A

PROJECT REPORT

on

EMOTION BASED MUSIC RECOMMENDATION SYSTEM

Submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

Session: - Jan-June 2023

Under Guidance of
Aaditya Maheshwari
Head of Industrial Project
Computer Science, and Engineering
Techno India NJR Institute of
Technology, Udaipur

Submitted by
Soumya Jaitawat (19ETCCS067)
Bhavini Jain (19ETCCS007)
Harshit Chaubisa (19ETCCS024)
Nandeshwari Ranawat
(19ETCCS041)
VIII Semester & CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001

MAY - 2023

A

PROJECT REPORT

on

EMOTION BASED MUSIC RECOMMENDATION SYSTEM

Submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

Session: - Jan-June 2023

Under Guidance of
Aaditya Maheshwari
Head of Industrial Project
Computer Science and Engineering
Techno India NJR Institute of Technology

Submitted by
Soumya Jaitawat (19ETCCS067)
Bhavini Jain (19ETCCS007)
Harshit Chaubisa (19ETCCS024)
Nandeshwari Ranawat (19ETCCS041)
VIII Semester & CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001

MAY – 2023

Department of Computer Science and Engineering
Techno India NJR Institute of Technology, Udaipur-313001

Certificate

This is to certify that project work titled **EMOTION BASED MUSIC RECOMMENDATION SYSTEM** by Soumya Jaitawat (19ETCCS067), Bhavini Jain (19ETCCS007), Harshit Chaubisa (19ETCCS024), Nandeshwari Ranawat (19ETCCS041) was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfilment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Aaditya Maheshwari
Head of Industrial Project
CSE,
TINJRIT,Udaipur
Date:27/05/2023.

Dr. Rimpy Bishnoi
Head of Department
CSE
TINJRIT, Udaipur
Date:27/05/2023.

TECHNO INDIA NJR
INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Techno India NJR Institute of Technology, Udaipur-313001

Examiner Certificate

This is to certify that the following student

Soumya Jaitawat
Bhavini Jain
Harshit Chaubisa
Nandeshwari Ranawat

of final year B.Tech. (Computer Science and Engineering), was examined for the project work titled

Emotion Based Music Recommendation System

during the academic year 2022 – 2023 at Techno India NJR Institute of Technology, Udaipur

Remarks:

Date:

Signature
(**Internal Examiner**)
Name :- Aaditya Maheshwari
Designation:- Head of Industrial Projects
Department: - CSE
Organization:- TINJRIT, Udaipur

Signature
(**External Examiner**)
Name :- Dr. Rimpy Bishnoi
Designation:- HOD
Department: - CSE
Organization:- TINJRIT ,Udaipur

PREFACE

In our proposed system, a mood-based music player is created which performs real time mood detection and suggests songs as per detected mood. This becomes an additional feature to the traditional music player apps that come pre-installed in our mobile phones. An important benefit of incorporating mood detection is customer satisfaction. The objective of this system is to analyse the user's image, predict the expression of the user and suggest songs suitable to the detected mood.

We have used a new approach for playing music automatically using facial emotion. Most of the existing approaches involve playing music manually, using wearable computing devices, or classifying based on audio features. Instead, we propose to change the manual sorting and playing.

In Chapter 1, we have given an overview of the music recommendation system and scope of emotion-based music recommendation system, along with the use cases.

Chapter 2 discusses the system requirements which includes functional requirement**,** non-functional requirements hardware requirements, software requirements.

Chapter 3 presents the study & weaknesses of current system along with the requirements of new system and feasibility study.

Chapter 4. It is all the literature review about the several emotions which this app can detect and provide the recommendation.

Chapter 5 and 6 are related to testing and app screenshots.

Chapter 7 talks about the conclusion and future scope of the project.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGTECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001

ACKNOWLEDGMENT

We take this opportunity to record our sincere thanks to all who helped us to successfully complete this work. Firstly, We are grateful to our **supervisor Mr. Aaditya Maheshwari** for his invaluable guidance and constant encouragement, support and most importantly for giving us the opportunity to carry out this work.

We would like to express our deepest sense of gratitude and humble regards to our

Head of Department Dr. Rimpy Bishnoi for giving invariable encouragement in our endeavours and providing necessary facility for the same. Also, a sincere thanks to all faculty members of CSE, TINJRIT for their help in the project directly or indirectly.

Finally, we would like to thank my friends for their support and discussions that have proved very valuable for us. We are indebted to our parents for providing constant support, love and encouragement. We thank them for the sacrifices they made so that we could grow up in a learning environment. They have always stood by us in everything we have done, providing constant support, encouragement and love

Soumya Jaitawat 19ETCCS067
Bhavini Jain 19ETCCS007
Harshit Chaubisa 19ETCCS024
Nandeshwari Ranawat 19ETCCS041

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1PURPOSE

1.1.1 INTRODUCTION

Communication is essential for sharing of messages or resource among individuals. Information can be shared among individuals either verbally or nonverbally. Facial Expression of an individual can be very useful in tapping the subject's mood and behavior. Human Emotion plays a crucial role in expressing the thought of an individual. They fall into one of the six basic types emotions which are sadness, happiness, anger, fear, disgust and surprise. These emotions can be determined by detecting changes in shape, size and movement of eyebrows, eyes and mouth. Modern devices that have access to internet contain millions of songs at any time. Music at any instant has the power to change the mood of an individual. Our main focus is to generate a music playlist which is automatically created by interpreting these human emotions.

Modern day technology includes automatic classification of a music based on Genres, artist, country, frequency and more. Mobile applications like Spotify, Saavn, wynk etc are some of the few applications who does automatic music classification. Here the users can create their own playlists or listen to the playlist generated by the application. Another modern-day application includes finding a song which a user is not able to identify. Mobile application like Shazam, trackID, Sound Hound can be very useful in such cases. Other Application like Musixmatch, YouTube music helps in displaying the lyrics of song at any instant and automatic translation of the lyrics from one language to another language. Music is said to be the greatest healing tool in the world. Therefore, we tap in that aspect of music to make wonders for the individual who is going through a certain emotion.

This Software Requirements Specification provides a complete description of all the functions and specifications of Emotion Based Music application (Emotion based music player).

The main objective of emotion-based music player is to help the user automatically play songs according to their emotions. It reduces user load to face the task of manually browsing the playlist of songs and choosing songs that support their current mood and behavior.

1.1.2 SCOPE

Scope of this project includes:

- It is a useful application for anyone who loves listening music.
- It can be used anywhere any time as it is a mobile application.
- User does not need to manually select song.

1.1.3 DOCUMENT OVERVIEW

The remainder of this document is 8 chapters, the first providing introduction of the project. It lists all the functions performed by the system. The second chapter consists of software requirements specification. The third chapter provides details about system analysis and design. The fourth chapter gives literature review information. The fifth chapter consists of snapshots of the complete project. The sixth chapter gives testing for the project. The seventh chapter talks about the conclusion and future enhancements of the project. The final chapter concerns with the bibliography.

This document is meant for describing all the features and procedures that were followed while developing the system. This document specially mentions the details of the project how it was developed, the primary requirement, as well as various features and functionalities of the project and the procedures followed in achieving these objectives.

Emotion based music player is proposed system based on real-time extraction of facial expressions from image captured into a specific emotion that will play song according to specified emotion such that the computation cost is relatively low.

With the effective use, any person can use the “Emotion based music player” for playing music according to their mood.

1.2 OVERALL DESCRIPTION

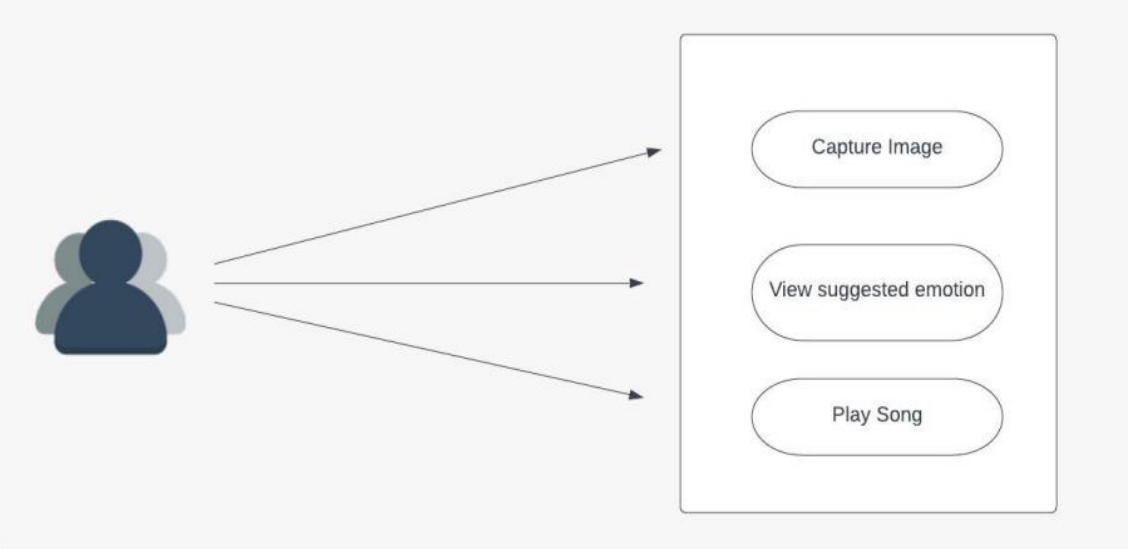
This project is an emotion-based music player which is a novel approach that helps the user to automatically play songs based on the emotions of the user. It recognizes the facial emotions of the user and plays the songs according to their emotion. In existing system user has to manually select the songs, randomly played songs may not match to the mood of the user, user has to classify the songs into multiple emotions and then for playing the songs user has to manually select a particular emotion. According to the emotion, the music will be played from the predefined playlist.

1.2.1 USE CASES

The user of the application can perform following actions;

- Capture image
- View detected emotion
- Play suggested song

Figure 1.1:



1.2.2 USER CHARACTERISTICS

Users are typically listeners of music. The user should be familiar with the Music Player related terminology like Play/Pause etc. The user should be familiar with the Internet and Mobile Application.

CHAPTER II

SYSTEM REQUIREMENT SPECIFICATION

2.1 FUNCTIONAL REQUIREMENT

In order for every software application to run properly, it needs to satisfy a lot of functions that are to be deployed in it. These functions are nothing but various operations that are performed in each step while developing the application. This step comes under the best practices of developing an application. Functional and Non-Functional Requirements together set a list of rules that govern the smooth running of an application and it also helps the developer and the user to determine the software and hardware requirements that are needed to run the application. Functional Requirements that are required to run this application is

Python: Python programming language was developed in the year 1991 by Guido Van Rossum. The syntaxes used in the language makes it very comfortable and easier for developers to work with. Because of this very reason, this programming language can be used both in small and large scale. They are dynamic and garbage collected.

OpenCV: It is a python-based library specifically used in image processing. It consists of many computer vision algorithms. It's a cross platform library and supports many deep learning algorithms and frameworks such as caffe, PyTorch, TensorFlow etc. Apart from image processing, OpenCV is also used for

- Video analysis
- Camera calibration

- Object detection
- Core functionality
- High-level GUI

Keras: Keras was developed by Francois Chollet which is a high-level neural network API. It can run on top of Theano, CNTK, TensorFlow etc. The Keras API is modular, user-friendly and extensible. The two most used Keras Models are

- Sequential
- Functional

2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are used to set conditions to monitor the performance characteristic of the application. It describes how a specific function in the application works.

They also determine the overall quality of the project and hence it is a very important aspect in any software development process.

The Non-Functional Requirements include

Usability: It refers to the easiness of the application and determines the ease with which it can be used by the user. Usability can be said to be high when the knowledge required to use the application is less and the efficiency of its functionality is high. It is also a main criterion which can determine the satisfaction of the user.

Accuracy: Accuracy determines the relative closeness of the value produced by the system to that of the ideal value. Less the difference between the system value and the ideal value, more is the accuracy. It is also one way to determine how the application works better compared to the other similar applications.

Responsiveness: Responsiveness is determined by completing the software operations with minimal errors or no errors. It is directly proportional to the stability and the performance of the application. The Robustness and Recoverability can also be determined by this criterion.

Scalability: Scalability is used to determine the growth of the project. It determines how much room the application can have in order to include more features in the future. It determines the sustainability of the project. This is one criterion which is used to develop long-term models for business growth.

2.3 HARDWARE REQUIREMENTS

Processor: Intel I3 processor

Storage Space: 500 GB.

Screen size: 15” LED

Devices Required: Web camera, Mouse and a Keyboard

Minimum Ram: 4GB and a good Internet connection.

2.4 SOFTWARE REQUIREMENTS

OS: Windows 7 and above /UBUNTU

Programming Language: Python

Software: JetBrains PyCharm Community Edition 2017.1.4 x 64

Backend: Keras

Additional requirements: TensorFlow

CHAPTER III

SYSTEM ANALYSIS AND DESIGN

3.1. STUDY & WEAKNESSES OF CURRENT SYSTEM

CURRENT SYSTEM

The Current system of listening music is highly complicated. In existing system user has to manually select the songs, randomly played songs may not match to the mood of the user.

WEAKNESSES IN CURRENT SYSTEM

The current system is as mentioned earlier very complicated and as compared to the new system. It also wastes the precious time of the user which can then be used in focusing on different activities and helping them improvise our mood.

Music plays a very primary role in elevating an individual ‘s life as it is an important medium of entertainment for music lovers and listeners. In today ‘s world, with the increasing advancements in the field of multimedia and technology, various music players have been developed with features like fast forward, reverse, variable playback speed, genre classification, streaming playback with multicast streams and including volume modulation, etc. These features might satisfy the user ‘s basic requirements, but the user has got to face the task of manually browsing the playlist of songs and choose songs supported their current mood and behavior. It would mean extra time and extra work for the user

Thus, the current system is in every way ineffective for listening music in these days when time is more costly than anything.

3.2. REQUIREMENTS OF NEW SYSTEM

3.2.1. USER REQUIREMENTS

The User requirements for the new system are to make the system fast, flexible, less prone to errors and save time.

- Time can be saved in creating playlist according to emotion of the user.

- A system that can play music according to specified emotion.

- The system doesn't need to have any records on hand which to use application.

- The New system should be more secure in managing user image data and reliable enough to be used in any condition.

- Finally, it should prove cost effective as compared to the current system.

3.3. FEASIBILITY STUDY

A key part of the preliminary investigation that reviews anticipated costs and benefits and recommends a course of action based on operational, technical, economic, and time factors. The purpose of the study is to determine if the systems request should proceed further.

3.3.1. DOES THE NEW SYSTEM CONTRIBUTE TO THE OVERALL OBJECTIVES OF THE USER?

The new system would contribute to the overall objectives of the User. It would provide a quick, error free and effective solution to the current process. The focus of this project will be entirely on the detection of facial expression and integrates it to the play music. As a prototype, the proposed model will detect only the basic emotion such as happy, sad, neutral, angry, fear, and surprise etc. The new system is flexible and scalable it can also be upgraded and extended to meet other complex requirements which may be raised in the future. However, it is up to the organization to upgrade or extend it.

3.3.2. CAN THE NEW SYSTEM BE IMPLEMENTED USING CURRENT TECHNOLOGY?

It would be very easy to set up the system in the current environment as the application can be used in mobile. The application is set up on the server, the system can be started as quick as required by the management.

3.4. FEATURES OF THE NEW SYSTEM.

The features available in the existing Music players present in computer systems are as follows: Manual selection of Songs, Party Shuffle, Playlists, Music squares where user has to classify the songs manually according to particular emotions for only four basic emotions. In today ‘s world, with ever increasing advancements in the field of multimedia and technology, various music players have been developed. The new system has been designed as per the user requirements so as to fulfill almost all them.

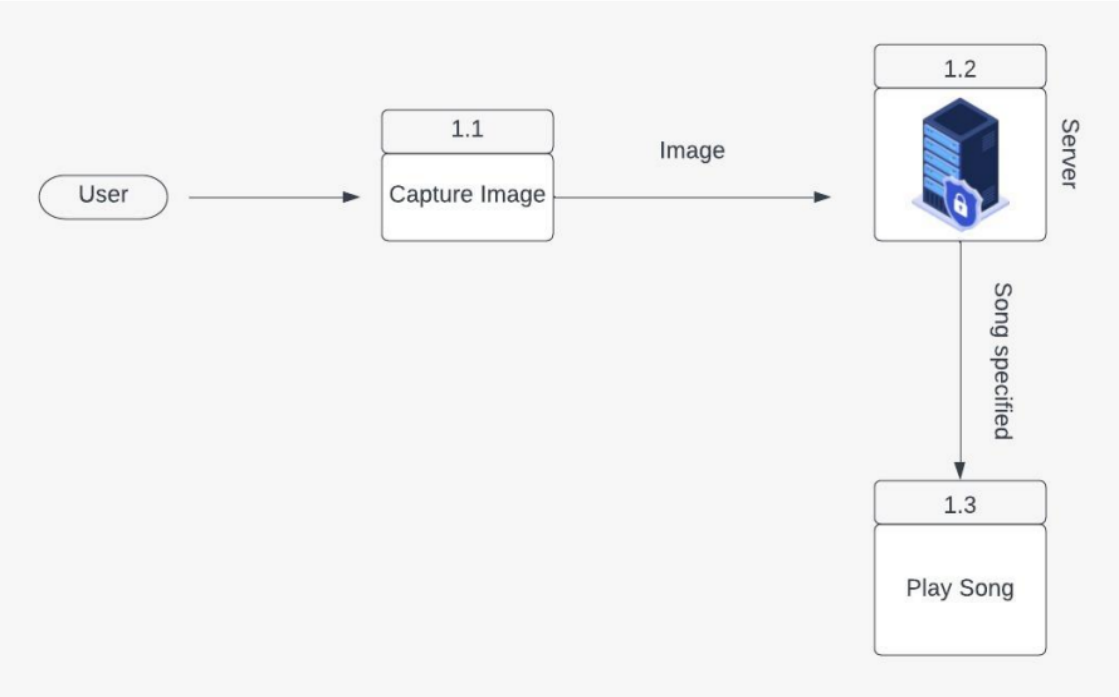
- It recognizes emotions through facial expressions.
- Recommend songs based on specified emotion.
- Easy to use.
- save time and efforts.

3.5. Data Flow Diagram (DFD)

The DFD (also known as bubble chart) is a simple graphical formalism that can be used to represent a system in terms of the input data into the system, various processes carried on these data, and the output data generated by the system. The main reason why the DFD technique is so popular is because the fact that the DFD is a very simple formalism – it is simple to understand and use. A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the 17 data flow among the

functions. Starting with a set of high-level functions that a system performs, a DFD model hierarchy represents various sub-functions.

FIGURE 3.1

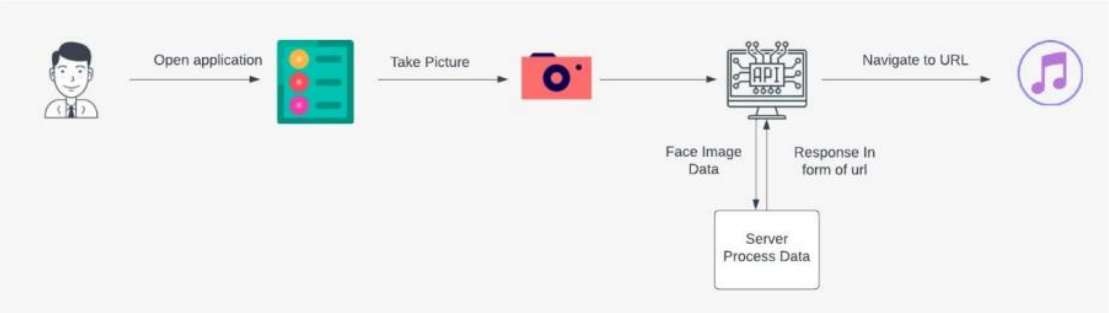


3.6 FLOW CHART

A flowchart is a graphical representation of a process. It is a diagram that illustrates the workflow required to complete a task or a set of tasks with the help of symbols, lines and shapes. Flowcharts are used to study, improve and communicate processes in various fields.

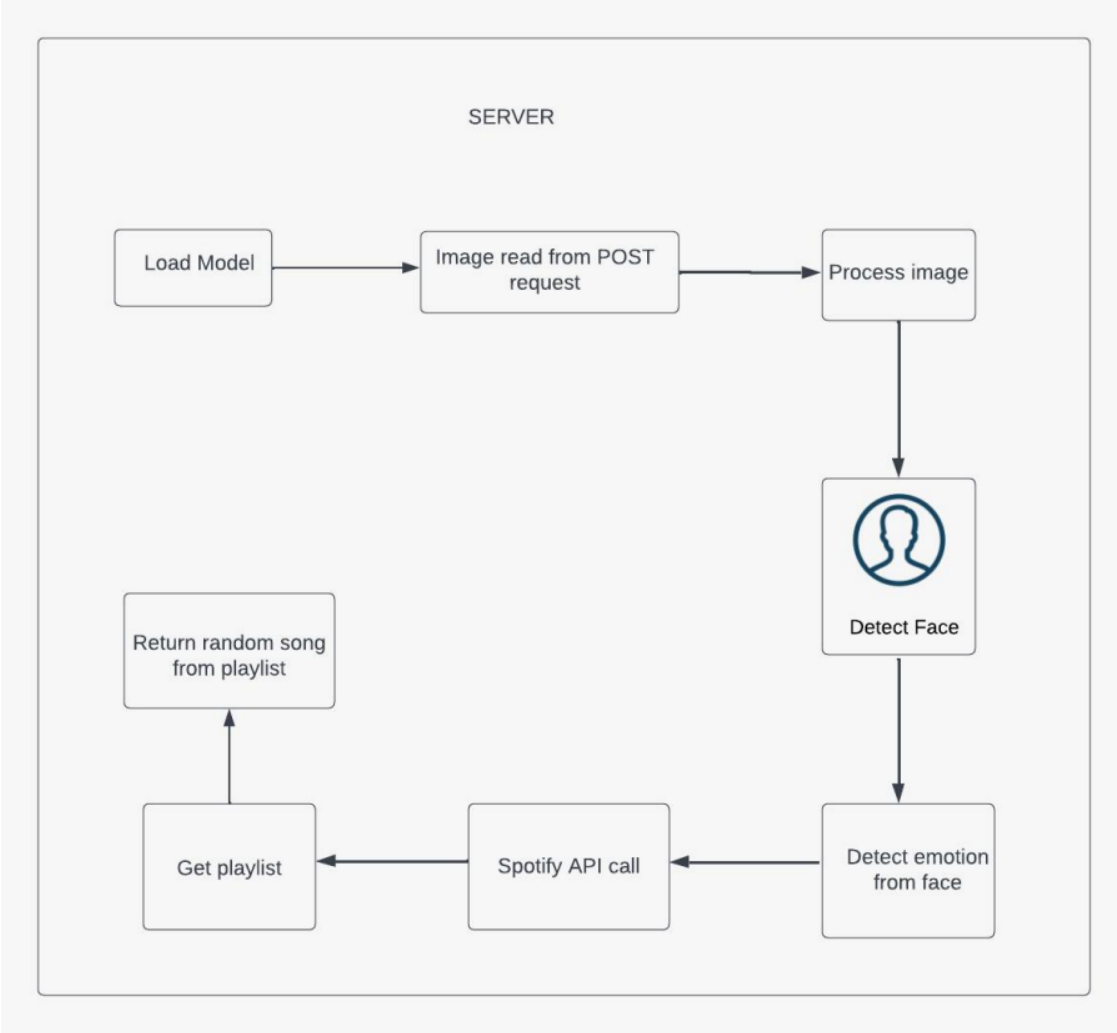
3.6.1 Client-Side Flow Chart

Figure 3.2



3.6.2 SERVER FLOW CHART

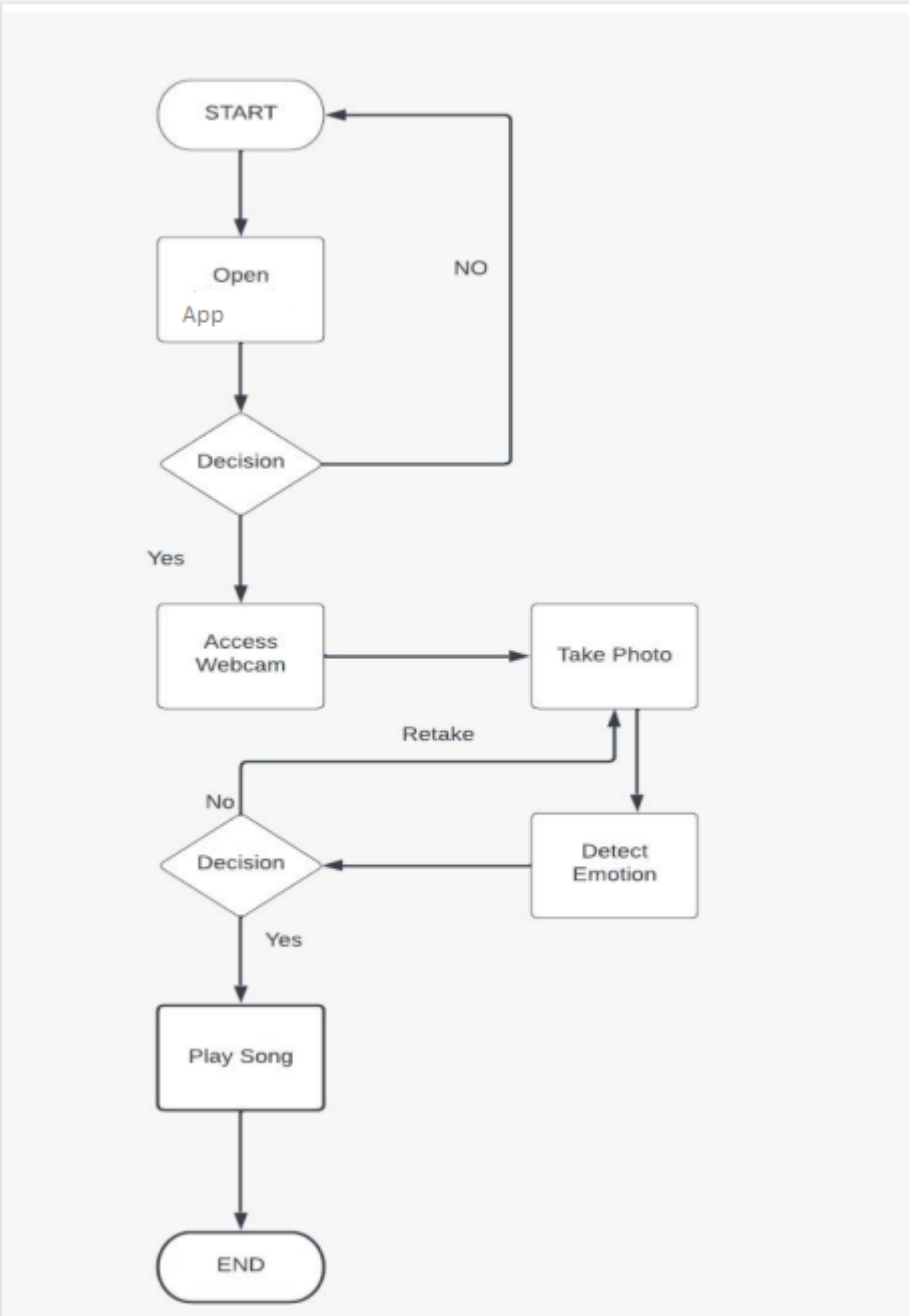
Figure 3.3



3.7. UML Modelling

3.7.1 ACTIVITY DIAGRAM

Figure 3.4: Activity Diagram for User



CHAPTER IV

LITERATURE REVIEW

4.1 LITERATURE REVIEW

There are several applications that provide facilities and services for music playlist generation or play a particular song and in this process, all manual work is involved. Now to provide there are various techniques and approaches have been proposed and developed to classify the human emotional state of behavior. The proposed approaches have only focused on only some of the basic emotions using complex technique.

Mary Duenwald had published an article which summarizes that scientists had did several studies and researches and shown that facial expressions across the globe fall roughly into seven categories:

- **Sadness:** The eyelids droop while the inner corners of the brows rise. When in extreme sadness, the brows will all push nearer together. As for the lips, both of its corners pull down and the lower lip may push up in a mope.

- **Surprise**: Both the upper eyelids and brows rise, and the jaw drops open.

- **Anger**: Both the lower and upper eyelids squeeze in as the brows move down and draw together. The jaw pushes forward, the upper and lower lip press on each other when the lower lip pushes upper a bit.

- **Neutral:** A blank expression is a facial expression characterized by neutral positioning of the facial features, implying a lack of strong emotion.

- **Disgust:** The individual's nose wrinkles and the upper lip rise while the lower lip protrudes.

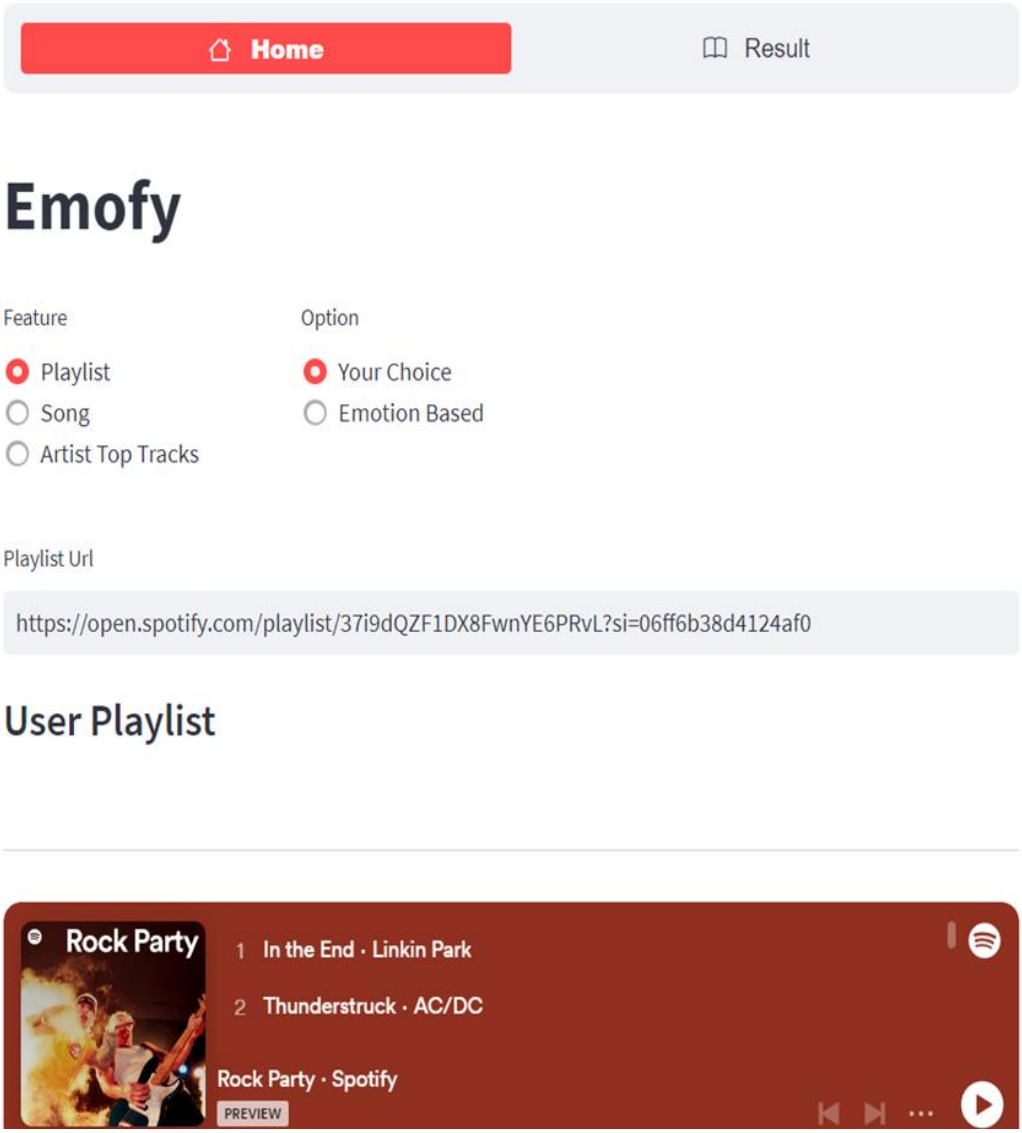
- **Fear:** The eyes widen and the upper lids rise. The brows draw together while the lips extend horizontally.

- **Happiness**: The corners of the lips lifted and shaped a smile, the eyelids tighten, the cheeks rise up and the outside corners of the brows pull down.

We primarily aim and focus on resolving the drawbacks involved in the existing system by designing an automated emotion-based music player for the playing song based on user extracted facial features and thus avoiding the employment of any additional hardware.

CHAPTER V

SCREENSHOTS



Home | Result

Emofy

Feature

- Playlist
- Song
- Artist Top Tracks

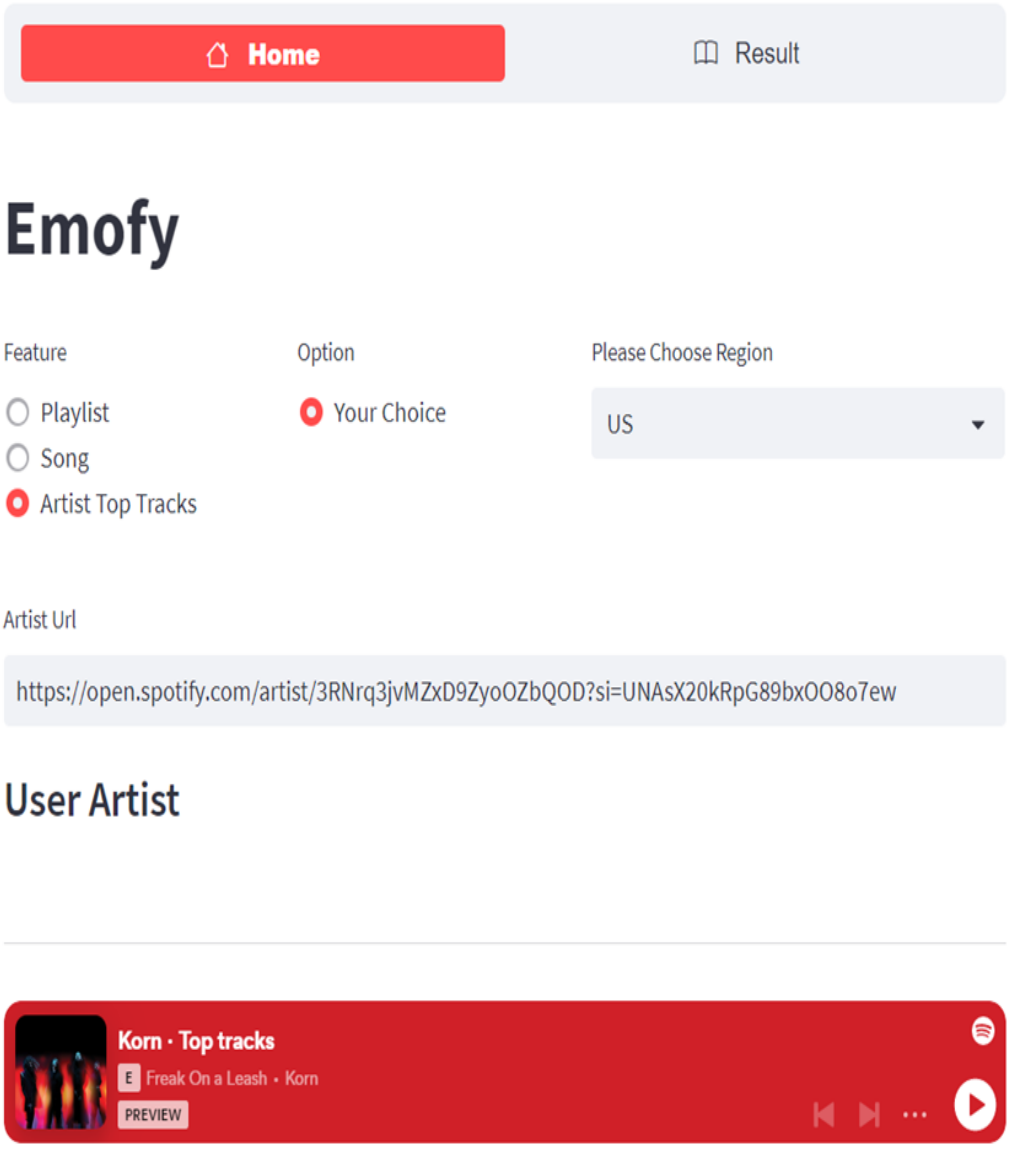
Option

- Your Choice
- Emotion Based

Song Url

https://open.spotify.com/track/5CQ30WqJwcep0pYcV4AMNc?si=ed4b04f153a24531

User Song


Home
Result
Emofy
Feature
Playlist
Song
Artist Top Tracks
Option
Your Choice
Please Choose Region
US
Artist Url
https://open.spotify.com/artist/3RNrq3jvMZxD9ZyoOZbQOD?si=UNAsX20kRpG89bxOO8o7ew
User Artist
Korn · Top tracks
E
Freak On a Leash · Korn
PREVIEW
Get Recommendations

Home | Result

Top 10 recommendations

Freak On a Leash
Korn
PREVIEW E

Coming Undone
Korn
PREVIEW

Falling Away from Me
Korn
PREVIEW E

Blind - Live at CBGB
Korn
PREVIEW E

Got the Life
Korn
PREVIEW E

Twisted Transistor

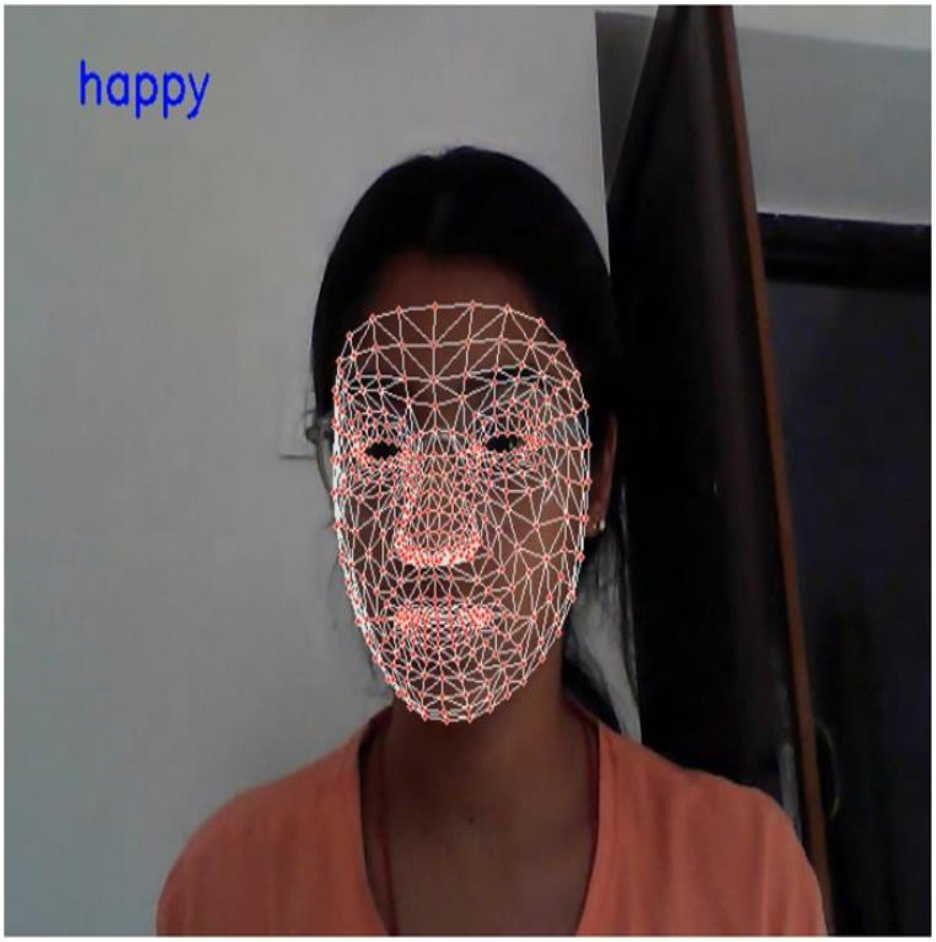
Emofy

Feature

- (•) Playlist
- ( ) Song
- ( ) Artist Top Tracks

Option

- ( ) Your Choice
- (•) Emotion Based



STOP

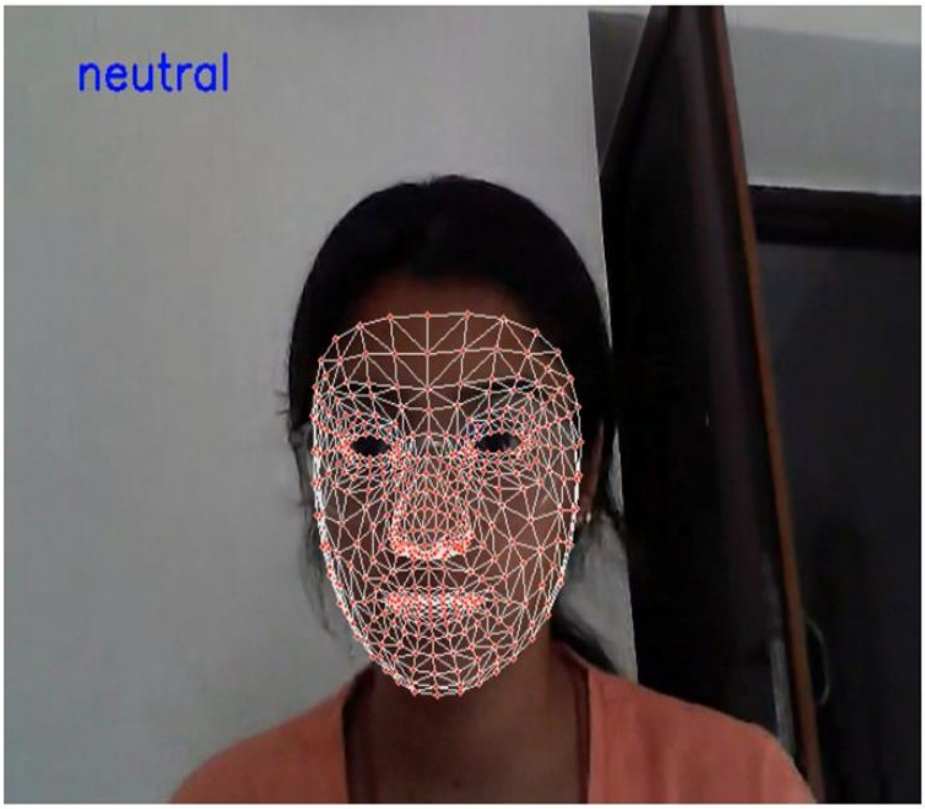
Emofy

Feature

- Playlist
- Song
- Artist Top Tracks

Option

- Your Choice
- Emotion Based



STOP

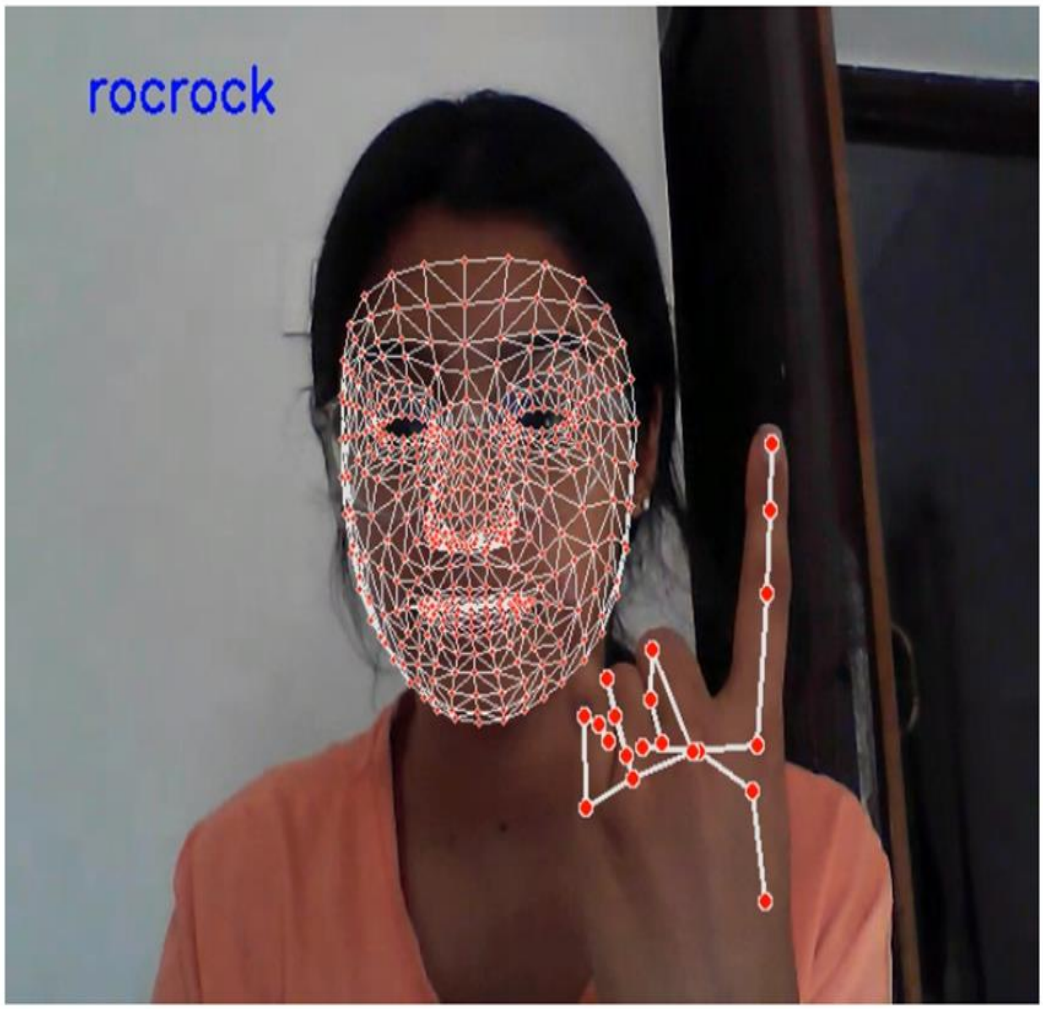
Emofy

Feature

- (•) Playlist
- ( ) Song
- ( ) Artist Top Tracks

Option

- ( ) Your Choice
- (•) Emotion Based



STOP

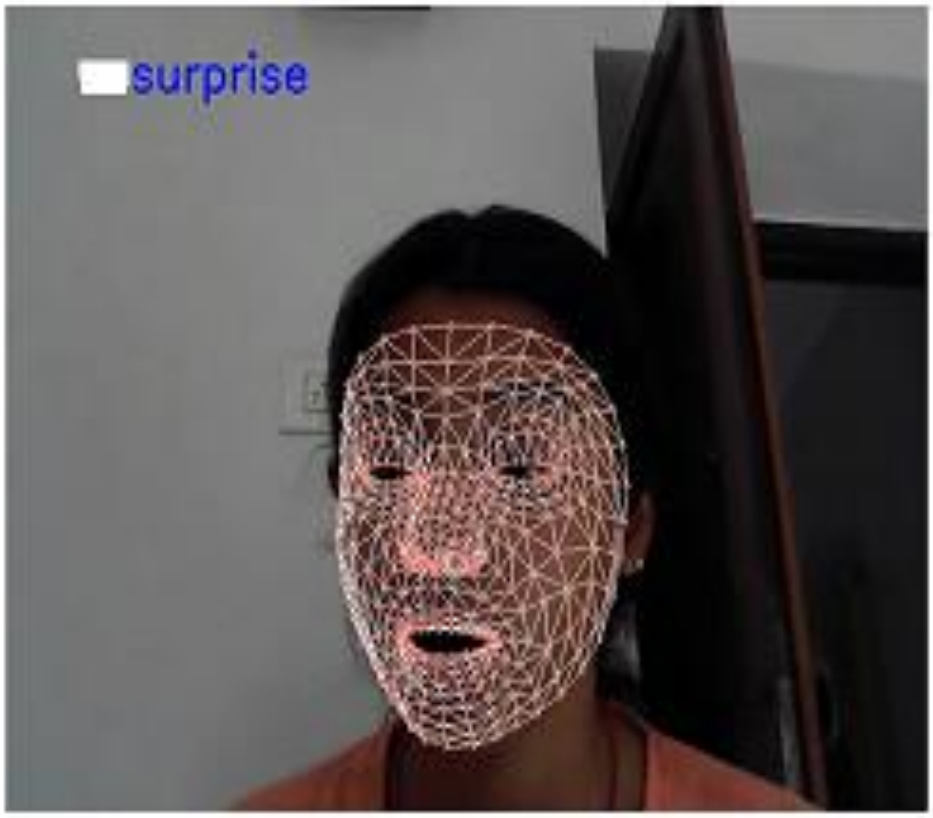
Emofy

Feature

- Playlist
- Song
- Artist Top Tracks

Option

- Your Choice
- Emotion Based



STOP

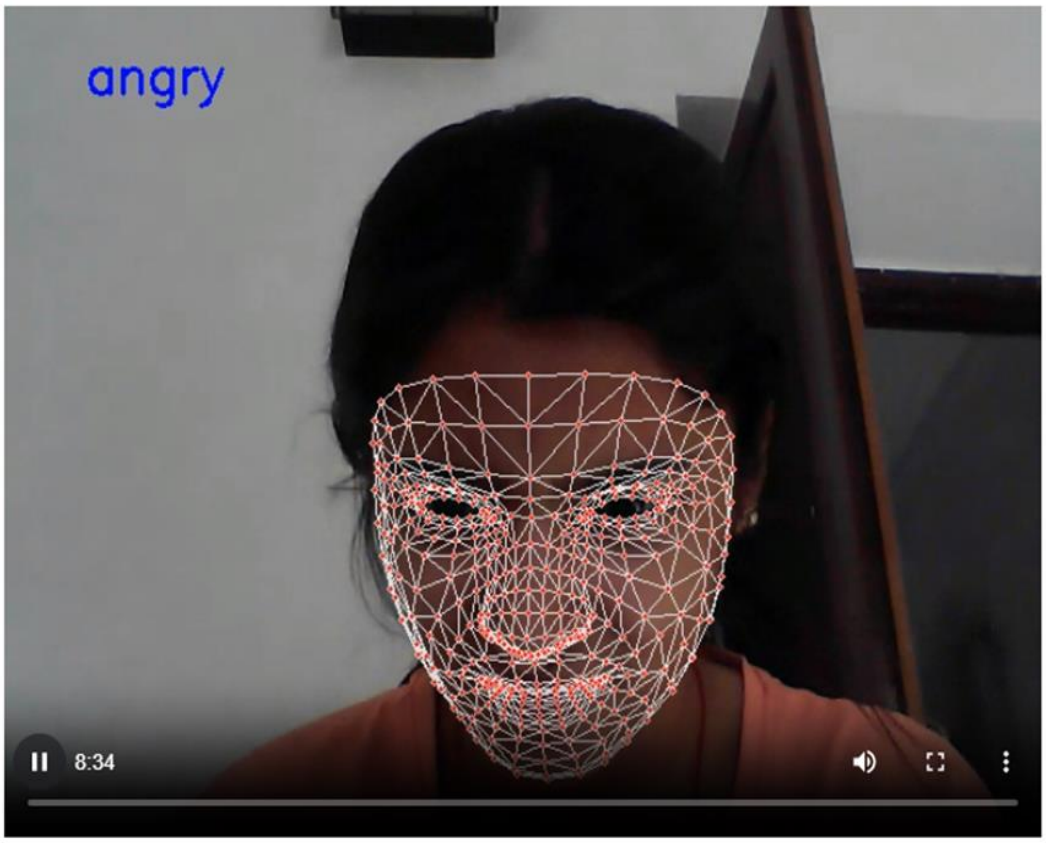
Emofy

Feature

- (•) Playlist
- ( ) Song
- ( ) Artist Top Tracks

Option

- ( ) Your Choice
- (•) Emotion Based



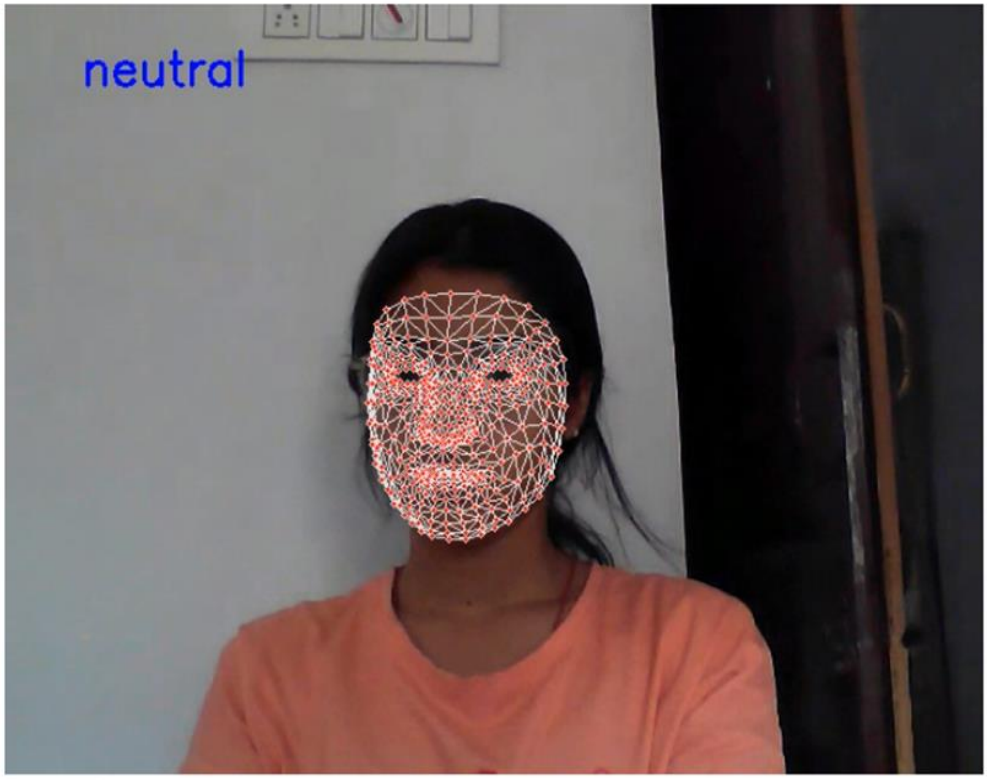
STOP

Feature

- (●) Playlist
- ( ) Song
- ( ) Artist Top Tracks

Option

- ( ) Your Choice
- (●) Emotion Based



STOP

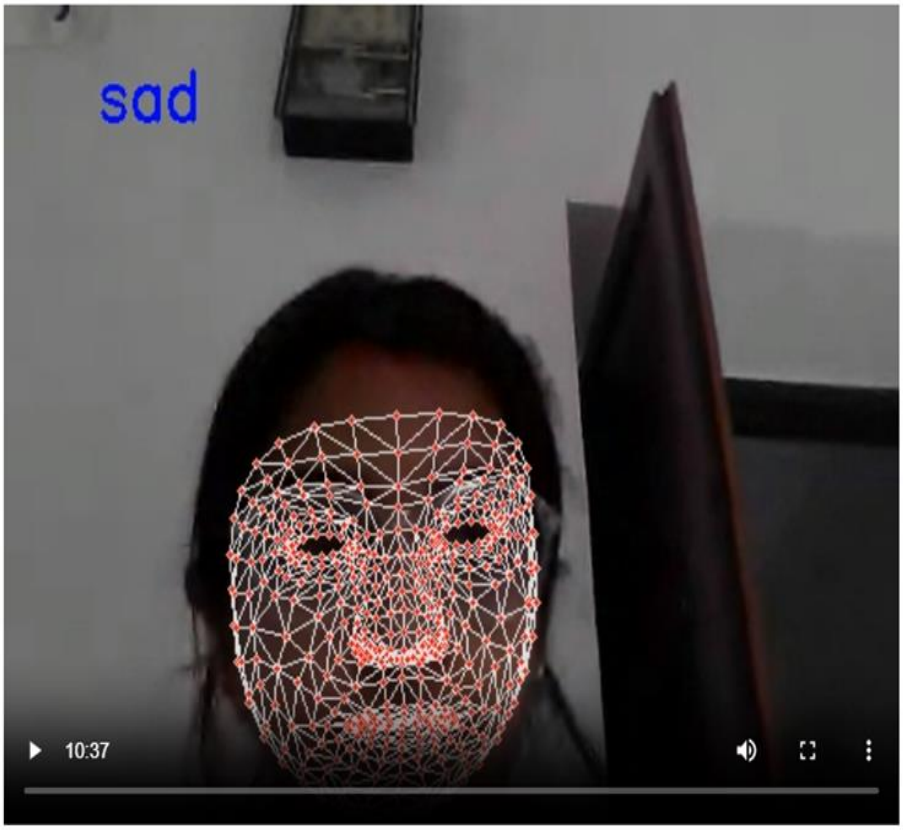
Recommend me songs

Feature

- (•) Playlist
- ( ) Song
- ( ) Artist Top Tracks

Option

- ( ) Your Choice
- (•) Emotion Based



STOP

Recommend me songs

Go to the Result page to view the Song recommendations

Home | **Result**

Top 5 recommendations

Chop Suey!
System Of A Down
PREVIEW

Killing In The Name
Rage Against The Machine
PREVIEW E

Dani California
Red Hot Chili Peppers
PREVIEW

Duality
Slipknot
PREVIEW

Uprising
Muse
PREVIEW

Home | Result

Top 5 recommendations

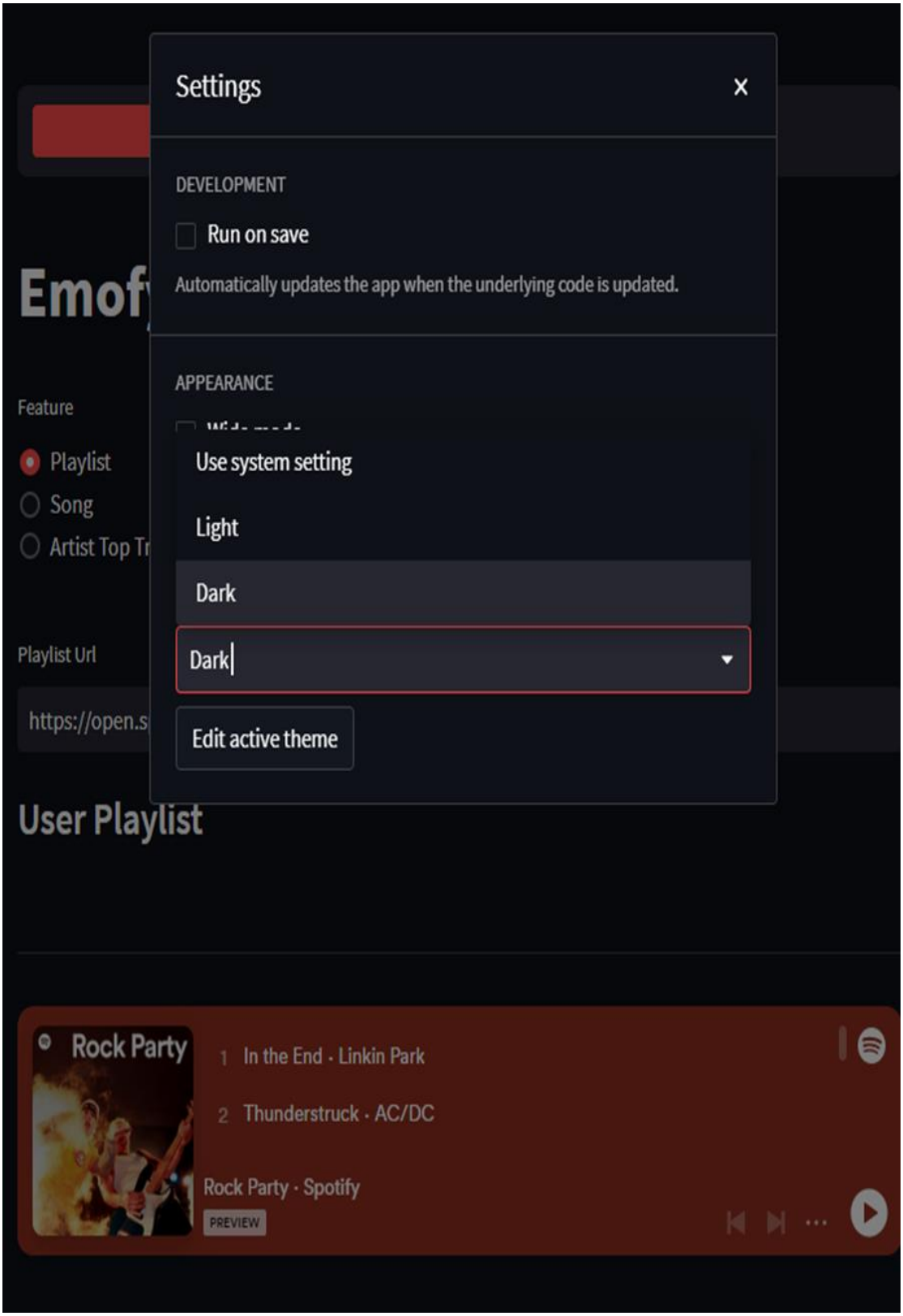
Pumped Up Kicks
Foster The People
PREVIEW

Africa
TOTO
PREVIEW

Take on Me
a-ha
PREVIEW

Highway to Hell
AC/DC
PREVIEW

Here Comes The Sun - Remastered 2009
The Beatles
PREVIEW


Settings
DEVELOPMENT
Run on save
Automatically updates the app when the underlying code is updated.
APPEARANCE
Use system setting
Light
Dark
Dark
Edit active theme
Feature
Playlist
Song
Playlist Url
User Playlist
Rock Party
1 In the End · Linkin Park
2 Thunderstruck · AC/DC
Rock Party · Spotify
PREVIEW

CHAPTER VI

TESTING

6.1 SYSTEM TESTING RESULTS

The user carried out system testing once the completion of the system development. The purpose of this testing is to check the functionalities system, whether if it is usable and well functioned. The results from the functional testing can be seen in the table below.

Table 6.1: System functional testing results

COMPONENT	EXPECTED FUNCTION	TESTING RESULT	
		POSITIVE	NEGATIVE
"TAKE A PICTURE" Button	To capture image & fetch image analysis results.	✓	
''RETAKE PICTURE" Button	To navigate user back to previous screen to recapture image.	✓	
"SETTING" Button	Enable user to set server URL	✓	
"PLAY MUSIC" Button	To open window to play suggested music.	✓	
"BACK " Button	To take back to home screen.	✓	

6.2 EMOTION ACCURACY TESTING RESULTS

Set of images for each emotion (normal, sad, surprise and happy) are saved in the proposed model for the comparison purposes. The newly load images will be compared with the saved dataset in order to detect the emotion of the users. Table below showed the set of images that saved in the proposed model.

Table 6.2: Testing result for various images

Image	Expected Output	Detected Emotion	Test Status
	Happy	Happy	Pass
	Neutral	Neutral	Pass
	Sad	Sad	Pass

	Angry	Sad	Pass
	No Face is Detected	No Face is Detected	Pass
	Fear	Sad	Fail
	Surprise	Surprise	Pass

CHAPTER – VII

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION

The significance of this project is the emotion detection of the images loaded into the proposed model. The main purpose is on its emotion detection functionality. Through the integration between emotion detection technology and music player, the proposed model is aimed to provide betterment in the individual's entertainment. The proposed application is able to detect the emotions i.e., anger, fear, happy, surprised, neutral, sad and disgust. Once the proposed model detected the emotion, music player will play the song(s) accordingly. As for the usability and accuracy, both system testing and emotion accuracy testing has been done to the proposed model and return a satisfying result. The proposed model is an application which can works well in all mobile devices. Thus, with this Emotion Based Music Player, users can have an alternative way of selecting songs, which is in a more interactive and simpler way. It can help the music lovers to search and play songs according to their emotions automatically.

Even though human emotions are complex and subtle, it is possible for a machine learning model to be trained to accurately detect a set of emotions which can be differentiated from each other with certain facial expressions. The expression on a person's face can be used to detect their mood, and once a certain mood has been detected, music suitable for the person's detected mood can be suggested. Our model, having the accuracy of approximately 85%, is able to detect seven moods accurately: anger, disgust, fear, happy, sad, surprise and neutral; and our android application is able to play the music that would be suitable for the detected mood. For accurate detection of fear and disgust moods, additional parameters such as heart rate or body temperature must also be considered rather than solely depending on facial expressions. In addition to that, finding suitable music to be played on detection of fear or disgust mood is also a challenge. As a result, it can be considered as a future

scope for our project. Our trained model is an overfit model, which can sometimes lead to fluctuations in accurate detection.

For example, the “disgust” mood is mostly classified as “angry” mood since the facial features (eyebrows, cheeks) are similar for both. Thus, for more accurate results it needs to be trained for more images, and for a greater number of epochs. Recommendation of movies and TV series on the basis of mood detection can also be considered as a future scope for our project.

7.2 FUTURE ENHANCEMENTS

Enhancements are the perquisite for development of a system. Every existing system has proposed enhancements which make it better and easier to use and more secure. The enhancements that have been proposed for this system are listed here:

- the future scope of the system would be to design a mechanism that would be helpful in music therapy treatment and help the music therapist to treat the patients suffering from disorders like mental stress, depression, trauma, and anxiety.
- A more compact device can be designed.
- A more accurate playlist can be generated.
- Voice/Facial recognition can be made more efficient.
- Humans tend to link the music they listen to; to the emotion, they are feeling. The song playlists though are, at times, too large to sort out 35 automatically. It can be a great relief if the music player was “smart enough” to sort out the music based on the current state of emotion the person is feeling. The project sets out to use various techniques for an emotion recognition system, analysing the impacts of different techniques used.
- We can develop parallel software that can be used anywhere with the help of providing the functionality of playing music according to the emotion detected. Developing a recommendation system could assist a user to decide which music one should listen to, helping the user to reduce his/her stress levels.
- Playing songs automatically, optimizing the algorithm by including additional features which help the system to categorize users based on many other factors

like location and suggesting the user travel to that location and play songs accordingly.

- There are a lot of limitations; creating a custom emotion recognition system that can be merged into the current application improves the functionality and performance of the system. Making the application run without needing an internet connection.

APPENDIX

CODE FOR EMOTION DETECTION.

```
import numpy as np
import argparse
import matplotlib.pyplot as plt
import cv2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

# command line argument
ap = argparse.ArgumentParser()
ap.add_argument("--mode",help="train/display")
mode = ap.parse_args().mode

# plots accuracy and loss curves
def plot_model_history(model_history):
    """
    Plot Accuracy and Loss curves given the model_history
    """
    fig, axs = plt.subplots(1,2,figsize=(15,5))
    # summarize history for accuracy

axs[0].plot(range(1,len(model_history.history['accuracy'])+1),model_history.his
tory['accuracy'])
```

```
axs[0].plot(range(1,len(model_history.history['val_accuracy'])+1),model_histor
y.history['val_accuracy'])
    axs[0].set_title('Model Accuracy')
    axs[0].set_ylabel('Accuracy')
    axs[0].set_xlabel('Epoch')

axs[0].set_xticks(np.arange(1,len(model_history.history['accuracy'])+1),len(mo
del_history.history['accuracy'])/10)
    axs[0].legend(['train', 'val'], loc='best')
    # summarize history for loss

axs[1].plot(range(1,len(model_history.history['loss'])+1),model_history.history['
loss'])

axs[1].plot(range(1,len(model_history.history['val_loss'])+1),model_history.hist
ory['val_loss'])
    axs[1].set_title('Model Loss')
    axs[1].set_ylabel('Loss')
    axs[1].set_xlabel('Epoch')

axs[1].set_xticks(np.arange(1,len(model_history.history['loss'])+1),len(model_
history.history['loss'])/10)
    axs[1].legend(['train', 'val'], loc='best')
    fig.savefig('plot.png')
    plt.show()

# Define data generators
train_dir = 'data/train'
val_dir = 'data/test'

num_train = 28709
num_val = 7178
batch_size = 64
```

```
num_epoch = 50

train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(48,48),
        batch_size=batch_size,
        color_mode="grayscale",
        class_mode='categorical')

validation_generator = val_datagen.flow_from_directory(
        val_dir,
        target_size=(48,48),
        batch_size=batch_size,
        color_mode="grayscale",
        class_mode='categorical')

# Create the model
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu',
input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
```

```
model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))

# If you want to train the same model or try other models, go for this
if mode == "train":
    model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001,
decay=1e-6),metrics=['accuracy'])
    model_info = model.fit_generator(
            train_generator,
            steps_per_epoch=num_train // batch_size,
            epochs=num_epoch,
            validation_data=validation_generator,
            validation_steps=num_val // batch_size)
    plot_model_history(model_info)
    model.save_weights('model.h5')

# emotions will be displayed on your face from the webcam feed
elif mode == "display":
    model.load_weights('model.h5')

    # prevents openCL usage and unnecessary logging messages
    cv2.ocl.setUseOpenCL(False)

    # dictionary which assigns each label an emotion (alphabetical order)
    emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4:
"Neutral", 5: "Sad", 6: "Surprised"}

    # start the webcam feed
    cap = cv2.VideoCapture(0)
    while True:
        # Find haar cascade to draw bounding box around face
        ret, frame = cap.read()
```

```
        if not ret:
            break
        facecasc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = facecasc.detectMultiScale(gray,scaleFactor=1.3,
minNeighbors=5)

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)
            roi_gray = gray[y:y + h, x:x + w]
            cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray,
(48, 48)), -1), 0)
            prediction = model.predict(cropped_img)
            maxindex = int(np.argmax(prediction))
            cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

        cv2.imshow('Video', cv2.resize(frame,(1600,960),interpolation =
cv2.INTER_CUBIC))
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```