EMOTION IDENTIFICATION USING AUDIO FILES

A PROJECT REPORT

Submitted by

GOKUL V	**(1905027)**
GOKUL PRASANTH R	**(1905028)**
KISSORE KUMAR M	**(1905040)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

SRI RAMAKRISHNA ENGINEERING COLLEGE

[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Reaccredited by NAAC with ‘A+’ Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and All Eligible Programmes Accredited by NBA]
Vattamalaipalayam, N.G.G.O. Colony Post,

COIMBATORE – 641 022

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

16IT266 FINAL YEAR PROJECT

Certified that this project report **“Emotion Identification using audio files”** is the bonafide work of **“Gokul V, Gokul Prasanth R** and **Kissore Kumar M”** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr.M.Senthamil Selvi	Mrs. N. Saranya
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor and head,	Assistant Professor (Sl.G),
Information Technology,	Information Technology,
Sri Ramakrishna Engineering College,	Sri Ramakrishna Engineering College,
Coimbatore-641022.	Coimbatore-641022.

Submitted for the Project Viva-Voce Presentation held on ____________________

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We put forth our heart and souls to thank the Almighty for being with us through our achievements and success. We would like to express our unfathomable thanks to our esteemed and Honorable Managing Trustee **Thiru.D.Lakshminarayanaswamy** and Joint Managing Trustee **Thiru.R.Sundar** for giving us the chance to be part of this elite team at Sri Ramakrishna Engineering College, Coimbatore.

We would like to express our sincere thanks to our honorable Principal **Dr.N.R.Alamelu** for the facilities provided to complete this project.

We take the privilege to thank the Head of the Department of Information Technology, **Dr.M.Senthamil Selvi,** for her consistent support, timely help and valuable suggestions during the entire period of our project.

We wish to convey our special thanks to our academic coordinator,**Dr.Preethi Harris,** Associate Professor, Information Technology for her consistentsupport, timely help and valuable suggestions during the entire period of our project.

We would like to express our sincere thanks to our Project coordinator **Dr.J.Angel Ida Chellam,** Associate Professor, Department of Information Technology for his valuable support in the completion of this project.

We would like to express our sincere thanks to our project guide **Mrs.N.Saranya,** Assistant Professor(Sl.G), Department of Information Technology, for her valuable support in the completion of this project.

We extent our sincere gratitude to all the teaching and non – teaching staff members of our department who helped us during our project.

ABSTRACT

Speech is the most natural way to express human emotions. As emotions play a vital role in communication, the detection and analysis of the same is importance in today’s digital world. Emotion Recognition is task of automatically identifying the emotional state of a speaker based on their speech signal. Explore various feature extraction techniques and machine learning algorithms to build a model that can accurately classify speech samples into different emotions. In this work a machine learning-based system was used for recognizing emotions from audio. The results of this work, will contribute to the advancement of human-computer interaction by enabling machines to understand and respond to human emotions in a more natural way. With the increasing availability of large datasets and advances in machine learning techniques, emotion recognition using audio is expected to improve and find new applications in the future. In this project, the multi-layer perceptron model was improved with 84% accuracy.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	iv
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
1	**INTRODUCTION**	1
	1.1 ARTIFICIAL INTELLIGENCE	1
	1.2 MACHINE LEARNING	2
2	**LITERATURE SURVEY**	6
3	**PROJECT DESCRIPTION**	9
	3.1 INTRODUCTION	9
	3.2 TERMINOLGIES	9
	3.3 PROPOSED SYSTEM	10
	3.4 BLOCK DIAGRAM	11
	3.5 WORKING PRINCIPLE	12
	3.6 ADVANTAGES	15
	3.7 APPLICATIONS	17
4	**RESULT**	19
5	**CONCLUSION AND FUTURE SCOPE**	22
	APPENDICES	
APPENDIX 1	**SOFTWARE AND MODULE DESCRIPTION**	23
APPENDIX 2	**SOURCE CODE**	29
	REFERENCES	40

LIST OF FIGURES

Fig No.	Name	Page No.
1.1	Speech Emotion Recognition	1
1.2	Multi-Layer	5
3.1	Block Diagram	11
4.1	Dataset_Image	19
4.2	Input_Image	20
4.3	Output_Image	20

LIST OF TABLES

Table No.	Table Name	Page No.
4.1	Accuray_measure	21

CHAPTER 1

INTRODUCTION

Speech is the most natural form of expression for humans. Being able to recognise and analyse emotions is essential in today's digital age of distant communication because they are so vital in communication. Emotion identification is challenging due to emotions' subjectivity. There is no prevailing consensus over how to assess or categorise. The sort of emotion can be predicted from audio in this case using machine learning algorithms. The features of the audio are extracted using a package called librosa.

Fig 1.1 Speech Emotion Recognition

1.1 Artificial Intelligence

Artificial intelligence is a technique for teaching a computer, computer controlled robot, or software to think effectively in the same way that humans do. AI is achieved by examining human brain patterns and analysing the cognitive process. These investigations result in the development of intelligent software and systems. There are four types of AI. They are

Purely Active

These machines have no data or memory to work with and specialise in only one area of work. In a chess game, for example, the machine observe moves and makes the best decision feasible to win.

Limited Memory

These machines collect previous info and kept in their memory. They have sufficient memory or experience to make sound choices, but their memory is limited. For example, based on the location data collected, this machine can recommend a restaurant.

Theory of mind

This type of AI can comprehend thoughts and feelings and interact socially. A machine of this sort, however, has yet to be built.

Self aware

The next iteration of these new technologies will be self-aware machines. They will be sentient, intelligent, and cognizant.

1.2 Machine Learning

Machine learning is a sub-field of artificial intelligence (AI). Understanding data structure and fitting into models that people can comprehend and use is the general goal of machine learning. Machine learning is a branch of computer science, but distinct from conventional computational methods. Algorithms in conventional computing are collections of explicitly designed instructions used by computers to compute or solve problems. Instead, machine learning algorithms let computers learn from data inputs and apply statistical analysis to produce numbers that fall inside a given range.

This allows computers to automatically automate decision-making processes based on data inputs by creating models from sample data. Machine learning is classified into four types based on the methods and ways of learning

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning

Supervised Machine Learning

Supervised machine learning is built on supervision, which means that in the supervised learning method, we train the machines using the "labelled" dataset, and the machine forecasts the output based on the training. The labelled data in this case indicates that some of the inputs have already been transferred to the output. To put on another way, we first train the machine with the input and matching output, and then we ask the machine to predict the output using the test dataset.

Unsupervised Machine Learning

Unsupervised learning differs from supervised learning in that there is no need for supervision, which implies that in unsupervised machine learning, the computer is trained using an unlabelled dataset and predicts the output without supervision. Unsupervised learning involves training models with data that is neither classified nor labelled, and then allowing the model to operate on that data without supervision.

Semi supervised Learning

Semi-supervised learning is a form of machine learning algorithm that falls somewhere between supervised and unsupervised learning, which bridges the gap between Supervised (with labelled training data) and Unsupervised (with no labelled training data) learning algorithms by combining labelled and unlabelled datasets during the training phase. Although semi-supervised learning works on data with a few labels and is the middle ground between supervised and unsupervised learning, mostly unlabelled data. Labels are expensive, but for corporate reasons, may only need a few labels. It differs from supervised and unsupervised learning in that are based on the presence or lack of labels.

Reinforcement Learning

Reinforcement learning is a feedback-based process in which an AI agent (a software component) explores its surroundings automatically by hitting and trailing, taking action, learning from experiences, and enhancing its performance. Because the agent is rewarded for every good action and punished for every bad action, the aim of the reinforcement learning agent is to maximise the rewards.

Librosa

A helpful Python music and sound analysis package called Librosa aids programmers in creating apps for working with sound and music document designs. This Python package for sound and music analysis is mostly used when working with sound data, such as in the music age (using Lstm's), Automatic Speech Recognition.

Multilayer Perceptron

The feed forward neural network is supplemented by the multilayer perceptron. The input layer, output layer, and hidden layer are the three different kinds of layers that make up. The input layer is where the input signal for processing is received. The output layer does the necessary tasks, such as classification and prediction. The real computational engine of the MLP consists of arbitrary number of hidden layers that are sandwiched between the input and output layers. The data flows from the input to the output layer in a forward direction, much like a feed forward network in an MLP. With the help of the back propagation learning algorithm, the MLP's neurons are taught. MLPs can resolve issues that are not linearly separable because they are made to approximate any continuous function.

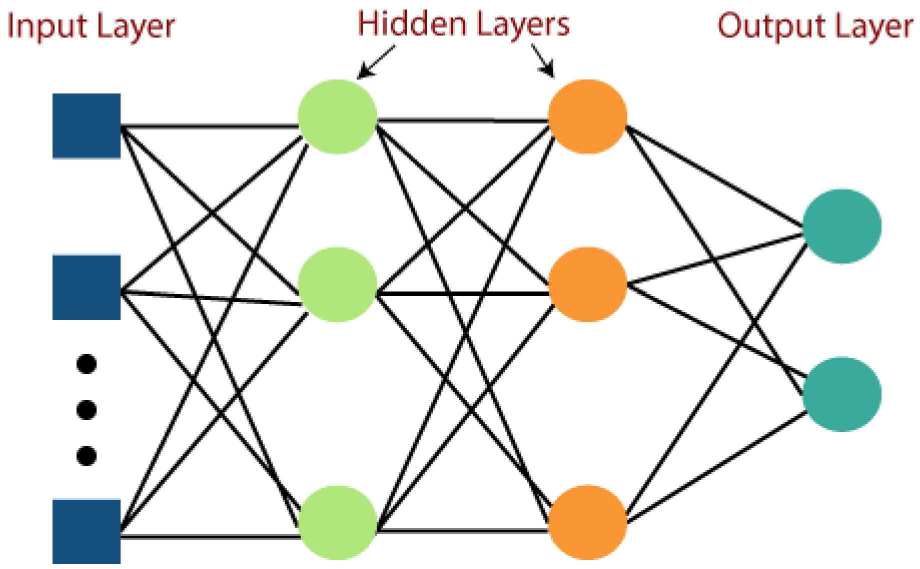


Fig 1.2 Multi-Layer Perceptron

CHAPTER 2

LITERATURE SURVEY

There are a lot of research works in this area, which helps to get deep insights about speech emotion recognition. Also, there are works which perform a single action using some dedicated computing devices. Some of the related contribution in speech emotion recognition related to the system in literature is summarized as follows:

2.1. Speech Emotion Recognition using Random Forest to trees method:

Rong et al. [1] presented an ensemble random forest to trees method with a high number of features for emotion identification without referring any linguistic information remains an unclosed problem. A modest amount of data with a lot of features is used in this strategy. An experiment using a dataset of Mandarin speakers' emotional speech was conducted to assess the suggested method, and the findings show that rate of emotion recognition improved.

2.2. Speech Emotion Recognition by utilizing speech signals:

Narayanan [2] proposed domain specific emotion identification by utilizing speech signals from call centre application. The fundamental goal of this research is to identify both negative and positive emotions, such as rage and happiness. Various forms of information include auditory, lexical, and discourse are employed for emotion recognition. In order to collect data on emotion information at the language level, information-theoretic content on emotional salience is also supplied.

2.3. Identifying emotions by representing hierarchical computational structure:

Lee et al.[3] represent a hierarchical computational structure to recognise emotions. This approach converts the input speech signal into one of the associated emotion classes using subsequent layers of binary classifications. The primary goal of each level in a tree is to carry out the categorization task as simply as possible while minimising error propagation.

2.4. Speech Emotion Recognition through three level SER method:

Chen et al.[4] aimed to improve speech emotion identification in speaker-independent with the three level speech emotion recognition method. This method groups various emotions into coarse, medium, and fine categories before choosing the best characteristic using the Fisher rate. The multi-level SVM-based classifier uses the Fisher rate output as one of its input parameters. In addition, four comparison experiments are classified and their dimensionality reduced using principal component analysis (PCA) and artificial neural networks (ANN), respectively. The Fisher + SVM, PCA + SVM, Fisher + ANN, and PCA + ANN four comparative trials. Fisher is superior to PCA in dimension reduction, and SVM is more extensible than ANN for classifying data for speaker independent emotion recognition. In the database of emotional speech maintained by Bei hang University, the recognition rates for the three levels are, respectively, 86.5%, 68.5%, and 50.2%.

2.5. Identify Speech Emotion Recognition through LFPC and discrete HMM method:

Nwe et al.[5] proposed a new system for emotion identification of utterance signals. The system used discrete HMM and short time log frequency power coefficients to characterise the classifier and speech signals, respectively. This technique divided the emotions into six groups. LFPC is contrasted with the Mel-frequency Cepstral coefficients (MFCC) and the linear prediction Cepstral coefficients in order to assess the effectiveness of the suggested strategy. The outcome shows that best and average categorization accuracy were 78% and 96%, respectively. Results also show that LFPC is a superior feature than standard characteristics for emotion classification.

2.6. Primitives-based evaluation and estimation of emotions in speech:

Grimm et al.[7] proposed a method for emotion recognition in speech using primitive-based evolution and estimation. Using genetic programming, a collection of rules that link the primitive features to various feelings is evolved by representing speech signals as a set of primitive features, such as pitch, energy, and spectral features. According to the study's findings, the primitive-based evolution and estimation method had a recognition rate of 70.6% overall, which was similar to other cutting-edge emotion recognition techniques at the time.

2.7. Automatic speech emotion recognition using modulation spectral features:

Wu et al.[22] proposed a method for automatic speech emotion recognition using modulation spectral features. The approach involves decomposing speech signals into a set of modulation components and extracting statistical features from the modulation spectra of these components. The results of the study showed that the modulation spectral features achieved an overall recognition rate of 66.2%.

2.8. Literature Findings

In all of these initiatives, the majority of the developers used supervised learning algorithms to predict the type of emotions. However, in this endeavour, we used the neural network concept to predict the type of emotion[15]. Most developers extracted the data set from speech signals, but in this project it contains a dataset of 24 distinct actors with different emotions.

CHAPTER 3

PROJECT DESCRIPTION

3.1 INTRODUCTION

For humans, speech is the most natural means of expression. Emoji are frequently used to communicate feelings in emails and text messages and in other kinds of social media communication. The neural network concept is utilised in this project to train the model and to make predictions. The trained model is integrated into the web application. The following are some of the terminologies used in this project.

3.2 TERMINOLOGIES

Audio signal:

The audio waveform captured from the speaker's voice that contains information about their emotions. In this project audio waveform is captured from 24 speaker’s voice which is used as the dataset.

Feature extraction:

Feature extraction is the process of extracting relevant features from the speech signal, such as pitch, duration, intensity, and spectral features. In this project a library named librosa is used for extracting the features from audio.

Mel-frequency cepstral coefficients (MFCCs):

A commonly used feature extraction technique that calculates the spectral envelope of the speech signal. It is the step which involves extracting relevant acoustic features from the speech signal. This is used in this project to train the model for recognizing different emotions.

Feature normalization:

This is the process of scaling the acoustic features to have zero mean and unit variance. Often performed before training the classifier to improve its performance.

Validation set:

A subset of the labelled dataset used to evaluate the performance of the machine learning algorithm during training. In this project we separated the dataset into training and testing with 80% data for training and 20% data for testing.

Test set:

A separate dataset used to evaluate the performance of the trained machine learning algorithm on unseen data. The dataset for this project is divided into training and testing, with 80% of the data for training and 20% of the data for testing.

Accuracy:

A measure of the performance of the machine learning algorithm that indicates the percentage of correct predictions. This project has gave an accuracy of 85% percentage after continuous adjustment in the neural network.

3.3 PROPOSED SYSTEM

Speech emotion recognition is a popular research area that aims to identify the emotional state of a person based on their speech signals. The proposed system that uses MFCC and Librosa for feature extraction and a multi-layer perceptron

(MLP) for training the model is a suitable approach for this task. MFCC (Mel Frequency Cepstral Coefficients) is a widely used feature extraction technique in speech processing. Librosa is a Python library for analysing and processing audio signals. Provides a range of functions for feature extraction, including MFCC. A Multi-layer perceptron (MLP) is a type of artificial neural network (ANN) that is commonly used in speech emotion recognition tasks. Using metrics such as accuracy, precision, recall etc, the model's performance will be assessed. With the help of this technique, it is possible to achieve a decent balance between the computing load and the real-time processes' performance accuracy. Overall, the proposed system is a suitable approach for speech emotion recognition.

3.4 BLOCK DIAGRAM

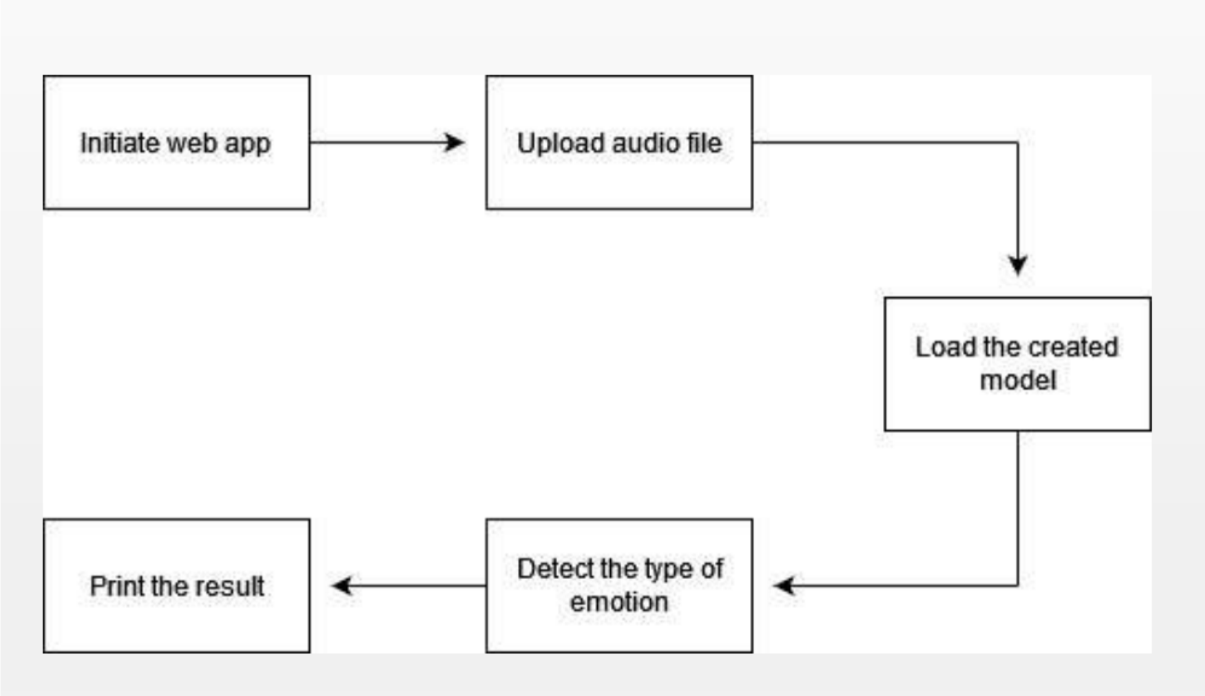


Fig 3.1 Block Diagram

The following is explained by the block illustration up top. The initial stage is to develop a web application that enables users to upload audio files. The programme in this project is made using the flask web framework written in python. The audio files must be uploaded as the next stage. Extract the features from an audio stream that a user has uploaded. Mel-frequency cepstral coefficients (MFCCs) and spectral contrast are examples of characteristics that can be extracted using the librosa library. These characteristics will be fed into the MLP model as input. The created model must be loaded as the next stage. In this stage, the MLP model that trained on a dataset of speech samples labelled with emotion classes needs to be loaded. Once the model has been trained and saved, use the proper library to import into web application. Finding the type of emotion is the next stage. The model can now be used to forecast the mood category of the speech sample after the audio file has been uploaded and loaded. Use the MLP model to forecast the emotion category using the audio signal's extracted features. Finally, the browser will display the predicted emotion category.

3.5 WORKING PRINCIPLE

Data Collection:

The first step in any emotion identification project is to collect a dataset of speech samples. The dataset should include speech samples from a variety of speakers and emotional states. Data collection is a crucial step in developing a speech emotion identification system as provides the necessary speech samples to train and test the system. Audio recording, Database acquisition, Annotation, Data augmentation are some of the common data collection techniques used in emotion identification. By carefully selecting and annotating speech samples, a high-quality dataset can be created for training and testing a speech emotion identification system.

Data Pre-processing:

Next comes the data pre-processing phase, Data pre-processing is an important step in speech emotion recognition, as helps to ensure that the speech samples are of high quality and free from any noise or artifacts that may interfere with the analysis of the speech signal. the collected speech samples are pre-processed to remove any noise or artifacts that may interfere with the analysis of the speech signal. This may include filtering, normalization, and segmentation. Noise removal, signal normalization, speech segmentation, speaker identification, feature extraction, feature scaling is some of the common data pre-processing techniques used in speech emotion recognition. By performing these data pre-processing techniques, the speech samples are prepared for further analysis and feature extraction, which can improve the performance of the speech emotion recognition system.

Feature Extraction:

Next phase is Featured extraction, the pre-processed speech samples are then analysed to extract relevant features that can be used to identify the emotional state of the speaker. These features may include pitch, duration, intensity, and spectral features such as Mel-frequency cepstral coefficients (MFCCs). The step which involves extracting relevant acoustic features from the speech signal that can be used to train a machine learning model to recognize different emotions. Mel-frequency cepstral coefficients (MFCCs), Pitch, Spectral features are commonly using features. These features are typically extracted from short segments of speech, and are then used as input to a machine learning model for emotion recognition.

Model Training:

In model training, a machine learning model is then trained on the selected features using a labelled dataset of audio samples. There are various model available, Multilayer perceptron (MLP) is a type of neural network architecture that has been widely used for speech emotion recognition. An MLP model consists of multiple layers of neurons, with each layer processing the input data in a non-linear way to extract higher-level representations of the data. In the case of speech emotion recognition, an MLP model takes acoustic features extracted from speech signals as input and learns to predict the corresponding emotion labels. The input acoustic features may include MFCCs, pitch, energy, and spectral features, as discussed earlier. The MLP model typically has a few hidden layers, each containing a set of neurons with a non-linear activation function. The number of neurons in each hidden layer and the number of hidden layers depend on the complexity of the input features and the size of the training dataset. During training, the MLP model learns to adjust the weights and biases of its neurons to minimize the prediction error between the predicted emotion labels and the ground-truth labels. This is typically done using a backpropagation algorithm that adjusts the weights and biases in the opposite direction of the gradient of the loss function with respect to the model parameters. Once the MLP model is trained, it can be used to predict the emotion label of new speech signals based on their acoustic features. The MLP model can be applied to real-time speech signals by using sliding windows to obtain the acoustic features in real-time and then feeding them into the MLP model for prediction.

Validation and Testing :

Validation and testing are crucial steps in evaluating the performance of an MLP model for speech emotion recognition. K-fold cross-validation, Holdout validation, Test set evaluation are some common methods used for validation and testing. In all of these methods, the performance of the MLP model is typically evaluated using metrics such as accuracy, precision etc. these metrics provide a quantitative measure of the model's ability to correctly predict the emotion labels from speech signals. Important to note that the MLP model should be tested on a diverse and representative dataset that includes a wide range of emotions and different speakers to ensure that the model generalizes well to real-world scenarios. Additionally, the performance of the MLP model should be compared with that of other state-of-the-art models to establish its effectiveness.

Model Deployment:

Model deployment is the process of integrating a trained machine learning model into an application or system for practical use. In speech emotion recognition, the deployment of a trained model can be done in various ways, depending on the application requirements and constraints. Standalone application, Web service, Embedded system, Cloud service are some common methods for model deployment. Web service deployment types is consumed in this model. The trained model can be deployed been a web service that accepts HTTP requests with speech signals as input and returns the predicted emotion label as output. This method allows the model to be used by multiple users over the internet without the need for installing any software on their devices. When deploying a trained model for speech emotion recognition, it is essential to ensure its performance, reliability, and security. The model should be tested extensively in real-world scenarios to verify its accuracy and robustness.

3.6 ADVANTAGES

Using Multi-layer Perceptron (MLP) and librosa for speech emotion detection has a number of benefits:

i) MLP is a potent deep learning algorithm that is well adapted for speech emotion recognition tasks because it can learn complex patterns in the data.

ii) MFCCs, which are frequently used for speech emotion detection, are just one of the many audio processing tools available in the Librosa Python package for analysing and processing audio signals.

iii) A dataset of speech samples with mood labels can be used to teach the popular MLP algorithm, which is used for classification tasks.

iv) When MLP and librosa are used together, it is possible to recognise various emotional states in speech signals with high accuracy. This makes the system helpful in a range of fields, including customer service, healthcare, and entertainment.

v) A non-intrusive and economical method of understanding human feelings, speech emotion recognition using MLP and librosa can be applied in a variety of contexts, including clinical psychology and education.

vi) The process of speech emotion recognition can be automated with the aid of MLP and librosa, which can save time and resources, particularly when compared to situations where human experts would have to manually analyse each audio sample.

Overall, speech emotion recognition with MLP and librosa has a number of benefits, including precision, effectiveness, and affordability, making a useful instrument for a variety of applications where comprehending human emotions is crucial.

3.7 APPLICATIONS

Speech emotion recognition has a wide range of applications, including in healthcare, education, customer service, and security.

Healthcare:

In the medical field, emotion can be used to identify and treat mental health conditions including depression and anxiety. It can be used to track and monitor patients' emotional states, giving clinicians immediate feedback so can modify treatment plans accordingly. Also, by identifying symptoms of suffering in speech signals, speech emotion recognition can be utilised to anticipate and prevent suicide.

Education:

In education, Speech emotion recognition can be used to improve the quality of online education by providing real-time feedback on student engagement and emotional states. Used to personalize the learning experience based on the emotional needs of students, such as adjusting the pace of instruction or providing additional support when needed. Additionally, speech emotion recognition can be used to detect signs of stress or frustration in students, allowing teachers to intervene and provide support.

Customer Service:

Speech emotion recognition can be used in customer service to improve the quality of customer interactions. Used to detect the emotional states of customers, providing real-time feedback to customer service representatives to adjust their approach and provide personalized services based on individual emotional needs. Additionally, speech emotion recognition can be used to analyse customer feedback and sentiment, allowing companies to improve their products and services based on customer needs.

Security:

In security, speech emotion identification can be used to detect signs of deception or stress in speech signals. They can be used for lie detection and security screening, such as in airport security or law enforcement. Additionally, speech emotion recognition can be used to analyse the emotional states of individuals in high-stress situations, such as emergency responders or military personnel, providing real-time feedback to optimize performance and reduce risk.

Entertainment:

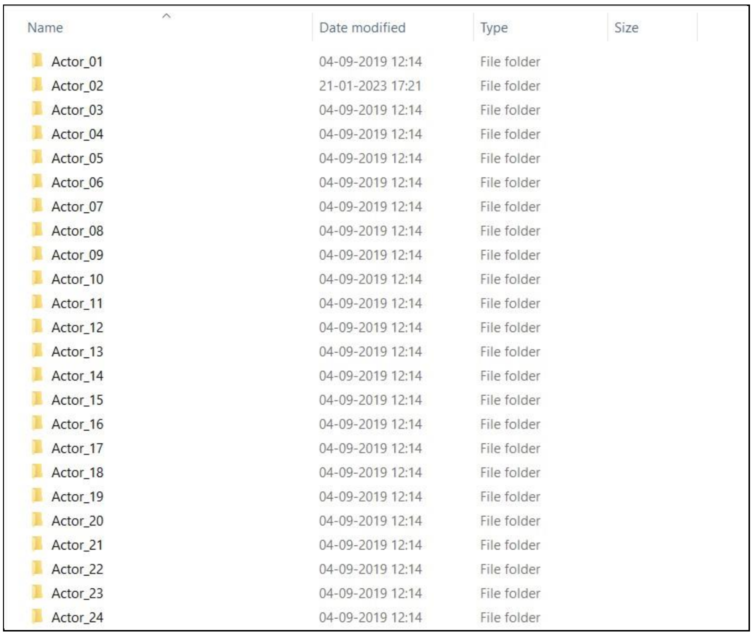
Speech emotion recognition can be used in entertainment to create immersive experiences that respond to the emotional states of users. Used in video games to adjust the difficulty level or storyline based on the emotional states of players. For example, the game can detect the player's emotional state and adjust the gameplay accordingly. If the player is feeling stressed or anxious, the game can reduce the difficulty level or provide more relaxing gameplay. Used in virtual reality (VR) applications to create more realistic and engaging environments. For example, the VR environment can change based on the user's emotional state, providing a more personalized experience. They also can be used in music applications to create playlists that match the user's emotional state. For example, if the user is feeling sad, the system can create a playlist of songs with slower tempos and more melancholic themes

CHAPTER 4

RESULT

Dataset:

A dataset is a collection of data that is organized in a specific way. A dataset usually consists of a set of observations, each of which includes one or more features or attributes that describe that observation. Datasets are a critical component of machine learning, used to train and test machine learning models. During the training phase, the model is presented with a subset of the dataset, and the algorithm attempts to learn patterns or relationships between the features and the target variable. Once the model is trained, evaluate using a separate subset of the dataset, called the test set, to see how well generalizes to new, unseen data.

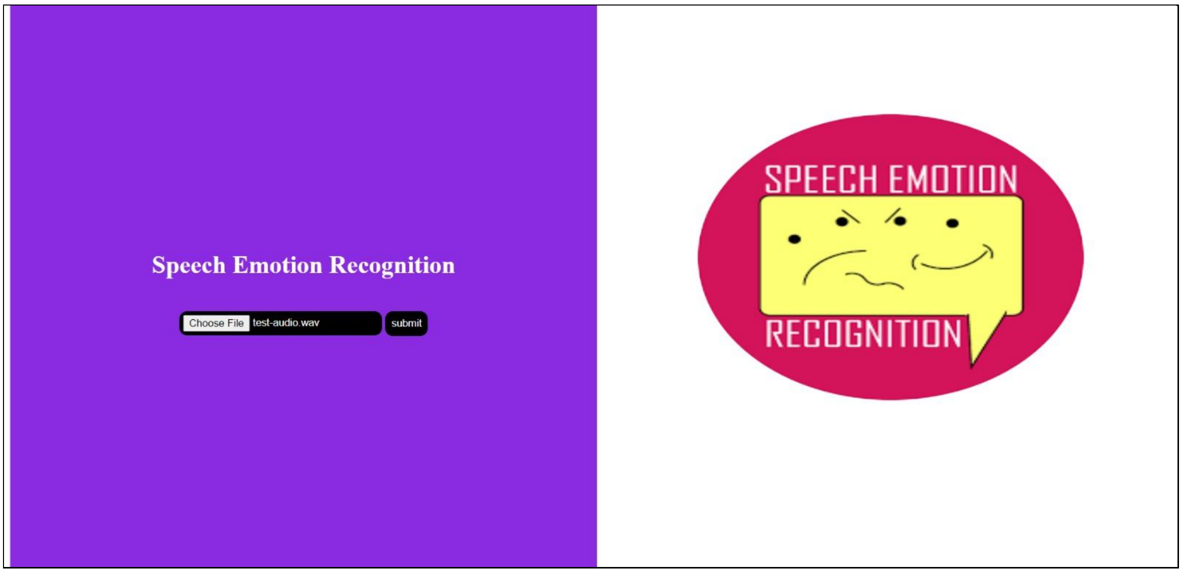


Name	Date modified	Type	Size
Actor_01	04-09-2019 12:14	File folder	
Actor_02	21-01-2023 17:21	File folder	
Actor_03	04-09-2019 12:14	File folder	
Actor_04	04-09-2019 12:14	File folder	
Actor_05	04-09-2019 12:14	File folder	
Actor_06	04-09-2019 12:14	File folder	
Actor_07	04-09-2019 12:14	File folder	
Actor_08	04-09-2019 12:14	File folder	
Actor_09	04-09-2019 12:14	File folder	
Actor_10	04-09-2019 12:14	File folder	
Actor_11	04-09-2019 12:14	File folder	
Actor_12	04-09-2019 12:14	File folder	
Actor_13	04-09-2019 12:14	File folder	
Actor_14	04-09-2019 12:14	File folder	
Actor_15	04-09-2019 12:14	File folder	
Actor_16	04-09-2019 12:14	File folder	
Actor_17	04-09-2019 12:14	File folder	
Actor_18	04-09-2019 12:14	File folder	
Actor_19	04-09-2019 12:14	File folder	
Actor_20	04-09-2019 12:14	File folder	
Actor_21	04-09-2019 12:14	File folder	
Actor_22	04-09-2019 12:14	File folder	
Actor_23	04-09-2019 12:14	File folder	
Actor_24	04-09-2019 12:14	File folder	

Fig 4.1 Dataset_Image

The dataset includes audio data of both men and women at different ages, so that the dataset can be diversified and helps in making model more precisely. Emotion of each actor was stored in different folders. The audio of each actor consists of various types of emotions such as fear, calm, happy and disgust.

Input:

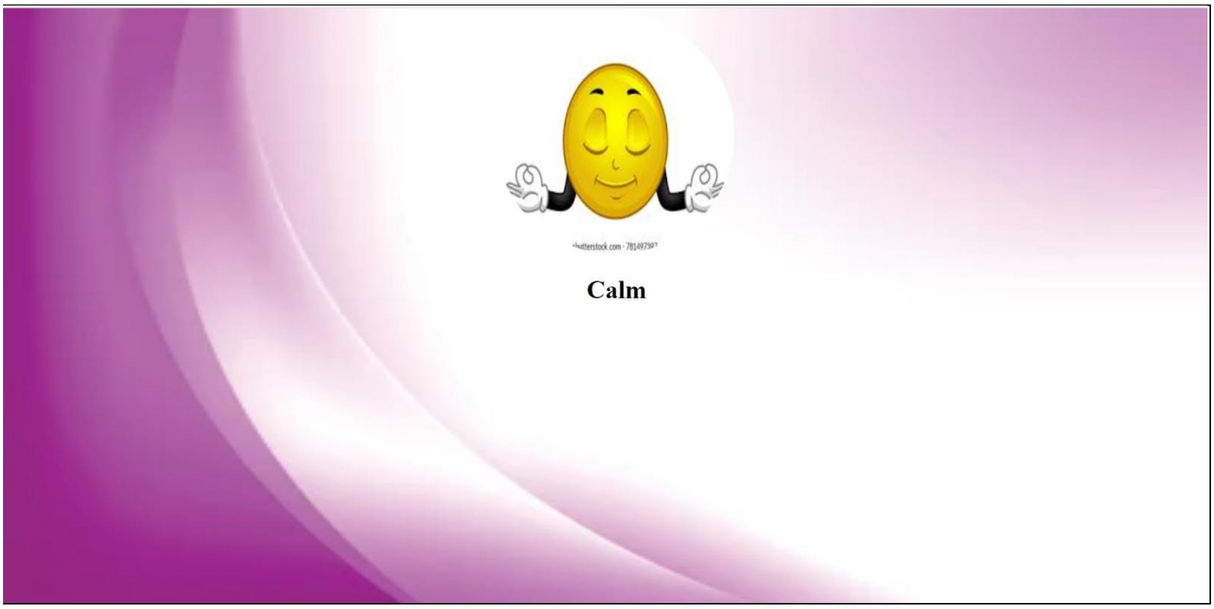


Speech Emotion Recognition

Fig 4.2 Input_Image

The above image depicts the UI Layout of the web application. The user has to choose the audio file to be predicted from user’s directory and click submit button. The audio will be passed as an input to the model created at background and the output will be displayed in next page.

Output:



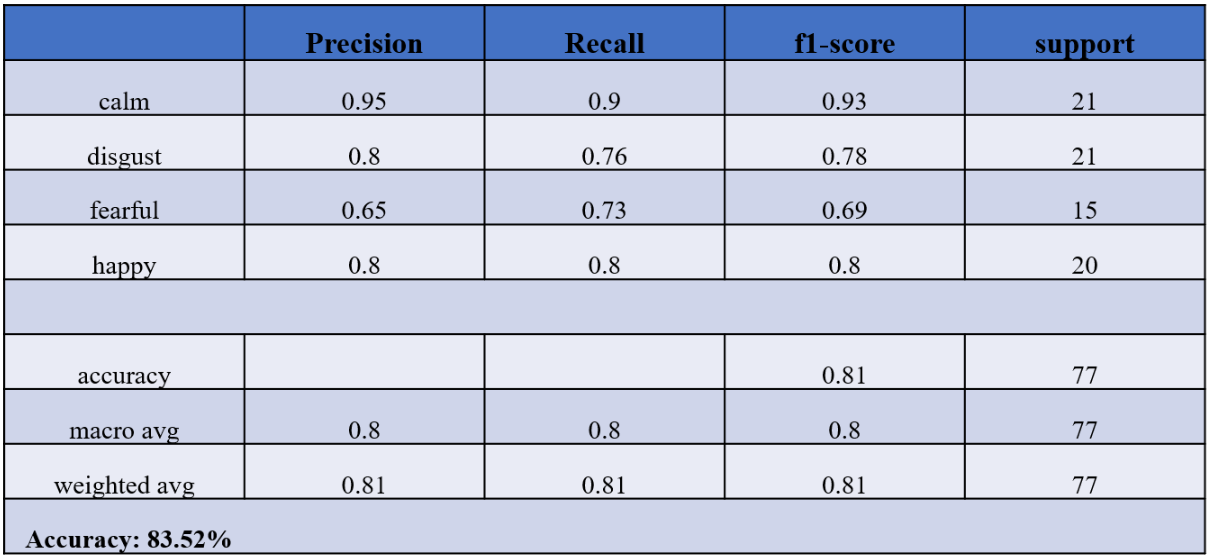
Calm

Fig 4.3 Output_Image

The output page displays the detected emotion based on user’s audio. It also displays the emoji based on the emotion predicted.

Accuracy measures:

Table 4.1 Accuracy_measures



	Precision	Recall	f1-score	support
calm	0.95	0.9	0.93	21
disgust	0.8	0.76	0.78	21
fearful	0.65	0.73	0.69	15
happy	0.8	0.8	0.8	20
accuracy			0.81	77
macro avg	0.8	0.8	0.8	77
weighted avg	0.81	0.81	0.81	77
Accuracy: 83.52%				

Accuracy are measures of how well a model is performing on a given dataset. The above image describes the metrics found on the generated model. The table shows the accuracy of the model created and model has generated high around 84% of accuracy, which is comparatively more than other traditional models.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Conclusion:

Emotion recognition is an exciting and rapidly evolving field with numerous potential applications in various fields. Recognising emotion has made significant progress in recent years, thanks to advances in machine learning algorithms, signal processing techniques, and the availability of large speech datasets. Today, emotion recognition systems can accurately recognize basic emotions such as happiness, fear, disgust, and calm. Emotion recognition using audio has numerous potential applications in areas such as healthcare, education, entertainment, and human-computer interaction, and is poised to revolutionize the way interact with machines and each other.

Future Scope:

Despite the significant progress made in emotion identification, there are still many challenges that need to be addressed. One major challenge is dealing with individual differences in speech and emotional expression, which can vary depending on factors such as culture, age, gender, and personality. In the future, emotion recognition systems can be more advance that can detect a wider range of emotions and incorporate other modalities such as physiological signals. Expect to see more personalized and context-aware emotion recognition systems that can adapt to individual users and their environments. Furthermore, it can expect to see more innovative and exciting applications in areas such as mental health, education, and entertainment.

APPENDIX

APPENDIX 1

SOFTWARE AND MODULE DESCRIPTION

- Python – v3.7
- Jupyter Notebook
- Flask Framework

PYTHON (v3.7):

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python is a robust programming language that is simple to learn. Its object-oriented programming methodology is straightforward but efficient, and includes good high-level data structures. Python is the perfect language for scripting and quick application development across the majority of platforms because of its clean syntax, dynamic typing, and interpreted nature. Also free to use and distribute. Additionally, the same website provides links to extra documentation as well as releases of numerous free third-party Python modules, programs, and tools. Simple to add new functions and data types to the Python interpreter that were developed in C or C++ (or other languages callable from C). Python is a good choice as an add-on language for flexible software. The reader is given a casual introduction to the fundamental ideas and capabilities of the Python system and language in this lesson. For practical experience, it is helpful to have a Python interpreter on hand, although the lesson can also be read offline because each example is self-contained. Python is also a highly versatile language that can be used for a wide range of applications. In web development, Python is often used with frameworks like Django and Flask to build scalable and secure web applications. In data science, Python has become the language of choice for many data analysts and scientists due to its ability to handle large datasets and its rich ecosystem of data analysis and visualization tools, such as NumPy, Pandas, and Matplotlib. Python's popularity in artificial intelligence and machine learning is also growing rapidly, thanks to frameworks like TensorFlow, Keras, and PyTorch. These frameworks provide powerful tools for building and training machine learning models, making easier for developers to create sophisticated applications that can learn from data and make intelligent decisions. In terms of performance, Python is generally slower than lower-level languages like C and C++. However, often fast enough for most applications and can be optimized using tools like NumPy and Cython. Introduces many of Python's most notes worthy features, and will give a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and will be ready to learn more about the various Python library modules described in library-index. If much work rely on computers, eventually try to find that there's some tasks would like to automate. For example, to perform a search-and-replace over a large number of text files, or rename and rearrange a bunch of photo files in a complicated way. Perhaps, write a small custom database, or a specialized GUI application or a simple game. If you're a professional software developer, then work with several C/C++/Java libraries but find the usual write/compile/test/re-compile cycle is too slow. Perhaps, writing a test suite for such a library and find writing the testing code a tedious task. Or maybe written a program that could use an extension language, and don't want to design and implement a whole new language for application.

Additionally, Python's ease of use and high-level abstractions makes a good choice for rapid prototyping and development, allowing developers to quickly iterate on their ideas and experiment with different approaches. Python's community is also one of its strengths. With a large and active community of developers, users, and contributors, Python has a wealth of resources available, including online forums, tutorials, and open-source projects. The community is also committed to making Python accessible and inclusive, with efforts to promote diversity and inclusivity in the language and its community.

JUPYTER NOTEBOOK:

Jupyter Notebook is an interactive computing environment that allows users to write, execute, and share code in a web-based interface. Developed to provide a platform for data science and scientific computing, but has since become popular among developers, researchers, and educators in a wide range of fields. At its core, Jupyter Notebook is built around the concept of "cells." Each cell can contain code, text, or images, and can be executed independently. This allows users to break their code into small, manageable chunks, and to test and debug their code incrementally. Jupyter Notebook also supports a range of programming languages, including Python, R, and Julia, making a powerful tool for data analysis and scientific computing.

One of the key advantages of Jupyter Notebook is its interactivity. Users can execute code directly in the notebook and see the results immediately, making easy to experiment with different approaches and to explore data interactively. The notebook also supports rich media, including images, videos, and interactive visualizations, which can be embedded directly into the notebook and shared with others. Jupyter Notebook also includes a number of features that make a powerful tool for collaboration and sharing. Notebooks can be saved as self-contained files that include all of the code, data, and outputs, allowing others to reproduce the analysis or experiment. Notebooks can also be shared online through services like GitHub, allowing users to collaborate and share their work with others. Another advantage of Jupyter Notebook is its extensibility. Users can create custom extensions that add new functionality to the notebook, such as additional widgets, data sources, or visualization tools. This has led to a growing ecosystem of third-party extensions and libraries that extend the functionality of Jupyter Notebook in new and interesting ways.

Jupyter Notebook also has several built-in features that makes a valuable tool for teaching and learning. For example, users can create "slides" from their notebooks that allow to present their work in a structured, interactive format. Notebooks can also be used to create tutorials, exercises, and assignments, providing a hands-on learning experience for students. Despite its many advantages, Jupyter Notebook is not without its limitations. One of the main challenges of working with Jupyter Notebook is managing dependencies and ensuring reproducibility. Since notebooks are often used to analyze data or run experiments, it's important to ensure that the notebook can be reproduced in the future, even if the dependencies or environment change. Another challenge of working with Jupyter Notebook is version control. Since notebooks are often modified over time, difficult to track changes and collaborate effectively. However, there are several tools available that can help with version control, such as Git and GitHub.

In conclusion, Jupyter Notebook is a powerful and flexible tool for data analysis, scientific computing, and teaching. Its interactivity, extensibility, and collaboration features makes a popular choice among developers and researchers in a wide range of fields. While it has some limitations, Jupyter Notebook remains an important tool for anyone working with data, code, or scientific research.

FLASK FRAMEWORK:

Flask is a lightweight web framework written in Python that is designed to make easy to build web applications quickly and easily. It is popular among developers for its simplicity, flexibility, and ease of use. One of the key features of Flask is its modular design. Flask is built on top of the Werkzeug WSGI toolkit and the Jinja2 template engine, which allows developers to create custom web applications with a high degree of flexibility. Flask is also designed to be extensible, with a large number of third-party extensions available to add functionality to applications. Flask is well-suited to building small to medium-sized web applications, such as personal blogs, simple e-commerce sites, and internal company tools. Used by many startups as a platform for building minimum viable products (MVPs), due to its ease of use and quick development time.

One of the reasons why Flask is so popular among developers is its simplicity. Flask does not include a lot of the built-in features and functionality that would find in larger frameworks like Django, which can be overwhelming for new developers. Instead, Flask is designed to be lightweight and easy to understand, allowing developers to quickly build and iterate on their applications. Flask also has a large and active community of developers who contribute to the project and create third-party extensions. This means that there are many resources available for developers who are just getting started with Flask, including tutorials, blog posts, and community forums. The community also contributes to the development of the framework itself, which helps to ensure that Flask remains up-to-date and relevant to developers' needs.

Another advantage of Flask is its built-in support for unit testing. Flask includes a test client that allows developers to write and run tests for their applications, which is an essential part of any development process. This helps to ensure that application is working correctly and that any changes make to the code, do not introduce bugs or other issues. Flask also supports a wide range of web development tasks, including routing, form handling, and database integration. Flask can be used with a variety of databases, including SQLite, PostgreSQL, and MySQL, which makes easy to store and retrieve data for application. Flask also includes built-in support for creating RESTful APIs, which is a common requirement for modern web applications.

In conclusion, Flask is a powerful and flexible web framework that is popular among developers for its simplicity and ease of use. Well-suited to building small to medium-sized web applications and is particularly useful for startups and developers who need to build and iterate quickly. Flask's modular design and large community of developers make a great choice for anyone who is looking for a lightweight and flexible web framework for their next project.

APPENDIX 2

SOURCE CODE:

index.html:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8'>
  <meta http-equiv='X-UA-Compatible' content='IE=edge'>
  <title>Speech Emotion Recognition</title>
  <meta name='viewport' content='width=device-width, initial-scale=1'>
  <link rel='stylesheet' href='../styles/main.css'>
  <style>
    body{
      overflow-y:hidden ;
    }
    .container {
      width:50%;
      height:98.5vh;
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      gap: 20px;
      background-color: blueviolet;
    }
    h1{
      color:white;
    }
    .btn {
      width: fit-content;
      padding: 5px;
      background-color: black;
      border: none;
      cursor: pointer;
      color: white;
      border-radius: 10px;
    }

    .submit-btn {
      padding: 8px;
    }

    .emotion-img {
      width: 500px;
```

```
            height: 50%;

            border-radius: 50%;

            position: relative;

            left:130px;

            top:150px;

        }

        .main-container{

            display: flex;

            flex-direction: row;

        }

        .side-nav{

            width:50%;

            height:100vh;

        }

    </style>

</head>



<body>

    <div class="main-container">

        <div class="container">

            <h1>Speech Emotion Recognition</h1>
            <form action="/predict" method="POST" enctype="multipart/form-
data">
                <input type="file" id="file" name="file" class="file-btn btn" />
                <button type="submit" class="submit-btn btn">submit</button>
            </form>
        </div>
        <div class="side-div">
            <img src="{{image}}" alt="emotion" class="emotion-img" />
        </div>
    </div>
    <script>
    </script>
</body>
</html>
```

predict.html:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Page Title</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <style>
        body{
            background-image: url('/static/pics/background.webp');
            background-size: 100% 110vh;
            background-repeat: no-repeat;
        }
        .img{
            width:300px;
            height:300px;
            border-radius: 50%;
            position: relative;
            left:40%;
        }
        h1{
            text-align: center;
        }
    </style>
</head>
<body>
    <div>
    {% block content %}
    {% if pred=='fearful' %}

    <img src="{{fear}}" class="img"/>

    <h1>Fearful</h1>

    {% elif pred=='happy' %}

    <img src="{{happy}}" class="img"/>

    <h1>Happy</h1>

    {% elif pred=='disgust' %}

    <img src="{{disgust}}" class="img"/>

    <h1>Disgust</h1>

    {% elif pred=='calm' %}

    <img src="{{calm}}" class="img"/>

    <h1>Calm</h1>

    {% endif %}

    {% endblock content %}

    </div>

</body>

</html>
```

app.py:

```
from flask import Flask,render_template,request

import pickle
```

```
from werkzeug.utils import secure_filename
import numpy as np
import os
from model import extract_feature
# import librosa
# import soundfile
# from model import extract_feature
app=Flask(__name__)
model=pickle.load(open('model.pkl','rb'))
picFolder=os.path.join('static','pics')
app.config['UPLOAD_FOLDER']=picFolder
@app.route('/')
def hello_world():
    pic1=os.path.join(app.config['UPLOAD_FOLDER'],'images.png')
    return render_template('index.html',image=pic1)

@app.route('/predict',methods=['POST','GET'])
def predict():
   pic2=os.path.join(app.config['UPLOAD_FOLDER'],'happy.jfif')
   pic3=os.path.join(app.config['UPLOAD_FOLDER'],'sad.jfif')
   pic4=os.path.join(app.config['UPLOAD_FOLDER'],'fear.jfif')
   pic5=os.path.join(app.config['UPLOAD_FOLDER'],'angry.webp')
```

```
    pic6=os.path.join(app.config['UPLOAD_FOLDER'],'disgust.jfif')
    pic7=os.path.join(app.config['UPLOAD_FOLDER'],'calm.jfif')
    pic8=os.path.join(app.config['UPLOAD_FOLDER'],'background.webp')
    file=request.files['file']
    file.save(secure_filename(file.filename))
    res=extract_feature(secure_filename(file.filename),mfcc=True, chroma=True, mel=True)
    res=res.reshape(1,-1)
    result=model.predict(res)
    print(result)
    return render_template('predict.html',pred=result,happy=pic2,sad=pic3,fear=pic4,angry=pic5,disgust=pic6,calm=pic7,bg=pic8)


if __name__=='__main__':
     app.debug=True
     app.run(debug=False,host='0.0.0.0')
```

model.py:

```
import librosa
import soundfile
import os, glob, pickle
import numpy as np
```

```
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
import pickle


def extract_feature(file_name, mfcc, chroma, mel):
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate=sound_file.samplerate
        if chroma:
            stft=np.abs(librosa.stft(X))
        result=np.array([])
        if mfcc:
            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate,
n_mfcc=40).T, axis=0)
            result=np.hstack((result, mfccs))
        if chroma:
            chroma=np.mean(librosa.feature.chroma_stft(S=stft,
sr=sample_rate).T,axis=0)
            result=np.hstack((result, chroma))
        if mel:
            mel=np.mean(librosa.feature.melspectrogram(y=X,
sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
        return result
emotions={
  '01':'neutral',
  '02':'calm',
  '03':'happy',
  '04':'sad',
  '05':'angry',
  '06':'fearful',
  '07':'disgust',
  '08':'surprised'
}

#DataFlair - Emotions to observe
observed_emotions=['calm', 'happy', 'fearful', 'disgust']



def load_data(test_size=0.2):
    x,y=[],[]
    for file in glob.glob("C:\\Users\\Lenovo\\Downloads\\speech-emotion-recognition-ravdess-data\\Actor_*\\*.wav"):
        file_name=os.path.basename(file)
        emotion=emotions[file_name.split("-")[2]]
        if emotion not in observed_emotions:
            continue
        feature=extract_feature(file, mfcc=True, chroma=True, mel=True)
        x.append(feature)
        y.append(emotion)
    return train_test_split(np.array(x), y, test_size=test_size, random_state=9)


x_train,x_test,y_train,y_test=load_data(test_size=0.2)
model=MLPClassifier(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500)
```

```
model.fit(x_train,y_train)


pickle.dump(model,open('model.pkl','wb'))
models=pickle.load(open('model.pkl','rb'))
```

REFERENCE

[1] Rong et al., (2017), ‘Speech emotion recognition methods’, AIP Conference Proceedings.

[2] Narayan, (2020), ‘Speech Emotion Recognition using Support Vector Machine’, Scholar, volume 1

[3] Lee et al., (2011), ‘Emotion recognition using a hierarchical binary decision tree approach’ Research Gate, volume 53

[4] L. Chen et al., (2012) “Speech emotion recognition: Features and classification models,” Digit. Signal Process., vol. 22, no. 6, pp. 1154–1160

[5] T. L. Nwe et al., (2003), “Speech emotion recognition using hidden Markov models,” Speech Commun., vol. 41, no. 4, pp. 603–623

[6] J. P. Arias et al., (2014) “Shape-based modeling of the fundamental frequency contour for emotion detection in speech,” Comput. Speech Lang., vol. 28, no. 1, pp. 278–294.

[7]. M. Grimm et al., (2007), “Primitives-based evaluation and estimation of emotions in speech,” Speech Commun., vol. 49, no. 10–11, pp. 787–800

[8] R. Banse and K. R. Scherer, “Acoustic profiles in vocal emotion expression.,” Pers. Soc. Psychol, vol. 70, no. 3, pp. 572–587, 1996.

[9] V. Hozjan and Z. Kačič, (2003), “Context-Independent Multilingual Emotion Recognition from Speech Signals,” Int. J. Speech Technol., vol. 6, no. 3, pp. 311–320.

[10] F. Burkhardt et al., (2005), “A database of German emotional speech.,” in Interspeech, vol. 5, pp. 1517–1520.

[11] A. Schuller et al., (2009), “The interspeech 2009emotion challengee,” Interspeech, pp. 312– 315.

[12] B. S. Atal, (2005), “Effectiveness of liner prediction characteristics of the speech wave for automatic speaker speech wave for automatic speaker identification and verification,” Acoust. Soc. Am., vol. 55, no. 6, pp. 1304–1312.

[13] Machine Learning - ‘https://www.ibm.com/in-en/topics/machine-learning’

[14] Librosa - ‘https://librosa.org/doc/latest/index.html’

[15] Multilayer perceptron - ‘https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/’

[16] Artificial Intelligence-‘https://en.wikipedia.org/wiki/Artificial_intelligence’

[17] Speech Emotion Recognititon - ‘https://www.kaggle.com/code/

shivamburnwal/speech-emotion-recognition’

[18] MFCC – ‘https://en.wikipedia.org/wiki/Mel-frequency_cepstrum’

[19] Flask – ‘https://flask.palletsprojects.com/en/2.2.x/’

[20] Neural networks – ‘https://en.wikipedia.org/wiki/Neural_network’

[21] Deep Learning – ‘https://en.wikipedia.org/wiki/Deep_learning’

[22] S. Wu, T. H. Falk, and W.-Y. Chan, (2011), “Automatic speech emotion recognition using modulation spectral features,” Speech Commun., vol. 53, no. 5, pp. 768–785

[23] H. Cao, R. Verma, and A. Nenkova, (2015) “Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech,” Comput. Speech Lang., vol. 28, no. 1, pp. 186–202.