Hamming window, is applied to each frame to reduce the spectral leakage caused by the abrupt transition at the edges of the frame.
- **Fourier transform:** The discrete Fourier transform (DFT) is applied to each windowed frame to obtain the power spectrum. **Mel-filterbank:** The power spectrum is passed through a series of triangular-shaped filters that are spaced in the mel-frequency scale, which is a perceptually relevant scale that approximates the human auditory system's frequency response.
- **Logarithmic compression:** The output of each filter is converted to a logarithmic scale to compress the dynamic range of the filterbank.
- **Discrete Cosine Transform (DCT):** Finally, the Discrete Cosine Transform (DCT) is applied to the log-filterbank energies, resulting in a set of coefficients that represent the spectral envelope of the signal in a compact and efficient way. The first few coefficients, known as the MFCCs, are typically used as features for speech recognition tasks.

MFCCs are widely used in speech processing tasks such as speaker identification, speech recognition, and emotion recognition. They are robust to noise and speaker variability and have been shown to be effective in various acoustic environments.

## 5.2 Model Implementation

### 5.2.1 Random forest model

Random Forest is a popular machine learning algorithm that is used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions. In Random Forest, multiple decision trees are created using a random subset of the features and a random sample of the training data. Each decision tree makes a prediction based on the feature subset it was trained on, and the final prediction is obtained by aggregating the predictions of all the decision trees. This process of aggregating the predictions is called ensemble learning. Random Forest has several advantages over traditional decision trees. Firstly, it can handle a large number of input features and provides a measure of feature importance. Secondly, it is robust to overfitting and can handle noisy data. Thirdly, it can provide an estimate of the generalization error, which indicates how well the model will perform on new, unseen data. Random Forest has numerous applications in various domains, including image and speech recognition, fraud detection, and healthcare. It is a powerful and versatile machine learning algorithm that is widely used in industry and academia due to its excellent performance and ease of use. Applied the random forest model, tuning the number of trees and their depth. To find the optimal hyperparameters we use cross validation grid search with KFold strategy, therefore the validation will be extracted from the training set.

#### Hyperparameter tuning

Random Forest is a powerful and flexible machine learning algorithm that can handle a wide range of datasets and classification or regression tasks. However, it has several hyperparameters that need to be optimized to obtain the best performance. Here are some common hyperparameters of Random Forest and their tuning techniques:

- `n_estimators`: This is the number of trees in the forest. A higher number of trees may lead to better performance, but it also increases the computational cost. The optimal value can be found using cross-validation or grid search techniques.
- `max_depth`: This is the maximum depth of each tree. A deeper tree can capture more complex patterns in the data, but it can also lead to overfitting. The optimal value can be found using cross-validation or grid search techniques.
- `min_samples_split`: This is the minimum number of samples required to split a node. A higher value can reduce overfitting, but it may also lead to underfitting. The optimal value can be found using cross-validation or grid search techniques.
- `max_features`: This is the number of features to consider when looking for the best split. A smaller value can reduce overfitting, but it may also reduce the model's ability to capture complex patterns in the data. The optimal value can be found using cross-validation or grid search techniques.

Overall, Random Forest hyperparameter tuning is an important step in building an accurate and robust model. It involves selecting the optimal values of the hyperparameters

by evaluating the model’s performance on a validation set or using cross-validation techniques. By tuning the hyperparameters, we can achieve the best performance of the Random Forest model on the new and unseen data.

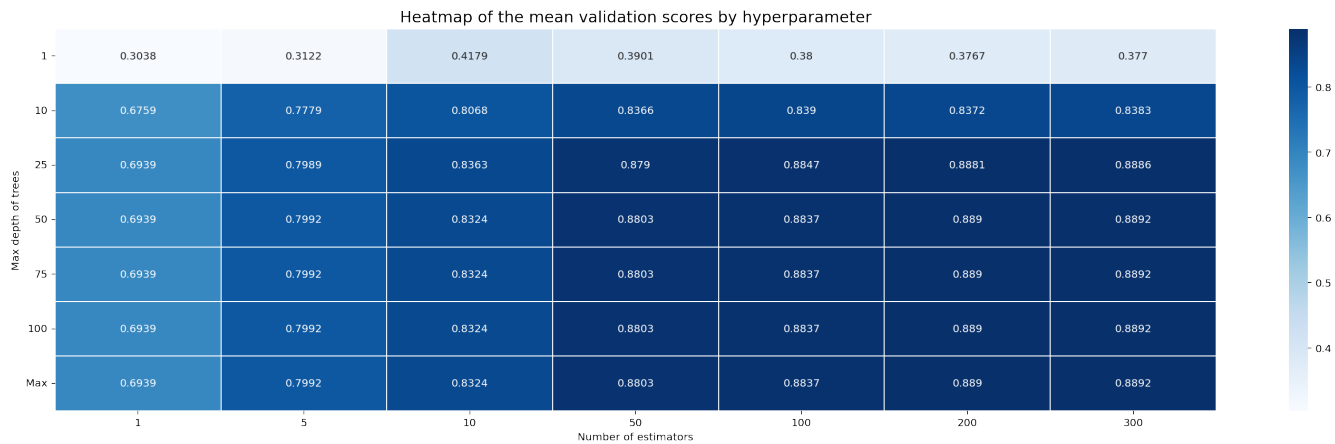


Figure 6: Random Forest Hyperparameter Tuning Heatmap

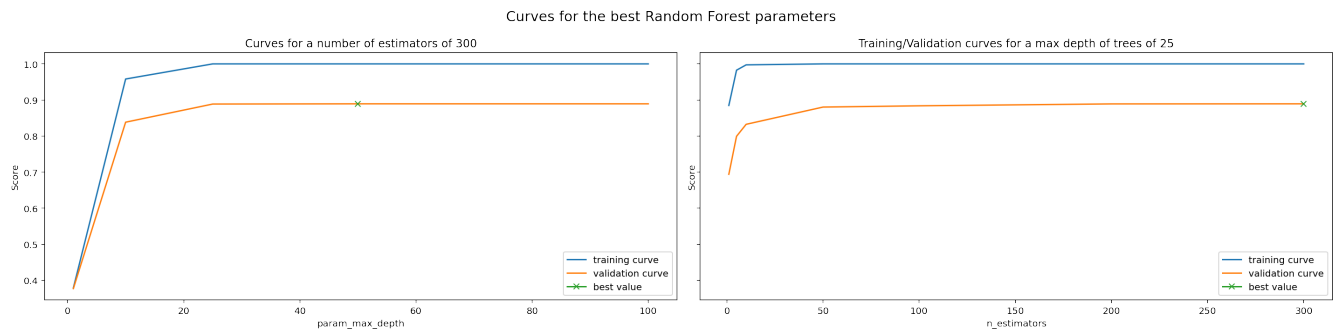


Figure 7: Random Forest Hyperparameter Tuning Graphs



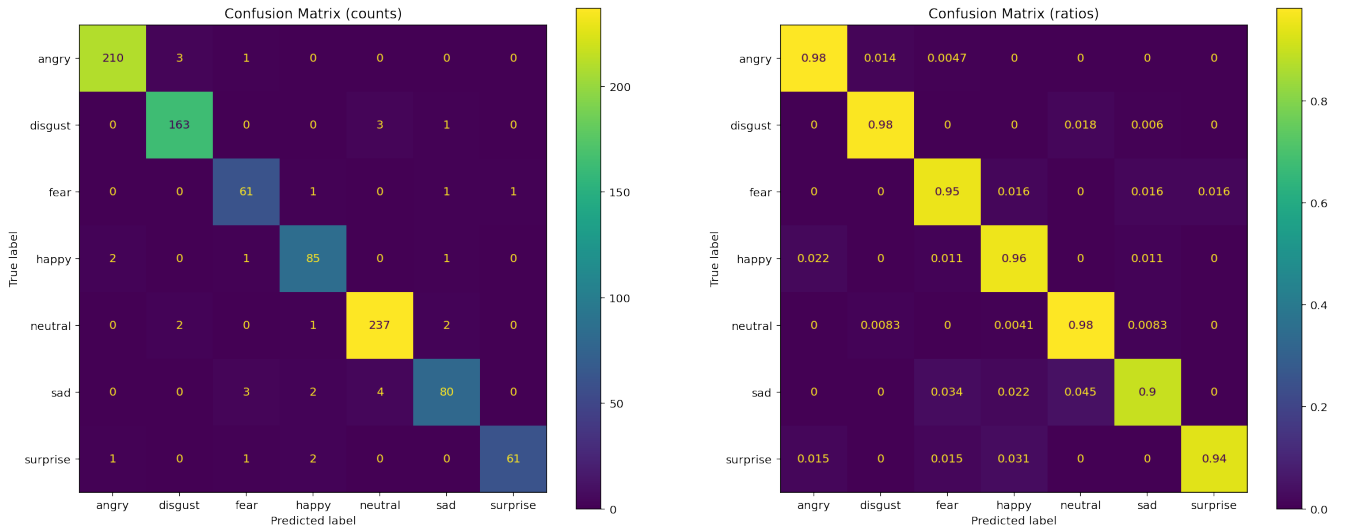


Figure 8: Random Forest Confusion Matrix

It can be seen that increasing the number of trees helps, as at each depth the best score always has the highest value, 300. The heatmap also suggests that increasing the depth of trees seems to improve the overall model but we see that after a value of 50 the score doesn't improve.

We can also infer from these curves that increasing both hyperparameters improves the score, but we also clearly note that there is not an improvement after a few steps. We can see that the training curves quickly reach the maximum score of 100%, a sign of overfitting. This especially for the max\_depth parameters, as the validation score is already defined after a depth of 20 30 and even slightly decreases after the best value 50. Hyperparameter tuning outcomes:

- The best hyperparameters for this random forest model are 300 trees with a depth of 25.
- Test accuracy for the random forest model after the tuning: 0.912

From the confusion matrix we can see that the best emotions identified are angry, neutral and disgust, while happy and sad scored as the worst.

### 5.2.2 Support vector machine

Support Vector Machine (SVM) is a well-known machine learning algorithm utilized for both classification and regression tasks. The main concept behind SVM is to identify the optimal hyperplane that can effectively separate data points into different classes, with the margin between the two classes being maximized.

SVMs can be employed for datasets that are either linearly separable or non-linearly separable, with the latter being accomplished by using kernel functions to transform the data into a higher-dimensional space, where separation between data points becomes feasible.

SVMs find wide-ranging applications in various fields, such as image classification, text classification, and bioinformatics, among others. The key benefits of SVMs include their strong theoretical basis, proven practical effectiveness, capability to handle high-dimensional data, and resistance to overfitting.

## Hyperparameter tuning

SVM has several hyperparameters that need to be tuned to obtain the best performance. Here are some common hyperparameters of SVM and their tuning techniques:

- **C:** This is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error. A higher value of C can lead to overfitting, while a lower value can lead to underfitting. The optimal value can be found using cross-validation or grid search techniques.
- **Gamma:** This is the kernel parameter that controls the width of the Gaussian kernel. A higher value can lead to overfitting, while a lower value can lead to underfitting. The optimal value can be found using cross-validation or grid search techniques.
- **Kernel:** This is the kernel function used to transform the input data into a higher-dimensional space. Common kernel functions include linear, polynomial, and radial basis function (RBF). The optimal kernel can be selected using cross-validation or grid search techniques.
- **Class weight:** This is the parameter used to balance the class weights in imbalanced datasets. It assigns a higher weight to the minority class and a lower weight to the majority class. The optimal value can be found using cross-validation or grid search techniques.

Overall, SVM hyperparameter tuning is an important step in building an accurate and robust model. It involves selecting the optimal values of the hyperparameters by evaluating the model's performance on a validation set or using cross-validation techniques. By tuning the hyperparameters, we can achieve the best performance of the SVM model on the new and unseen data.

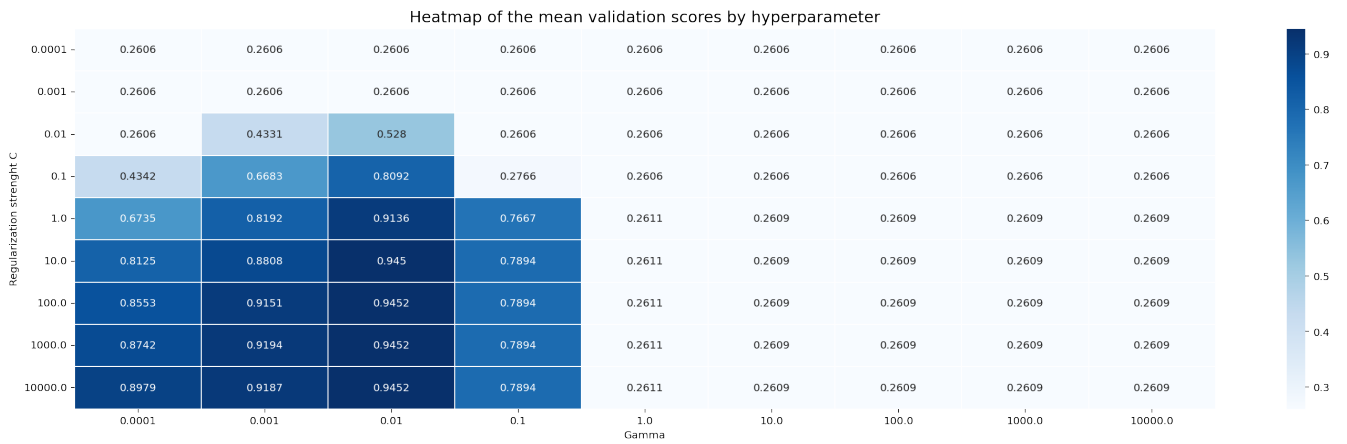


Figure 9: SVM Hyperparameter Tuning Heatmaps

The Support Vector Machine model with the RBF kernel yielded optimal results when tuned with a regularization strength (C) of 100 and a gamma value of 0.01. Interestingly,

employing a gamma value of 1 led to a significant decrease in model accuracy, while using a C value below 0.1 also resulted in a noticeable decline in performance. These findings emphasize the importance of carefully selecting hyperparameters, particularly C and gamma, to achieve optimal performance in the SVM model with the RBF kernel.

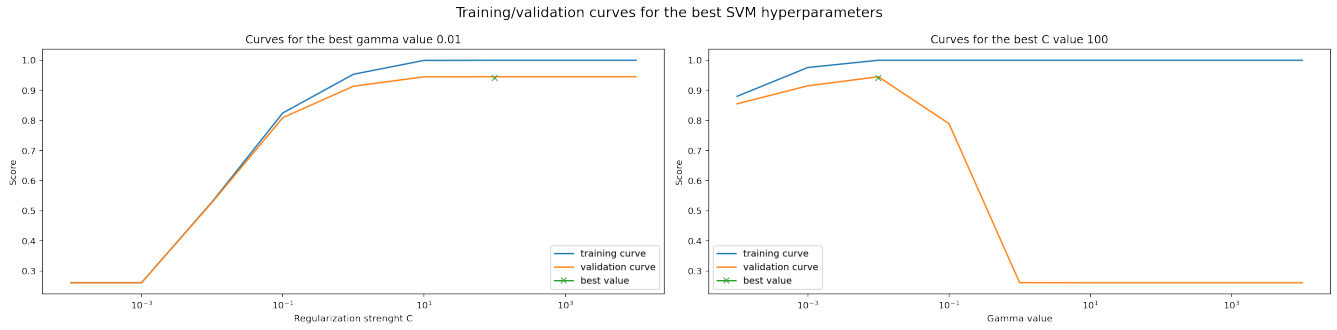


Figure 10: SVM Hyperparameter Tuning Graphs

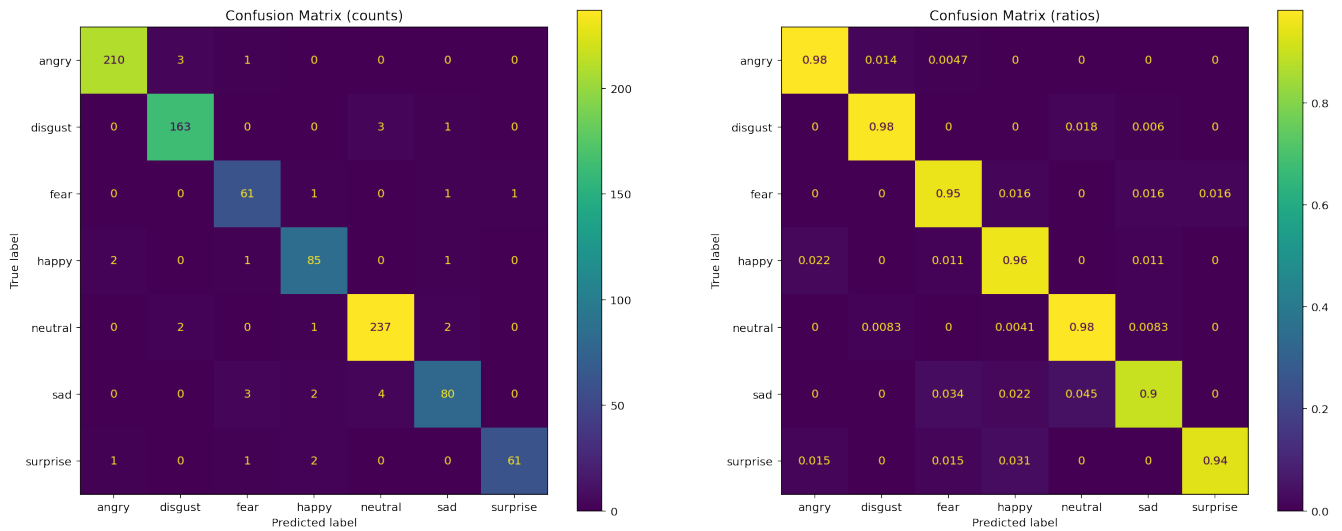


Figure 11: SVM Confusion Matrix

As shown in the first graph with the training curve, we can see that the classifier is improving for high values of C, however after a while (10-100) both curves becomes flat. We also see the model overfits when the gamma value exceed 0.1 as the training curve remains very high, while the validation curve sharply loses accuracy.

## RBF kernel

RBF (Radial Basis Function) kernel is a popular kernel function used in SVM (Support Vector Machine) for classification and regression tasks. The RBF kernel measures the similarity between two samples by computing the distance between them in a higher-dimensional space. It uses a Gaussian function to transform the input features into a higher-dimensional space,

where it becomes easier to find a hyperplane that separates the data into different classes. The RBF kernel has several advantages over other kernel functions. Firstly, it can handle non-linearly separable data by transforming it into a higher-dimensional space. Secondly, it can capture complex patterns in the data by adjusting its shape and width. Finally, it can provide a measure of uncertainty by giving a probability estimate for each classification decision. However, the RBF kernel also has some limitations. Firstly, it can be computationally expensive, especially when dealing with large datasets. Secondly, it may require careful tuning of the hyperparameters, such as the regularization parameter  $C$  and the kernel parameter  $\gamma$ , to achieve optimal performance. Overall, RBF kernel SVM is a powerful and versatile machine learning algorithm that has numerous applications in various domains, including image and text classification, bioinformatics, and finance.

The SVC classifier with the RBF kernel achieved outstanding results, exhibiting a remarkable test accuracy of 0.961. Notably, the RBF kernel even outperformed its performance on the validation set, indicating its robustness and effectiveness. Analyzing the outcomes, it is evident that numerous categories attained high scores, indicating accurate classification. However, the happy and sad categories demonstrated the lowest scores, suggesting greater difficulty in predicting these emotions.

### 5.2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of deep learning architecture that is predominantly used for tasks related to recognizing images and videos. The fundamental concept behind CNN involves utilizing convolutional layers to learn the features of the input data automatically. This is followed by pooling layers that help to reduce the dimensionality of the data.

CNNs can learn progressively more abstract features of the input data in a hierarchical fashion, with each layer of the network learning higher-level features. This approach allows CNNs to achieve impressive performance levels on numerous computer vision tasks.

Typically, CNNs consist of various types of layers, including convolutional layers, pooling layers, activation functions, fully connected layers, and a softmax output layer. The convolutional layers are responsible for extracting the features, while the pooling layers help to reduce the dimensionality of the data. The activation functions introduce non-linearity to the network, and the fully connected layers combine the features extracted by the convolutional layers.

CNNs have found wide-ranging applications in numerous fields, such as image classification, object detection, and image segmentation. They have also been used in natural language processing tasks, including text classification and sentiment analysis, among others.

The training and validation accuracies are both improving during the first 25 epochs, then we start to see signals of overfitting.

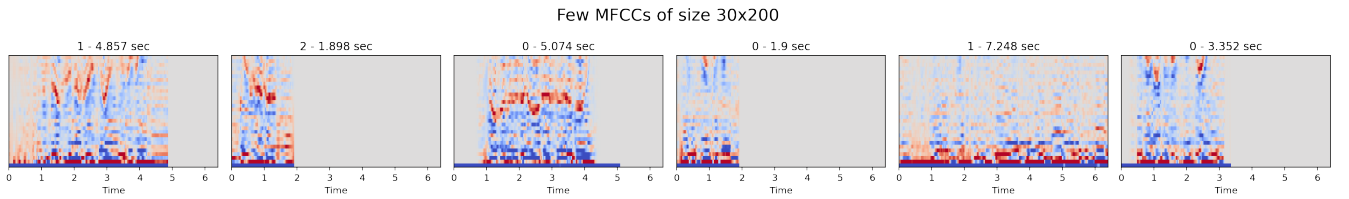


Figure 12: Images made from MFCCs

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 13, 98, 256)	6656
max_pooling2d (MaxPooling2D)	(None, 6, 49, 256)	0
batch_normalization (Batch Normalization)	(None, 6, 49, 256)	1024
conv2d_1 (Conv2D)	(None, 2, 46, 128)	524416
max_pooling2d_1 (MaxPooling2D)	(None, 1, 23, 128)	0
batch_normalization_1 (Batch Normalization)	(None, 1, 23, 128)	512
flatten (Flatten)	(None, 2944)	0
dropout (Dropout)	(None, 2944)	0
dense (Dense)	(None, 128)	376960
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 7)	455
=====		
Total params: 918,279		
Trainable params: 917,511		
Non-trainable params: 768		

Figure 13: Convolutional Neural Network Model Architecture

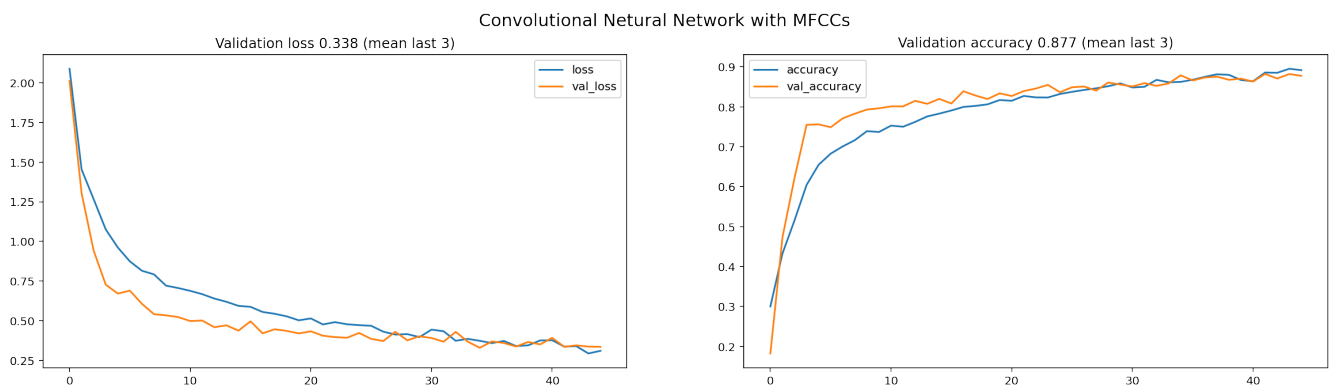


Figure 14: Convolutional Neural Network Accuracy Loss Graphs

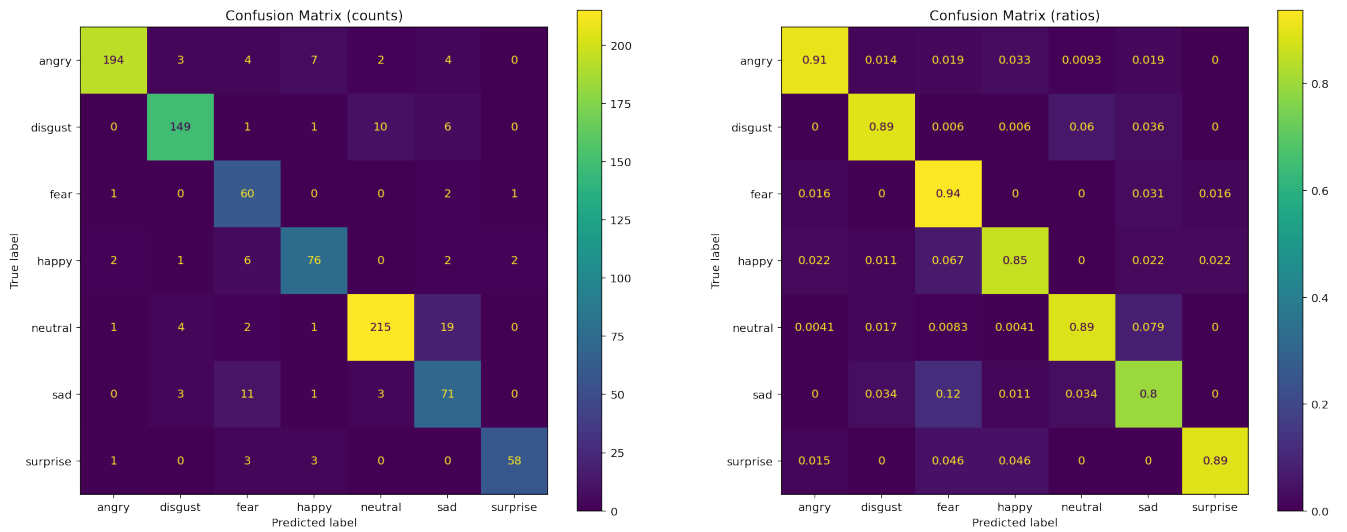


Figure 15: Convolutional Neural Network Confusion Matrix

The CNN model achieved promising results with a test loss of 0.31, indicating a relatively low level of error. Additionally, the model demonstrated an impressive test accuracy of 88.49%, highlighting its ability to accurately classify emotions within the dataset. These findings underscore the effectiveness and reliability of the CNN model in emotion categorization tasks.

### 5.3 Speaker Diarization

Speaker diarization is the process of automatically separating an audio or video recording into segments based on who is speaking. The goal of speaker diarization is to identify the different speakers in a recording and group the segments of the recording that correspond to each speaker.

The algorithms typically use a combination of audio signal processing techniques and machine learning models to separate speech segments by speaker. Some common approaches include clustering techniques to group similar segments together, classification models to identify the speaker of each segment, and modeling the speaker characteristics such as voice pitch and speaking style. The output of a speaker diarization system is typically a timeline of speaker turns, indicating when each speaker is speaking in the recording.

In short, Speaker diarization is the process of identifying "who spoke when" in an audio recording, typically a conversation involving multiple speakers. It involves segmenting the audio into different speaker segments and labeling each segment with the corresponding speaker's identity. This process can be useful in a variety of applications, such as transcribing speech, identifying speakers in a surveillance video, or analyzing the dialogue in a movie, speech processing, including transcription, speaker recognition, and language modeling.

There are several approaches to speaker diarization, including:

- **Clustering-based approaches:** Clustering is the most common approach to speaker diarization. It involves segmenting the audio into short frames and clustering the frames based on acoustic features such as MFCCs or i-vectors. Each cluster is assumed to correspond to a different speaker, and the clusters are grouped into speaker segments based on a set of clustering criteria.
- **GMM-based approaches:** Gaussian Mixture Models (GMMs) can be used to model the speaker-specific characteristics of the audio signal. Each speaker is modeled by a separate GMM, and the speaker identity of each frame is assigned based on the likelihood of the frame belonging to each GMM.
- **Deep Learning-based approaches:** Deep learning-based methods, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been used for speaker diarization. These methods use neural networks to extract high-level features from the audio signal and classify each frame into different speaker segments.



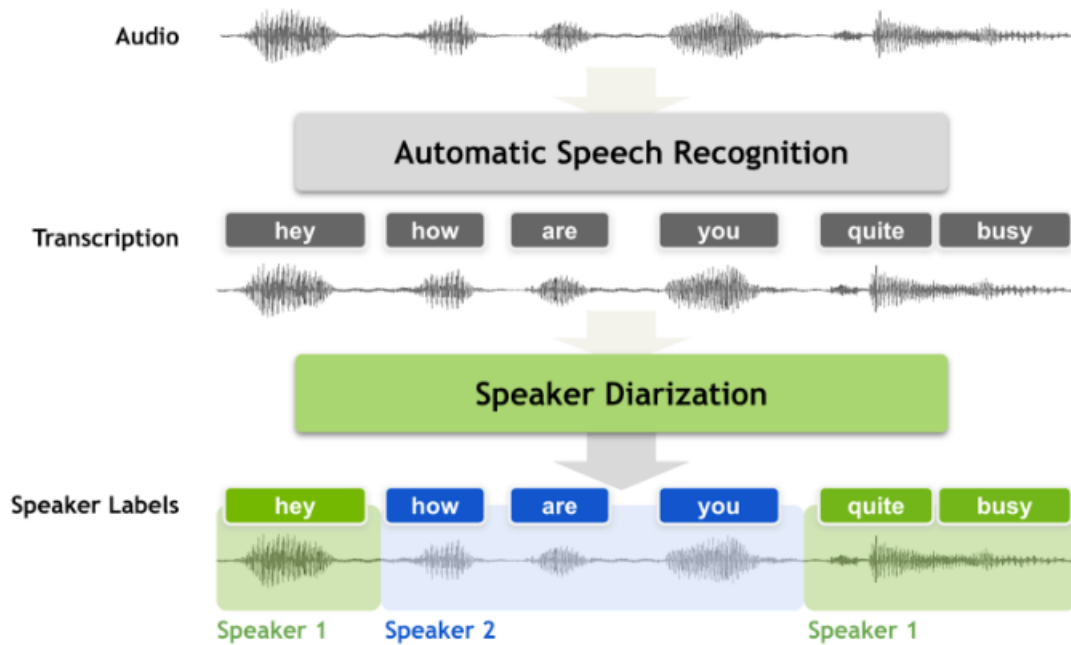


Figure 16: Project Timeline

The performance of speaker diarization depends on several factors, including the quality of the audio signal, the number of speakers, and the duration of each speaker segment. Speaker diarization is a challenging task, particularly in the case of overlapping speech, speaker variability, and background noise.

We are trying out different kinds of methods and have decided to go with the deep learning based approach as followed and researched as per the highly reputed state of the art level paper. [12]

A Speaker Diarization system consists of the following subsystems :

- **Speech Detection:** In this step, we use a Voice Activity Detector (VAD) module to separate out speech from non-speech. This is required to trim out silences and non-speech parts from your audio recording. The authors have used the VAD module from pyannote.metrics library. A VAD is basically a neural network trained to distinguish speech signals from non-speech signals. It is used to trim out silences from the audio file.
- **Speech Segmentation:** In this step, we extract out small segments of your audio call (typically about 1 second long) in a continuous manner. Speech segments typically contain just one speaker. This is achieved by segmenting the audio into windows with overlap. The size of the window determines the size of your segment. Think of it as a magnifier which works on a specific segment of your audio file. So, if your window size is 2 seconds, and you set an overlap of 0.5 seconds, your first window would be : (start = 0.0s , stop = 2.0s), next window will be: (start = 0.5s, stop = 2.5s) ... and so on until your full audio is covered.

- **Embedding Extraction:** This system creates a neural-network based embedding of your speech segments extracted in the previous step. An embedding is a vector representation of data which could be used by the deep learning framework. We can create embeddings for audio, text, images, documents etc. The authors extracted embeddings of each of the audio segments which we found before. First, we find the MFCC (Mel Frequency Cepstral Coefficient) of the audio segment. These are basically feature coefficients which capture the variations in the speech like pitch, quality, intone etc of the voice in a much better way. They are obtained by doing a specialized Fourier Transform of the speech signal. Don't worry, the SciPy library of python has a separate module for finding MFCCs for us. In the next step, the authors use an LSTM based network which takes in the MFCCs and outputs a vector representation (embedding) which they call a d-vector.
- **Clustering:** After creating embeddings of the segments, we next need to cluster these embeddings. After clustering, the embeddings of the segments belonging to the same speakers are part of one cluster, and assigned the label of the speaker. This step is crucial as it assigns the labels to our embeddings, as well as the number of clusters, which indicates to us the number of speakers in the audio file. Clustering is an Unsupervised machine learning method which tries to create clusters (or groups) of your data in an n-dimensional space. There are many clustering algorithms being used by machine learning researchers, most famous among them being K-means.

## 6 Project Plan and Timeline

In this section, we present the overall plan and timeline for the execution of the project. This section outlines the specific tasks, milestones, and deliverables to be accomplished throughout the project's duration. Attached below is the diagram for our project plan and timeline.

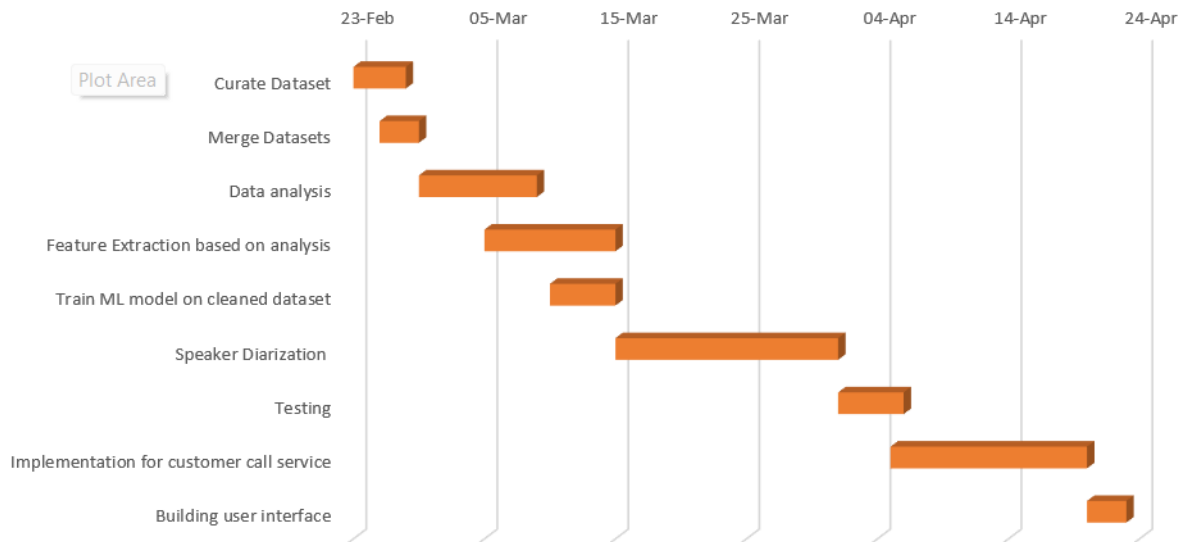


Figure 17: Project Timeline

## 7 Implementation

We have curated a web app that renders our machine learning model. This web-app is developed using Django, DbSqlite3, Html5, CSS3, and Javascript. We have integrated our model.h5 file in the web interface. On this interface, the user can upload an audio file. The audio file gets stored in the database with its name and content. Then, this file gets processed (noise removal and segmentation). After the processing, we can see the segmented audios with their timestamps (start and end time) and the speaker (0 or 1) that is speaking for that particular duration. Now, on these segments, we run our machine learning model to analyse the audio segment. Based on the classified result we can determine how the conversation has been from the start. On this basis, we can conclude whether the customer service person did a good job or not.

First, we start with visualization of our speaker diarizer.

### 7.1 Speaker Diarizer

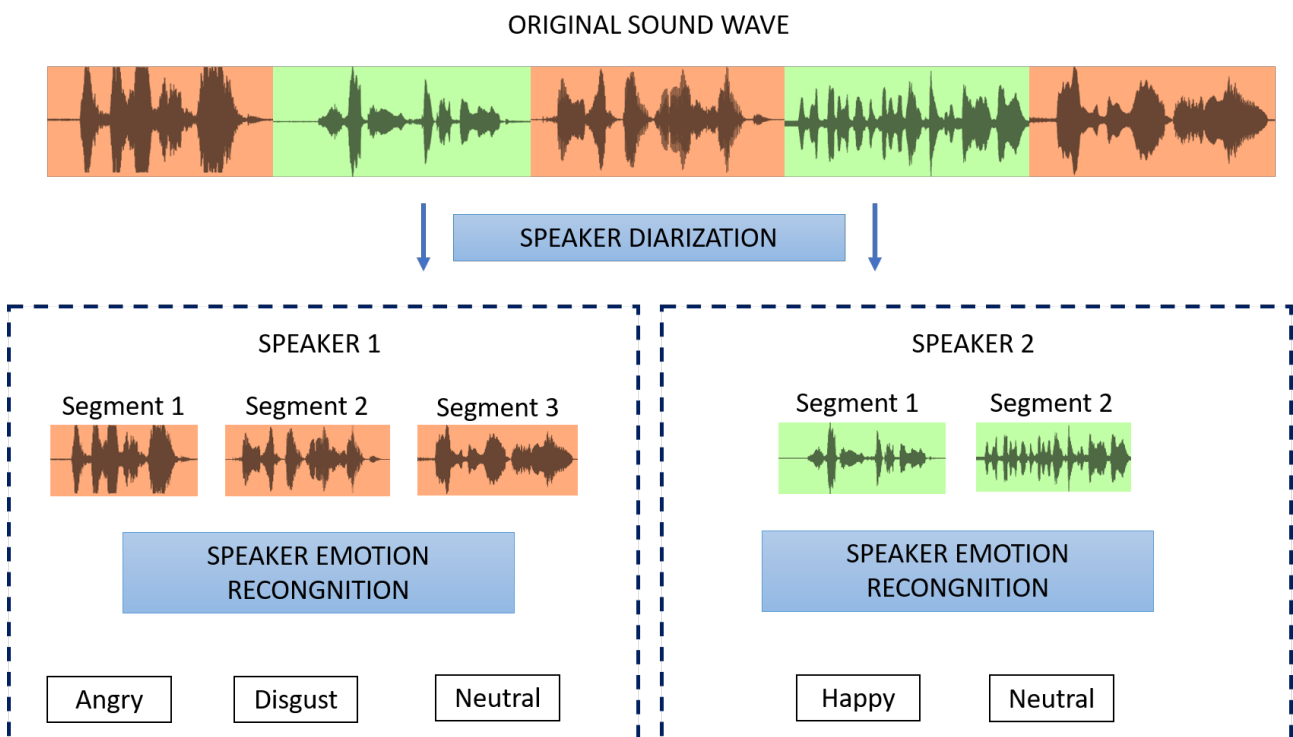


Figure 18: Speaker Diarizer

Fig.18 shows how an audio file has been broken down into 2 segments. There are 2 speakers in that audio file and the parts in which the speaker 1 and 2 speak, are separated and grouped. Now, on these parts, we run our speech recognition model which detects the emotion of an individual segment. Based on this, we can conclude the emotion throughout the call of both individuals.

We have 2 audio files that we work on namely, test\_bro\_new.wav (referred to as first audio file) and CustomerCare.mp3 (referred to as second audio file)

## 7.2 Segmenting first audio file

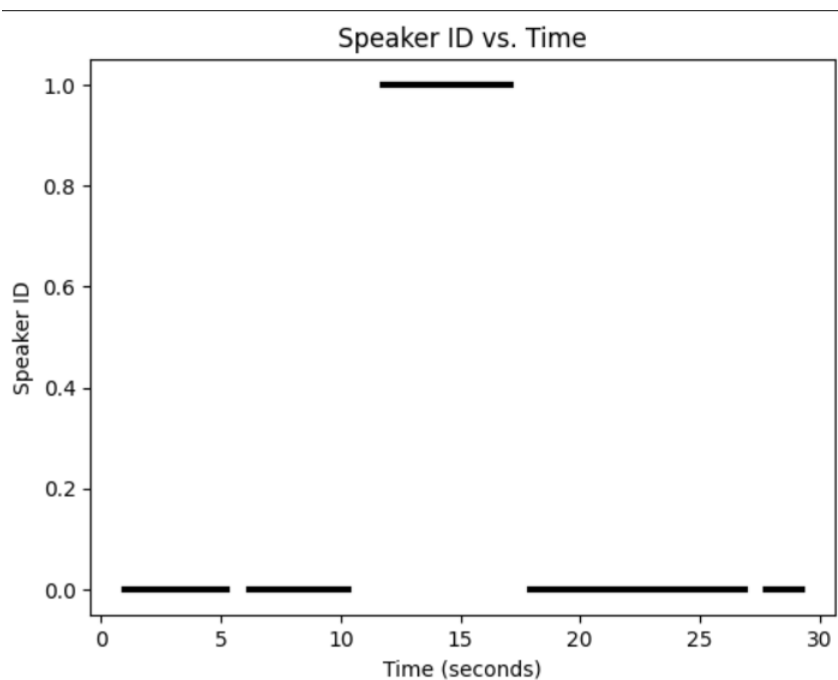


Figure 19: Speaker ID VS Time Test Bro New

In Fig.19, we run our speaker diarization on the first audio file. As we can see, we have speaker ID on the X-axis and time (seconds) on the Y-axis. The speaker 0 speaks from 0.9 to 5.2 seconds approximately, and then again from 6.2 to 10.3 seconds. Whereas, the speaker 1 speaks from 11.7 to 17.1 seconds. We save these segments as separate audio files and they contain the speaker ID in the database as we can see in Fig.20

	Start	End	Speaker
0	0.970	5.239	0
1	6.218	10.352	0
2	11.753	17.119	1
3	17.947	26.857	0
4	27.785	29.253	0

Figure 20: Start Time, End Time and Speaker ID Test Bro New

### 7.3 Segmenting second audio file

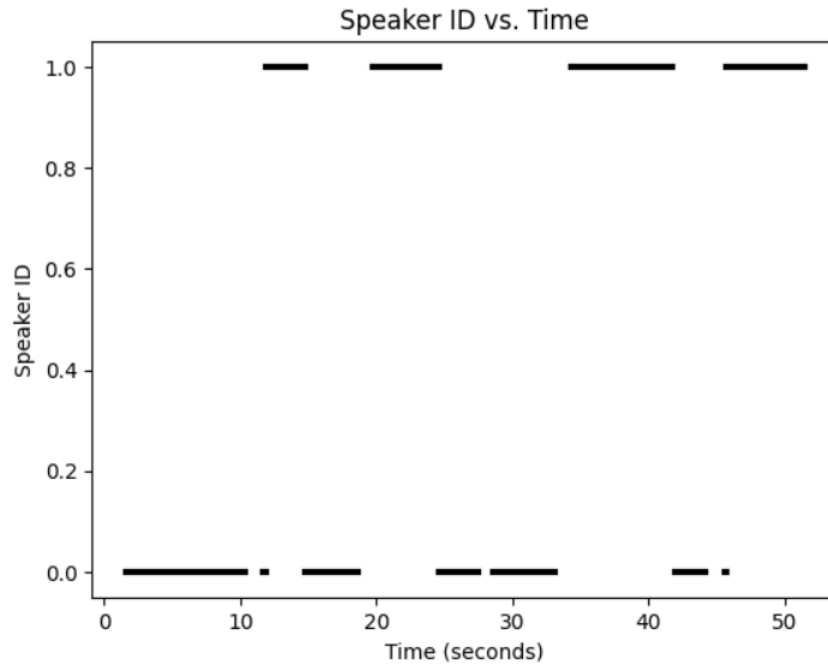


Figure 21: Speaker ID VS Time Customer Care

In Fig.21, we run our speaker diarization on the second audio file. As we can see, we have speaker ID on the X-axis and time (seconds) on the Y-axis. The speaker 0 speaks from 0.5 to 11 seconds approximately. The speaker 1 speaks from 12 to 14 seconds and so on. We, again, save these segments as separate audio files and they contain the speaker ID in the database as we can see in Fig.22

	Start	End	Speaker
0	1.595	10.370	0
1	11.703	11.872	0
2	11.872	14.741	1
3	14.740	18.638	0
4	19.786	19.803	0
5	19.803	24.629	1
6	24.629	27.515	0
7	28.544	33.151	0
8	34.383	41.774	1
9	41.926	44.170	0
10	45.655	45.689	0
11	45.689	51.460	1

Figure 22: Start Time, End Time and Speaker ID Customer Care

## 7.4 Detecting emotion on first audio file using Web-Based UI

We will upload the audio file on our webapp to do segmentation and detection.

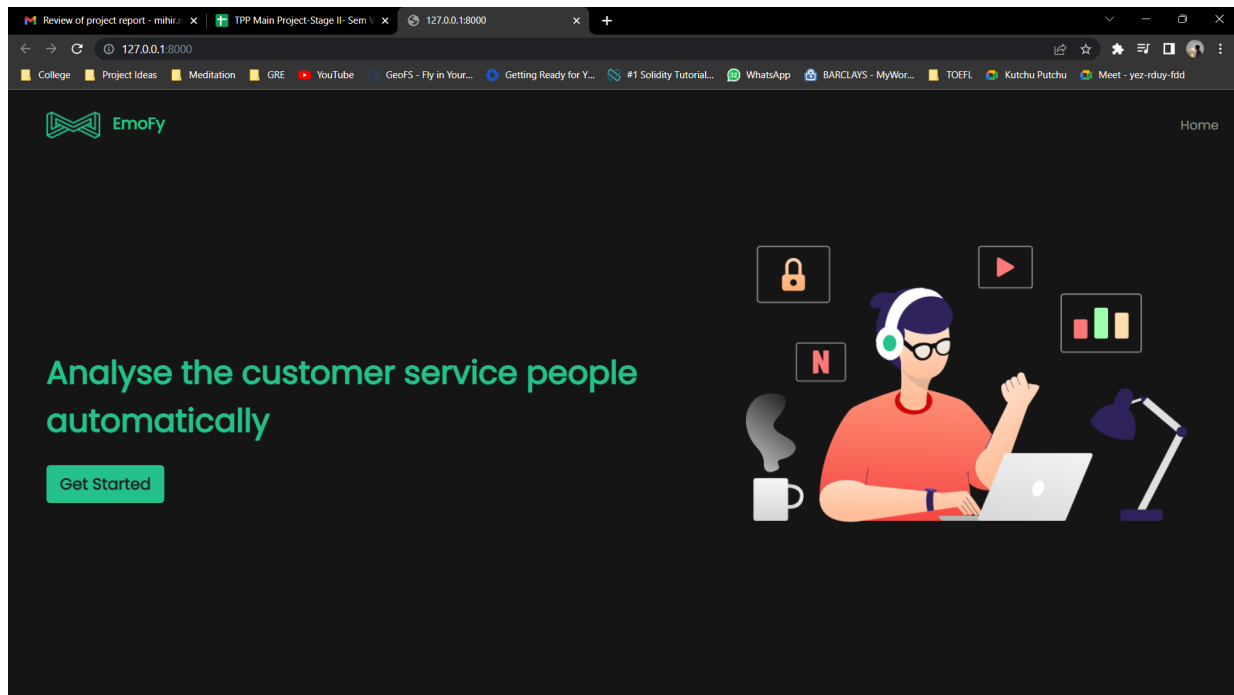


Figure 23: Home Page

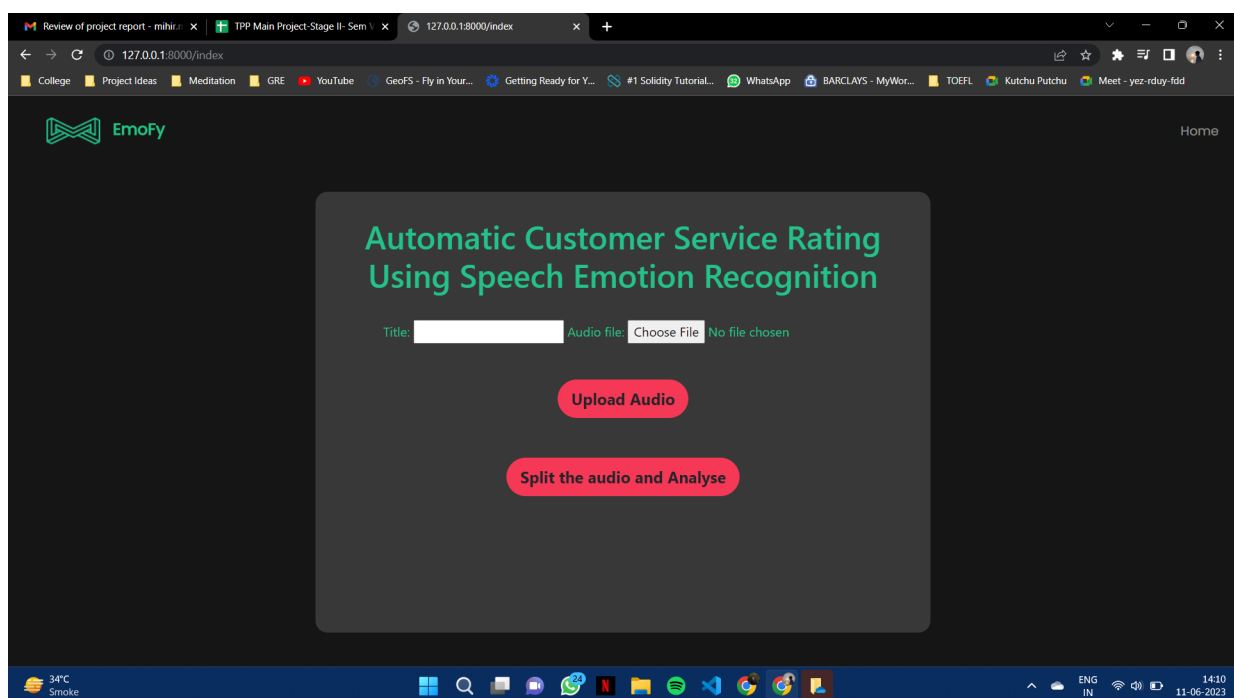
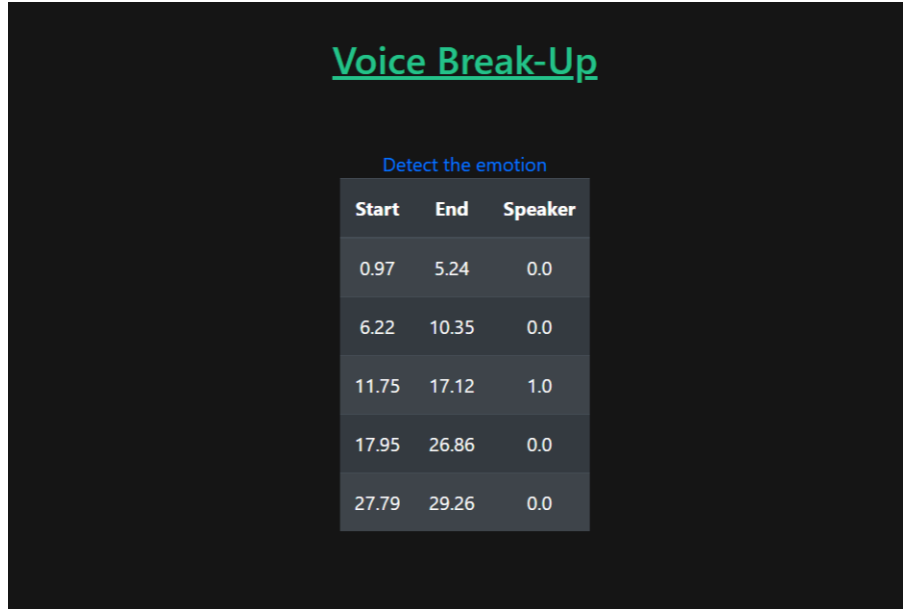


Figure 24: Upload audio file

In Fig. 24 we can choose a file from our file storage/hard drive and upload it with a given title. This same title is used to retrieve the audio file.

Post this, we select the file that we want to split our audio based on the number of speakers. We have assumed that there are 2 speakers, but this application can be extended to have more speakers than that. Once we click on the 'split the audio and analyse' button, we get the Fig. 25.



The screenshot shows a web application interface with a dark background. At the top, the title 'Voice Break-Up' is displayed in a green, monospace-style font. Below the title, there is a blue button labeled 'Detect the emotion'. Underneath the button is a table with three columns: 'Start', 'End', and 'Speaker'. The table contains five rows of data, representing segmented audio segments.

Start	End	Speaker
0.97	5.24	0.0
6.22	10.35	0.0
11.75	17.12	1.0
17.95	26.86	0.0
27.79	29.26	0.0

Figure 25: Segmented Audio Time Stamps with Speaker ID

We can see in Fig. 25 the Start Time, End Time and the Speaker ID. This completes our segmentation/diarization phase of the project.

Now, we move on to the analysis phase of it. Each of these segments are saved in the database in chunks. Each chunk is treated as an individual audio file. These files are served as an input to our emotion recognition model which gives us the probability of different emotions. We choose the maximum probability among the given emotion and conclude that as the emotion of that segment. We can see these probabilities for Segment 1, 2, and 3 in Fig. 27, Fig. 28, and Fig. 29 respectively.

Finally, we display the final emotion to each of the segments. These emotions can be analysed from start to end. The general trend for a good call should be that the customer should go from sadness/disgust/anger to neutral/happy. If that is seen in the call, we can say that the the customer service guy did a good job.



### Voice Break-Up

Home

Start	End	Speaker	Emotion
0.97	5.24	0	sadness
6.22	10.35	0	disgust
11.75	17.12	1	disgust
17.95	26.86	0	sadness
27.79	29.26	0	anger

Figure 26: Emotion Detected on the created segments

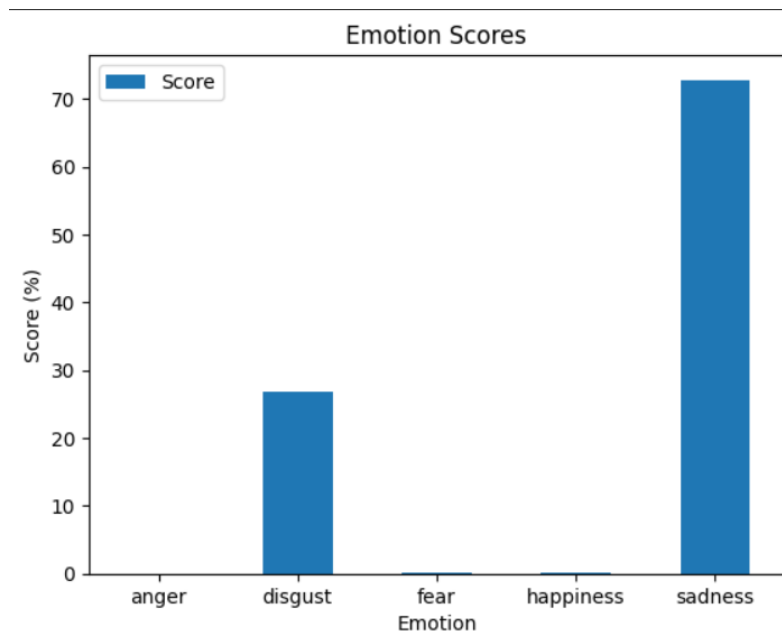


Figure 27: Emotion Analysis on Segment 1 with Speaker ID 0

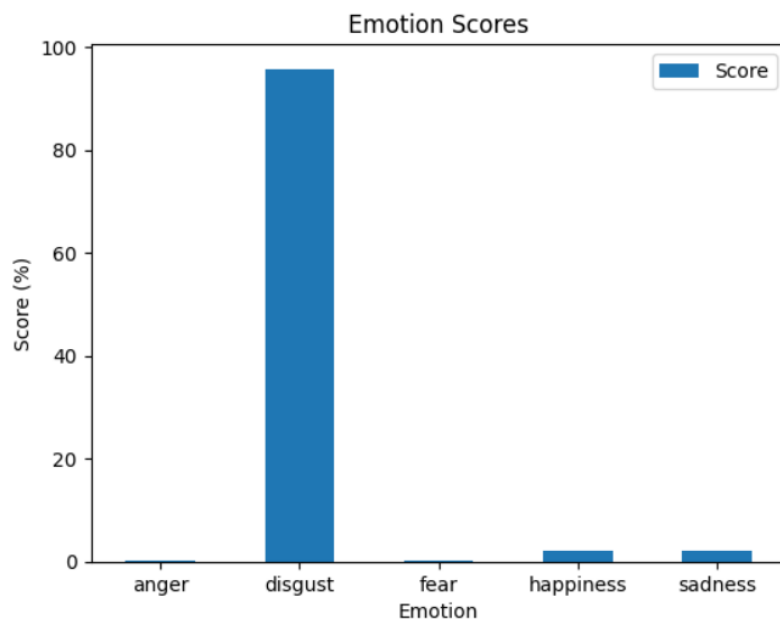


Figure 28: Emotion Analysis on Segment 2 with Speaker ID 0

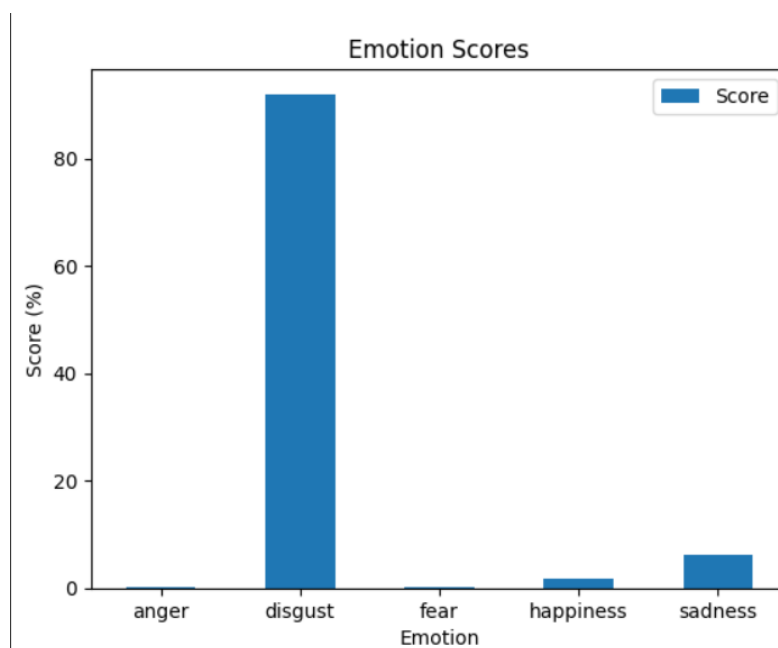


Figure 29: Emotion Analysis on Segment 3 with Speaker ID 1

## 8 Conclusion

The comparison table shows the performance of three models: SVM, Random Forest, and CNN, in terms of accuracy. Among the three models, SVM demonstrates the highest accuracy, with Random Forest and CNN following closely behind. The results suggest that SVM may be the most suitable model for the given task, but further analysis and evaluation are required to make a definitive conclusion.

Table 1: Comparison of model accuracy

Model	Testing Accuracy
SVM	95.7
Random Forrest	90
ConvNet MFCCs	87.6

All models succeeded in beating the baseline, with Support Vector Machine being the best overall.

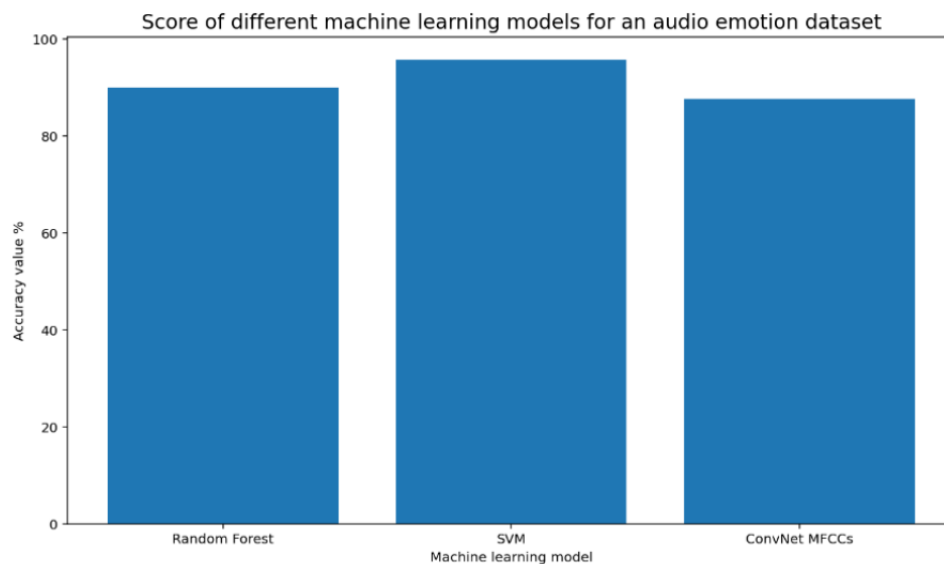


Figure 30: Accuracy values for our models

Let's also compare the results of the models by each emotion:

	angry	disgust	fear	happy	neutral	sad	surprise
<b>Random Forest</b>	96	92	88	86	96	75	89
<b>SVM</b>	98	98	94	96	98	90	92
<b>ConvNet MFCCs</b>	91	89	94	85	89	80	89

Figure 31: Comparing results of our model for each emotion

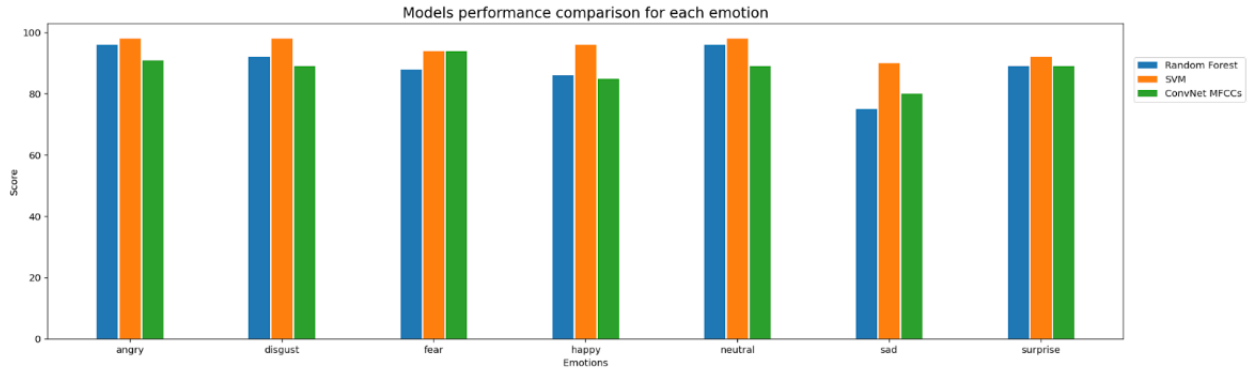


Figure 32: Comparing model performances for each emotion

The data indicates that the emotions of "angry," "disgust," and "neutral" demonstrated the highest categorization accuracy. Moreover, the "sad" category posed the greatest challenge for prediction. The Support Vector Machine model consistently outperformed all other models in each instance.

In conclusion, this project highlights the importance of data quality in achieving accurate classification results. The detrimental effects of inadequate data were evident from the Crema-D dataset, which introduced noise and negatively impacted performance. Techniques like PCA and T-SNE were instrumental in visualizing and understanding the data's quality.

Furthermore, the significance of data quantity cannot be understated. While the project utilized around 12,000 samples for model training and evaluation, augmenting the dataset with more samples is necessary to enhance the generalization capabilities of the machine learning algorithms and build more robust models.

Considering the time required for feature extraction, it is crucial to explore alternative algorithms and libraries to optimize the process. Certain methods, such as obtaining fundamental frequencies using the 'librosa' package, proved to be time-consuming, and finding more efficient approaches can significantly reduce analysis time.

The effectiveness of using Mel-frequency cepstral coefficients (MFCC) in sound analysis was confirmed in this project. MFCC spectrograms exhibited clear distinctions with the target emotions during exploratory data analysis. The Random Forest model even selected MFCC components as top features for classification. Additionally, the convolutional neural network model, relying solely on MFCC spectrograms, achieved commendable results.

Throughout this project, it became apparent that knowledge is key. A deep understanding of the research domain, in this case, digital signal processing, is essential for successful machine learning projects. However, it is important to acknowledge the vast amount of unknowns that exist, including various features and the significance of MFCC in extracting meaningful information from sound. Beyond the programming aspect, understanding different aspects of a topic is crucial when approaching a problem.

## References

- [1] Latif, Siddique Rana, Rajib Younis, Shahzad Qadir, Junaid Epps, Julien. (2018). Transfer Learning for Improving Speech Emotion Classification Accuracy. 257-261. 10.21437/Interspeech.2018-1625.
- [2] Murugan, Harini. (2020). Speech Emotion Recognition Using CNN. International Journal of Psychosocial Rehabilitation. 24. 10.37200/IJPR/V24I8/PR280260.
- [3] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar and T. Alhussain, "Speech Emotion Recognition Using Deep Learning Techniques: A Review," in IEEE Access, vol. 7, pp. 117327-117345, 2019, doi: 10.1109/ACCESS.2019.2936124.
- [4] S. Yoon, S. Byun, S. Dey and K. Jung, "Speech Emotion Recognition Using Multi-hop Attention Mechanism," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 2822-2826, doi: 10.1109/ICASSP.2019.8683483.
- [5] S. Zhang, S. Zhang, T. Huang and W. Gao, "Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching," in IEEE Transactions on Multimedia, vol. 20, no. 6, pp. 1576-1590, June 2018, doi: 10.1109/TMM.2017.2766843.
- [6] Sahu, Gaurav. (2019). Multimodal Speech Emotion Recognition and Ambiguity Resolution.
- [7] Pepino, L., Riera, P., Ferrer, L. (2021) Emotion Recognition from Speech Using wav2vec 2.0 Embeddings. Proc. Interspeech 2021, 3400-3404, doi: 10.21437/Interspeech.2021-703
- [8] Tripathi, Suraj Kumar, Abhay Ramesh, Abhiram Singh, Chirag Yenigalla, Promod. (2019). Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions.
- [9] A. Shirian and T. Guha, "Compact Graph Architecture for Speech Emotion Recognition," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 6284-6288, doi: 10.1109/ICASSP39728.2021.9413876.
- [10] P. Shi, "Speech emotion recognition based on deep belief network," 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Zhuhai, China, 2018, pp. 1-5, doi: 10.1109/ICNSC.2018.8361376.
- [11] Deshmukh, Girija Gaonkar, Apurva Golwalkar, Gauri Kulkarni, Sukanya. (2019). Speech based Emotion Recognition using Machine Learning. 812-817. 10.1109/ICCMC.2019.8819858.
- [12] H. Bredin, A. Larcher, C. Barras and G. Gelly, "PYANNOTE.AUDIO: Neural building blocks for speaker diarization," in Proceedings of Interspeech 2017, Stockholm, Sweden, Aug. 2017, pp. 3092-3096, doi: 10.21437/Interspeech.2017-1138.