**MINI PROJECT REPORT**

ON

**ABUSIVE COMMENT IDENTIFICATION IN MALAYALAM LANGUAGE**



**School of Digital Sciences**

[**Kerala University of Digital Sciences, Innovation and Technology**](https://duk.ac.in/)

**(Digital University Kerala)**

**Abusive Comment Identification In Malayalam Language**

Submitted by

**ABHISHANAV M (Register Number: 223025)**

**GAYATHRI ANIL (Register Number: 223027)**

**HARSHIN RONALDO (Register Number: 223028)**

*In partial* *fulfilment of the requirements for the award of*

*Master of Science in COMPUTER SCIENCE WITH DATA ANALYTICS of*

****

**School of Digital Sciences**

[**Kerala University of Digital Sciences, Innovation and Technology**](https://duk.ac.in/)

**(Digital University Kerala)**

**Technocity Campus, Thiruvananthapuram, Kerala – 695317**

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled “**Abusive Comment Identification In Malayalam Language”** Submitted by **---------- (Reg. No:2230--)** in partial fulfillment of the requirements for the award of **Master of Science in Computer Science with Specialization in Data Analytics** is a bonafide record of the work carried out at “**Kerala University of Digital Sciences, Innovation and Technology”.**



**Supervisor Course Coordinator**

**DR. ANOOP V.S Prof. MANOJ KUMAR T.K**

**School of Digital Sciences School of Digital Sciences**

**DUK DUK**

**Head Of Institution**

**Prof. SAJI GOPINATH**

**Vice Chancellor**

**DUK**

**DECLARATION**

I, ----------**,**astudent of **MSc Computer Science with Specialization in Data Analytics**, hereby declare that this report is substantially the result of my own work, except where explicitly indicated in the text, and has been carried out during the period **April 2023 – September 2023**.

Place: Trivandrum ……………………….

Date: 08-01-2024 Student’s signature

**ACKNOWLEDGEMENT**

I want to express my deep gratitude to **Dr. Anoop V.S**, Professor at Digital University Kerala, Trivandrum, for his invaluable guidance and mentorship, which were instrumental in the successful completion of this project. I would like to extend my heartfelt thanks to **Dr. Manoj T.K**, Course Coordinator, School of Digital Sciences, Digital University Kerala, Trivandrum for creating a supportive environment and facilitating access to educational resources that contributed to this achievement. I extend my appreciation to **Prof. Saji Gopinath**, Vice Chancellor, Digital University Kerala, for granting access to the university’s facilities and resources. Furthermore, I am deeply grateful to my friends and family for their steadfast support, inspiration, and aid throughout the project’s journey.

**ABSTRACT**

YouTube is a video-sharing and social media platform where users create profiles and share videos for their followers to view, like, and comment on. Abusive comments on videos or replies to other comments may be offensive and detrimental for the mental health of users on the platform. It is observed that often the language used in these comments is informal and does not necessarily adhere to the formal syntactic and lexical structure of the language. With the increasing presence of abusive language and toxicity in online platforms, particularly YouTube, the need for efficient and accurate identification of such content in regional languages like Malayalam has become imperative. Therefore, creating a rule-based system for filtering out abusive comments is challenging.

This project aims to utilize natural language processing and deep learning approaches for identifying abusive comments posted to the YouTube that are written in Malayalam, which is one of the agglutinative languages spoken in the state of Kerala. For this, we use datasets of abusive comments in Malayalam and code-mixed Malayalam-English languages that are extracted from YouTube videos. Different machine learning approaches with pre-trained language models will be used to implement the classifier. Overall, this project may help in detection of abusive comments in Malayalam and may help in creation of comment-filters for Malayalam language on YouTube

CONTENTS

[CHAPTER 1 INTRODUCTION 1](#_Toc145333871)

[**1.2 PROBLEM DEFINITION** 2](#_Toc145333872)

[CHAPTER 2: METHODOLOGY 3](#_Toc145333873)

[**2.2 EXPERIMENTAL ANALYSIS** 9](#_Toc145333874)

[CHAPTER 3: RESULTS AND INSIGHTS 18](#_Toc145333875)

[**3.2 EmoDiarize: An Emotion-Aware Speaker Diarization App** 23](#_Toc145333876)

[**3.3 CONCLUSION** 25](#_Toc145333877)

[REFERENCE 26](#_Toc145333878)

**TABLE OF FIGURES**

[**Figure 1**: Emotion-Aware Speaker Diarization model building process 8](#_Toc145333721)

[**Figure 2**: Emotions against count 9](#_Toc145333722)

[**Figure 3:** Normal audio 9](#_Toc145333723)

[**Figure 4**: Stretched audio 10](#_Toc145333724)

[**Figure 5**: Audio with noise 10](#_Toc145333725)

[**Figure 6**: Shifted audio 10](#_Toc145333726)

[**Figure 7**: Audio with pitch 11](#_Toc145333727)

[**Figure 8**: Features to extract 11](#_Toc145333728)

[**Figure 9**: Feature extraction code 12](#_Toc145333729)

[**Figure 10**: Loading pre-trained speaker diarization piplines 12](#_Toc145333730)

[**Figure 11**: Perform diarization and save output 13](#_Toc145333731)

[**Figure 12**: Audio segments after diarization 13](#_Toc145333732)

[**Figure 13**: Graph showing speaker diarization output 14](#_Toc145333733)

[**Figure 14**:Splitting into training and testing sets 14](#_Toc145333734)

[**Figure 15**: CNN Model Architecture 15](#_Toc145333735)

[**Figure 16**: Summary of the model 17](#_Toc145333736)

[**Figure 17:** Confusion matrix of the model 18](#_Toc145333737)

[**Figure 18**: Classification report 19](#_Toc145333738)

[**Figure 19**: Predicted labels and Actual labels 20](#_Toc145333739)

[**Figure 20**: Training vs Testing loss plot 21](#_Toc145333740)

[**Figure 21**: Training vs Testing accuracy plot 22](#_Toc145333741)

[**Figure 22**: Selecting Audio from the drop-down menu 23](#_Toc145333742)

[**Figure 23**: Choose an audio and click Go 23](#_Toc145333743)

[**Figure 24**: Streamlit Demo Output 24](#_Toc145333744)

# 

# **CHAPTER 1**

# **INTRODUCTION**

Speech Emotion Recognition (SER) is a specialized branch of Natural Language Processing (NLP) that centres around the interpretation of emotions conveyed through spoken language. The human voice carries a wealth of emotional information in its pitch, intensity, rhythm, and timbre, which makes it a valuable resource for understanding the emotional state of an individual. Emotion recognition from audio samples is a valuable application of machine learning and signal processing, with numerous potential use cases in areas like entertainment, market research, mental health assessment, and more. This process involves analysing audio data, often in the form of speech or vocal expressions, to determine the emotional state of the speaker. It involves the use of machine learning and deep learning techniques to discern the emotional state of a person from their voice, whether it is in spoken language, song, or other vocalizations.

The practical applications of this technology are far-reaching. In the realm of entertainment, emotion recognition can transform how we interact with media, making it more immersive and responsive. In marketing and market research, it can provide invaluable insights into consumer sentiment and preferences, enhancing product development and advertising strategies. Additionally, in the field of mental health, it holds the potential to assist in early diagnosis, monitoring, and treatment, offering a new dimension in patient care. However, amidst these promises, we must navigate challenges related to accuracy, privacy, and the ethical implications of emotion recognition technology. This report aims to navigate this multifaceted landscape, shedding light on the methods, opportunities, and responsibilities inherent in the fascinating world of emotion recognition from audio.

Speaker diarization is the process of segmenting audio recordings by speaker labels and it effectively tells us "Who spoke when?” in the audio recording. The process is far from straightforward. It faces challenges, particularly in situations with multiple speakers talking at the same time, varying speech patterns, background noise or low-quality audio recordings. It has a wide range of applications, including transcribing multi-speaker conversations accurately, indexing large audio and video archives, and enhancing the performance of speech recognition systems when multiple speakers are involved. Various methods have been employed for speaker diarization over the years. Traditional techniques include Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs). More recently, deep learning-based methods, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have shown promise in improving diarization accuracy, especially in complex scenarios.

**1.2 PROBLEM DEFINITION**

The problem addressed in this report is the development of a comprehensive machine learning system for audio diarization and emotion prediction. The objective is to create a robust and accurate model that can process audio recordings containing multiple speakers, segment them into distinct speaker segments and predict the emotional state of each speaker by using some recorded audio samples.

# **CHAPTER 2**

# **METHODOLOGY**

***LIBRARIES***

**Pandas:**

Pandas is an open-source library in Python that is mainly intended for working with relational or labelled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. Pandas is fast and has high performance & productivity for users.

**NumPy:**

NumPy is a general-purpose array-processing package. It provides high-performance multidimensional array objects, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It can also be used as an efficient multi-dimensional container of generic data.

**Librosa:**

Librosa is a Python package for music and audio analysis. It is typically used when working with audio data like in music generation (using LSTM’s) and Automatic Speech Recognition.

**Seaborn:**

Seaborn is a library mostly used for statistical plotting in python. It provides beautiful default styles and colour palettes to make statistical plots more attractive.

**Matplotlib:**

Matplotlib is an amazing visualization library in python for 2D plots of array. Pyplot is a Matplotlib module that provides a MATLAB-like interface.

**Scikit-learn:**

Scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface.

**Keras:**

Keras is a deep learning API written in python language, which is running on the top of the machine learning platform TensorFlow. It provides industry-strength performance and scalability. It focuses on ease of use, debugging speed, code elegance & conciseness, maintainability.

***DATASETS***

The following datasets were used to build the machine learning model. The first four are datasets available for public use. The final dataset was created specifically for the purpose of this project.

**RAVDESS**: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains 7,356 files (total size: 24.8 GB). The database contains 24 professional actors (12 female, 12 male), vocalizing two lexically matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. For the purpose of building this specific model, only the 1440 WAV files containing speech audio were used.

**CREMA-D**: CREMA-D is a data set of 7,442 original clips from 91 actors. These clips are from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified).

**TESS**: There are a set of 200 target words were spoken in the carrier phrase "Say the word \_' by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total. All the audio files are in WAV format.

**SAVEE**: The SAVEE database was recorded from four native English male speakers (identified as DC, JE, JK, KL) - postgraduate students and researchers at the University of Surrey aged from 27 to 31 years. Emotion has been described psychologically in discrete categories: anger, disgust, fear, happiness, sadness and surprise.

**Movie clips**: The dataset was compiled by randomly selecting audio clips from movies and then annotating them based on the expressed emotions. It has a total of 166 clips. The emotions identified are anger, happy, fear, disgust, sad, surprise and neutral.

***SPEAKER DIARIZATION***

Speaker diarization is the process of partitioning an audio stream containing human speech into homogeneous segments according to the identity of each speaker. It can enhance the readability of an automatic speech transcription by structuring the audio stream into speaker turns and, when used together with speaker recognition systems, by providing the speaker’s identity. It is used to answer the question “who spoke when?”

Pyannote audio is an open-source toolkit written in python for speaker diarization. Version 2.1 introduces a major overhaul of the default speaker diarization pipeline, made of three main stages: speaker segmentation applied to a short sliding window, neural speaker embedding of each speaker, and agglomerative clustering.

***DATA AUGMENTATION***

Data augmentation is the process of creating new synthetic data samples by adding small perturbations on our initial training set. Syntactic data can be generated for audio, by applying noise injection, shifting time, changing pitch and speed. The objective is to make the model invariant to those perturbations and enhance its ability to generalize. For this to work, adding the disturbance must conserve the same label as the original training sample.

* Pitch Shift: Altering the pitch of audio clips to simulate different vocal tones.
* Time Stretch: Changing the duration of audio clips while preserving their content.
* Noise Addition: Adding background noise to audio recordings.
* Shifting: Simulates the effect of slight timing variations in the audio clips.

***FEATURE EXTRACTION***

Feature extraction is a crucial step in analysing and establishing relationships between various elements. Audio data, in its raw form, cannot be directly comprehended by models. Therefore, it's necessary to convert it into a format that models can understand, and this is where feature extraction comes into play. The audio signal is a 3-dimensional signal in which the three axes represent time, amplitude and frequency. With the help of sample rate and the sample data, several transformations can be performed to extract valuable features out of it.

* Zero Crossing Rate: The rate of sign-changes of the signal during the duration of a particular frame.
* MFCC: Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel scale.
* RMS (Root Mean Square) value: It is the square of the arithmetic mean or the square of the function.

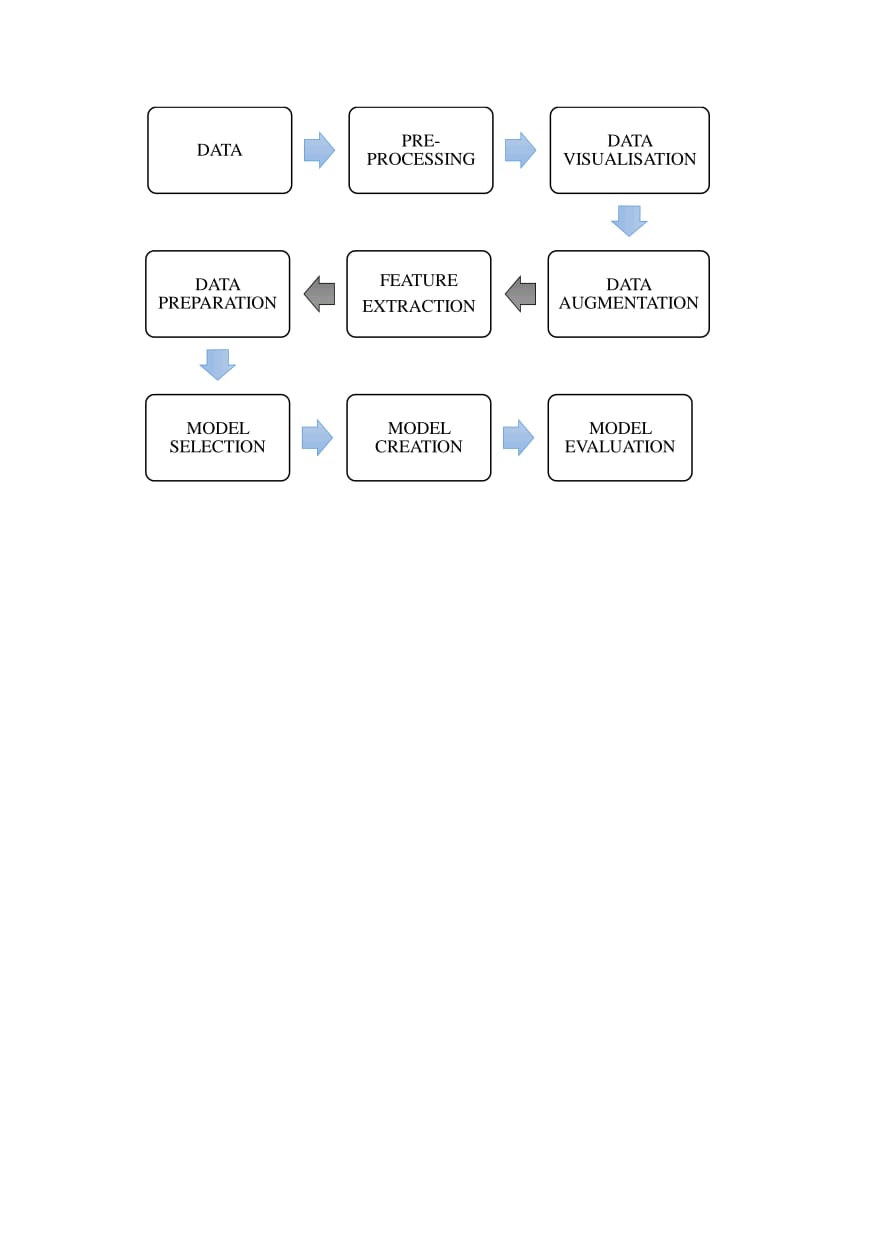
***CNN MODEL***

Convolutional Neural Networks (CNNs) are a fundamental component of machine learning. These networks consist of interconnected layers designed to automatically learn and extract intricate patterns and features from images. Convolutional layers apply learnable filters to scan the input image, while activation layers introduce non-linearity, enabling the network to capture complex relationships. Pooling layers reduce the spatial dimensions, and fully connected layers make the final classification. By training on labelled data and adjusting internal parameters, CNNs excel at recognizing objects, shapes, and patterns in images, making them a cornerstone technology in computer vision and image analysis within the realm of machine learning.

**Sequential model:**

A sequential model in machine learning provides a structured framework for organizing and applying processing steps in a specific sequence, ensuring that data flows through the model in a predetermined order. The term "sequential" refers to the specific way in which layers are stacked in the model, where one layer follows another in a sequential order. Sequential models are commonly used in various types of neural networks, including CNNs. CNNs often have a sequence of layers that perform operations like convolution, pooling, flattening, and fully connected layers. These layers are typically arranged sequentially in the order they are applied to the input data.

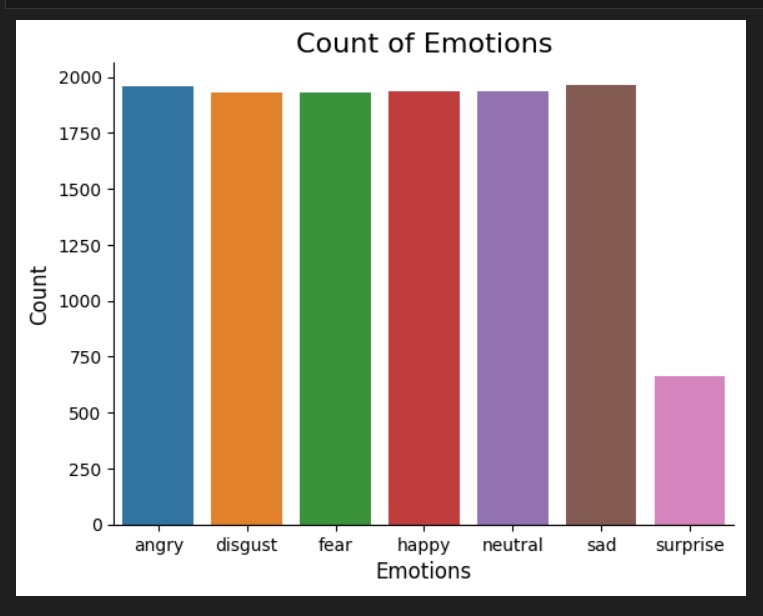
**Emotion-Aware Speaker Diarization Model Building Process**



**Figure 1**: Emotion-Aware Speaker Diarization model building process

**2.2 EXPERIMENTAL ANALYSIS**

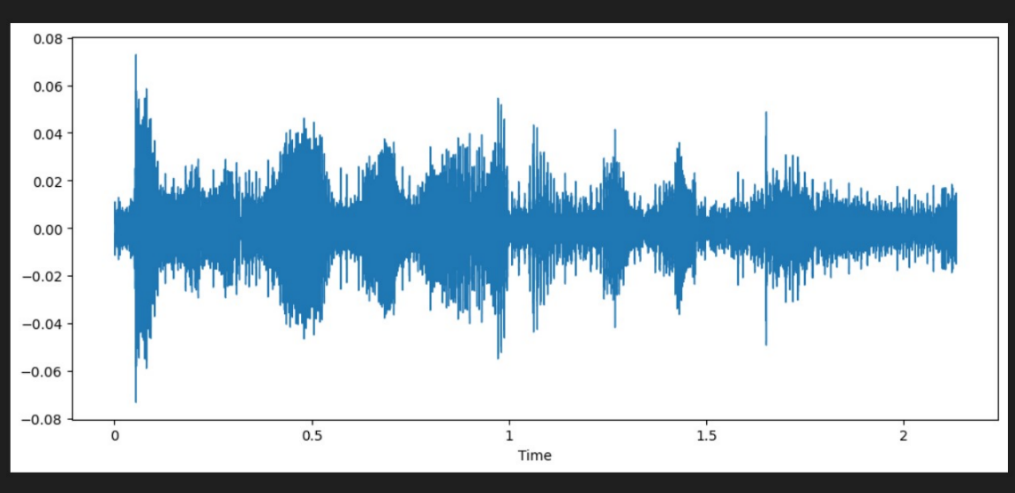
Among the five datasets used, four are publicly available: RAVDESS, CREMA-D, SAVEE and TESS. The final dataset, the movie clips dataset, was created specifically for the purpose of this project. All of them are integrated into a single dataset where each audio file is classified into one of seven emotions - sad, fear, disgust, neutral, happy, surprise, anger. These are plotted against the count of emotions.



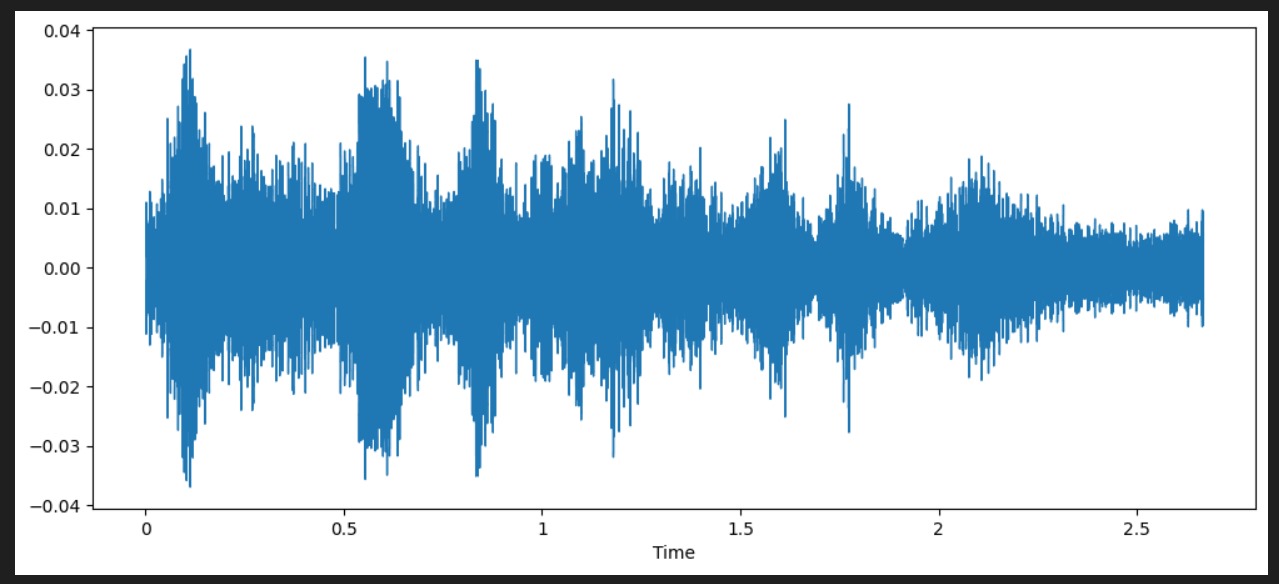
**Figure 2**: Emotions against count

**Data Augmentation:**

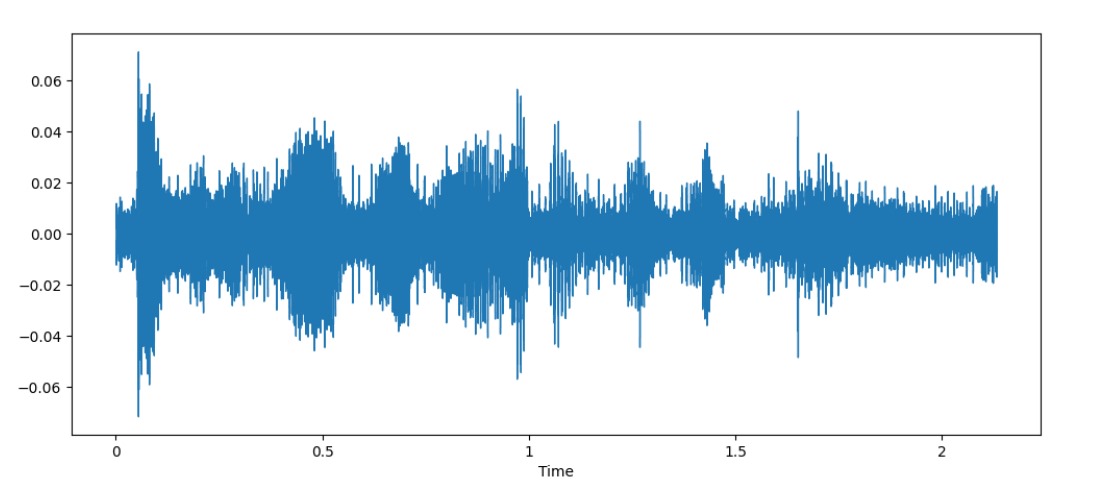
Each audio signal in the sample undergoes data augmentation.



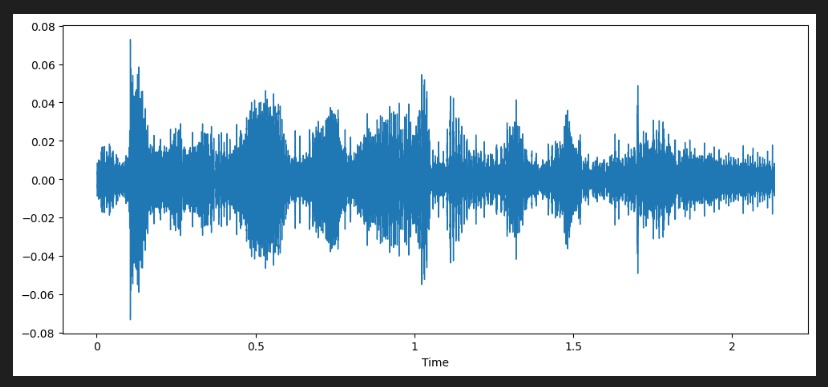
**Figure 3:** Normal audio



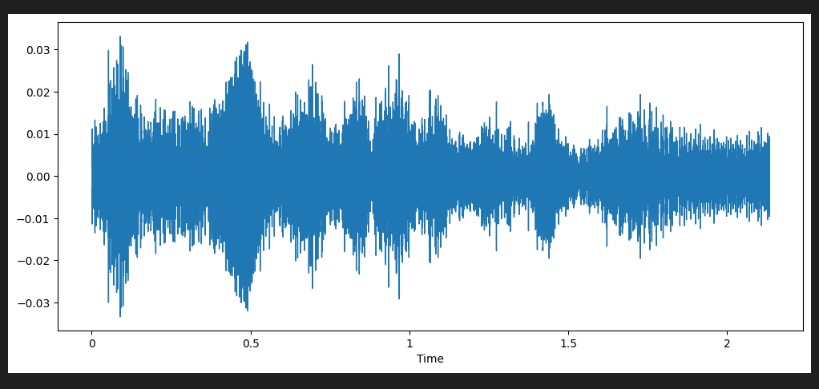
**Figure 4**: Stretched audio



**Figure 5**: Audio with noise



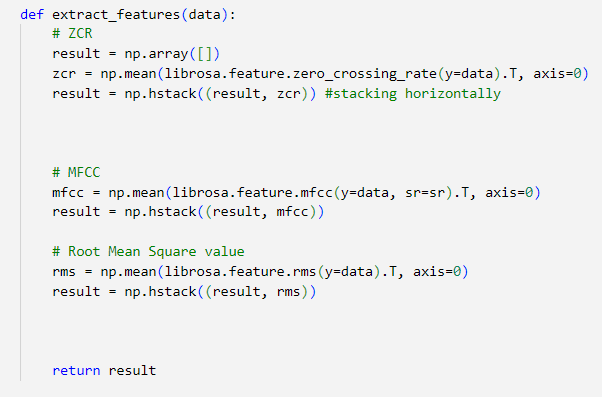
**Figure 6**: Shifted audio



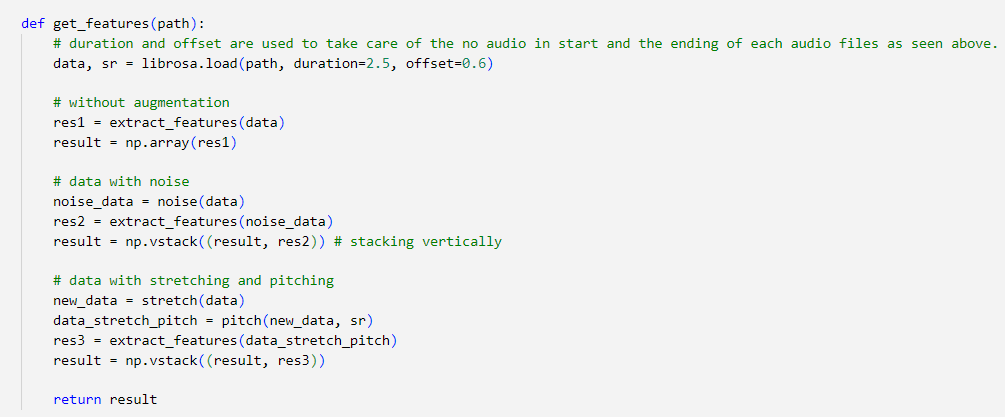
**Figure 7**: Audio with pitch

**Feature extraction:**

During the feature extraction phase, three features - Zero Crossing Rate, MFCC and Root Mean Square value - are extracted. A function called 'extract\_features' is created, with the duration for the feature extraction process set to 2.5 seconds and the offset to 0.6 seconds to accommodate the varying lengths of the data. A total of 22 features are extracted from each audio, containing one ZCR value, one Root Mean Square Value and 20 MFCC coefficients.



**Figure 8**: Features to extract

****

**Figure 9**: Feature extraction code

**Speaker diarization:**

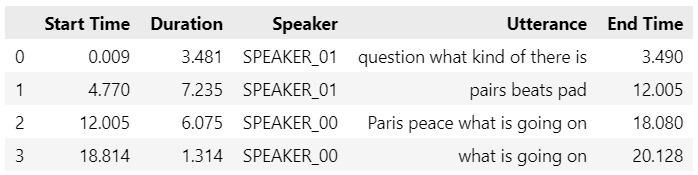
Speaker diarization is the process of identifying and separating individual speakers in an audio stream so that, in the automatic speech recognition (ASR) transcript, each speaker's utterances are identified and separated. Speaker diarization is used in scenarios where multiple speakers are involved. Here, a pre-trained speaker diarization pipeline is used to identify and separate two speakers from a given set of audio samples.

**Figure 10**: Loading pre-trained speaker diarization pipelines



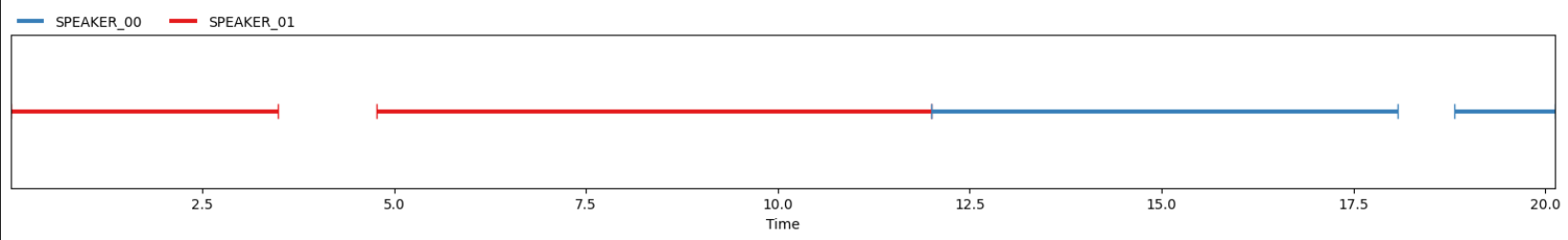
**Figure 11**: Perform diarization and save output

The integrated dataset is run through the pre-trained model and the output contains utterances of each speaker separately along with their start time and end time.



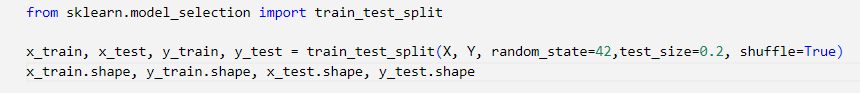
**Figure 12**: Audio segments after diarization

The result of the diarization process can be represented as a diagram indicating when each person speaks



**Figure 13**: Graph showing speaker diarization output

**Splitting the dataset for machine learning:**



**Figure 14**:Splitting into training and testing sets

The code is intended to split a given dataset into training and testing sets. This is a common practice in machine learning to assess the performance of models on unseen data and avoid overfitting, thereby enabling the development and evaluation of machine learning models. Proper data splitting helps in assessing a model's generalization performance and its ability to make accurate predictions on new, unseen data.

It takes several parameters to control how the data should be split. ‘X’ typically contains the independent variables or features that the model uses to make predictions. ‘Y’ typically contains the corresponding target values or labels that your model is trying to predict. Random state parameter is important for reproducibility. It sets a seed for the random number generator, ensuring that the same random split will be generated every time you run this code with the same seed. In this case, the random state is 42. The test size parameter determines the proportion of the data that will be allocated to the testing set. In this case, 20% of the data will be used for testing, and the remaining 80% will be used for training. Shuffling is usually set to ‘True’ to prevent any inherent order or bias in the data from affecting the model's performance.

**Speech emotion recognition:**

The emotion expressed in each audio segment obtained after speaker diarization is identified in this step. We employed multiple models, namely LSTM, CNN, and CLSTM, for emotion prediction. After testing these models, it was determined that the CNN model provided the most accurate predictions, and as a result, we selected the same. In this case, a sequential model is employed to construct the CNN.

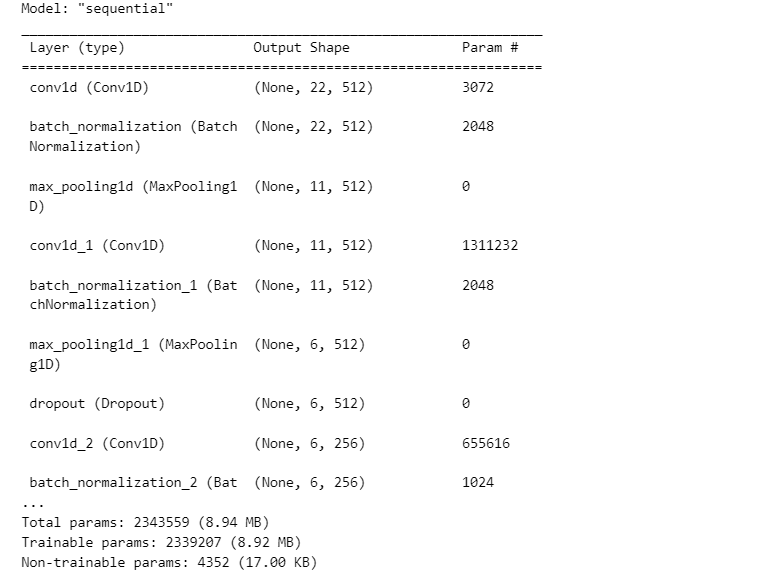


**Figure 15**: CNN Model Architecture

The CNN model uses TensorFlow library and its sub-library Keras for the classification task where Keras provides tools for building and training neural networks. A sequential neural network model is created, which is defined as a sequence of different layers, one after the other in a specific order.

It starts with convolutional layer (conv1D) which scans over the data to detect the patterns. After each convolutional layer there is a batch normalization step that helps the network to learn more effectively. This is followed by a maxpooling layer which reduces the size of the data and helps to focus on the most important features. A dropout layer is then added to prevent the overfitting during training. After several of these layers, a flatten layer is added, which takes the data and turns it into a flat list of numbers. Dense layers are traditional neural network layers where each neuron connects to every neuron in the previous layer. Finally, there is an output layer with a SoftMax activation which gives the final prediction as probabilities for different categories. The model is then configured for training using the ‘compile’ method. It specifies the optimizer, loss function and the metrics to track during training.

The summary of a neural network model provides a concise overview of the architecture, showing details about each layer, including the layer type, output shape, and the number of trainable parameters.

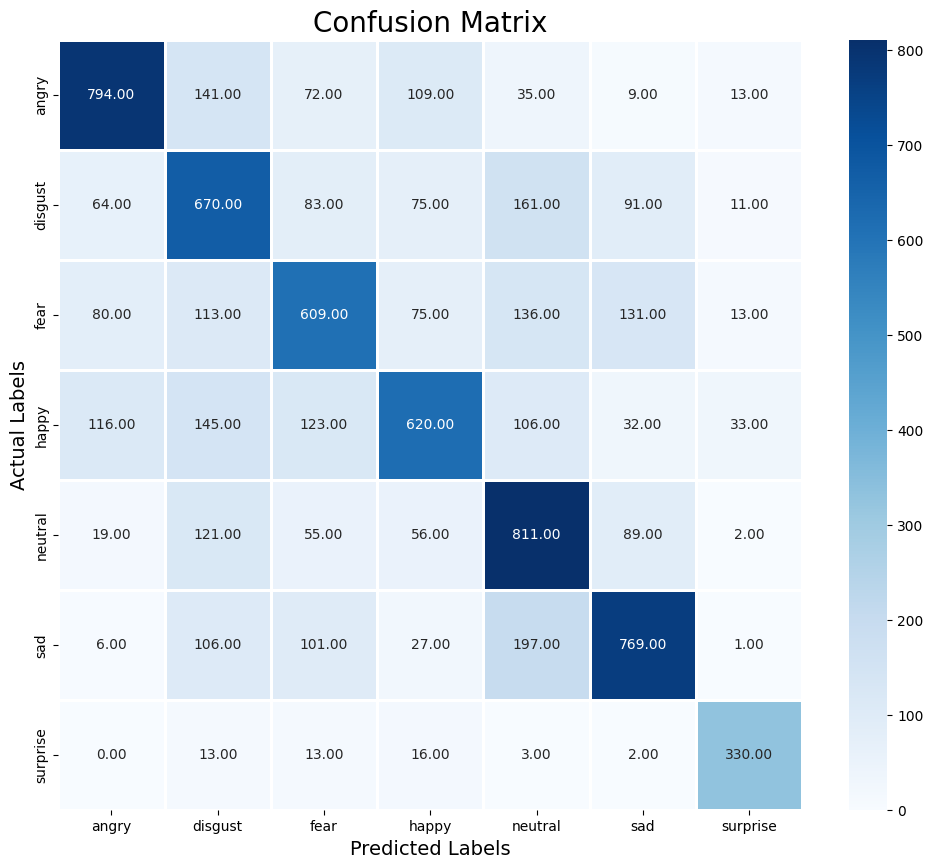


**Figure 16**: Summary of the model

**CHAPTER 3**

**RESULTS AND INSIGHTS**

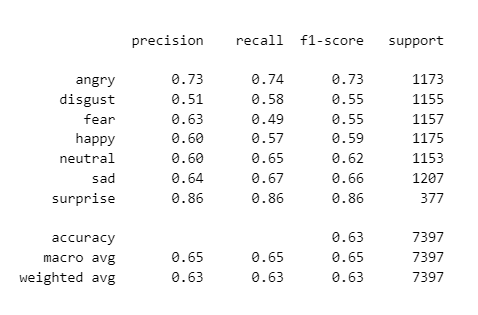
The confusion matrix generated for the model is given below:



**Figure 17:** Confusion matrix of the model

We can see that the model is more accurate in predicting surprise and angry emotions. It makes sense because audio files of these emotions differ from all the other audio files in terms of pitch, speed and more. We achieved an overall accuracy of 61% on our test data. We can improve it further by applying more augmentation techniques and using other feature extraction methods. In a heatmap, the diagonal cells from the top-left to bottom-right represents the correct predictions The darker these cells, the better your model performs. Evidently, all the diagonal cells of the confusion matrix are darker than rest of the cells, meaning the model is performing well and making predictions accurately.

The classification report obtained is given below:



**Figure 18**: Classification report

As we investigate the classification report, we can see that the model predicts the emotion “surprise” with a precision of 0.86 and this is maximum precision value in the report. This means that most of the “surprise” emotions are predicted accurately. Moreover, recall value and F1-score of the emotion are also 0.86.

Accuracy of the model is calculated by:

**Accuracy = (TP + TN) / (TP + TN + FN + FP)**

where ,

TP is True Positive

TN is True Negative

FN is False Negative

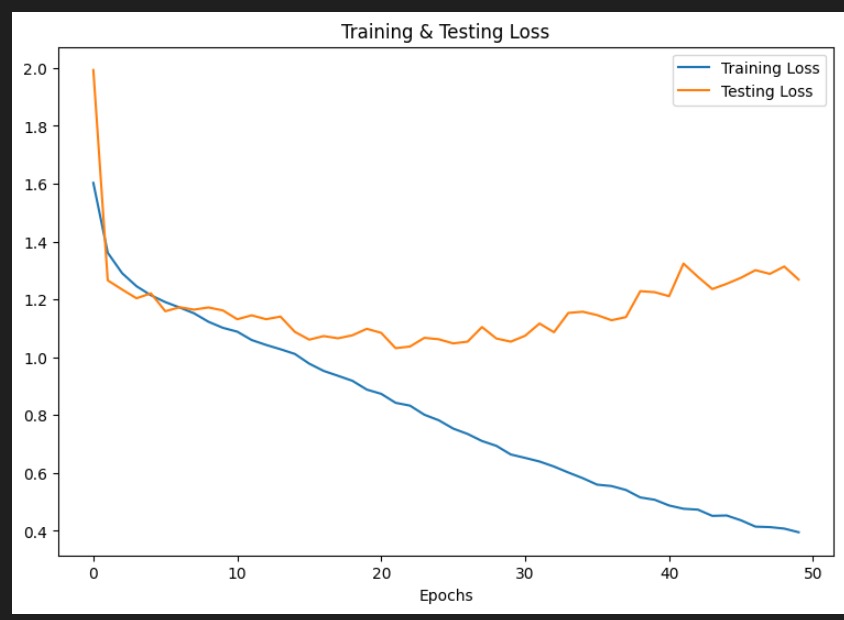
FP is False Positive

Here, the accuracy obtained is 0.63 i.e, 63%.



**Figure 19**: Predicted labels and Actual labels

The training vs testing loss is plotted in the graph given below:

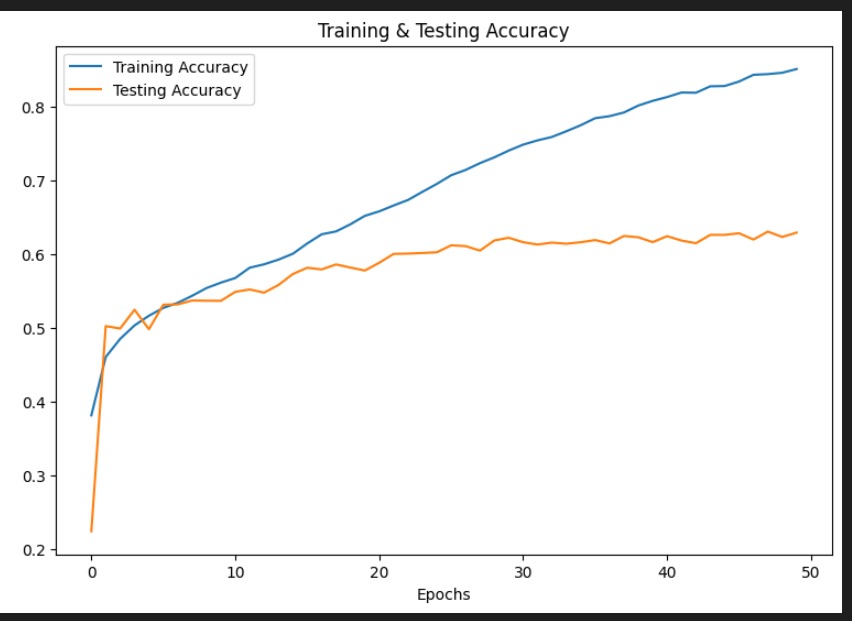


**Figure 20**: Training vs Testing loss plot

Initially, at epoch 1, training loss is comparatively high. This is because the model has not yet learned any patterns in the data. As training progresses (from epochs 2 to 50), the training loss decreases rapidly. This indicates that the model is improving its fit to the training data and is becoming proficient at recognizing patterns within it.

Similarly, the testing loss also starts at higher values during epoch 1, as the model has not been exposed to the testing data. However, as training continues (from epochs 2 to 50), the testing loss initially decreases. This is a positive sign, indicating that the model is generalizing well to unseen data. While the testing loss graph may exhibit some fluctuations, there is a slight upward trend. This suggests that there may be a risk of overfitting, where the model is becoming too specialized in fitting the training data and may not perform as well on new data.

The training vs testing accuracy graph is plotted below:



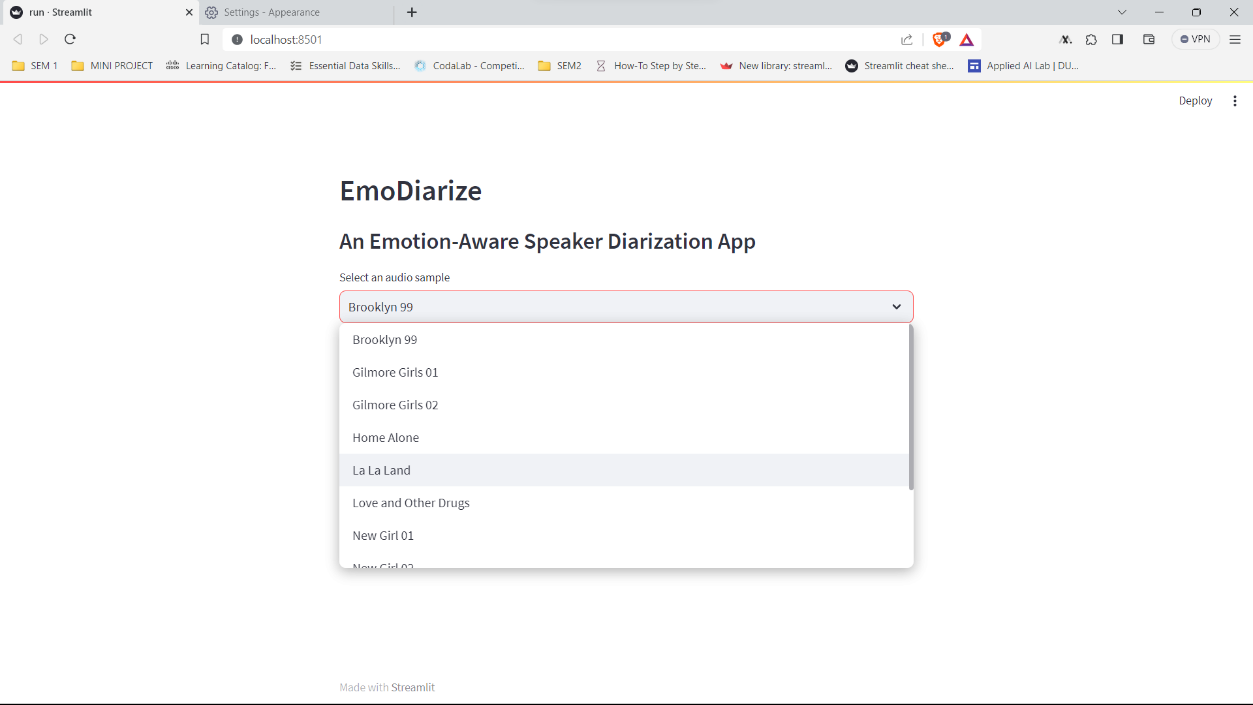
**Figure 21**: Training vs Testing accuracy plot

At the beginning of the training (epoch 1), the training accuracy is relatively low since the model has not learned about the features of the data. As the training progresses (epoch 2 to 50), the training accuracy increases rapidly. It indicates that the model is getting better at correctly predicting the emotions from the training dataset. Like the training accuracy, the testing accuracy is comparatively low at the starting of the training. As the training continues, the testing accuracy increases. This is a positive sign that the model is learning to recognize the features. As we investigate the plot, it becomes clear that the testing accuracy plot becomes almost stable by the end of the training. This means that the model is generalizing to unseen data and has learned the underlying patterns in the data.

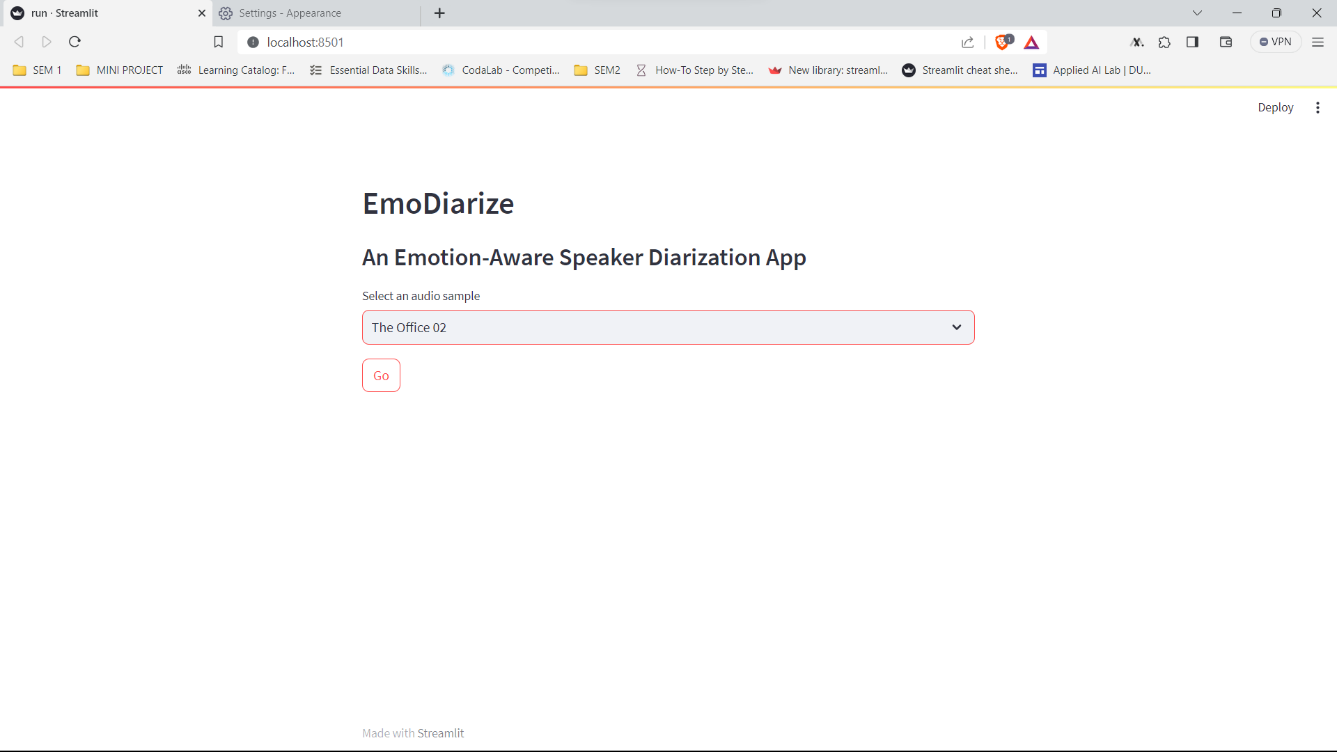
**3.2 EmoDiarize:**

**AN EMOTION-AWARE SPEAKER**

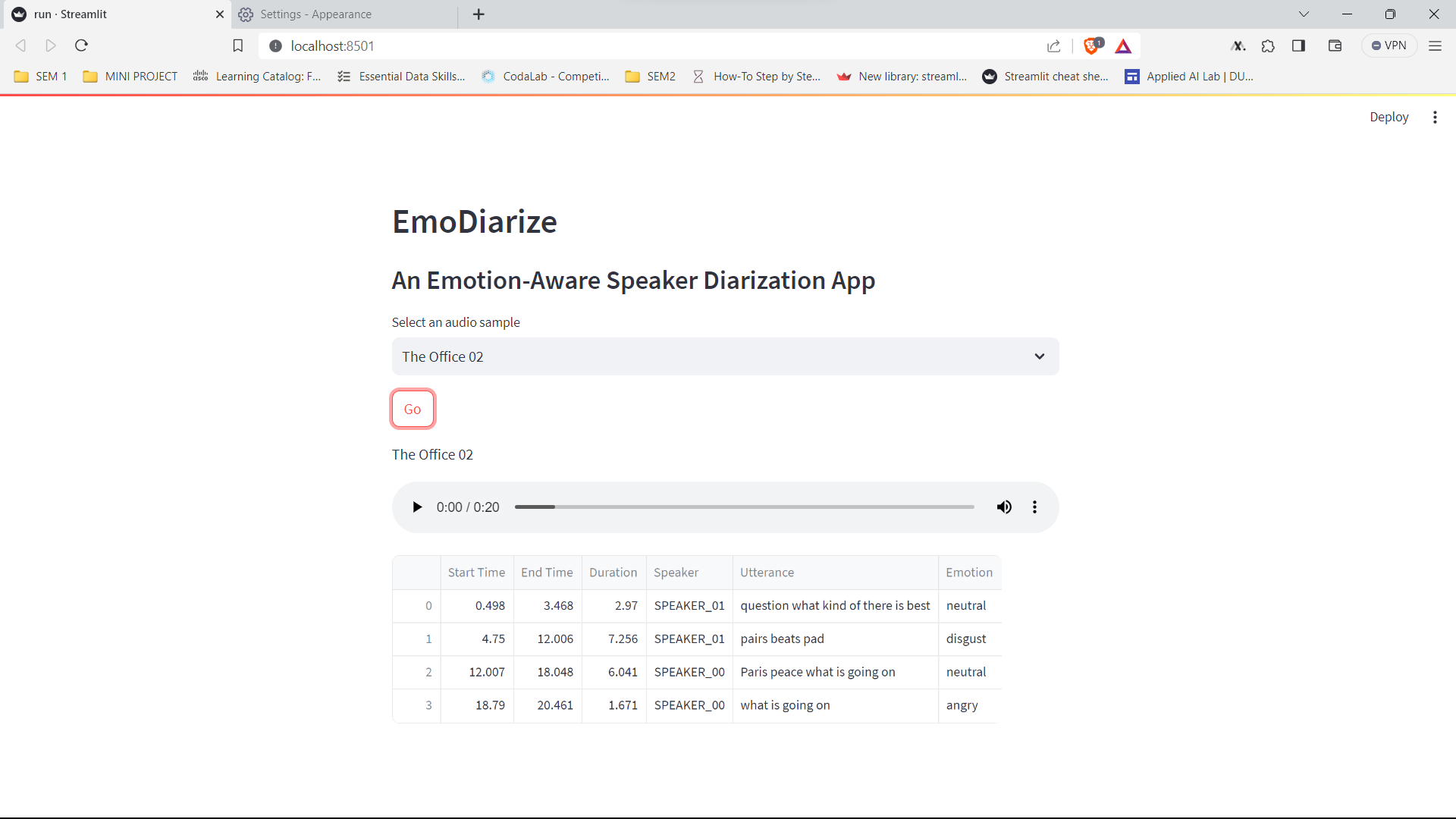
**DIARIZATION APP**



**Figure 22**: Selecting Audio from the drop-down menu



**Figure 23**: Choose an audio and click Go



**Figure 24**: Streamlit Demo Output

**3.3 CONCLUSION**

Emotions play a crucial role in how we communicate, allowing us to convey complex feelings when we speak. Our project aimed to create a smart system that can understand these emotions in speech. We developed this system by combining two technologies: one that can separate speakers in audio recordings, and another that can predict the emotions of each speaker.

This system has wide-ranging uses. For instance, in customer service centres, it can help companies understand how customers feel during interactions, enabling them to improve services in real-time. It goes beyond simple satisfaction surveys, providing a deeper insight into customer emotions. In the context of psychological care, therapists can use this system to better understand their clients' emotions during therapy sessions, allowing for more targeted and empathetic interventions. This has the potential to enhance the quality of mental health care, potentially speeding up the recovery process for clients. Our project highlights the exciting possibilities of combining audio separation and emotion prediction to improve how people interact with technology and each other.

The future of speaker diarization, which identifies and separates speakers in audio recordings, holds exciting prospects. Improved accuracy can also be expected with advanced technology like deep learning, making it more reliable in various settings. Real-time applications, like live transcription and teleconferencing, will become more efficient. Speaker diarization will expand to handle multiple languages, combine different types of media (text, video), and find uses in healthcare, education, and entertainment. Addressing privacy concerns will be crucial, and adapting to complex audio environments with multiple speakers is a priority. Customization for specific contexts, standardized evaluation metrics, and integration with language and emotion recognition will also shape the future of this technology, making it more accessible and versatile for a wide range of applications.

# 

# **REFERENCES**

1. *Speaker diarization — OpenVINOTM  documentation*. (n.d.). [https://docs.openvino.ai/2023.0/notebooks/212-pyannote-speaker-diarization-with-output.html#](https://docs.openvino.ai/2023.0/notebooks/212-pyannote-speaker-diarization-with-output.html)
2. McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. *Proceedings of the Python in Science Conferences*. <https://doi.org/10.25080/majora-7b98e3ed-003>
3. Sharma, P. (2021). Training Neural Network with Keras and basics of Deep Learning. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/11/training-neural-network-with-keras-and-basics-of-deep-learning/>
4. *Build software better, together*. (n.d.). GitHub. <https://github.com/topics/ravdess-dataset>
5. crema\_d. (n.d.). *TensorFlow*. <https://www.tensorflow.org/datasets/catalog/crema_d>
6. *Toronto emotional speech set (TESS) | TSpace Repository*. (n.d.). <https://tspace.library.utoronto.ca/handle/1807/24487>
7. *Papers with Code - SAVEE Dataset*. (n.d.). <https://paperswithcode.com/dataset/savee>
8. Khoma, V., Khoma, Y., Brydinskyi, V., & Konovalov, A. I. (2023). Development of supervised speaker diarization system based on the PyAnnote Audio Processing Library. *Sensors*, *23*(4), 2082. <https://doi.org/10.3390/s23042082>
9. Saxena, R. (2021, December 14). Who spoke when! How to Build your own Speaker Diarization Module. *Medium*. <https://medium.com/saarthi-ai/who-spoke-when-build-your-own-speaker-diarization-module-from-scratch-e7d725ee279>
10. Mostafaabdlhamed. (2023). Speech emotion recognition 97.25% accuracy. *Kaggle*. <https://www.kaggle.com/code/mostafaabdlhamed/speech-emotion-recognition-97-25-accuracy#Test-script>
11. Park, T. J. (2021, January 24). *A Review of Speaker Diarization: Recent Advances with Deep Learning*. arXiv.org. <https://arxiv.org/abs/2101.09624>
12. Park, T., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. S. (2022). A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, *72*, 101317. <https://doi.org/10.1016/j.csl.2021.101317>
13. R. (2021, March 5). *What Is Speaker Diarization? (How It Works With Real-Life Examples) | Rev*. Rev. <https://www.rev.com/blog/transcription-blog/what-is-speaker-diarization>
14. R. (2023, July 22). *Speaker diarization using Whisper ASR and Pyannote*. Medium. <https://medium.com/@xriteshsharmax/speaker-diarization-using-whisper-asr-and-pyannote-f0141c85d59a>
15. Bredin, H. (2023). pyannote. audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In *Proc. Interspeech* (Vol. 2023).
16. Amersh. (2022). *Speech Emotion Recognition.*

<https://www.studocu.com/in/document/chandigarh-university/disruptive-technologies-2/ser-final-report/27338335>