PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

6th Semester Project Report on

EMOTION RECOGNITION ON VIDEO SPEECH

Submitted by

RIYA MANDAL (PES1201802215)

Jan – May, 2021

under the guidance of

Internal Guide

Tamal Dey
Assistant Professor
Department of Computer Applications,
PESU, Bengaluru - 560085

FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER APPLICATIONS
PROGRAM – MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project entitled

EMOTION RECOGNITION ON VIDEO SPEECH

is a bonafide work carried out by

RIYA MANDAL - PES1201802215

in partial fulfillment for the completion of 6th semester project work in the Program of Study MCA with specialization in Data Science under rules and regulations of PES University, Bengaluru during the period Jan. 2021 – May 2021. The project report has been approved as it satisfies the 6th semester academic requirements in respect of project work.

Internal Guide
Prof Tamal Dey
Assistant Professor,
Department of Computer Applications,
PES University, Bengaluru - 560085

Chairperson
Dr. Veena S

Dean-Faculty of Engineering& Technology
Dr. B K Keshavan

Name and Signature of Examiners:

Examiner 1: *Examiner 2:* *Examiner 3:*

DECLARATION

I, **Riya Mandal,** hereby declare that the project entitled, ***Emotion Recognition on Video Speech,*** is an original work done by me under the guidance of **Mr. Tamal Dey,** Assistant Professor, Dept of Computer Application, PES University and is being submitted in partial fulfillment of the requirements for completion of 6th Semester course work in the Program of Study **MCA**. All corrections/suggestions indicated for internal assessment have been incorporated in the report. The plagiarism check has been done for the report and is below the given threshold.

PLACE: Bengaluru

DATE: 20/08/2021

RIYA MANDAL

ACKNOWLEDGEMENT

I would like to acknowledge all who have given the guidance and encouragement that has made me to do what is done so far. I express my sincere thanks and gratitude to Dr. M.R. Doreswamy, Honorable Chancellor, PES University for providing us the excellent infrastructure to carry out this project work. I am thankful to Prof. Jawahar Doreswamy, Pro Chancellor, Dr. Suryaprasad J, Vice Chancellor, Dr. K. S Sridhar, Registrar, PES University, Bangalore for their support and encouragement for this work.

I take this opportunity to express my heartfelt thanks to Dr. Veena S, Chairperson, Department Of MCA, PES University for her encouragement to complete this project successfully. I also thank my guide and project coordinator Mr. Tamal Dey, Assistant Professor, Department of MCA, PES University who helped me for the successful completion of this project.

I avail this opportunity to express my deep sense of gratitude and sincere thanks to Faculty members of our Department of MCA for their continuous support. I also thank my friends for their valuable suggestions. I am thankful to them for their cooperation for the successful completion of this project. Last but not the least I would like to thank my parents for being supportive in all the activities and without whom it wouldn’t be possible to complete this project.

ABSTRACT

Human emotion recognition is a long ongoing research in the field of Machine Learning. The idea of sensing human emotions automatically by the machines is intriguing. However, it is no easy work to teach a machine how to recognize emotions which is complex and confusing as well. Some machine learning algorithms make this complex task easier and give an approximate emotion of people even if it is not cent percent correct all the time.

The idea of this project is to build a machine learning model that can detect emotions like Happiness, Anger, Sadness, Surprise, etc. from video speech. The main processes involved are Preprocessing audio chunks from YouTube Videos, extract the acoustic features and classify them discretely using a trained model with the help of a pre-trained dataset. The main purpose of this project is to identify the emotions of the speaker in the video from their speeches.

CONTENTS

LIST OF FIGURES

List of Tables

INTRODUCTION

1. INTRODUCTION

1.1 Project Description

A human emotion recognition system is the need of modern tech dominated era where everything is automatic and customers are targeted through AI. It is very important for Computers to understand the human sentiments as people does. So that in near future, they can understand better about emotions and can be trained according to people's needs. The work in this field is ongoing and it will take about a decade before the human emotion recognition comes anywhere near accurate. However, there are many research paper and software addressing to this complex problem and someday machines can understand us better than other people. The idea behind this project is to calculate human emotions of videos like anger, sadness, etc. through machine learning algorithm. In doing so, You Tube videos comes handy to download. These can be processed into audios and audio features such as its tune and wavelengths to detect the underlying emotions. Pre-trained dataset needs to be used for training the model to understand the emotions and features better.

- **Problem Definition**

  To build a Machine Learning model that can detect human emotions like Happiness, Anger, Sadness, Surprise from the Video speeches. An interactive user interface of dynamic web page to be made available that can be used by the user easily.

- **Proposed Solution**

  Audios can be extracted from the downloaded videos to get the audio features from the speech signals. These audio features are to be fed to a trained machine learning model that will detect the emotions from the audio signals and give the output as observed emotion. An interactive user-interface to be made available for easy upload of audio clips and prediction of emotion.

- **Purpose**

  The main purpose of this application is to identify the emotions of the speaker in the video from their speeches. Instead of predicting emotions on acted datasets present online, a self-made dataset has been created for prediction and analysis of video speech. This shall allow to find out the underlying emotions of the speaker at a given point of time during entire video. This will also lead to get the dominating emotions of the Speakers in a video.

- **Scope**

  Human emotions are to be predicted on unacted raw videos from YouTube. Analysis shall be done on the given predicted emotions with respect to each video. User can predict emotions of the speaker for any given audio clip uploaded in the system.

LITERATURE SURVEY

2.LITERATURE SURVEY

2.1 Background Study

Emotion Recognition is the process of detecting human sentiments by the computer software through Machine Learning. In this era of AI and Automotive Industry detection of emotions plays a vital role to understand the people and customers. Human emotions such as Happiness, Anger, Sadness, Excitement, etc... are very complex sentiments and differs from person to person and situation wise as well. People have different perspective to respond to same situation. However, the new progress in Machine Learning Algorithms has shown promising results in this area. So far, the progress is very ground level and the works are going on to analyze human emotion better but perfection is a long road and the research works continues. [4][5]

There are many papers written in this field that has been referenced for this project. They are following:

TITLE: Speech Emotion Recognition with Deep Learning

Author: Harar, Radim Burget and Malay Kishore Dutta

Conference Paper, February 2017

Summary: This paper deals with Speech Emotion Recognition i.e SER with the help of Deep Neural Network architecture using convolutional, pooling and fully connected layers of DNN. German Corpus dataset has been used in this paper with 3 class subsets angry, neutral and sad.

Implementation strategy: The files were split into 20ms chunks and removed the silent segments. The dataset is divided into 3 sets of training-79.56%, validation-9.84% and testing-10.6%. The layers of DNN architecture used first are Convolutional layers of different kernel sizes. Then the network is branched into pooling layers which is later

merged together. After this, the network consists only of fully connected layers with different sizes. The Stochastic Gradient Descent algorithm is used for training purpose. The input data were presented in batches of size 21 in multiple epochs. The maximum value of predicted probabilities has been used to give the predicted class. To give the final outcome as a single emotion for the entire file, the average probability is considered.

Result: This experiment has shown 79.14% validation accuracy and 96.97% accuracy on testing files for the DNN architecture. 69.55% was the average confidence of file prediction. [4]

TITLE: Human Emotion Recognition from Audio and Video Signals

Author: B. R. K. Madhur, Sai Nikhil Chennoor, Dr. T. Kishore Kumar and Moujiz Ali

Department of Electronics and Communication Engineering, National Institute of Technology, Warangal, Telangana, India - 506004

Summary: This paper deals with different audio and video dataset such as Berlin Emo database and JAFEE database. The paper has demonstrated a model for emotion recognition from audio and video signals that could be deployed in devices with nominal capabilities.

Implementation strategy: The video signals are transformed to frames and emotion on these frames are recognized and makes the overall emotion of video signals. The LBP feature extraction and SVM classification model is used for video frames. Similarly, Audios are extracted from videos and SVM model is used on features like pitch and entropy to calculate the emotion. At the merging stage a threshold value is given from 0 to 10 and if the difference between first 2 maximum frame counts is greater than threshold then emotion from video signals is counted else emotion from audio signal is counted.

Result: The overall model accuracy is 77.33%. The video accuracy is 74.66% for 75 files. Same files were taken for emotion detection audio signals giving an accuracy rate of 69.33%. [5]

TITLE: Emotion Recognition in Speech using Cross-Modal Transfer in the Wild

Author: Andrea Vedaldi, Samuel Albanie, Andrew Zisserman and Arsha Nagrani

Visual Geometry Group, Department of Engineering Science, University of Oxford, Seoul, Republic of Korea, October 22-26-2018

Summary: This paper is based on the hypothesis that the emotions of speech correlate with the facial expression of that speaker. The VoxCeleb Dataset has been used for the experiments.

Implementation strategy: Teacher-student method is used where a poweful teacher network performs emotion recognition from face images training teaches the student model that is tasked for performing emotion recognition from audios. The CNN network has been trained for 50 epochs.

Result: The student model had a mean ROC AUC of 0.69 over the teacher-predicted emotions present in the unheard identities consisting of all emotions except some like disgust and fear. A mean ROC AUC of 0.71 has been achieved on validation set of heard identities on those emotions.[6]

2.2 FEASIBILITY STUDY

2.2.1 Economic Feasibility

This application does not need any costly computer and hence it is affordable. All the libraries used are open source and hence no license needs to be purchased. All the software and technologies are free of cost.

2.2.2 Operational Feasibility

The website is user-friendly so that users can easily use this website without any confusion and users can select the audio from their local files to predict emotions.

2.2.3 Technical Feasibility

The software used for the completion of the application are open-sourced and easy to install and use. However, all that a user needs to access this web-based application is any browser like Chrome or Microsoft edge and a system that support the browser.

2.3 ABOUT TOOLS AND TECHNOLOGIES

Python is the best choice of programming language for machine learning. It contains libraries that are very useful for data analysis and computation purposes. It is very easy to use for its simplicity and readable codes. It saves a huge amount of time for the developers which can be used for developing rather than stuck in complex codes. Python provides libraries like Matplotlib that can be used to draw graphs in Analysis.

Flask is a web-based framework which is written in python. It is used for smaller project. It doesn't require any libraries. Its light weight property makes it very easy to use and programmer can make the web application faster. It also allows to run machine learning models.

HTML is the standard markup language for Web pages. It is easy to learn and helps to create Website. The tags used in html makes it easy to create website and put contents in proper position. It also allows CSS to give styling to the website. It allows to play audios and the tags like <form>, <input>, are used for uploading audio files.

CSS is used to style the HTML pages to give them a better look. It also helps in positing the elements and also does a little bit of animation. CSS provides a large scale of options for colour, sizes, padding etc.

JupyterLab is a web-based interactive development environment for Jupyter notebooks. The interactive outputs in a single notebook are what makes it preferable for machine learning projects.

PyCharm is an IDE that supports Flask framework for web-development. It also supports HTML, CSS and JavaScript. Debugging and inspection is easy to do with this IDE. Various inputs can be given to the system and see result in output windows which is very useful for testing purpose.

Pandas package is available in python that is used to work with dataset.

Numpy package is used to deal with the arrays of audio features and data.

Sklearn is a free library in Python used for training the model with the help of functions like train_test_split, MLPClassifier and accuracy_score.

MLP Classifier is an artificial neural network used to that is used for emotion prediction. The model is given audio signal features as input array which is processed in the hidden layers and gives output as predicted emotion.

Librosa is a python package used to load and play the audio. It is also used for plotting graphs of audio features like Soundwaves and MFCC.

Pickle is the module used to save and load the trained model.

Wavfile module under Scipy.io is used for read/write the audio files of .wav format.

PyTube is used download YouTube videos.

Pydub is used to convert the videos into .wav audio format.

Moviepy is used to chunk the audio into smaller audio clips.

Glob is used to retrieve files and pathnames of the local system.

HARDWARE AND SOFTWARE REQUIREMENTS

3. HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware Requirements

- Processor: intel i3
- Hard Disk: 100GB and above.
- Ram: 4GB and above.
- Standard Keyboard and Mouse

3.2 Software Requirements

- Language: Python 3.9
- IDE: JUPYTERLAB v3.0, PyCharm 2021.1.3
- Front-end: HTML5, CSS3
- Flask 2.0.1

SOFTWARE REQUIREMENTS SPECIFICATION

4. SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Functional Requirements

4.1.1 Data Preparation

- **Download YouTube Videos:** Download a few videos from YouTube with clear audios and speech of English-speaking speakers.
- **Convert to .wav format:** Convert the downloaded videos into .wav audio format.
- **Clip Audio:** The audios need to be clipped at the start and end to avoid unnecessary noisy part of the videos.
- **Extract Audio Chunks:** The toughest part of data preparation of dividing a long audio file in smaller audio chunks of few seconds duration. To chunking audios manually one by one is a very long tiring process. So, to make the process easier the audios are split on the basis of silence period and threshold using python libraries.
- **Cleaning and masking Audio Chunks:** The chunks are further cleaned and saved in different folder where the trained model will be applied.

4.1.2 Data Visualization

- **Play Audio:** The audios can be played in Jupyter notebook to check the quality and tone.
- **Plotting Graphs:** The graphs are plotted for the audios to see in depth qualities of sounds and checks on features such as MFCC, MEL on which the emotions of speaker are to be classified.

4.1.3 Training RAVDESS dataset:

A Ryerson Audio Visual Database of Emotional Speech and Song i.e., RAVDESS is pretrained dataset containing 1440 audio files which is used to train the model for Emotion recognition in this project.

- **Download RAVDESS Dataset:** The RAVDESS dataset is available online for the sole purpose of emotion recognition.
- **Extract Features:** Extract features like MFCC, MEL, CHROMA from audios.
- **Select Emotions:** The emotions on which features to be observed and applied to Model like Happy, Angry, etc.
- **Split into Train and Test data:** Split the whole dataset into train and test.
- **Apply MLP model:** The Multi-Layer Perceptron classifier is used for classification of emotions.
- **Predict Test Set:** The test set of data is predicted.
- **Check Accuracy:** Accuracy of the model is important deciding factor. More the accuracy rate better the model works on data.
- **Save the model:** Save the trained model into local file for future use.

4.1.4 Apply Model into Dataset created from You Tube

- **Load Saved Model:** Load the saved model file into Jupyter Notebook,
- **Extract Features:** Extract features Such as MFCC, MEL, CHROMA of the audios.
- **Deploy Model:** Apply the model into the self-created dataset to predict emotion of the speaker.
- **Save Predicted Emotion into CSV File:** The predicted emotions are to be saved for future use like visualization.

4.1.5 Predicted Emotion Visualization

The predicted emotions are visualized using bar graph to show the emotion distribution in each video and overall dataset created from different videos. It also helps to show the dominated Emotion in a video. It gives a detailed analysis on the prediction of created dataset.

4.1.6 User Interface

A simple interactive user interface is created for the user to select the audio from local folder and predict the emotion.

4.2 Non-Functional Requirements

- **Compatibility:** This application is compatible in any system with internet browser like Chrome, Microsoft edge, etc.
- **Scalability:** The application can be scaled for more data with clean audios of various lengths.
- **Manageability:** Very easy to manage all components are already available the user only needs to give input as audio chunks.
- **Usability:** It is very easy to use with the interactive user interface provided. The simplicity of the interface makes sure that user can use the application without any hassle.
- **Performance:** Any good application is determined by their performance. The web application for Emotion recognition on video speech is a single web page that opens fast depending on internet connection. User has to upload an audio and the application provides the output in a matter of as fast as 5 seconds.

SYSTEM DESIGN

5. SYSTEM DESIGN

5.1 Architectural Diagram

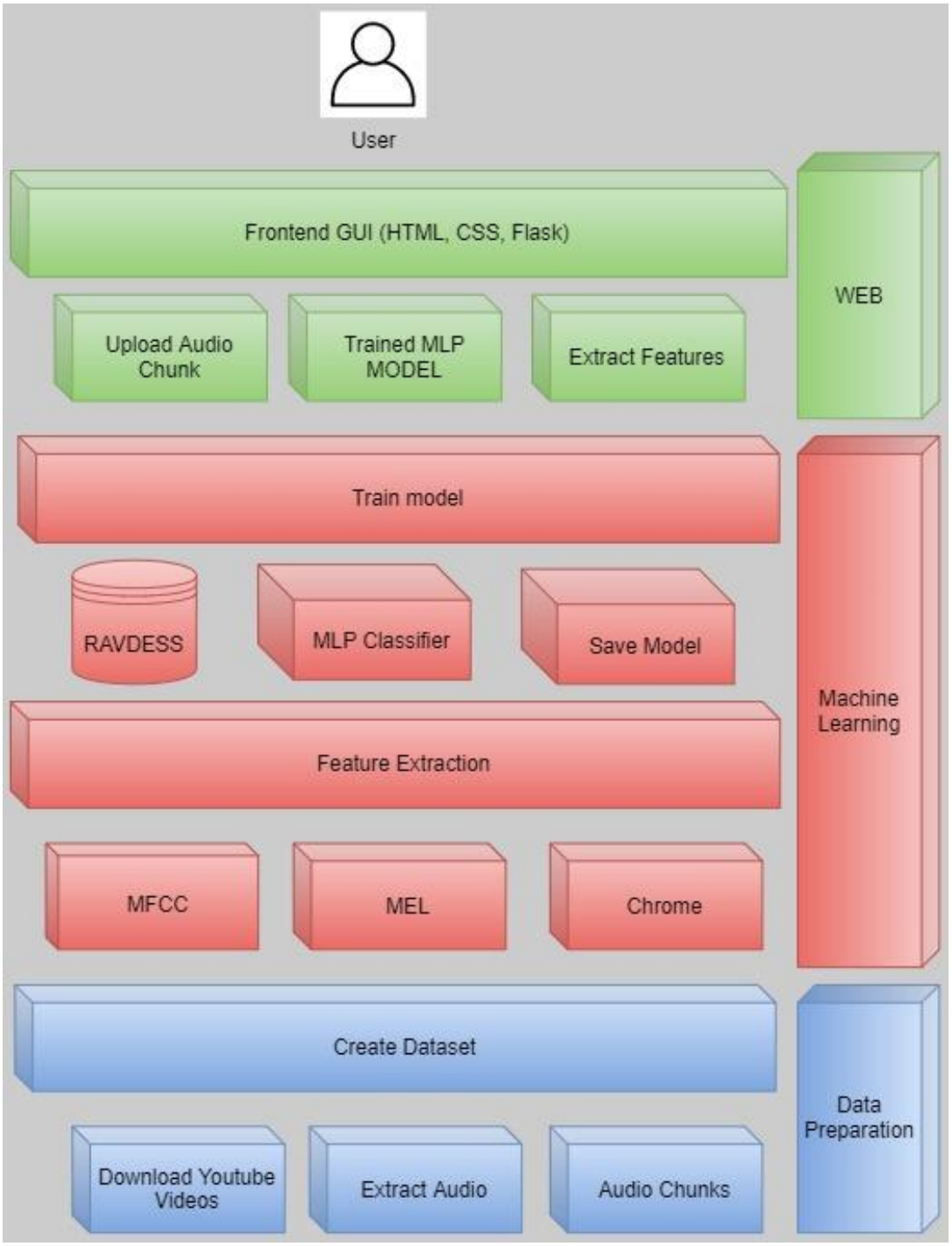


Fig 5.1 Architectural Diagram

The above diagram in fig 5.1 is a 3-tier vertical architectural diagram of the application. As it shows the user has access only to the frontend GUI. The User interface consist of HTML and CSS along with Flask to apply the business logic to it and apply the model. The user needs to upload an audio file and audio signal features will be extracted. The trained model will predict the emotion of the speaker in the audio clip.

The machine learning part is where the actual computation of emotion prediction is done with the help of MLP model. It consists of 2 parts including Feature extraction and model training. Features like MFCC, MEL and CHROMA are extracted from the audio signals and the array of features are given to MLP classifier network to process. The model gives emotion as output.

The first part which is in the bottom of diagram is dataset creation. This is the dataset upon which prediction and analysis shall be done. It consists of 3 main functionalities that is downloading YouTube Videos, extracting audios and chunking audio file into smaller clips.

5.2 Context Diagram

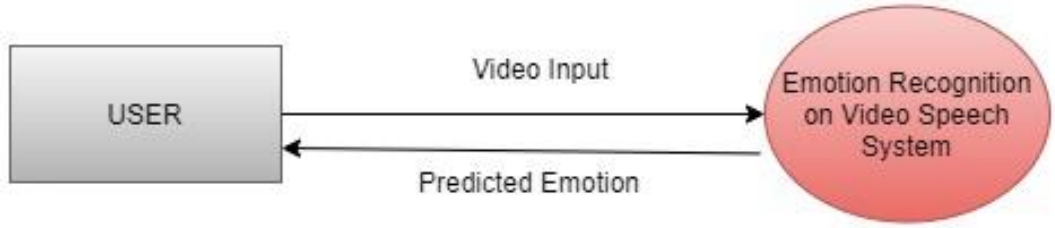


Fig 5.2 Context Diagram

The above context diagram shows the relationship between User and the system. The user will give video input which is downloaded YouTube Videos and the System that is the web page as end product will give predicted emotion of the audio clip extracted from the video.

5.3 Use Case Diagram

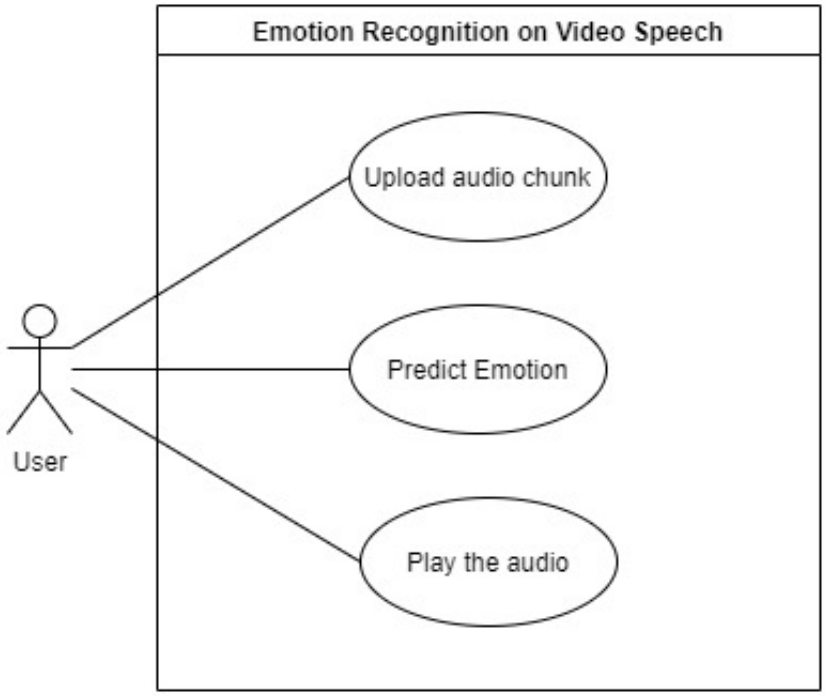


Fig 5.3 Use case Diagram for Emotion Recognition on Video Speech

The above use case diagram shows that there is one actor for the system that is the user. The user has three functionalities upload audio chunk, Predict Emotion and Play the audio. On Upload audio chunk the user has to upload an audio of .wav format to the application. On clicking submission button, the application will provide the emotion of the audio file. The used can also play the audio and listen to the audio uploaded by them.

DETAILED DESIGN

6. DETAILED DESIGN

6.1 Flow Diagram

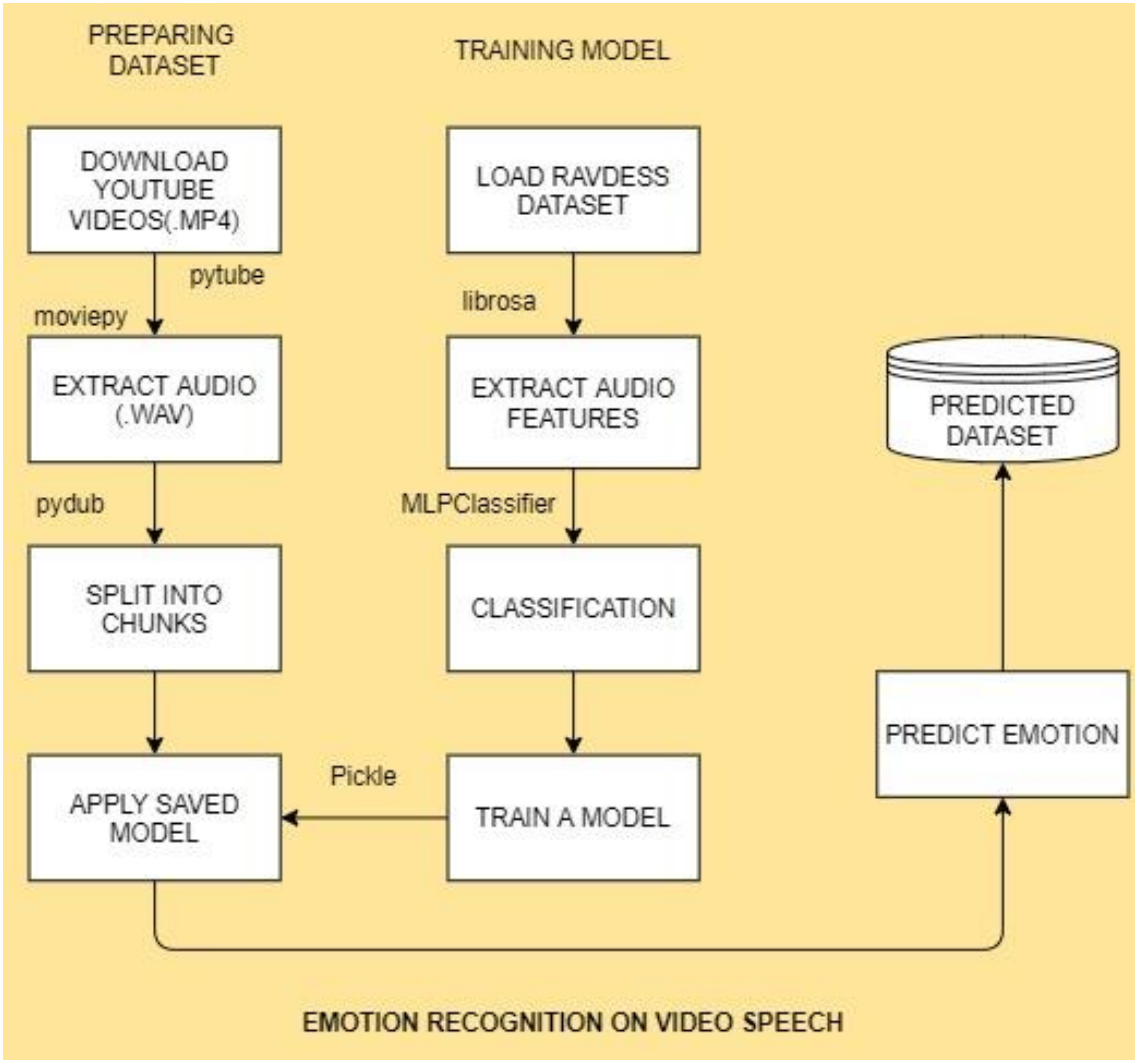


Fig 6.1 Flow diagram of the system.

The above flow diagram shows the detailed working of the system. All the packages used are also shown in the figure for each process.

First of all, a dataset has been created from the downloaded YouTube Videos with the help of Pytube. Audios are extracted from the videos using moviepy module of python.

These audios are further clipped into small audio chunks which will apply trained MLP classifier. This process is known Preparation of dataset.

The MLP classifier is trained with the help of RAVDESS dataset available online. It is the most important phase of the application as the accuracy of the model decides the credibility of the whole application. The features of audios are extracted which is then fed to input layer of MLP classifier. The classifier gives output as predicted emotion. Pickle module is used to save and load the model whenever required. This is how a model is trained and applied to a dataset.

The predicted emotion, files and speaker are saved and formed into a dataset. It is saved in the form of csv files so that it is user to store use. This dataset is further used for exhaustive analysis to see how the emotion prediction works which shall be discussed later.

IMPLEMENTATION

7. IMPLEMENTATION

7.1 Source Code

7.1.1 Creating Dataset

Download You Tube Videos:

```
from pytube import YouTube
link="https://youtu.be/9cA2KlV8zyQ"
try:
    yt = YouTube(link)
    filters = yt.streams.filter(progressive=True, file_extension='mp4')

filters.get_lowest_resolution().download(output_path='C:/Users/R.N.Mandal/Desktop/AVER/videos', filename='video_TonySamara')
    print('Video Downloaded Successfully')
except Exception as e:
    print(e)
```

Convert to .wav format:

```
import moviepy.editor as mp
video=
mp.VideoFileClip(r"C:\Users\R.N.Mandal\Desktop\AVER\videos\video_TonySamara.mp4")
video.audio.write_audiofile(r"C:\Users\R.N.Mandal\Desktop\AVER\audios\audio_TonySamara.wav")
```

7.1.2 Training Model

Feature Extraction:

```
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(X, sr=sample_rate).T, axis=0)
            result = np.hstack((result, mel))
    return result
```

Fig 7.1 Feature extraction code

The above figure 7.1 is the feature extraction code for the application the array ‘result’ consists of all audio features like MFCC, CHROMA and MEL.

MLP model training:

```
#Emotions to observe
observed_emotions=['calm', 'happy', 'angry','disgust']

#Load the data and extract features for each sound file
def load_data(test_size=0.2):
    x,y=[],[]
    for file in glob.glob("C:\\Users\\R.N.Mandal\\Desktop\\AVER\\clean_ravdess\\*.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) # split the data to training and testing and return it

#Split the dataset
x_train,x_test,y_train,y_test=load_data(test_size=0.10)

#Initialize the Multi Layer Perceptron Classifier
model=MLPClassifier(alpha=0.01, batch_size=256, epsilon=1e-08, hidden_layer_sizes=(300,), learning_rate='adaptive', max_iter=500)

#Train the model
model.fit(x_train,y_train)

#Predict for the test set
y_pred=model.predict(x_test)

#Calculate the accuracy of our model
accuracy=accuracy_score(y_true=y_test, y_pred=y_pred)

#Print the accuracy
print("Accuracy: {:.2f}%".format(accuracy*100))
```

Fig 7.2 MLP model training code

The above figure 7.2 is the code for training MLP classifier neural network. 4 emotions are taken for training purpose. RAVDESS data is divided into test and train set. The MLP is initialized and trained with train set. The accuracy is checked on test dataset.

7.1.3 Apply Model into created dataset

Save Model:

```
#save the file into the local file system
Pkl_Filename = "MlpModel3.pkl"
with open(Pkl_Filename, 'wb') as file:
    pickle.dump(model, file)
```

Load Model:

```
# Load the Model back from file
with open("MlpModel3.pkl", 'rb') as file:
    Emotion_Model = pickle.load(file)
Emotion_Model
```

Apply model to clean Audio chunks:

```
import glob,pickle
from tqdm import tqdm
from scipy.io import wavfile
a=[]
c=[]
for file in glob.glob
("C:\\Users\\R.N.Mandal\\Desktop\\AVER\\clean_audiochunks\\TonySamar*\\*.wav"):
    file_names = os.path.basename(file)
    print(file_names)
    a.append(file_names)
    new_features = extract_feature(file, mfcc=True, chroma=True, mel=True)
```

```
    emotion=Emotion_Model.predict([new_features])
    b=emotion[0]
    c.append(b)
    print(emotion)
```

Save data into a csv file:

```
dataset = pd.DataFrame({'file_name': a, 'predicted_emotion': c})
dataset['speaker']='TonySamara'
dataset['predicted_emotion'].value_counts()
dataset.to_csv('Prediction.csv',index=False)
#for appending
dataset.to_csv('Prediction.csv', mode='a', index=False, header=False)
data = pd.read_csv("Prediction.csv")
data
```

Visualization:

```
dataset['predicted_emotion'].value_counts().plot(kind='bar')
sns.catplot(y="speaker", hue="predicted_emotion", kind="count",
            palette="pastel", edgecolor=".6",
            data=data, height=10)
```

7.1.4 User Interface

App.Py

```
@app.route('/', methods=['POST'])
def upload_predict():
    if 'file' not in request.files:
        flash('No audio file given')
        return redirect(request.url)
    file = request.files['file']
    if file.filename == '':
        flash('No audio selected yet. please select an audio')
        return redirect(request.url)

    if file and allowed_file(file.filename):
        prediction = ""
        filename = secure_filename(file.filename)
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(file_path)
        new_feature = extract_feature(file_path, mfcc=True, chroma=True, mel=True)
        arr = model.predict([new_feature])
        prediction = arr[0]
        flash(filename)
        return render_template('index3.html', filename=filename, prediction= prediction)
    else:
        flash('Allowed audio types are only .wav')
        return redirect(request.url)


@app.route('/display/<filename>')
def display_audio(filename):
    print('display_audio filename: ' + filename)
    return redirect(url_for('static', filename='uploads/' + filename), code=301)


if __name__ == "__main__":
    app.run(debug=True, threaded=True)
```

Fig 7.3 App.py code for business logic using Flask

The above figure 7.3 is the business logic of the application. Flask has been used for this purpose. The first module is used for uploading audio and its verification along with the prediction code using the trained and saved model. The second module is used for displaying the audio control for the uploaded audio.

Index.html

```
<h3>Select a file to upload and predict</h3>
<p>
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        <ul>
        {% for message in messages %}
          <li>{{ message }}</li>
        {% endfor %}
        </ul>
      {% endif %}
    {% endwith %}
</p>
{% if filename %}
    <div >
        <audio controls="controls" src="{{ url_for('display_audio', filename=filename) }}" type="audio/wav"></audio>

        <p style="line-height: 1.8">The speaker is : <span style="color:rgb(255, 77, 77);"> {{prediction}}</span></p>
    </div>
{% endif %}
<form method="post" action="/" enctype="multipart/form-data">
    <dl>
        <p>
            <input type="file" name="file" class="form-control" autocomplete="off" required style="width:300px;">
        </p>
    </dl>
    <p>
        <input type="submit" value="Submit" class="btn btn-info">
    </p>
</form>

</div>
</div>
<footer style="text-align:center;">&copy; Copyright 2021 Riya Mandal, MCA -
<a href="mailto:riyamandal.dm@gmail.com">riyamandal.dm@gmail.com</a>
</footer>
</body>
```

Fig 7.4 Index.py code for html page

The above figure 7.4 is the code for interface designing. The page consists of form where user can upload the audio file and predict emotion.

7.2 Screen Shots

Visualization:

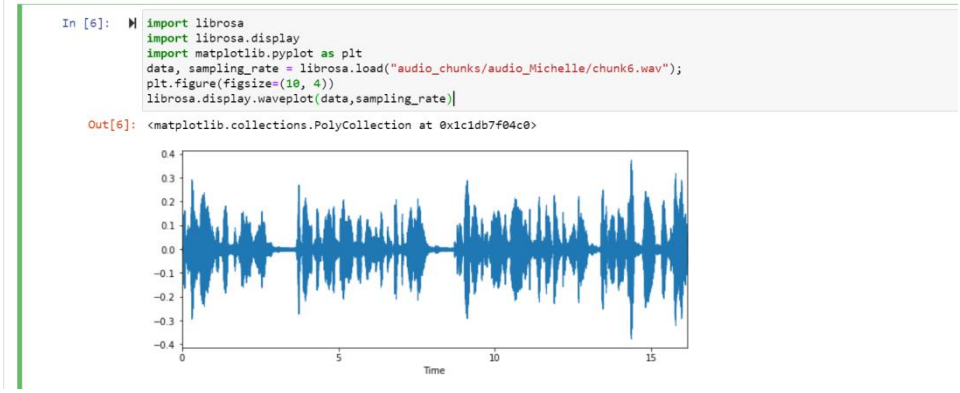


Fig 7.5 Audio Wave plot

The above figure 7.5 shows the audio wave plot where the audio file is converted into time-frequency graph. The graph shows the loudness of the audio with respect to time for the entire file.

MFCC Features:

```
In [6]: x, sr = librosa.load('audio_chunks/audio_Michelle/chunk6.wav')
        mfccs = librosa.feature.mfcc(x, sr=sr)
        print(mfccs.shape)
        plt.figure(figsize=(13, 4))
        librosa.display.specshow(mfccs, sr=sr, x_axis='time')
        plt.title("Mel Frequency Capstrum  Coefficients")

        (20, 867)

Out[6]: Text(0.5, 1.0, 'Mel Frequency Capstrum  Coefficients')
```

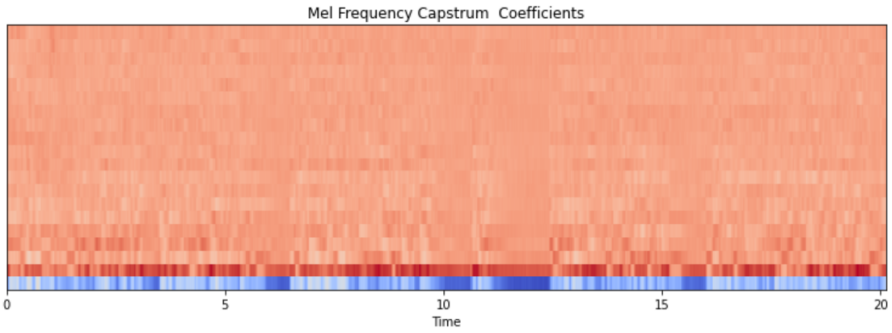


Fig 7.6 MFCC features in graph

The above figure 7.6 shows the MFCC feature of audio file that shows the Mel Frequency Cepstral Coefficients. It converts the frequency into Mel Scale which is used to measure human sensitivity. This is why it makes an ideal choice of feature for Emotion Recognition on Video Speech.

You Tube video

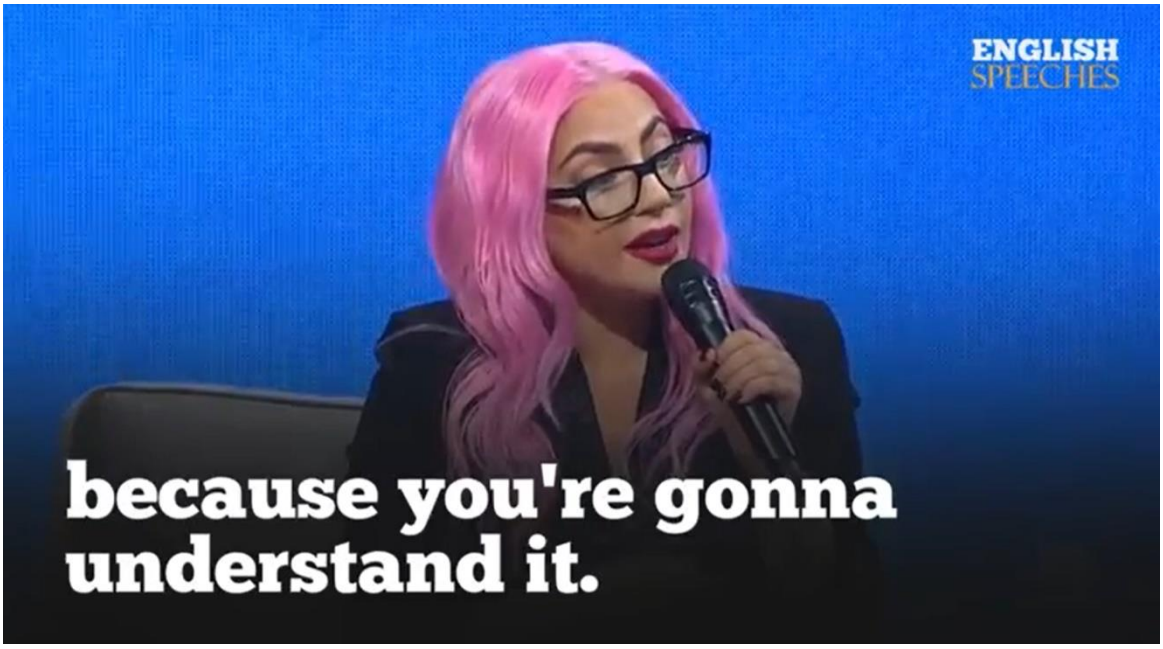


Fig 7.7 Playing a You Tube video

The above figure 7.7 is of a downloaded YouTube video for the prediction task. It is one of the many such videos where the speaker is talking in public. These videos are less noisy which makes them best choice for the emotion prediction.

Clean audio

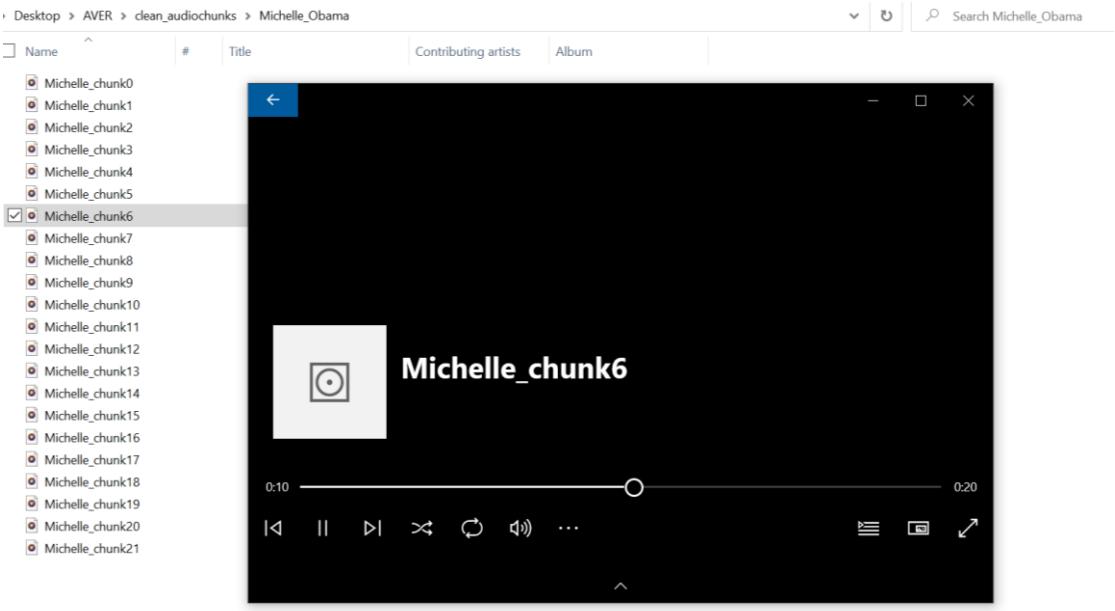


Fig 7.8 Playing an audio chunk

The above figure 7.8 shows the audio chunks being played locally. These audio chunks are extracted from the YouTube Videos. They are made by clipping the audio when silenced for some time thus making the chunking process faster.

User Interface

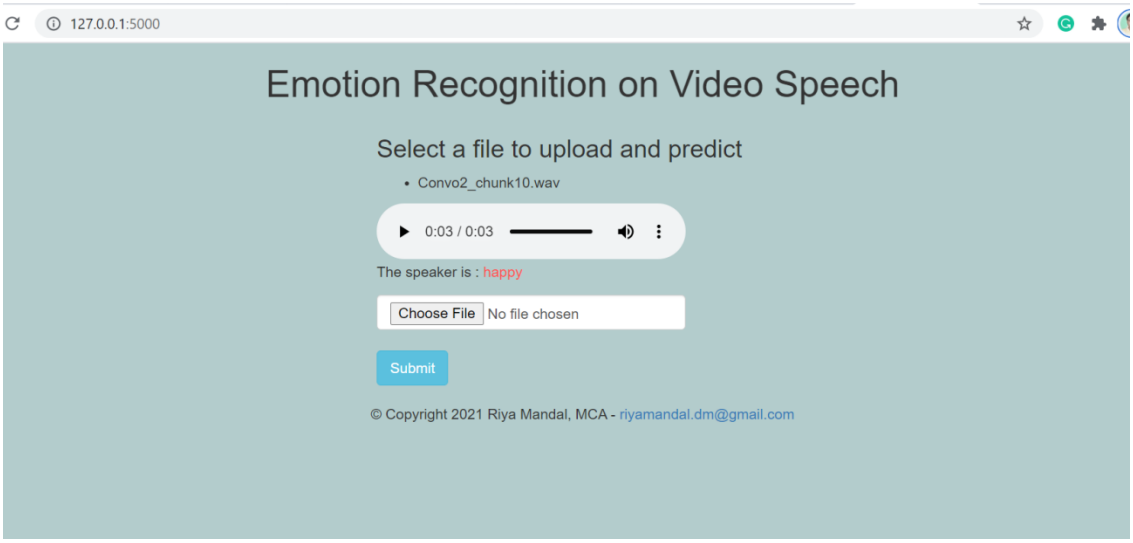


Fig 7.9 User Interface to choose a file and predict the emotion

The above fig 7.9 shows the user interface where user will upload the audio with the help of button choose file. And on submission, the predicted emotion will show up.

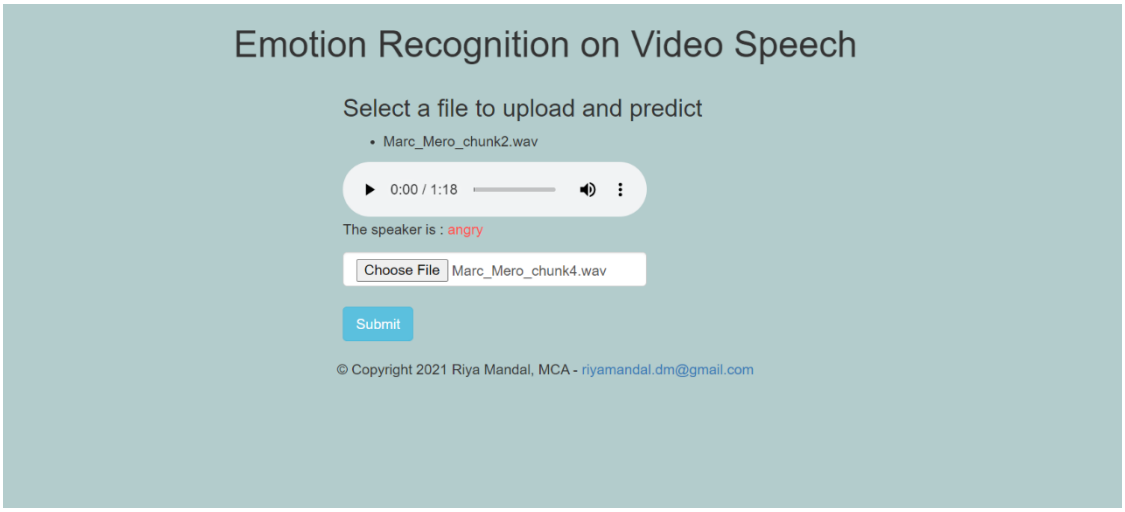


Fig 7.10 User Interface predicting ‘angry’ emotion

The above figure 7.10 shows the file name above “Marc_Mero_chunk2.wav” and a audio control where user can play the audio and change the volume. The emotion predicted here is “Angry”. So, it means that the speaker during this interval of time was angry.

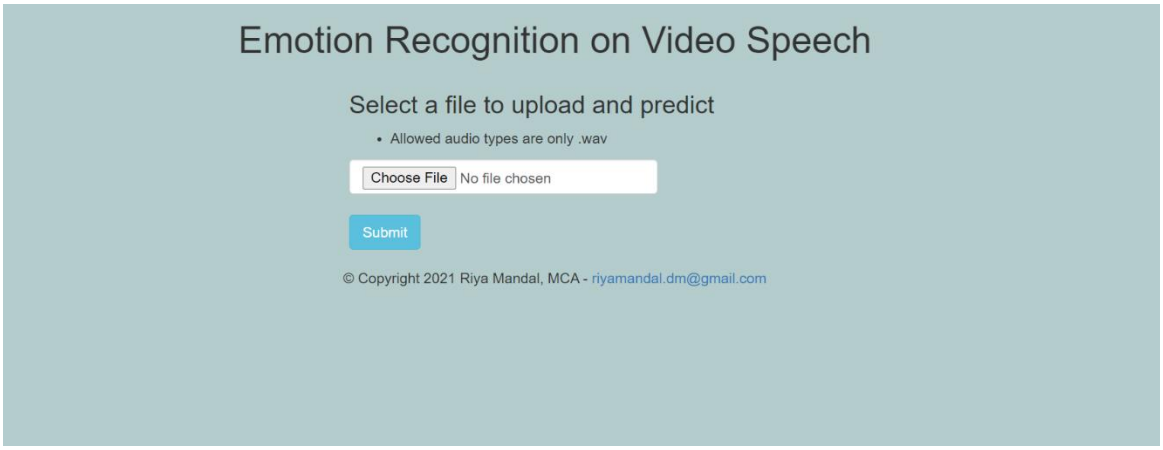


Fig 7.11 User Interface showing message when uploaded different file format

The above figure 7.11 shows the same user interface when the user chooses a file other than .wav audio file. This gives a message that says “Allowed audio types are only .wav”. This is to showcase the test case and for user to understand what type of file the application will take.

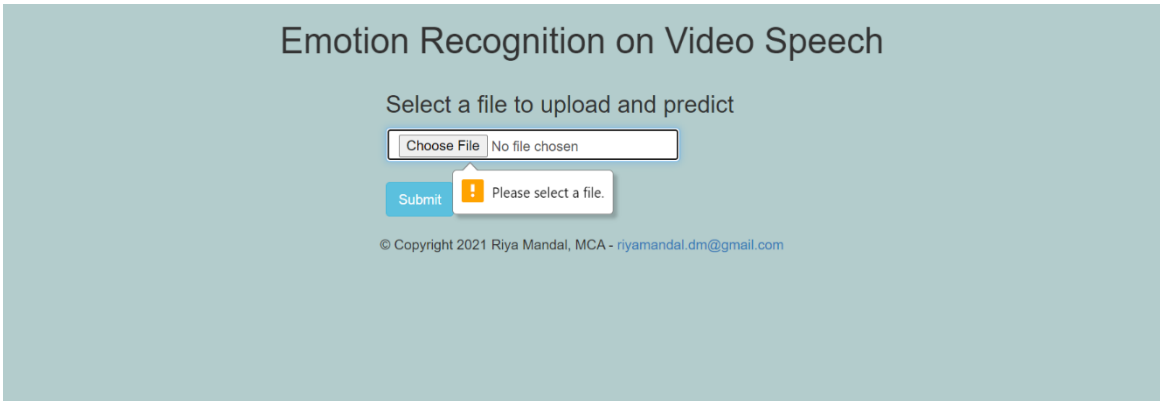


Fig 7.12 User Interface showing message when not selected any file

The above figure 7.12 shows another situation when the user does not select any file but clicks on “submit” button. It is mandatory to upload a file to the application of .wav audio type to get the emotion predicted.

SOFTWARE TESTING

8. SOFTWARE TESTING

8.1 Manual Testing

8.1.1 Test Case for selecting file for Emotion Prediction

Test Case ID 01		**Test Case Description**		File Selection
Created By	Riya Mandal	**Reviewed By**	Riya Mandal	**Version** 1.3
QA Log This testing is done for the selection of audio files for prediction. The testing is done in the output terminal each test cases.				
Tester Name	Riya Mandal	**Date of testing**	July 1, 2021	**Test Case (Fail/Pass)** Pass

Sl. No.	***Prerequisites:***		**Sl. No**	**Information about Data**
1	***http://127.0.0.1:5000/***		1	Username: NA
2	Internet Connectivity		2	Password: NA

Sl. no	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/Not executed**
1	Navigate to "http://127.0.0.1:5000/"	EMOTION RECOGNITION ON VIDEO SPEECH website should open.	EMOTION RECOGNITION ON VIDEO SPEECH website opened.	Pass
2	Click Submit without any file selected	A message should show "Please select a file".	A message pops up saying "Please select a file".	Pass
3	Click on "Choose File"	Local file system should open.	Local file system opened.	Pass
4	Select a picture of .png format.	Message shows "Allowed audio types are only .wav"	Message showed "Allowed audio types are only .wav"	Pass

8.1.2 Test Case for Predicting Emotions

Test Case ID 02		**Test Case Description**		Predict Emotion Of Audios	
Created By	Riya Mandal	**Reviewed By**	Riya Mandal	**Version** 1.3	
QA Log This testing is done for the prediction of audio files as output. The testing is done in the output terminal each test cases.					
Tester Name	Riya Mandal	**Date of testing**	July 1, 2021	**Test Case (Fail/Pass)**	Pass
Sl. No.	***Prerequisites:***		**Sl. No**	**Information about Data**	
1	***http://127.0.0.1:5000/***		1	Username: NA	
2	Internet Connectivity		2	Password: NA	

Sl. no	**Step Details**	**Expected Results**	**Actual Results**	**Pass/Fail/Not executed**
1	Navigate to "http://127.0.0.1:5000/"	EMOTION RECOGNITION ON VIDEO SPEECH website should open.	EMOTION RECOGNITION ON VIDEO SPEECH website opened.	Pass
2	Click on "Choose File" button and select audio file of .wav format.	User should see the selected file name in the interface.	User can see the audio file name.	Pass
3	Click on 'Submit' button.	An audio interface and name along with predicted emotion is shown.	The audio name and interface are shown and predicted emotion "The speaker is: angry"	Pass
4	Play the audio	The audio should get played.	The audio is played.	Pass

MODEL EVALUATION AND PERFORMANCE

9. MODEL EVALUATION AND PERFORMANCE

Multi-layer Perceptron classifier:

The multi-layer perceptron (MLP) classifier is very widely used classifier in Machine Learning. It is a feedforward artificial neural network model that takes input data sets and process in the hidden layers which then gives set of appropriate outputs.

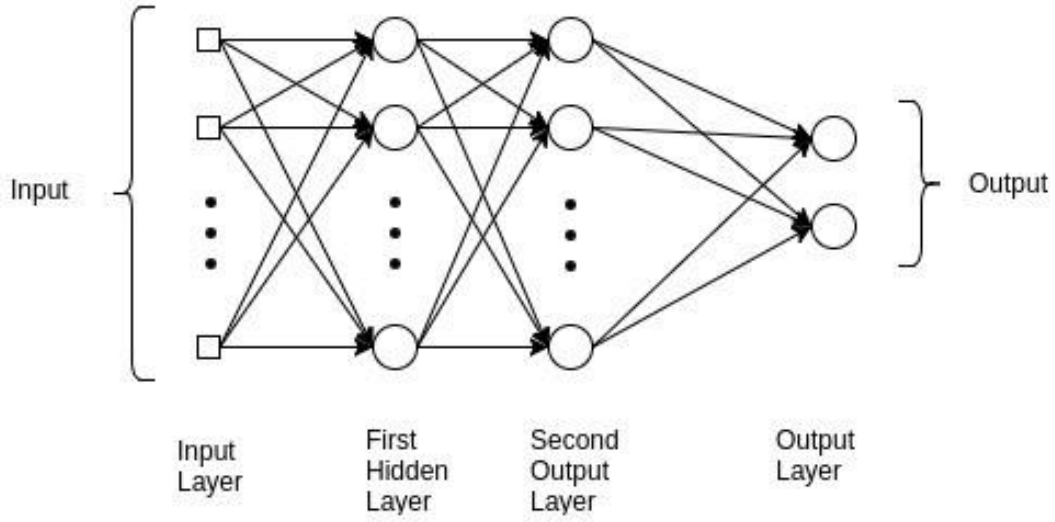


Fig 9.1 A MLP Classifier model

The fig 9.1 shows that the MLP model consists of multiple layers and every layer is fully connected to the next layer. The nodes of each layer are called neurons that has activation functions to each of them except for the neurons in input layer. The layers present in between the input and output layer are known as hidden layers.

In this project the dataset is very large and multi-class emotion classification is need to be performed. So, the artificial neural network fits best for this kind of project and that is why MLP classifier is used here.

The MLP model is initialized as:

alpha=0.01

batch_size=256

epsilon=1e-08

hidden_layer_sizes=300

learning_rate='adaptive'

max_iter=500

Here there is only one hidden layer with 300 nodes. In this model the input is given as array of features and output is the prediction of emotions.

Performance:

The MLP classifier gives an accuracy rate of 81.82% on test data. This shows the performance of model is very good with high accuracy rate

```
In [24]: from sklearn.metrics import classification_report
         print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
angry	0.80	0.76	0.78	21
calm	0.92	0.92	0.92	24
disgust	0.65	0.81	0.72	16
happy	0.92	0.75	0.83	16
accuracy			0.82	77
macro avg	0.82	0.81	0.81	77
weighted avg	0.83	0.82	0.82	77

Fig 9.2 Classification report of MLP model

Here, the report in fig 9.2 gives overall report for each class of prediction such as happy, angry, calm and disgust in the testing data.

Precision determines not to label an output positive that is actually negative for the given MLP classifier. Accuracy of positive predictions for happy and calm is highest.

Precision is defined by the formula TP/ (TP + FP) where TP is True Positive and FP is False Positive.

Recall is the ability of a classifier to find all positive instances. Fraction of positives that were correctly identified. Calm has highest percentage for this.

Recall is defined by the formula TP/(TP+FN) where TP is True Positive and FN is False Negative.

The F1 score is a weighted harmonic mean of precision and recall where the best score is 1.0 and the worst score is 0.0.

F1 Score is defined by the formula 2*(Recall * Precision) / (Recall + Precision)

Support is defined by the number of actual occurrences of the output class in the given dataset. It diagnoses the entire evaluation process.

This report shows the entire performance of the MLP model used for this application. The performance is very satisfactory as the accuracy and precision of the model is very high which is good for any machine learning application that means the machine is more accurate.

RESULT AND DISCUSSION

10. RESULT AND DISCUSSION

The trained MLP model is applied on the self – created Dataset of audio chunks made from YouTube videos containing some public speeches and conversation videos.

File_Name	Predicted_Emotion	Speaker
Michelle_chunk0.wav	angry	Michelle Obama
Michelle_chunk1.wav	happy	Michelle Obama
Michelle_chunk10.wav	happy	Michelle Obama
Michelle_chunk11.wav	angry	Michelle Obama
Michelle_chunk12.wav	happy	Michelle Obama
Michelle_chunk13.wav	happy	Michelle Obama
Michelle_chunk14.wav	happy	Michelle Obama
Michelle_chunk15.wav	happy	Michelle Obama
Michelle_chunk16.wav	happy	Michelle Obama
Michelle_chunk17.wav	angry	Michelle Obama
Michelle_chunk18.wav	happy	Michelle Obama
Michelle_chunk19.wav	happy	Michelle Obama
Michelle_chunk2.wav	disgust	Michelle Obama
Michelle_chunk20.wav	angry	Michelle Obama
Michelle_chunk21.wav	angry	Michelle Obama
Michelle_chunk3.wav	happy	Michelle Obama
Michelle_chunk4.wav	happy	Michelle Obama
Michelle_chunk5.wav	happy	Michelle Obama
Michelle_chunk6.wav	happy	Michelle Obama
Michelle_chunk7.wav	happy	Michelle Obama
Michelle_chunk8.wav	happy	Michelle Obama
Michelle_chunk9.wav	happy	Michelle Obama

Table 10.1 Predicted emotion of a video

The above table 10.1 shows the predicted emotions of a video saved in a file. The above table has 3 columns i.e... File_name, Predicted_Emotion and Speaker. The *File_Name* consist name of audio chunks during splitting phase where the full audio of a video is

converted into audio chunks very small sizes. It is done to make the feature extraction and prediction easier. The *Predicted_Emotion* is the emotion dictated by the MLP model with the features like MFCC, MEL and Chroma of the assigned audio chunk. The *Speaker* is the name of the speaker whose video is downloaded and it is also included in the file names. All these data are saved into a .csv file where it is combined with the data of all the other videos making one big predicted dataset ready for analysis.

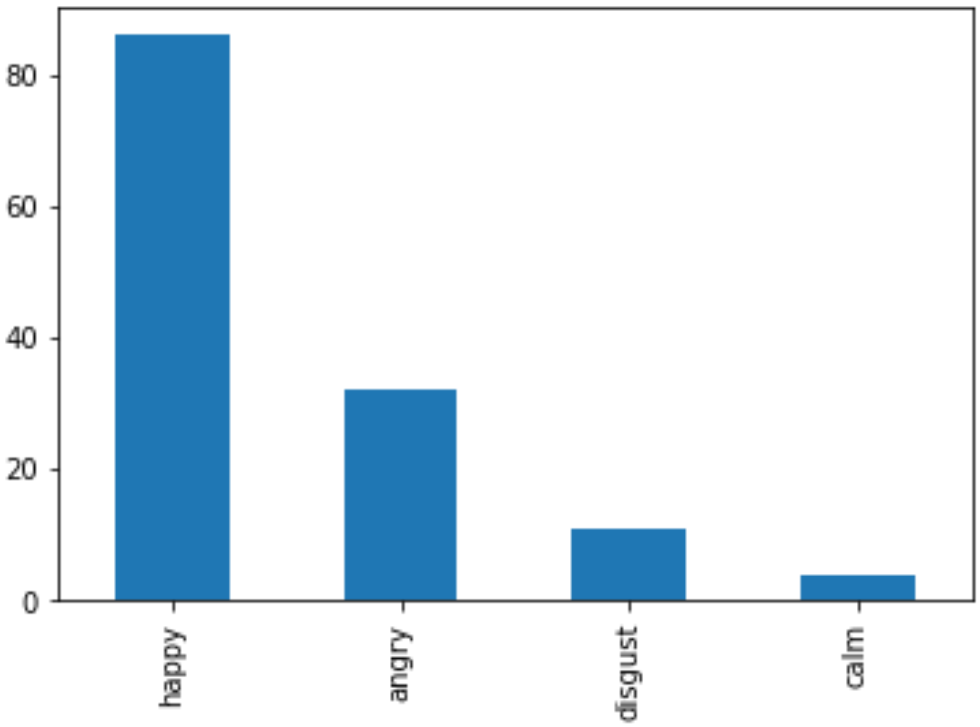


Fig 10.1 Bar graph of predicted emotions of a video

Emotion	**No of Audio chunks**
Happy	86
Angry	32
Disgust	11
Calm	4

Table 10.2 Distribution of emotions of a video

The above Fig 10.1 and Table 10.2 shows the distribution of emotions for a video. A video is converted number of smaller audio chunks which are put through a MLP model. This gives different emotions for all the audio chunks. This emotion distribution is very uneven because of the emotions like Happy and Angry that has dominated the entire video. This is the case for most of the videos with an exception of one or two videos.

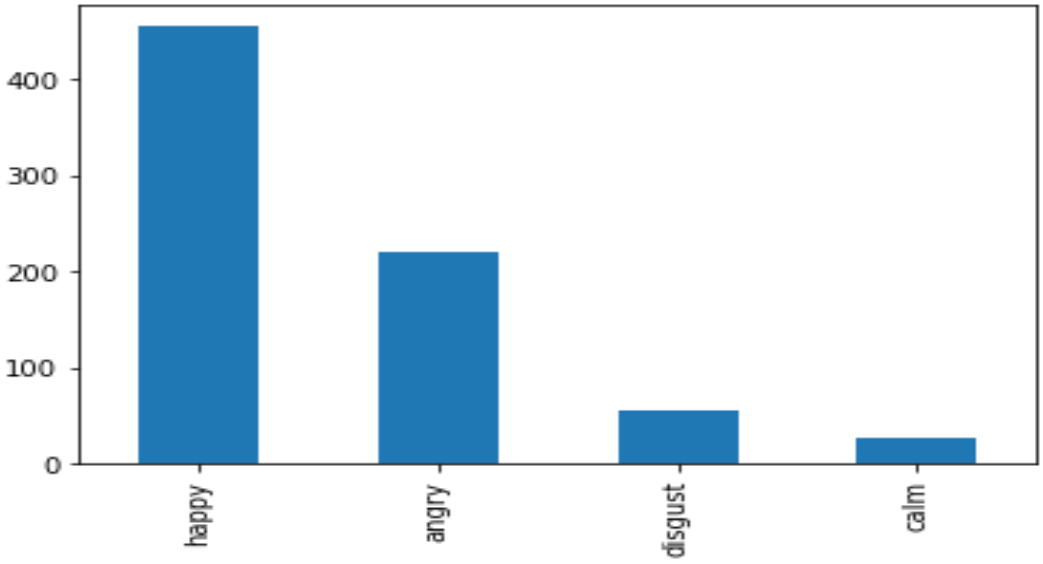


Fig 10.2 Bar graph for emotion distribution on overall dataset

The above graph in fig 10.2 shows the distribution of emotions of audio chunks in all the video files. This graph is the extension of fig 10.1 and shows the similar outcome where the dominated emotions remain *Happy* and *Angry*.

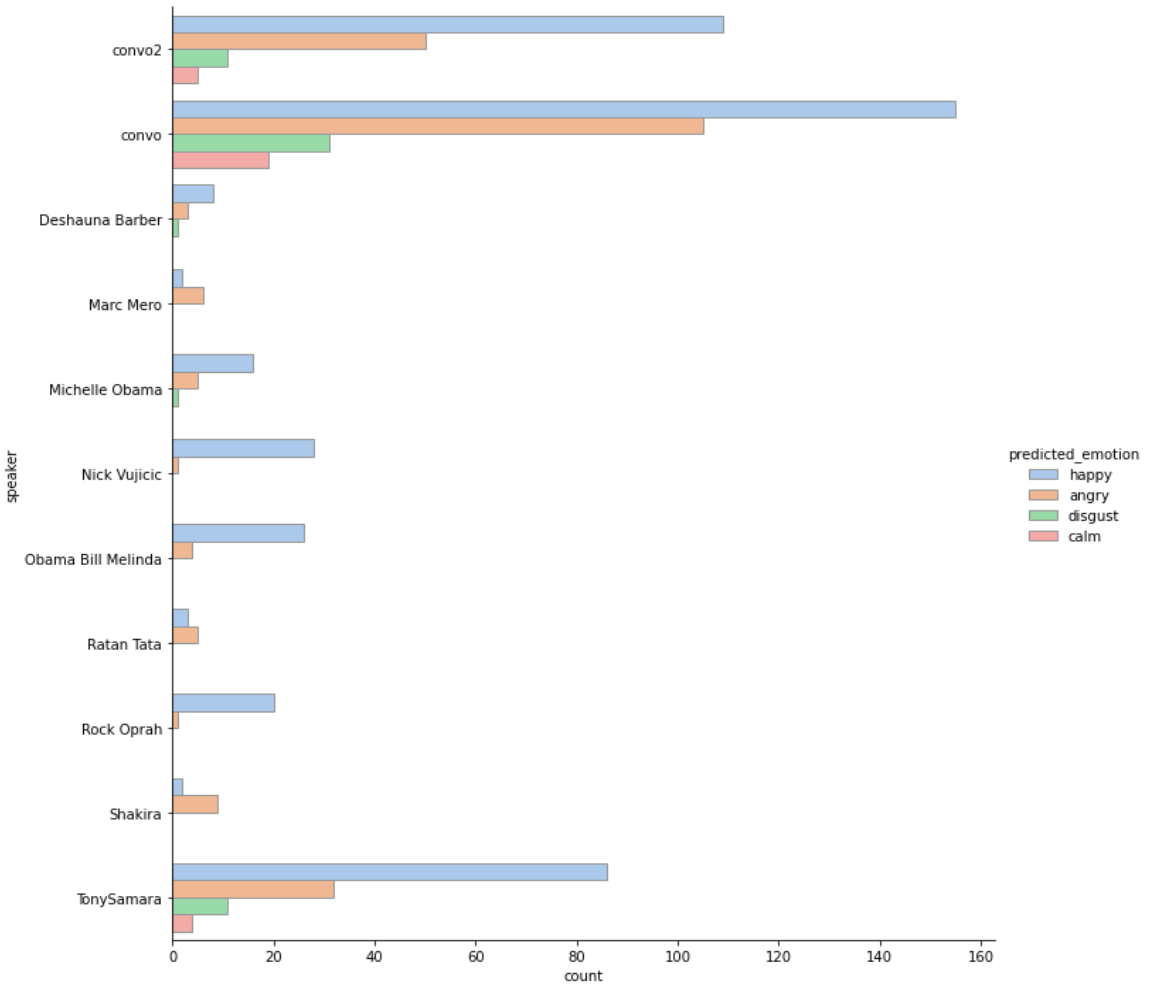


Fig 10.3 A graph depicting Speaker's emotions and no of audio chunks for each video

The fig 10.3 shows the distribution of emotions among each video very clearly. In the *y-axis* it has name of the speaker in the video and the *x-axis* represents the no. of audio chunks for each emotion.

The above result shows some very important analysis like the uneven distribution of emotions. The emotion like happy and angry is dominated. It could be because the videos mainly consist of public speeches and conversation that has less emotion. It could also be because the model takes tones and pitches and not words as input for emotion prediction. This leads to the emotion distribution of created dataset as it is.

CONCLUSION

11. CONCLUSION

The accuracy rate of 81.82% is achieved with MLP model and 4 emotions i.e... Happy, Calm, Angry and Disgust which is very satisfactory. The accuracy rate differs for the combination of different Emotions. The emotion of the speaker changes in audio chunks over a period of time in video. Emotions like Happy and Angry is dominated in most videos. This application is tested for a wide range of dataset created from YouTube videos and a detailed analysis has been done.

The web application is an easy-to-use dynamic webpage where user can upload an audio chunk in .wav format and on submission the predicted emotion will be displayed. The output will be any of four emotions otherwise gives error on upload of different format data.

FUTURE ENHANCEMENT

12. FUTURE ENHANCEMENTS

This project work can be enhanced in future with following:

- The web application can have more feature of clipping and extracting chunks.
- More detailed analysis can be done with larger dataset and different videos.
- The same process and model can be applied to another dataset.
- The accuracy can be increased with different combination of emotions.
- The prediction can be done with other audio signal features as well.

BIBLIOGRAPHY

BIBLIOGRAPHY

Website References:

Download YouTube videos:

[1] https://www.geeksforgeeks.org/pytube-python-library-download-youtube-videos/

Extract Audio from Videos:

[2] https://www.codespeedy.com/extract-audio-from-video-using-python/

Features Description:

[3] https://www.analyticsinsight.net/speech-emotion-recognition-ser-through-machine-learning/

Research Paper References:

[4] S. N. Chennoor, B. R. K. Madhur, D. T. K. Kumar, and M. Ali, "HUMAN EMOTION RECOGNITION FROM AUDIO AND VIDEO SIGNALS." Department of Electronics and Communication Engineering, National Institute of Technology, Warangal, Telangana, India- 506004

[5] Harár, Radim Burget and Malay Kishore Dutta "SPEECH EMOTION RECOGNITION WITH DEEP LEARNING PAVOL" Conference Paper · February 2017

[6] Samuel Albanie, Arsha Nagrani, Andrea Vedaldi, Andrew Zisserman "EMOTION RECOGNITION IN SPEECH USING CROSS-MODAL TRANSFER IN THE WILD" Visual Geometry Group, Department of Engineering Science, University of Oxford, Seoul, Republic of Korea, October 22-26-2018

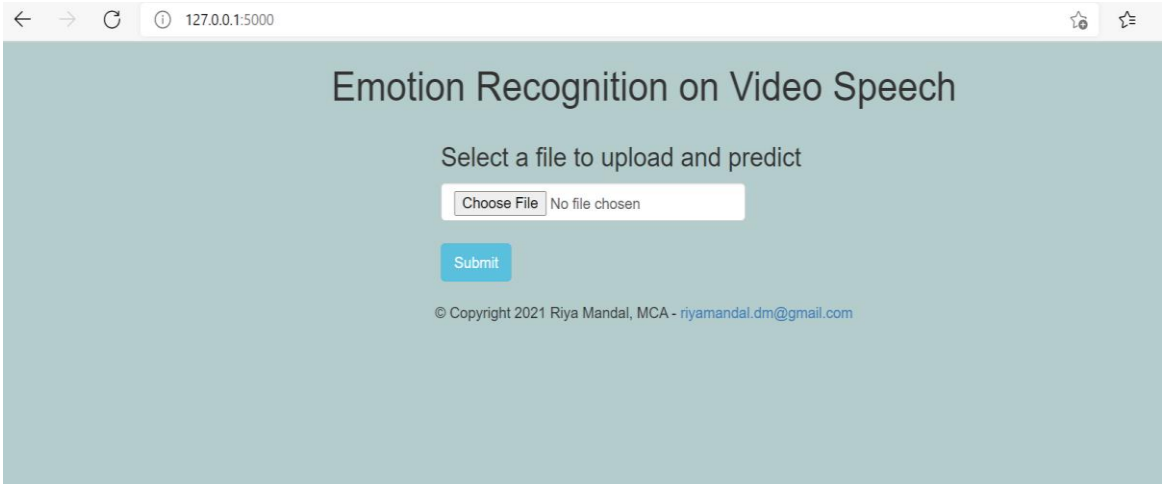
Text Book References:

[7] **Object Oriented Systems Analysis and Design Using UML by** Simon Bennett (4th Edition), McGraw Hill,2010

APPENDIX

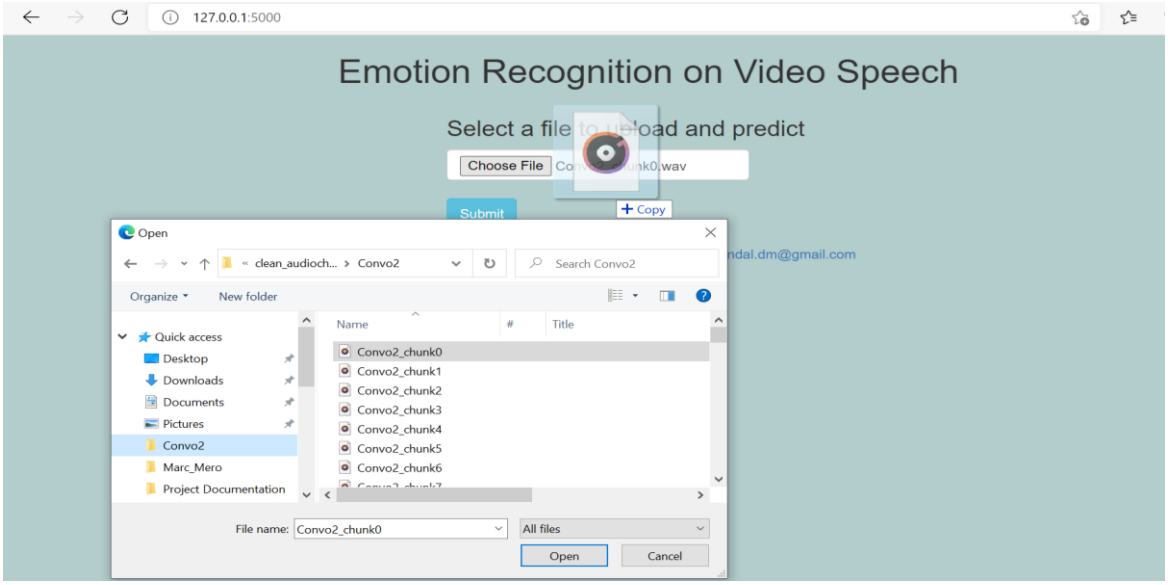
Appendix A

USER MANUAL

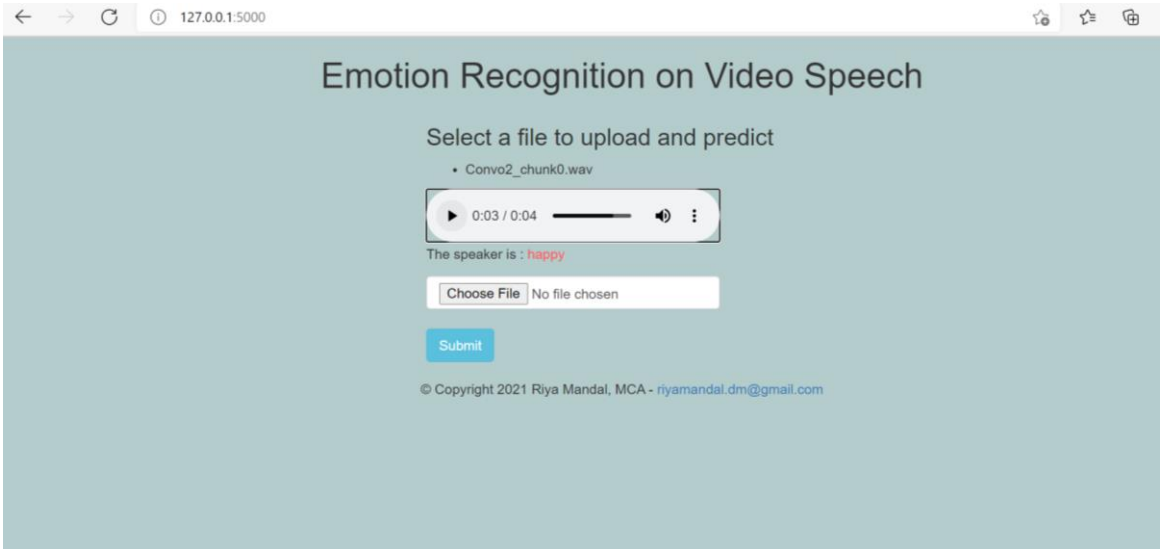


Step 1- Visit the web page at the URL http://127.0.0.1:5000/

Step 2- Click on "Choose File". A window with local file system will open.



Step 3- Select an audio file of .wav format and click open. The file can also be dragged and drop in the file box.



Step 4- Click on the submit button. On submission, the audio file will be uploaded and the emotion predicted by the model will be shown the page.

Step 5- An audio control will also be displayed in the page. Click on play button to listen to the file.

NOTE: Only .wav audio files can be uploaded for prediction. For any other format media, the application will give an error message.

Appendix B

PLAGIARISM REPORT