Speech Emotion Recognition

Akarsh Mahajan - 20ucc011@lnmiit.ac.in
Akshat Mathur - 20ucc014@lnmiit.ac.in
Vivek Satishrao Nilkanthwar - 20ucc116@lnmiit.ac.in

Abstract

Speech is the method of expression that comes most naturally to us. We rely entirely on our speech because it conveys our emotions and reveals how we will act toward others. When we have to face someone or present ourselves to someone, we understand its significance. In the modern era, everything is done online. In their stead, we have M.L.-trained robots or bots, which can perform all human tasks except recognize emotions. In this project, we trained our M.L. model with two labeled audio datasets because it is challenging for humans to recognize human emotion from a person's voice, and it is even more challenging for M.L. to recognize emotions from audio alone; We were able to train this model to 70% accuracy because of this. Also, we are uploading the speech emotion data on Thingspeak in real time from which further analysis and visualization are performed.

1.1 Introduction

In order to train any M.L. model, we need features of the input data, which in our project is audio or voice.

Three classes can be used to categorize the speech:

- Lexical features - include the vocabulary used
- Visual features - includes the speaker's expressions
- Acoustic features - includes the characteristics of the sound (such as frequency, pitch, and loudness)

We can identify an emotion from a speech by extracting and analyzing any of one or many features; however, selecting lexical features requires a further step of text extraction from the audio transcript. However, we can extract acoustic features from audio in real-time and it doesn't require any additional steps, unlike visual features where we will have to convert video to speech or text, which is not always possible. Therefore, we decide to analyze our speech using acoustic features.

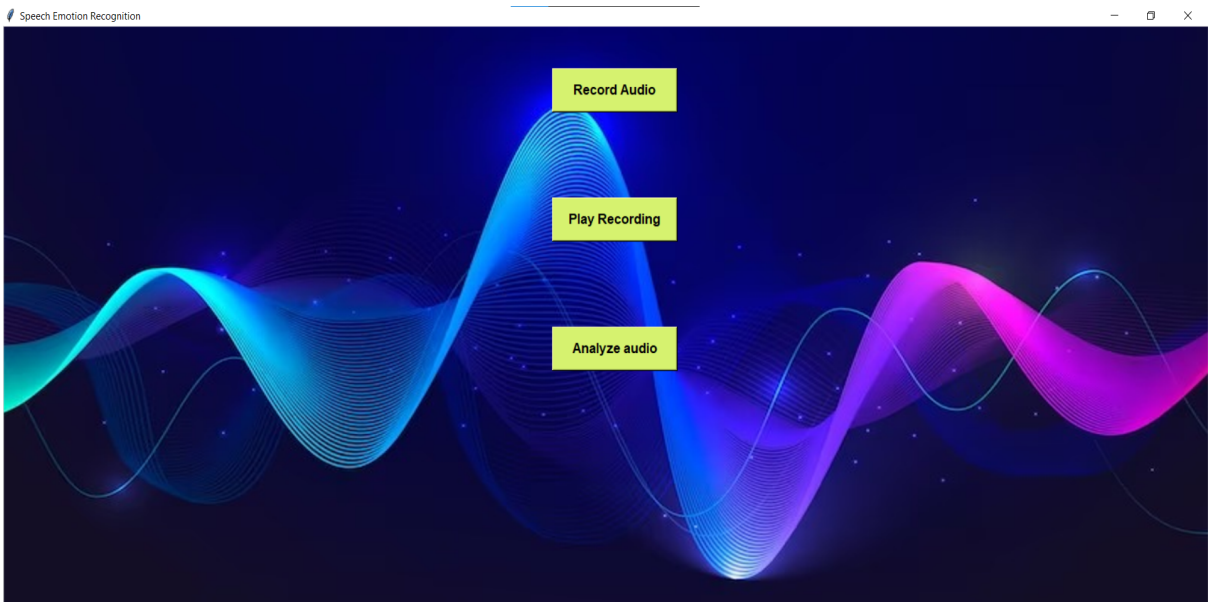


Figure 1.1 Interface of the Speech Emotion Detector

A quality dataset is required for every M.L. model. As a result, we employed a RAVDESS dataset to increase efficiency.

- RAVDESS - This dataset contains approximately 1500 audio files that were provided by 24 different actors.

We chose MLP(multi-layer perceptron) Classifier over other classification algorithms and neural networks because it trains using Backpropagation, which makes quick predictions after training.

We have used several libraries including librosa for audio processing, TensorFlow, keras for the machine learning model and matplotlib and numpy for data visualization.

Also, we have used Thingspeak which is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud.

You can also send data to ThingSpeak from your devices and create instant visualizations of live data in real-time.

The applications of speech emotion detection are numerous and diverse. It can be used to enhance human-computer interaction and aid in the development of virtual assistants, chatbots, and other conversational agents. It can also be used in fields such as marketing, customer service, education, and mental health to understand and respond to human emotions and needs.

SWOT Analysis: -

1.2 Methodology

In this project, deep learning is being used to identify emotions including calm, fear, anger, sadness, and happiness. We need an emotion analyzer that can detect our feelings and make recommendations based on our emotions because personalization is essential in the modern world.

We needed a dataset to train and test the project, so we used the RAVDESS dataset, also known as the Ryson Audio-Visual Database of Emotional Speech. This dataset contains 500 audio files input from 24 different actors, 12 of whom are male and 12 of whom are female, who recorded brief audios in 5 different emotional states, including calm, happy, sad, angry, and fearful.

This is a Python-based project that imports several libraries, including **librosa** for audio processing, **TensorFlow** and **Keras** for the machine learning model, and **matplotlib** and **numpy** for data visualization. It defines several functions for recording and playing audio, and for classifying the emotion in the audio. Here we used the **Tkinter** library to create the user interface, and the **pyaudio** library to record and play audio. The recorded audio is saved as a .wav file, and then the speech emotion recognition model is used to classify the

emotion in the audio.

The user interface is created using **Tkinter** and consists of three buttons for recording, playing, and analyzing the emotion in the recorded audio. When the 'Analyze audio' button is pressed, the script loads a pre-trained machine learning model for speech emotion recognition, reads the recorded audio from the .wav file, processes it with librosa, and feeds it to the model for classification. The recognized emotion is then displayed on a label on the user interface.

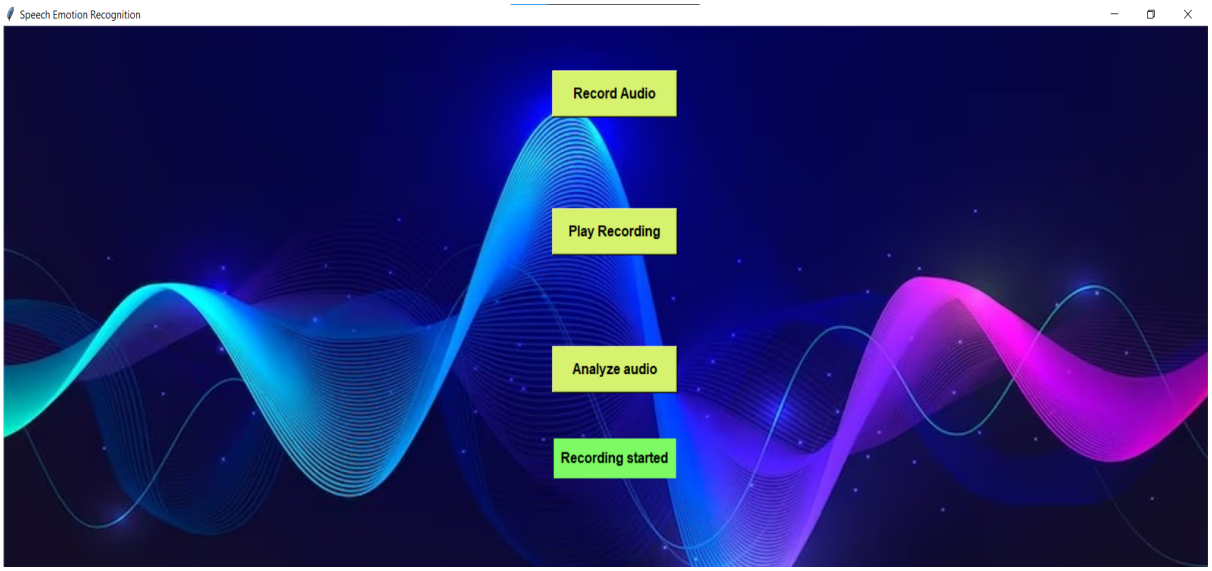


Figure 1.2 Recording audio

Figure 1.3 Recording complete

The novelty added to the project is that we send the data to **ThingSpeak** for visualization. First, it defines the Channel ID and Write API Key for the ThingSpeak channel you want to send data. Then, it defines the BASE_URL for the ThingSpeak API, which includes the Write API Key. Then, it sends the data to ThingSpeak using a POST request with the requests.post() method. Here we use visualization tools to create graphs, charts, and maps to display our data in real time. This will help us to gain insights and make informed decisions based on the data.

The ThingSpeak Channel ID for our Channel is: **2124840**

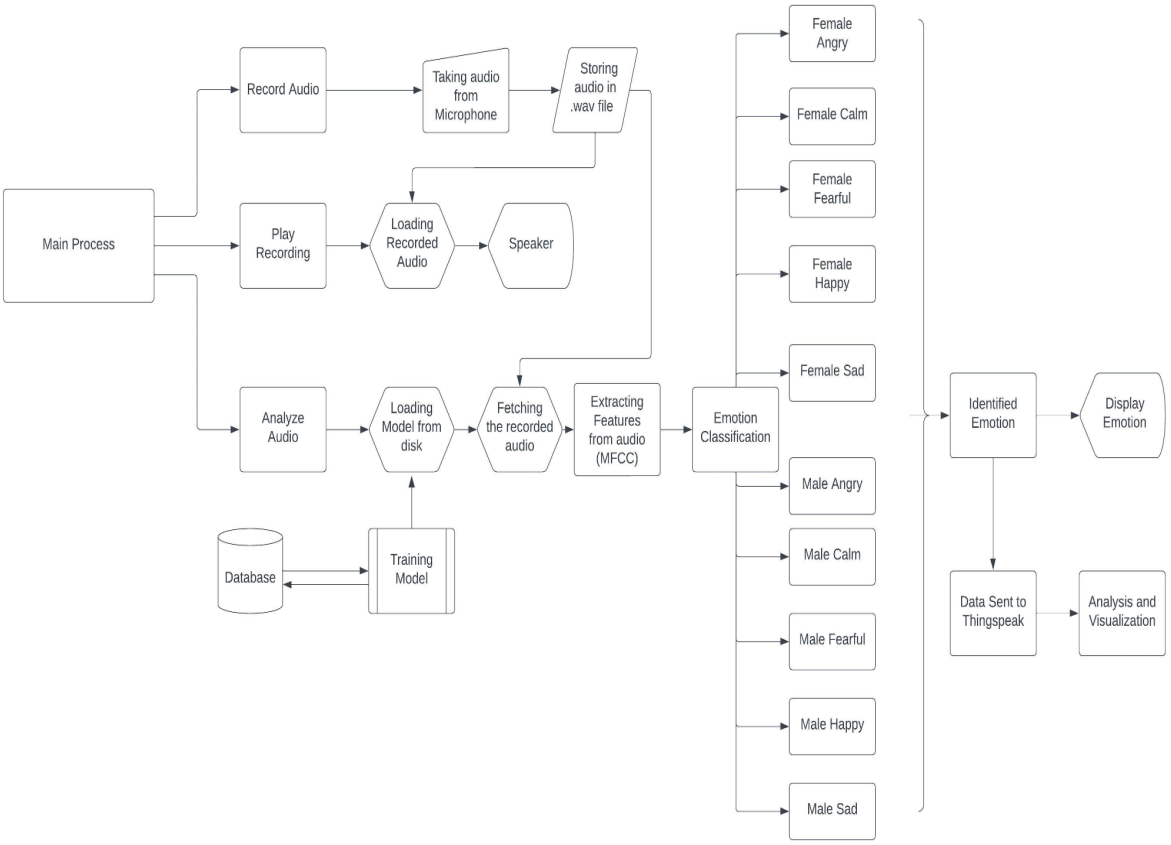
The ThingSpeak Write API Key for our Channel is: **TMOYUJPJ1KUESPVY**

MATLAB Code

```
readChannelID = 2124840;

% Emotion Field ID
EmotionFieldID = 1;

% Channel Read API Key
% If your channel is private, then enter the read API
% Key between the '' below:
readAPIKey = 'N4BQV1QRT3TAYVAC';

tempF = thingSpeakRead(readChannelID,'Fields',EmotionFieldID,...
'NumMinutes',15*60, 'ReadKey',readAPIKey);

% Create an empty categorical array of the same size as tempF
emotions = categorical(zeros(size(tempF)), [0:9], ...
    {'Female Angry', 'Female Calm', 'Female Fearful', 'Female Happy', 'Female Sad', ..
    'Male Angry', 'Male Calm', 'Male Fearful', 'Male Happy', 'Male Sad'});

for i = 1:length(tempF)
    % Store the emotion value in emotions array
    if tempF(i) == 0
        emotions(i) = "Female Angry";
    elseif tempF(i) == 1
        emotions(i) = "Female Calm";
    elseif tempF(i) == 2
        emotions(i) = "Female Fearful";
    elseif tempF(i) == 3
        emotions(i) = "Female Happy";
    elseif tempF(i) == 4
        emotions(i) = "Female Sad";
    elseif tempF(i) == 5
        emotions(i) = "Male Angry";
    elseif tempF(i) == 6
        emotions(i) = "Male Calm";
    elseif tempF(i) == 7
        emotions(i) = "Male Fearful";
    elseif tempF(i) == 8
        emotions(i) = "Male Happy";
    elseif tempF(i) == 9
        emotions(i) = "Male Sad";
    end
end

histogram(emotions);
xlabel('Emotion');
ylabel('Number of Measurements\newline for Each Emotion');
title('Histogram of Emotion Variation');

```

Figure 1.4 Matlab Code for Visualizing Emotions corresponding to real-time data in histogram

1.2.1 Call Flow Diagram



1.3 Results

1.3.1 Emotion Detection on GUI

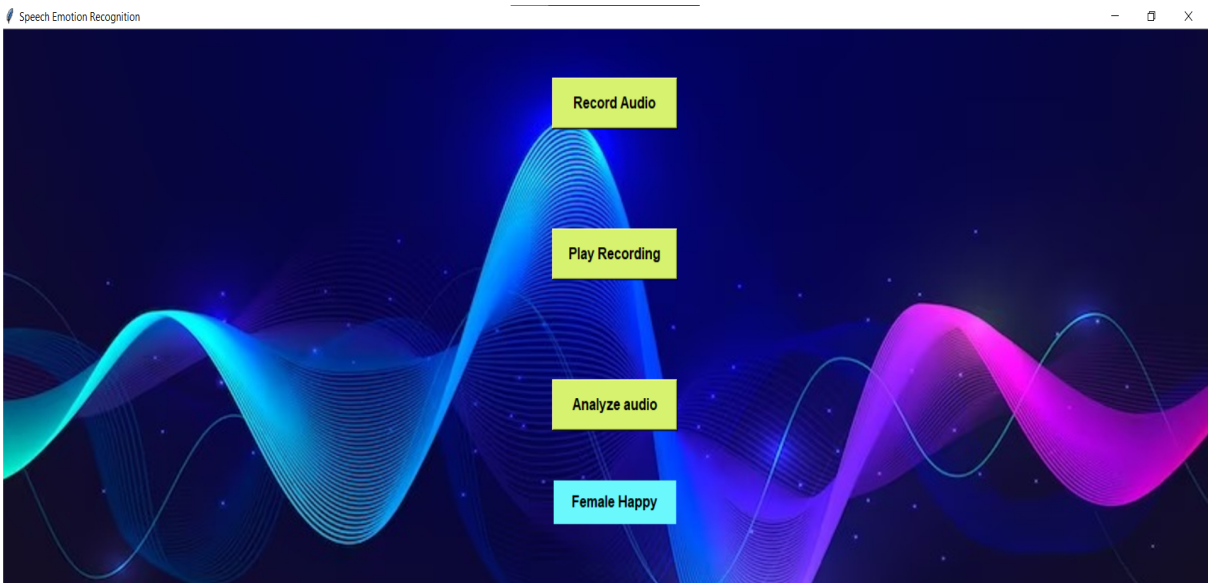


Figure 1.5 Result for Emotion Detection

1.3.2 Data Visualization and Analysis on ThingSpeak: -

The user is using an emotion detector to track their emotional state over the course of a day. The detector is mapping each emotion of male & female to an index i.e. **0- Female Angry, 1-Female Calm, 2-Female Fearful, 3-Female Happy, 4-Female Sad, 5-Male Angry, 6-Male Calm, 7-Male Fearful, 8-Male Happy, 9-Male Sad.**

Now we are plotting them on a line graph, with time on the x-axis and the emotion index on the y-axis. By the end of the day, the line would show the overall trend of the user's emotions over the course of the day. This type of visualization could be helpful for users to understand patterns in their emotional state and to identify triggers or events that might be affecting their emotions. It could also be used by therapists or mental health professionals to track the emotional progress of their patients over time.

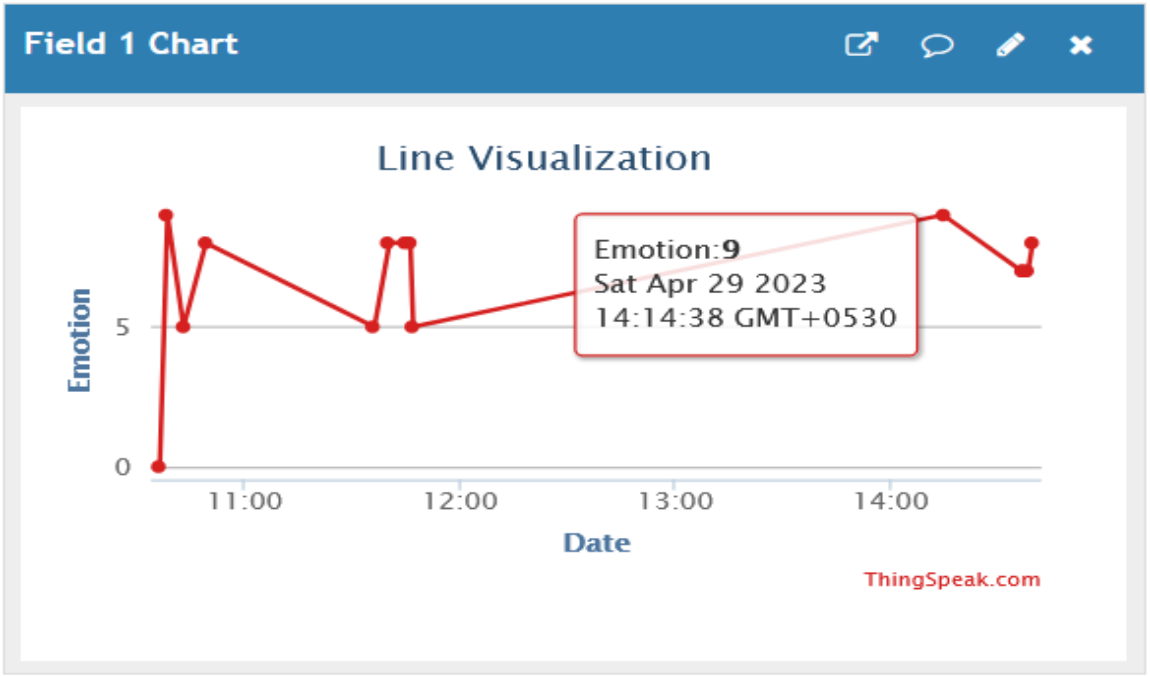


Figure 1.6 Real Time Line Visualization of Data

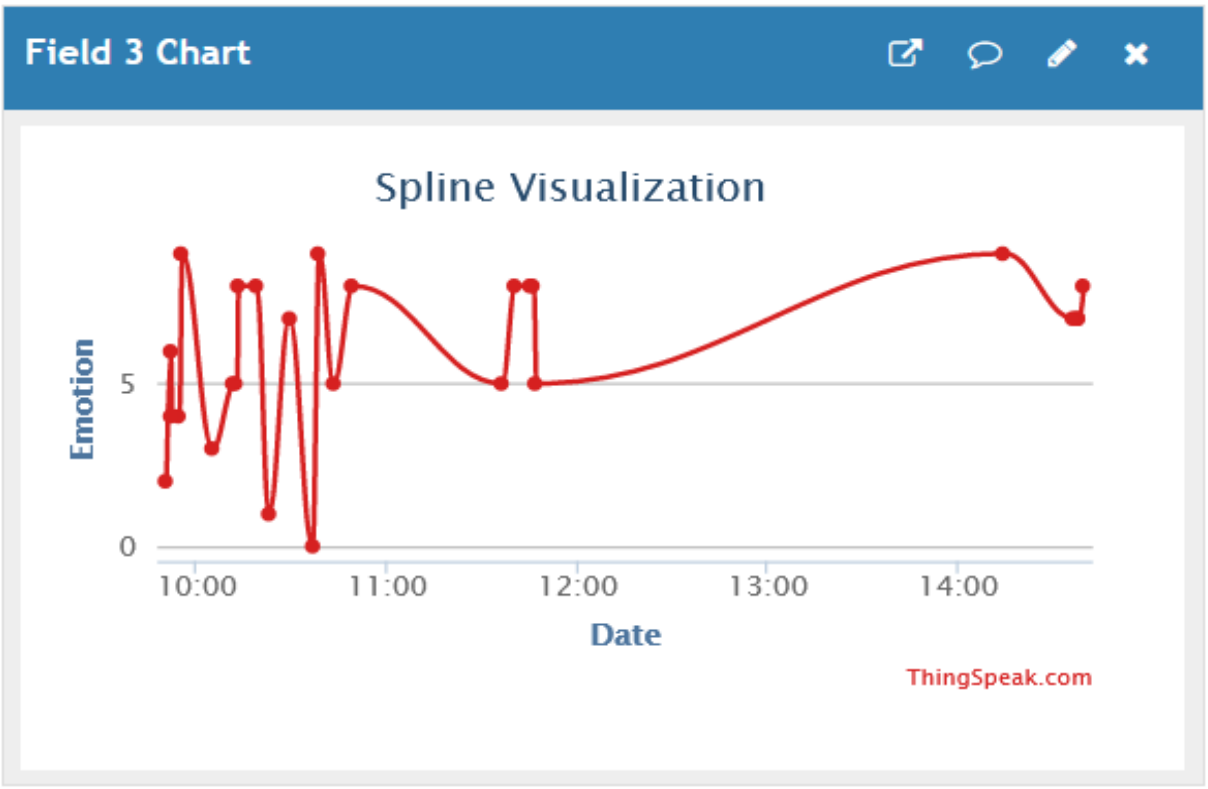


Figure 1.7 Real Time Spline Visualization of Data

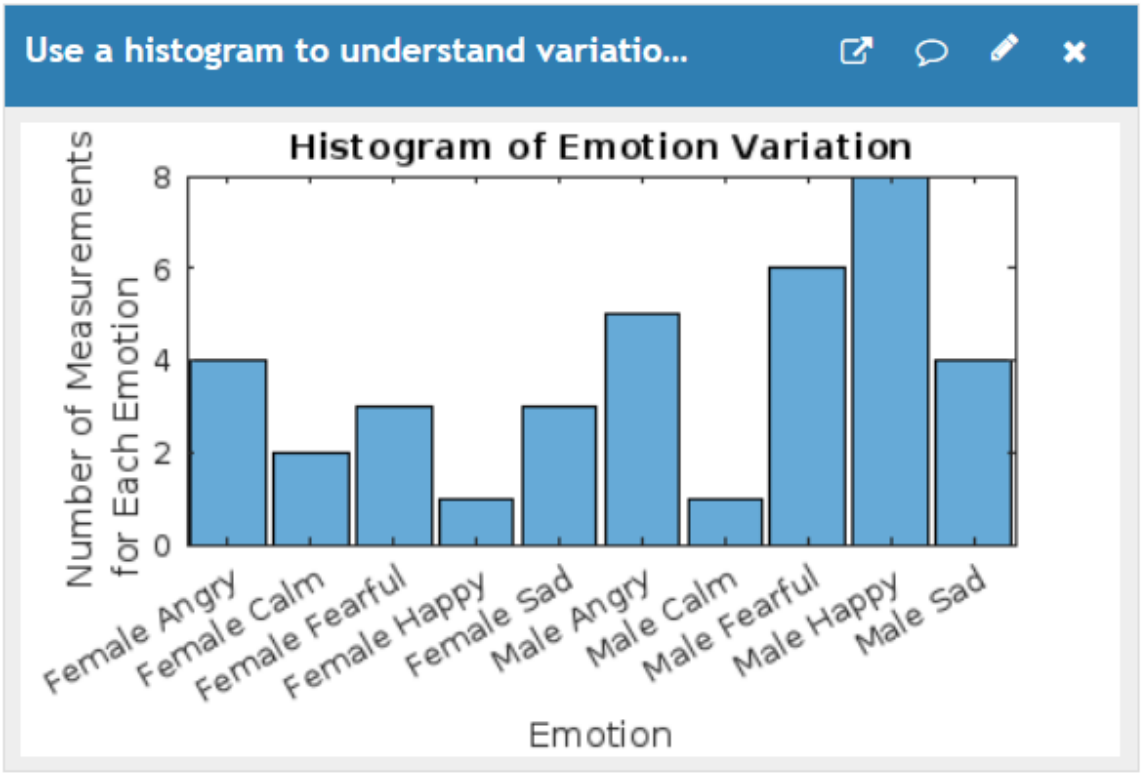


Figure 1.8 Histogram representation of Frequency of Emotions detected

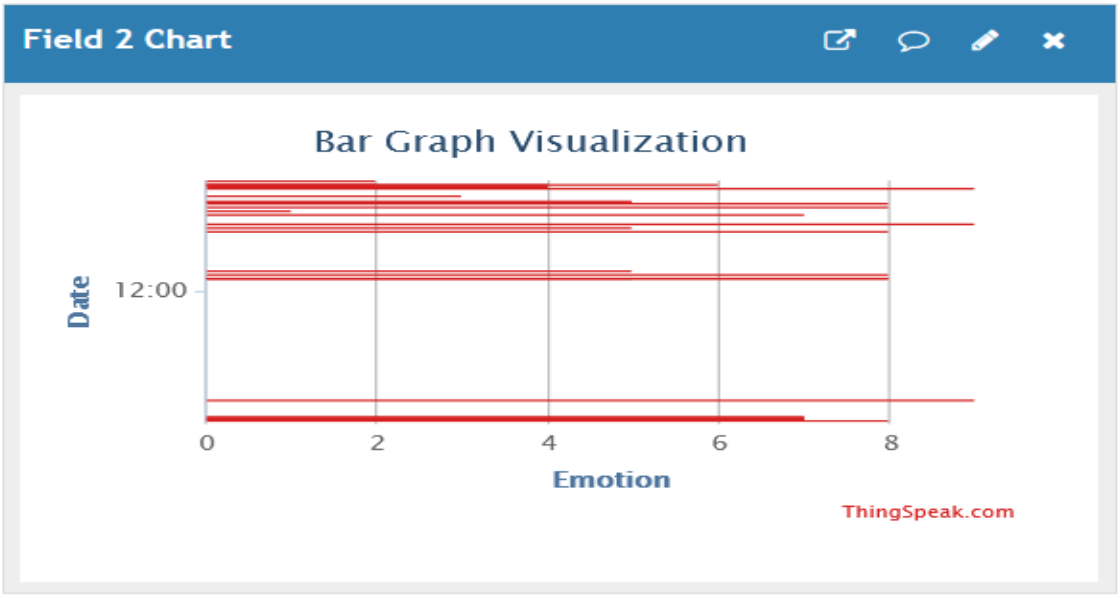


Figure 1.9 Real Time Bar Graph Visualization of Data

Our Emotion Detector can distinguish between male and female when the audio input belongs to a male then the blue light will glow in Male Gender Detector and when the audio input belongs to a female then the pink light will glow in Female Gender Detector.

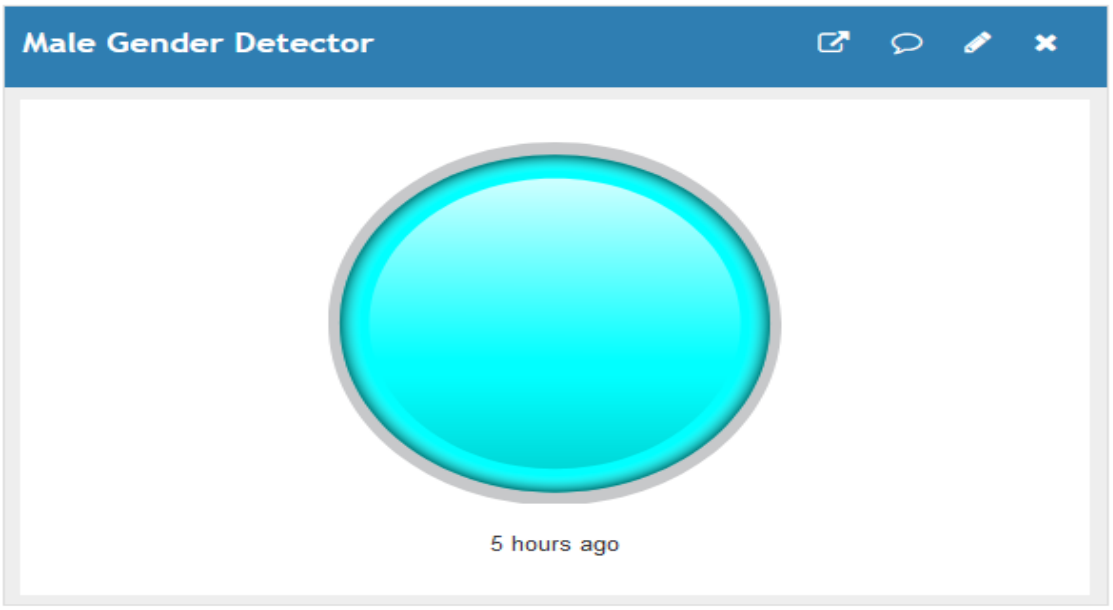


Figure 1.10 Male Gender Detector

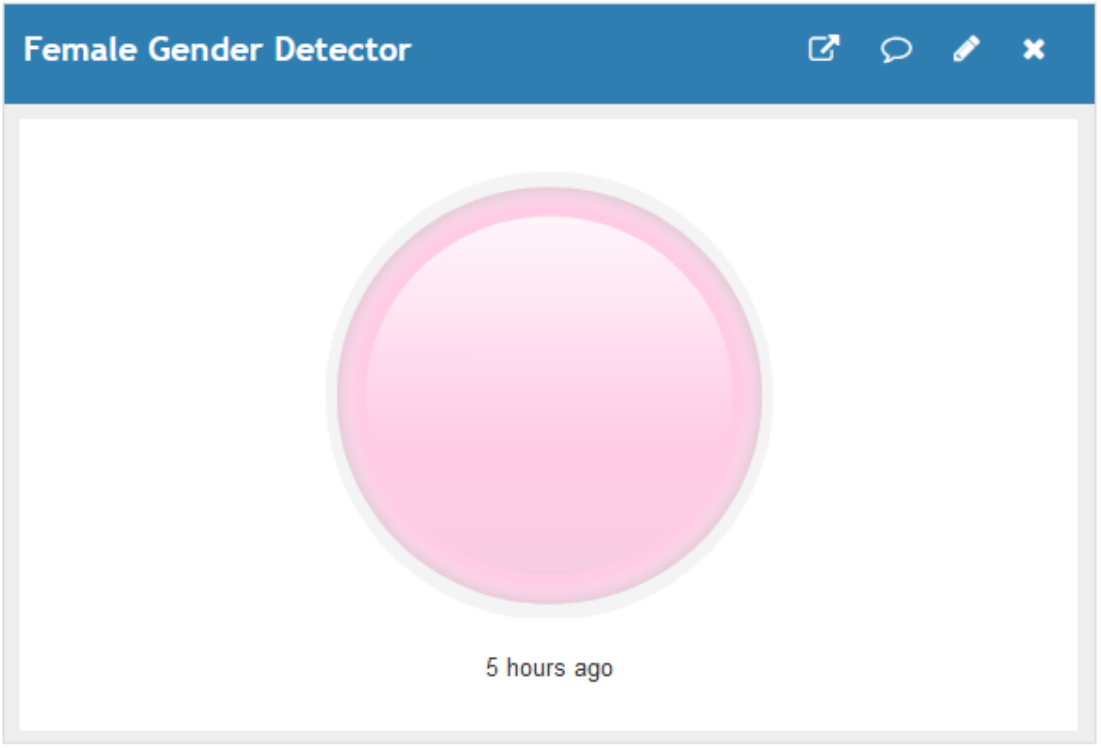


Figure 1.11 Female Gender Detector

1.3.3 Applications

Data on emotion variation gathered by a speech emotion detector can be used for various purposes, including

Research: The data can be used to conduct research on emotional responses in different contexts, such as analyzing emotional responses to specific words or phrases. This can help to improve our understanding of emotions and their impact on communication.

Marketing: Emotion variation data can be used by marketing professionals to understand how their customers respond emotionally to their products, services, or marketing messages. This can help them to create more effective marketing campaigns and messaging.

Education: The data can be used in educational contexts to understand how students respond emotionally to different learning materials or teaching methods. This can help educators to adapt their teaching approach to better meet the emotional needs of their students.

Healthcare: The data can be used in healthcare settings to monitor patients' emotional states and track changes over time. This can be particularly useful in mental health treatment, where emotional changes can be important indicators of treatment progress.

Personalization: Emotion variation data can be used to personalize interactions with users, such as chatbots or virtual assistants. This can help to create more natural and empathetic interactions that better meet the emotional needs of users.

1.4 Conclusion

In this project, we've demonstrated how to train an M.L. model to listen to our speech and categorize it according to various emotions. This model is 100% accurate at differentiating between male and female voices and it is 70% accurate at identifying emotions from the audio input file.

Following efforts can be taken to increase the model's reliability and accuracy:

- Figuring a way to silence the noise and create a quiet place.
- Assemble more data on various voice accents
- Concentrate on various feature extraction techniques

We have successfully uploaded real-time emotion data from Speech Emotion Detector to ThingSpeak and further analyzed that data by using various visualization techniques and graphs.

Overall, data on emotion variation can be a valuable resource for understanding and improving communication, marketing, education, healthcare, and personalization.

Acknowledgement: - We want to thank Dr.Abhishek Sharma for helping us with the project and serving as our mentor. Only because of him, did we have the opportunity to work on this project and working with him and learning more about this field was a wonderful experience.

References: -

Tkinter: -

[1.] “Graphical User Interfaces with Tk”, https://docs.python.org/3/library/tk.html

[2.] “Tkinter Modules”, https://docs.python.org/3/library/tkinter.html#tkinter-modules

[3.] “The Tkinter Cookbook ”, https://www.dvlv.co.uk/pages/the-tkinter-cookbook.html

Database: -

[4.] “RAVDESS Emotional Speech audio”, https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio

Python Models: -

[5.] “Machine Learning - Train/Test”, https://www.w3schools.com/python/python_ml_train_test.asp#:~:text=Train%2FTest%20is%20a%20method,model%20using%20the%20training%20set

[6.] “Machine Learning Tutorial”, https://www.geeksforgeeks.org/machine-learning/

Keras: -

[7.] “The Model class”,https://keras.io/api/models/model/

[8.] “Keras layers API”,https://keras.io/api/layers/

[9.] “Callbacks API”,https://keras.io/api/callbacks

ThingSpeak: -

[10.] “Analyze Your Data”, https://in.mathworks.com/help/thingspeak/analyze-your-data.html