VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi – 590018, Karnataka State, India

PROJECT ENTITLED

"A Classifier For Facial Emotion Recognition (FER) which Suggests Music Based on the Emotion Detected "

Submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

For the academic year 2018-2019

Submitted by:

RAHUL SAILESH WADHWA (1MV16CS074)

V BINU ALDRIN (1MV16CS113)

Project Carried out at

Sir M. Visvesvaraya Institute of Technology

Bengaluru - 562157

Under Guidance of

Mrs. Mayuri K P

Assistant Professor

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

HUNASAMARANAHALLI BENGALURU – 56215

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

Krishnadevaraya Nagar, International Airport Road,
Hunasmaranahalli, Bengaluru – 562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

It is certified that the project work entitled " **A Classifier For Facial Emotion Recognition (FER) which Suggests Music Based on the Emotion Detected** " is carried out by **RAHUL SAILESH WADHWA (1MV16CS074), V BINU ALDRIN(1MV16CS113)** bonafide students of **Sir M Visvesvaraya Institute of Technology** in partial fulfilment for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the year **2019-2020**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

Name & Signature of Guide	Name & Signature of HOD	Name & Signature of Principal
Mrs. Mayuri KP	**Dr. G. C. Bhanu Prakash**	**Dr. V.R. Manjunath**
Asst. Prof & Internal Guide	HOD, Dept. Of CSE,	Principal,
Dept. Of CSE, Sir MVIT	Sir MVIT	Sir MVIT
Bengaluru - 562157	Bengaluru - 562157	Bengaluru - 562157

External Examination:

Name of Examiner | Signature with Date

1)

2)

DECLARATION

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru

Date:

Signature of Students:

RAHUL SAILESH WADHWA	V BINU ALDRIN
(1MV16CS074)	(1MV16CS113)

ACKNOWLEDGMENT

It gives us immense pleasure to express our sincere gratitude to the management of **Sir M. Visvesvaraya Institute of Technology,** Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide **Mrs. Mayuri KP,** Assistant Professor, Dept. of CSE, for her invaluable help and guidance.

Heartfelt and sincere thanks to **Dr. G. C. Bhanu Prakash,** HOD, Dept. of CSE, for his suggestions, constant support and encouragement.

We would also like to convey our regards to **Dr. V.R. Manjunath,** Principal, Sir. MVIT for providing us with the infrastructure and facilities needed to develop our project.

We would also like to thank the staff of Department of Computer Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

RAHUL SAILESH WADHWA	1MV16CS074
V BINU ALDRIN	1MV16CS113

ABSTRACT

The human face is an important part of an individual's body and it especially plays an important role in knowing an individual's mood. Extracting the required input from the human face can now be done directly using a camera. This input can then be used in many ways. One of the applications of this input can be for extracting the information to deduce the mood of an individual. This data can then be used to get a list of songs that comply with the mood derived from the input provided earlier. This eliminates the time-consuming and tedious task of manually Segregating or grouping songs into different lists and helps in generating an appropriate playlist based on an individual's emotional features. Various algorithms have been developed and proposed for automating the playlist generation process. Facial Expression Based Music Player aims at scanning and interpreting the data and accordingly creating a playlist based the parameters provided. The scanning and interpreting includes audio feature extraction and classification to get a list of songs belonging to a similar genre or to get a list of similar sounding songs. Human emotions are meant for mutual understanding and sharing feelings and intentions. The emotions are manifested in verbal and facial expressions.

Face detection has been around for ages. Taking a step forward, human emotion displayed by face and felt by brain, captured in either video, electric signal (EEG) or image form can be approximated. Human emotion detection is the need of the hour so that modern artificial intelligent systems can emulate and gauge reactions from face. This can be helpful to make informed decisions be it regarding identification of intent, promotion of offers or security related threats. Recognizing emotions from images or video is a trivial task for human eye, but proves to be very challenging for machines and requires many image processing techniques for feature extraction. Several machine learning algorithms are suitable for this job. Any detection or recognition by machine learning requires training algorithm and then testing them on a suitable dataset.

People of today’s world are very indecisive even when it comes to the simplest of things like music. This project’s purpose is to design a classifier that could be used to eliminate this confusion exhibited by people.

This project is about building a music recommendation system for users. Such a system can not only be used to brighten up one's mood on a rainy weekend; especially in hospitals, other medical clinics, or public locations such as restaurants, this classifier could be used to spread positive mood among people.

The project has to be implemented in 2 phases-facial emotion recognition and then suggestion of music based on the detected emotion in the previous phase.The emotional reaction to music is different for every person, so analysing it will not likely yield perfect results. The method used then is to decide upon certain base songs that very closely embody a certain mood, and to match songs to these specific categories.

We can also study the efficacy of music stimuli to modify (reduce rather than induce) naturally occurring negative mood states (like sad, depressed, etc.) and to gradually bring about a change in this negative mood state turning it into a positive one. It could also simply be used just to prevent certain songs from having a negative impact on a person’s mood.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1
INTRODUCTION

Human emotion detection is implemented in many areas requiring additional security or information about the person. It can be seen as a second step to face detection where we may be required to set up a second layer of security, where along with the face, the emotion is also detected. This can be useful to verify that the person standing in front of the camera is not just a 2-dimensional representation.

Music is present in all cultures, providing a construct through which mood can be expressed. The ability of music to evoke mood makes it is a potentially powerful resource for mood modulation.

Human emotions can be classified as: fear, contempt, disgust, anger , surprise, sad, happy, and neutral. These emotions are very subtle. Facial muscle contortions are very minimal and detecting these differences can be very challenging as even a small difference results in different expressions.

Recognition of facial expressions is used to identify the basic human emotions. Facial expressions give important rules about emotions. Computer systems based on affective interaction could play an important role in the next generation of computer vision systems. Face emotion can be used in areas of security, entertainment and human machine interface (HMI). A human can express his/her emotion through lip and eye. Generally people have a large number of songs in their database or playlists. Thus to avoid trouble of selecting a song, most people will just randomly select a song from their playlist and some of the songs may not be appropriate for the current mood of the user and it may disappoint the user. As a result, some of the songs are not matching to the user's current emotion. Moreover, there is no commonly used application which is able to play songs based on the current emotions of the user. Music plays a very important role in enhancing an individual's life as it is an important medium of entertainment for music lovers and listeners and sometimes even imparts a therapeutic approach. In today's world, with ever increasing advancements in the field of multimedia and technology, various music players have been developed with features like fast forward, reverse, variable playback Although these features satisfy the user's basic requirements, yet the user has to face the task of manually browsing through the playlist of songs and select songs based on his current mood and behaviour.

The main objective of this project is to design an efficient and accurate classifier that would generate a playlist based on current emotional state and behaviour of the user. Face detection and facial feature extraction from image is the first step in emotion based music player. For the face detection to work effectively, we need to provide an input feed. We have used algorithm that is used for face detection and facial feature extraction. We have generated landmarks points for facial features. The next step is the classification of emotion for which we have used a pre-trained HAAR classifier. The generated landmarks points are provided to the HAAR classifier for training purpose. The emotion classified by classifier is then passed to music player and accordingly music will be played.

CHAPTER 2

LITERATURE SURVEY

1. **"Facial emotion detection using modified eyemap-mouthmap algorithm on an enhanced image and classification with tensorflow"**– springer.com, 2019

   The main components are pre-processing, face detection, mouth detection, and eye detection. Pre-processing- The input image is first pre-processed in order to enhance the clarity of the image. The input RGB image is first converted to HSV image.
   Face detection- The enhanced image is used for face detection. Algorithm: Viola Jones algorithm. Modified mouth and eye map: The cropped image is then converted to the YCbCr color space. The different planes are then separated individually as Y , Cb, and Cr. The separated plane is used to find the mouthmap. Eye detection is also performed using Viola Jones algo-rithm.
   Algorithm: MoMMEM algorithm for mouth and eye mapping and Viola Jones for face detection.
   The proposed algorithm is optimal in detecting the geometry of the face which in turn can be used for identifying the emotion with good accuracy. The analysis reveals that proper machine learning algorithm is also required to identify the emotions.
   Karolinska Directed Emotional Faces (KDEF) database- the proposed method showed an accuracy of 98.1%.

2. **Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition** – Elsevier.com, 2019.

   In the pre-processing stage, facial regions are detected by the Viola-Jones faces detector. The proposed DAM-CNN model consists of three modules, i.e., the VGG-Face network for extracting features, SERD for refining CNN features and highlighting salient expressional regions, and MPVS-Net for generating a high-level representation robust to multiple variations. SERD is forced to discriminate features that are not only related to expressions but also effective to the final classification task. MPVS-Net is constructed by a three-layer network, where the first layer is based on a Multi-Path Auto Encoder (MPAE) and the following are two fully connected layers. In our method, we replace all fully-connected layers of VGG-SERD with the MPVS-Net. we train DAM-CNN by two steps. In the first step, we fine tune the VGG-SERD with expressional images. In the second step, we introduce the MPVS-Net into VGG-SERD to construct DAM-CNN. In these two training stages, dropout is applied to the fully-connected layers as well as the MPAE to enhance the generalization ability of the model. Algorithm: Viola Jones face detector for face recognition.

The proposed DAM-CNN can still be improved in some aspects. In addition, DAM-CNN is a generic model for classification. It can be extended to other recognition task such as face recognition, which can be one of our future work to further investigate the effectiveness of DAM-CNN.

DAM-CNN can obtain an averaged recognition accuracy of 93.65% and 93.20% in the task of six class and seven class classification respectively when using unconstrained datasets.

3. **An Integrated approach to emotion recognition and gender classification** - Elsevier, 2019

Geometric feature based approach is employed which calculates the distance between detected facial features. The viola Jones algorithm is used to detect different facial regions and those are extracted from the frontal face. Yoav Freund and Robert Schapire proposed an AdaBoost algorithm which is used for the classification of gender. In this work, the emotions for which MSER features are extracted and classified using SVM (support vector machine).SVM classifier is used to classify the emotions and AdaBoost for gender classification.

This study analyses effective implementation of bi-modal emotion recognition system with its strengths and the limitations of the system based on facial expressions and speech features. This paper focused on integrating a decision based fusion approach with multi-class classification using Support Vector Machines (SVM).

Emotion Recognition from facial Expression: The classifier’s overall performance is 73.01%. The emotions with highest recognition rates are Surprise and Sad (74%).Emotion recognition from speech features. The classifier overall performance is 77%. The emotions with highest recognition rates are Happiness and Disgust (78%).

4. **Hybrid deep neural networks for face emotion recognition -** Elsevier.com, 2018

For the preprocessing, represent fluctuating lighting conditions (specifically, crosswise over datasets). Utilized the adjusted appearances gave by the coordinators to remove highlights from the CNN. By processing one change for each dataset the nose, eyes, and mouth are generally in a similar area holding a slight measure of variety.

For the processing, when the image passed to the network, 200 dimensional vectors will be extracted from the fully connected layers. For the assumed time t, take P frames from the past (i.e. [t − P, t]). Then passes every frame from time t − P to t to the CNN and extract P vectors fully for each image. Each and every vector goes through a node of the RNN model. Then every node of the RNN returns some results of valence label. The mean squared error has been used for the cost function while optimizing.

Algorithm: Momentum and Adam (deep learning initializer algorithms) for parameter updates.

A hybrid deep CNN and RNN model was proposed. It has been found that the combination of the two types of neural networks (CNN-RNN) cloud significantly improve the overall result of detection, which verified the efficiency of the proposed model.

Based on the experiments, the best results obtained by using 6 hidden layers. Hybrid CNN-RNN model could achieve an accuracy of 94.72%, while a single CNN can reach only to 71.42%.

5. **Facial Emotion Recognition using Machine Learning-** SJSU scholar works, 2018

Tools and Libraries used: OpenCV, Dlib, Python, Scikit-learn, Jupyter Notebook Database- CK+ and Radbound Faces Database (RaFD). Support Vector Machines to predict the emotions. Sklearn machine library was used to implement the Support Vector Machines (SVM) and Logistic Regression algorithms. The multiclass strategy used was One-Vs-Rest for all the algorithms. Logistic regression algorithm was fine tuned for penalty.The linear kernel was fine tuned to rbf and poly to see the variation in results. Cross-validation technique was used along with SVM to remove any biases in the databases.

Main focus was on feature extraction and analysis of the machine algorithm on the dataset. Algorithms like logistic regression, linear discriminant analysis and random forest classifier can be fine-tuned to achieve good accuracy and results.Feature extraction accuracy for CK+ database- 89%.The results of the emotion detection algorithm gave average accuracy up to 86% for RaFD database and 87% for CK+ database

6. **An Emotion Recognition Based on Facial Recognition in Virtual Learning Environment –** Elsevier, 2018.

Haar Cascades method is used to identify whether a face exists in the image frames.If the face exists, eyes and mouth need to be located and eye and mouth regions need to be cropped.Filter and edge detection is carried out using Sobel edge detection method, followed by feature extraction. We train the feature extraction using the neural network method.

This paper provided a proposed model to solve the problems of emotion recognition based on facial recognition in virtual learning environments, and the efficiency and accuracy are considered at the same time. It can be applied to real distance education.

Proposed Solution: Sad (78.54%), Surprise (93.26%), Happy (95.25%), Anger (91.22%), Disgust (84.32%), Fear (82.58). Current Solution: Sad (76%), Surprise (87.72%), Happy (94%), Anger (87.66%), Disgust (82.76%), Fear (79.73

7. **Facial Expression Recognition Using Enhanced Deep 3D Convolution Neural Networks** – IEEE conference on CVPR,2017

A 3D Inception-ResNet architecture extracts both spatial and temporal features of the sequences in an end-to-end neural network. Another component is incorporating facial landmarks in an automated manner during training in the proposed neural network. The final part of our proposed method is an LSTM unit which takes the enhanced feature map resulted from the 3D Inception-ResNet (3DIR) layer as an input and extracts the temporal information from it. The LSTM unit is followed by a fully-connected layer associated with a soft-max activation function.

Facial Landmarks-to extract the facial landmarks, OpenCV face recognition is used to obtain bounding boxes of the faces. We initially resize all of the images in the sequence to their corresponding filter size in the network. Afterwards, we assign weights to all of the pixels in a frame. The closer a pixel is to a facial landmark, the greater weight is assigned to that pixel. Algorithm: Manhattan distance with a linear weight function was used.

Experiments show that the proposed method outperforms many of the state-of-the-art methods and provides a general solution for the task of FER.

The recognition rates obtained for subject independent tasks were as follows-CK+ was 93.21±2.32 MMI was 77.50±1.76 FERA was 77.42±3.67 DISFA was 58.00±5.77 Whereas those for cross-database tasks were- CK+-67.52%MMI-54.76% FERA-41.93% DISFA-40.51%

8. **Real-time Convolutional Neural Networks for Emotion and Gender Classification** - arxiv.org, 2017

First, the accurate position of dense facial landmarks is located with face alignment method, and the typical active appearance model (AAM) is adopted in this paper. Based on the characteristic of facial expressions, 51 landmarks of the inner face are chosen because of their high accuracy and reliability. Then, the descriptors from the patches centred at landmarks are extracted and the high-dimensional feature is formally composed by concatenating all descriptors. We set the size of patches to a relative small value so as to describe the local features more accurately. A linear dimension reduction method called as the principal component analysis (PCA) approach is utilized. The feature compressed by PCA is utilized as the input data of the deep sparse auto-encoders.

This proposed a novel approach for facial expression recognition using deep sparse autoencoders (DSAE), which can automatically distinguish the expressions with high accuracy. Both the facial geometric and appearance features have been introduced to compose a high dimensional feature with accurate and comprehensive information of emotions. The generalization performances of the DSAE based on the HOG feature are 95.79% for 7-class recognition and 89.84% for 8-class recognition, respectively.

9. **DAGER: Deep Age, Gender and Emotion Recognition using Convolutional Neural Networks**- arxiv.org, 2017.

Slighthound software for Facial recognition, Age estimation and Gender and emotion recognition. In this paper, an end to end system for age, gender and emotion recognition is presented. Emotion Recognition accuracy on slighthound's dataset: Slighthound-76.1% Microsoft-61.3% Age Classification Accuracy: Slighthound-top1 (70.5%), 1-off (96.2%) Hou et al-top1 (65.0%), 1-off (96.1%) Rothe et al-top1 (62.3%), 1-off (94.3%) Dong et al-top1 (56.0%), 1-off (92.0%) Gallegher et al-top1 (42.9%), 1-off (78.1%)

10. **Dexpression: Deep convolutional neural network for expression recognition** - arxiv.org,2016.

Convolutional Neural Networks-The proposed deep Convolutional Neural Network architecture consists of four parts. The first part automatically pre-processes the data. This begins with Convolution 1, which applies 64 different filters. The next layer is Pooling 1, which down-samples the images and then they are normalized by LRN 1. The next steps are the two FeatEx (Parallel Feature Extraction Block) blocks. The features extracted by theses blocks are forwarded to a fully connected layer, which uses them to classify the input into the different emotions. They made use of 2 training datasets- CKP dataset and MMI dataset. An application built on DeXpression which is used in a real environment could benefit from distinguishing between more emotions such as Nervousness and Panic. An approach to enhance emotion recognition could be to allow for composed emotions.

For the CKP Dataset, the lowest accuracy is achieved by the emotion Surprise with 98.79% while Contempt/Sadness are both recognized with 100%. For the MMI Dataset, the lowest accuracy is achieved by the emotion Fear with 93.75%. Happiness is recognized with 98.21%.

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATON

A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software or any kind of application.

3.1 Development Requirements

3.1.1 Hardware Requirements

PROCESSOR	:	Intel i5-3230 or Higher
RAM	:	4GB
HARD DISK	:	16GB
CPU	:	4 x 2.60GHz
Webcam		

3.1.2 Software Requirements

OPERATING SYSTEM	:	Linux/Windows
SOFTWARE	:	Python 3.7 (IDE: Spyder)
PYTHON LIBRARIES	:	Pandas, numpy, scikit-learn, opencv, tkinter, mutagen, pygame.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

Currently, there are many existing music player applications. Some of the interesting applications among them are:

• Saavan and Spotify – These application gives good user accessibility features to play songs and recommends user with other songs of similar genre

• Moodfuse - In this application, user should manually enter mood and genre that wants to be heard and moodfuse recommends the songs-list.

• Steromood - User should select his mood manually by selecting the moods from the list and the application plays music from YouTube .

• Musicovery - This application has High quality songs and comprehensive music recommendations. It also suggest predefined play-list for the user.

4.2 Proposed System

The project has 2 main phases that have to be constructed and integrated. Each one deals with the one part of the problem.

Phase 1 consists of the following:

- Pre-Processing of Dataset (CK+)
- Face Detection
- Emotion Recognition

Phase 2 consists of the following:

- Formulation of the Dataset (Music)
- Suggestion of Music based on the recognized emotion.

4.2.1 Phase 1- Emotion Detection

- Obtain a face from the webcam and pre-process the image.
- Make use of haar cascade classifier to detect the face and pre-process the image.
- Detect the emotion from the pre-processed image.
- Send the detected emotion for classification of Music.

4.2.2 Phase 2-Suggestion of Music Based on Emotion

- Check the database and obtain list of songs based on the detected emotion.
- Generate a random shuffled playlist.
- Allow user to play music from the created playlist.

CHAPTER 5

SYSTEM DESIGN

5.1 System Architecture

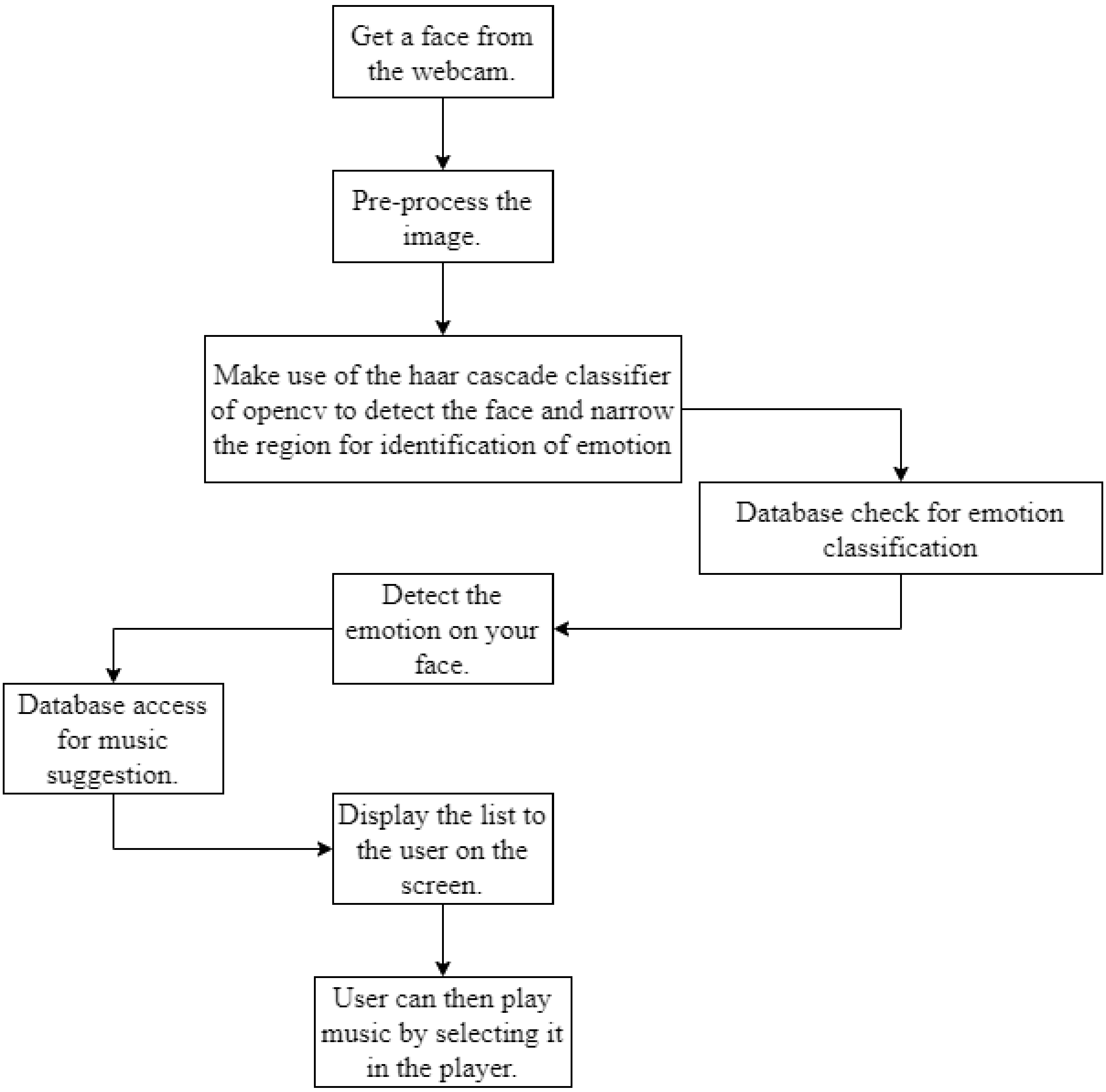


FIG. 5.1 – SYSTEM ARCHITECTURE OF EMOTION BASED CLASSIFIER.

5.2 Sequence Diagram

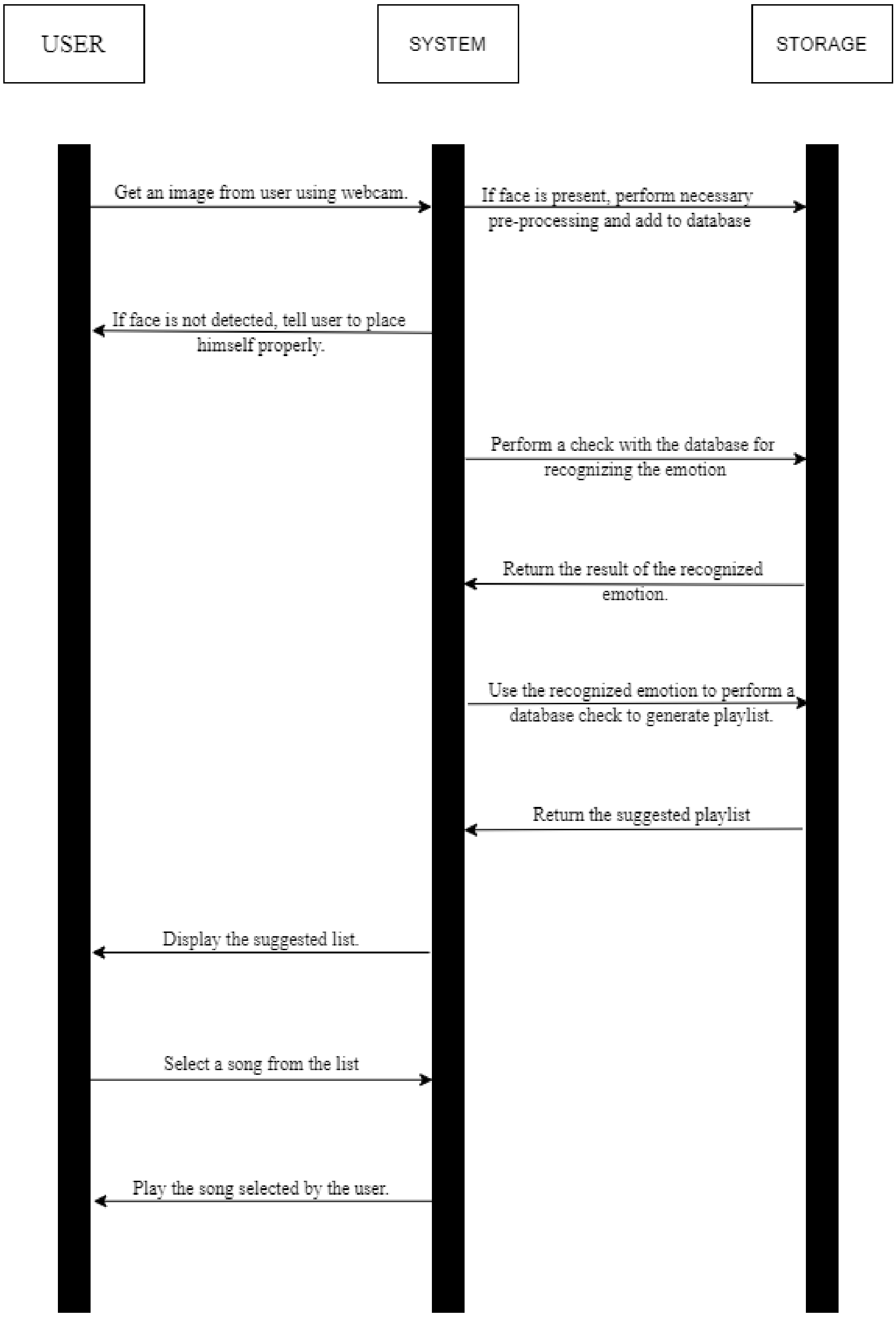


FIG. 5.2 – SEQUENCE DIAGRAM

CHAPTER 6
IMPLEMENTATION

6.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

-

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is also sometimes termed the off-side rule.

6.1.1 Features of Python

- **Easy-to-learn**: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read**: Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain**: Python's source code is fairly easy-to-maintain.
- **A broad standard library**: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode**: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable**: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable**: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases**: Python provides interfaces to all major commercial databases.
- **GUI Programming**: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable**: Python provides a better structure and support for large programs than shell scripting.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.

6.2 Python Libraries

- **Pandas:** Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.
- **Numpy:** NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

- **Scikit Learn**: Scikit-Learn, a package that provides efficient versions of a large number of common algorithms. Scikit-Learn is characterized by a clean, uniform, and streamlined API, as well as by very useful and complete online documentation. A benefit of this uniformity is that once you understand the basic use and syntax of Scikit-Learn for one type of model, switching to a new model or algorithm is very straightforward. Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. The library is focused on modeling data.
- **OpenCV**: OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day. Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations. OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.
- **Tkinter**: *Tkinter* is a Python binding to the Tk GUI toolkit. Tk is the original GUI library for the Tcl language. Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. There are several other popular Python GUI toolkits. Most popular are wxPython, PyQt, and PyGTK.
- **Mutagen:** Mutagen is a Python module to handle audio metadata. It supports ASF, FLAC, MP4, Monkey's Audio, MP3, Musepack, Ogg Opus, Ogg FLAC, Ogg Speex, Ogg Theora, Ogg Vorbis, True Audio, WavPack, OptimFROG, and AIFF audio files. All versions of ID3v2 are supported, and all standard ID3v2.4 frames are parsed. It can read Xing headers to accurately calculate the bitrate and length of MP3s. ID3 and APEv2 tags can be edited regardless of audio format. It can also manipulate Ogg streams on an individual packet/page level.
- **Pygame**: Game programming is very rewarding nowadays and it can also be used in advertising and as a teaching tool too. Game development includes mathematics, logic,

physics, AI and much more and it can be amazingly fun. In python, game programming is done in pygame and it is one of the best modules for doing so.

CHAPTER 7

SOFTWARE IMPLEMENTATION

The project has 2 main phases that have to be constructed and integrated. Each one deals with the one part of the problem.

Phase 1 consists of the following:

- Pre-Processing of Dataset (CK+)
- Face Detection
- Emotion Recognition

Phase 2 consists of the following:

- Formulation of the Dataset (Music)
- Suggestion of Music based on the recognized emotion.

7.1 PHASE 1

Phase 1 involves detection of the face and recognition of the emotion portrayed by the user.

7.1.1 Pre-Processing of Dataset

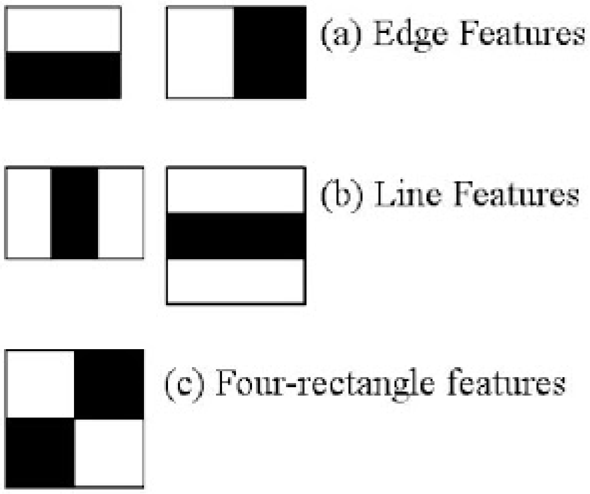
The CK+ dataset has to be unzipped and all the images for the various emotions depicted by the users has to be cleaned and any duplicate images for the same emotion have to be removed in-order to make the dataset more easily manageable. This pre-processing involves segregating the different images into a set of folders that represent the different emotions which is either done manually or by making use of an existing classifier. The haar cascade XML file is formed in this stage.

7.1.2 Collection of Image sample

The system needs image input that comprises of the different emotions that can be used to depict the different moods of the user. For that, the system has a predefined set of images that are previously classified by the CK+ dataset. So, the emotion that the system needs to classify, is done using the live input given by the user. On receiving the image sample, the system will pre-process it and store it into the various classified folders.

7.1.3 Face Detection

We will use a pre-trained HAAR classifier supplied with OpenCV. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.



First we need to load the required XML classifiers. Then load our input image (or video) in grayscale mode. We make use of 'haarcascade_frontalface_default.xml' in order to obtain a box around the face detected. Now we find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h). Once we get these locations, we can create a ROI for the face.

7.1.4 Emotion Detection

To detect the actual emotion on your face we use an Individual model rather than a generalized model. Models trained on a single individual work much better when used on the same individual, often because in that case there is less variance between the data (here :facial features). If we minimise the variance by keeping the face the same, most of the detected differences will be due to the fact that a different emotion is expressed. You need to collect varied images of yourself to make this robust. We make use of Fisherfaces algorithm for emotion classification.

Fisherfaces is a supervised classification method used for face recognition. It is used to classify samples of unknown classes based on training samples with labelled classes. The goal of Fisherfaces is to maximize between-class (across users) variance and minimize within-class (within user) variance. Here, C class is the face with mean of class j denoted by N j – and the ith image in class j denoted as j xi . These scatter matrix are sb and sw respectively and are calculated as follows:

$$S_b = \Sigma_{j=1}^{C} (\bar{N}_j - N)(\bar{N}_j - N)^T$$

$$S_w = \Sigma_{j=1}^{C} \Sigma_{i=1}^{Nj} (x_i^j - \bar{N}_j)(x_i^j - \bar{N}_j)^T$$

$$W_{Frisher-Descri\min at} = \arg\max \frac{|W^T\ S_b W|}{|W^T\ SwW|}$$

7.2 PHASE 2

Now that we have successfully detected the emotion on your face playing the right music is actually not that hard.

7.2.1 Formulation of Music Dataset

The music dataset has to be created and hosted on the system. This includes sorting the various songs that are .mp4 files and storing them in folders that will be assigned with the emotion labels. There is no limit on the number of songs that can be stored on the system. This stage involves getting together the entire music dataset and sorting them according to emotion in the folders as well as in a .csv file.

7.2.2 Suggestion of Music based on Recognized Emotion

Finally after the emotion has been recognized, a playlist has to be generated according to the emotion recognized by the classifier. This is done by a simple retrieval of data from the .csv file which already consists of the music that has been categorized into the various emotional states defined. Once this is done the user can select a song from the list displayed and the music can then be played easily. The user can also add or delete songs to the list as per his/her whims and fancies.

CHAPTER 8

SOURCE CODE

Update_Model.py music_player.py emotion_recognition.py

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Apr  4 12:57:11 2020

@author: rahul
"""

import cv2, numpy as np, argparse, time, glob, os, sys, subprocess, pandas, random, Update_Model, math

camnumber = 0
facedict = {}
actions = {}
actionlist=[]
emotions = ["angry", "happy", "sad", "neutral"]
df = pandas.read_excel("EmotionLinks.xlsx") #open Excel file
actions["angry"] = [x for x in df.angry.dropna()]
actions["happy"] = [x for x in df.happy.dropna()]
actions["sad"] = [x for x in df.sad.dropna()]
actions["neutral"] = [x for x in df.neutral.dropna()]
video_capture = cv2.VideoCapture(0)
facecascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
fishface = cv2.face.FisherFaceRecognizer_create()

def main():
    fishface.read("trained_emoclassifier.xml")
    actionlist.clear()
    video_capture.open(camnumber)
    run_detection()
    video_capture.release()
    return actionlist

def crop_face(clahe_image, face):
    for (x, y, w, h) in face:
        faceslice = clahe_image[y:y+h, x:x+w]
        faceslice = cv2.resize(faceslice, (350, 350))
    facedict["face%s" %(len(facedict)+1)] = faceslice
    return faceslice

def update_model(emotions):
    print("Model update mode active")
    check_folders(emotions)
    for i in range(0, len(emotions)):
```

FIG. 8.1 – EMOTION RECOGNITION.PY-PART 1

Update_Model.py music_player.py emotion_recognition.py

```python
    for i in range(0, len(emotions)):
        save_face(emotions[i])
    print("collected images, looking good! Now updating model...")
    Update_Model.update(emotions)
    print("Done!")

def check_folders(emotions):
    for x in emotions:
        if os.path.exists("dataset\\%s" %x):
            pass
        else:
            os.makedirs("dataset\\%s" %x)

def save_face(emotion):
    print("\n\nplease look " + emotion + ". Press enter when you're ready to have your pictures taken")
    input() #Wait until enter is pressed with the raw_input() method
    video_capture.open(camnumber)
    while len(facedict.keys()) < 16:
        detect_face()
    video_capture.release()
    for x in facedict.keys():
        cv2.imwrite("dataset\\%s\\%s.jpg" %(emotion, len(glob.glob("dataset\\%s\\*" %emotion))), facedict[x])
    facedict.clear()

def recognize_emotion():
    global actionlist
    predictions = []
    confidence = []
    for x in facedict.keys():
        pred, conf = fishface.predict(facedict[x])
        cv2.imwrite("images\\%s.jpg" %x, facedict[x])
        predictions.append(pred)
        confidence.append(conf)
    recognized_emotion = emotions[max(set(predictions), key=predictions.count)]
    actionlist = [x for x in actions[recognized_emotion]]#get list of actions/files for detected emotion
    random.shuffle(actionlist)#Randomly shuffle the list
    return actionlist

def grab_webcamframe():
    ret, frame = video_capture.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
```

FIG 8.2 – EMOTION RECOGNITION.PY-PART 2

```python
    clahe_image = clahe.apply(gray)
    return clahe_image

def detect_face():
    clahe_image = grab_webcamframe()
    face = facecascade.detectMultiScale(clahe_image, scaleFactor=1.1, minNeighbors=15, minSize=(10, 10), flags=c
    if len(face) == 1:
        faceslice = crop_face(clahe_image, face)
        return faceslice
    else:
        print("no/multiple faces detected, passing over frame")

def run_detection():
    global actionlist
    while len(facedict) != 10:
        detect_face()
    actionlist=recognize_emotion()
```

FIG. 8.3 – EMOTION RECOGNITION.PY-PART 3

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Apr  3 17:04:44 2020

@author: rahul
"""
import cv2, numpy as np, argparse, time, glob, os, sys, subprocess, pandas, random, Update_Model, math
from tkinter import *
import tkinter.messagebox
from pygame import mixer
from tkinter import filedialog
import os
import sys
from mutagen.mp3 import MP3
import time
import threading
from tkinter import ttk
from emotion_recognition import main

paused=FALSE
muted=FALSE
playlist=[]
lst=[]
index=0
filename=""
refresh=FALSE

def about_us():
    tkinter.messagebox.showinfo('About BoomBox','This is a program for a music player that can suggest music bas

def browse_file():
    global filename
    filename=filedialog.askopenfilename()
    add_to_playlist(filename)

def add_to_playlist(f):
    global index
    f=os.path.basename(f)
    playlistbox.insert(index,f)
    playlist.insert(index,filename)
    index+=1
```

FIG. 8.4 – MUSIC_PLAYER.PY-PART 1

Update_Model.py music_player.py emotion_recognition.py

```
def del_song():
    selected_song=playlistbox.curselection()
    selected_song=int(selected_song[0])
    playlistbox.delete(selected_song)
    playlist.pop(selected_song)

def show_details(cur_song):
    file_data=os.path.splitext(cur_song)

    if file_data[1]=='.mp3':
        audio=MP3(cur_song)
        totlength=audio.info.length
    else:
        a=mixer.Sound(cur_song)
        totlength=a.get_length

    mins,secs=divmod(totlength,60)
    mins=round(mins)
    secs=round(secs)
    timeformat='{:02d}:{:02d}'.format(mins,secs)
    lengthlabel['text']='Total length- '+timeformat

    t1=threading.Thread(target=start_count,args=(totlength,))
    t1.start()

def start_count(t):
    global paused
    while t and mixer.music.get_busy():
        if paused:
            continue
        else:
            mins,secs=divmod(t,60)
            mins=round(mins)
            secs=round(secs)
            timeformat='{:02d}:{:02d}'.format(mins,secs)
            timelabel['text']=timeformat
            time.sleep(1)
            t-=1

def  play_music():
    try:
        stop_music()
```

FIG. 8.5 – MUSIC_PLAYER.PY-PART 2

Update_Model.py music_player.py emotion_recognition.py

```python
        time.sleep(1)
        selected_song=playlistbox.curselection()
        selected_song=int(selected_song[0])
        play_it=playlist[selected_song]
        mixer.music.load(play_it)
        mixer.music.play()
        statusbar['text']="Playing music : "+os.path.basename(play_it)
        show_details(play_it)
    except:
        tkinter.messagebox.showerror('File not found.','BoomBox could not find the file. Please check again.')

def stop_music():
    mixer.music.stop()
    statusbar['text']="stopped"

def pause_music():
    global paused
    if(paused==TRUE):
        mixer.music.unpause()
        paused=FALSE
        statusbar['text']="Playing music : "+os.path.basename(filename)
    else:
        paused=TRUE
        mixer.music.pause()
        statusbar['text']=os.path.basename(filename)+' paused'

def set_vol(val):
    volume=float(val)/100
    mixer.music.set_volume(volume)

def mute_music():
    global muted
    if(muted==TRUE):
        mixer.music.set_volume(0.5)
        volumebtn.configure(image=volumephoto)
        volcontrol.set(50)
        muted=FALSE
    else:
        mixer.music.set_volume(0)
        volumebtn.configure(image=mutephoto)
        volcontrol.set(0)
        muted=TRUE
```

FIG. 8.6 – MUSIC_PLAYER.PY-PART 3

Update_Model.py music_player.py emotion_recognition.py

```python

def fersuggest():
    global filename
    global refresh
    playlistbox.delete(0,END)
    playlist.clear()
    lst=[]
    lst=main()
    for filename in lst:
        add_to_playlist(filename)
    emodetect=lst[0]
    if(emodetect[6]=='H'):
        emotion='Happy'
    elif(emodetect[6]=='N'):
        emotion='Neutral'
    elif(emodetect[6]=='S'):
        emotion='Sad'
    else:
        emotion='Angry'
    statusbar['text']=emotion+" emotion detected"
    refresh=TRUE

def on_closing():
    stop_music()
    root.destroy()


root=Tk()
#creating blank menubar
menubar=Menu(root)
root.config(menu=menubar)
#creating submenus
subMenu=Menu(menubar,tearoff=0)
menubar.add_cascade(label='File',menu=subMenu)
subMenu.add_command(label='Open',command=browse_file)
subMenu.add_command(label='Exit',command=on_closing)

subMenu=Menu(menubar,tearoff=0)
menubar.add_cascade(label='Help',menu=subMenu)
subMenu.add_command(label='About Us', command=about_us)

mixer.init()
```

FIG. 8.7 – MUSIC_PLAYER.PY-PART 4

```python
root.title("BoomBox")
root.iconbitmap(r'boombox.ico')

statusbar=ttk.Label(root,text='',relief=SUNKEN,anchor=W,)
statusbar.pack(side=BOTTOM,fill=X)

leftframe=Frame(root)
leftframe.pack(side=LEFT,padx=20)

playlistbox=Listbox(leftframe,width=50,height=30)
playlistbox.pack()

addbtn=ttk.Button(leftframe,text="+ Add",command=browse_file)
addbtn.pack(side=LEFT)

delbtn=ttk.Button(leftframe,text="- Delete",command=del_song)
delbtn.pack(side=LEFT)

rightframe=Frame(root)
rightframe.pack()

topframe=Frame(rightframe)
topframe.pack()

lengthlabel=ttk.Label(topframe,text='Total length- --:--')
lengthlabel.grid(row=0,column=0,pady=10)

timelabel=ttk.Label(topframe,text='--:--',relief=GROOVE)
timelabel.grid(row=1,column=0)

ferphoto=PhotoImage(file='fer.png')
ferbtn=ttk.Button(topframe,text='music suggestion',image=ferphoto,command=fersuggest,compound=TOP)
ferbtn.grid(row=0,rowspan=2,column=2,pady=10,padx=20)

middleframe=Frame(rightframe)
middleframe.pack()

playphoto= PhotoImage(file='play.png')
pausephoto= PhotoImage(file='pause.png')
stopphoto=PhotoImage(file='stop.png')

playbtn= ttk.Button(middleframe,image=playphoto,command=play_music)
```

FIG. 8.8 – MUSIC_PLAYER.PY-PART 5

```
playbtn.grid(row=0,column=1,padx=10,pady=10)

pausebtn= ttk.Button(middleframe,image=pausephoto,command=pause_music)
pausebtn.grid(row=0,column=0,padx=10,pady=10)

stopbtn= ttk.Button(middleframe,image=stopphoto,command=stop_music)
stopbtn.grid(row=0,column=2,padx=10,pady=10)

bottomframe=Frame(rightframe)
bottomframe.pack()

volumephoto=PhotoImage(file='volume.png')
mutephoto=PhotoImage(file='mute.png')

volumebtn=ttk.Button(bottomframe,image=volumephoto,command=mute_music)
volumebtn.grid(row=0,column=0,padx=20)

volcontrol=ttk.Scale(bottomframe,from_=0,to=100,orient=HORIZONTAL,command=set_vol)
volcontrol.set(50)
mixer.music.set_volume(0.5)
volcontrol.grid(row=0,column=1,pady=15)

root.protocol("WM_DELETE_WINDOW",on_closing)

root.mainloop()
```

FIG. 8.9-MUSIC_PLAYER PY-PART 6

Update_Model.py music_player.py emotion_recognition.py

```python
import cv2
import glob
import random
import numpy as np

fishface = cv2.face.FisherFaceRecognizer_create()

data = {}

def make_sets(emotions):
    training_data = []
    training_labels = []

    for emotion in emotions:
        training = training = glob.glob("dataset\\%s\\*" %emotion)
        for item in training:
            image = cv2.imread(item)
            gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            training_data.append(gray)
            training_labels.append(emotions.index(emotion))

    return training_data, training_labels

def run_recognizer(emotions):
    training_data, training_labels = make_sets(emotions)

    print("training fisher face classifier")
    print("size of training set is: " + str(len(training_labels)) + " images")
    fishface.train(training_data, np.asarray(training_labels))

def update(emotions):
    run_recognizer(emotions)
    print("saving model")
    fishface.write("trained_emoclassifier.xml")
    print("model saved!")
```

Fig. 8.10 – Update_Model.py-part 1

CHAPTER 9

SYSTEM TESTING

9.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

During the development of software, errors can be injected at any stage. However, requirements and design errors are likely to remain undetected. During testing, the program to be tested is executed with a set of test cases and output of program for test cases are evaluated to describe the program performance to expected level. No system design is perfect because of communication problem, programmer‘s negligence or constraints create errors that must be eliminated before the system is ready for use of acceptance. There are various types of test. Each test type addresses a specific testing requirement. Following sections describes the system testing and test cases.

9.2 Unit Testing

This is the first level of testing, where different components are tested against the requirement specification for the individual components. Unit testing is essential for verification of the code produced during the coding phase and hence the goal is to test internal logic of those components.

9.3 System Testing

System testing is performed on the entire system in the context of functional requirements specification and system requirement specifications. System testing tests not only the design, but also the behaviour and the expectations. It is also to test up to and beyond the bounds defined in the software requirement specification. System testing for our platform is accomplished by the following test cases. These test cases show how each and every component interacts with each other as a dynamic system.

TEST CASE ID	PAGE	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	STATUS
TC01	Home screen	Face Detection	Face is detected	Face is detected	PASS
TC02	Home screen	Face not present	Show error	Show error	PASS
TC03	Home screen	Play music without selection of music.	Show error	Show error	PASS
TC04	Home screen	Happy Emotion Recognition	Emotion Recognized	Emotion Recognized	PASS
TC05	Home screen	Neutral Emotion Recognition	Emotion Recognized	Emotion Recognized	PASS
TC06	Home screen	Sad Emotion Recognition	Emotion Recognized	Emotion Recognized	PASS
TC07	Home screen	Anger Emotion Recognition	Emotion Recognized	Emotion Recognized	PASS
TC08	Playlist screen	Playlist generation based on emotion	Playlist Generation for respective emotion detected	Playlist Generation for respective emotion detected	PASS
TC09	Music screen	Play selected music	Play the selected music	Play the selected music	PASS
TC10	Home screen	Detect new face and emotion	New emotion detected	New emotion detected	PASS

TABLE 9.1 - TEST CASES

CHAPTER 10

RESULTS AND SCREENSHOTS

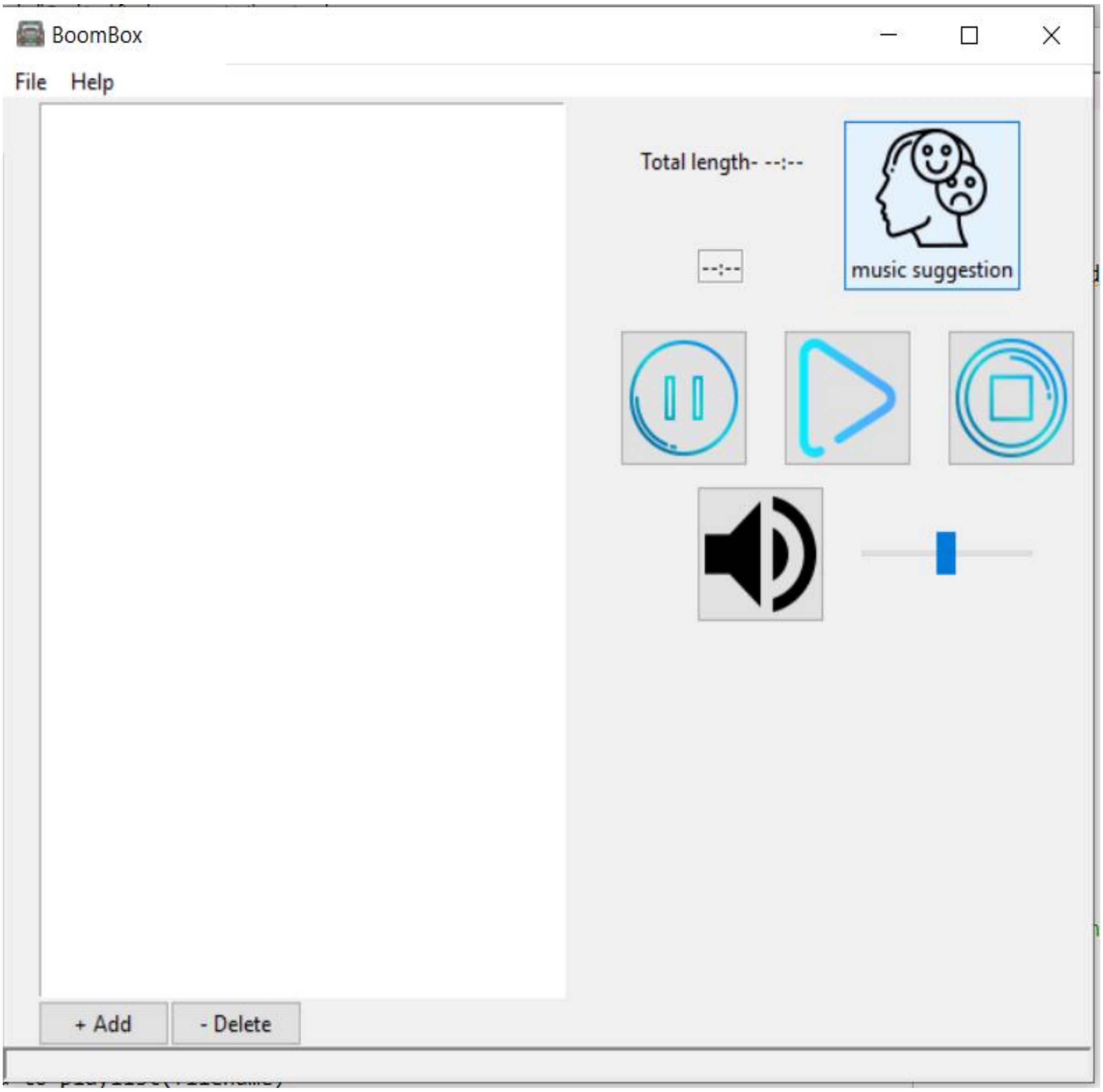


FIG 10.1 – HOME SCREEN

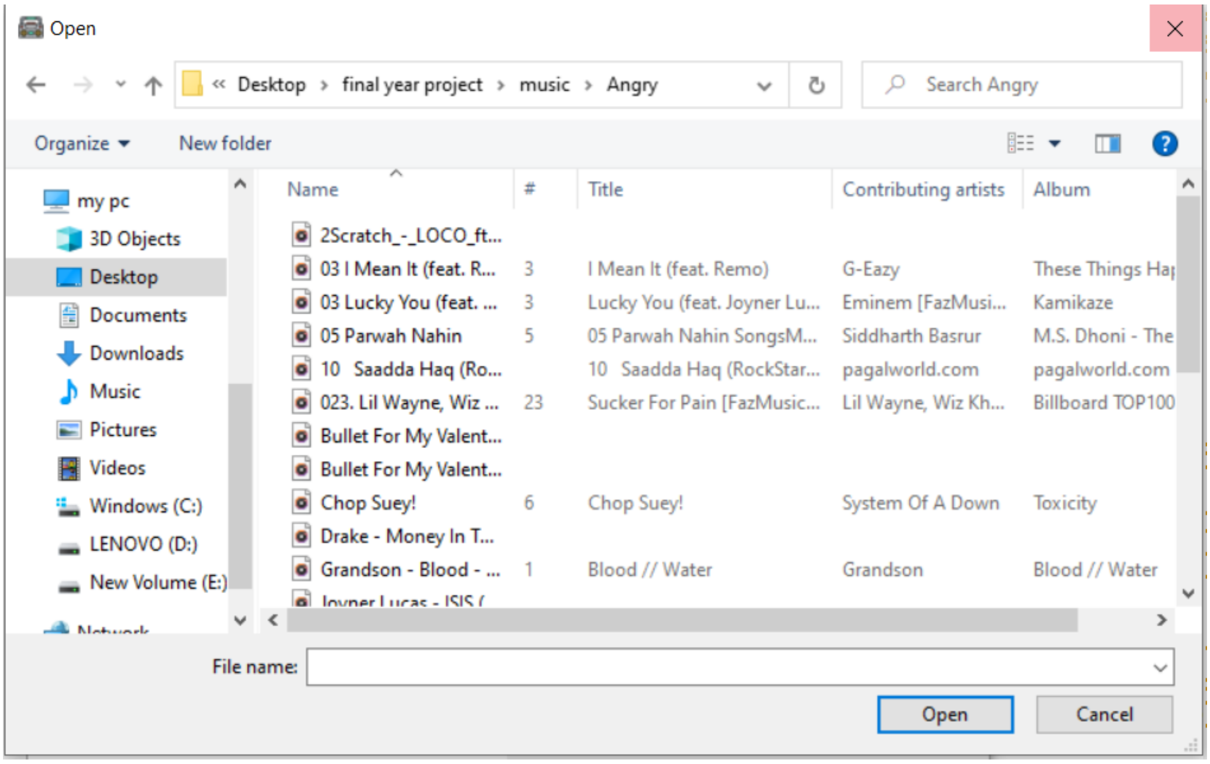


FIG 10.2– MUSIC FILES IN THE LIBRARY

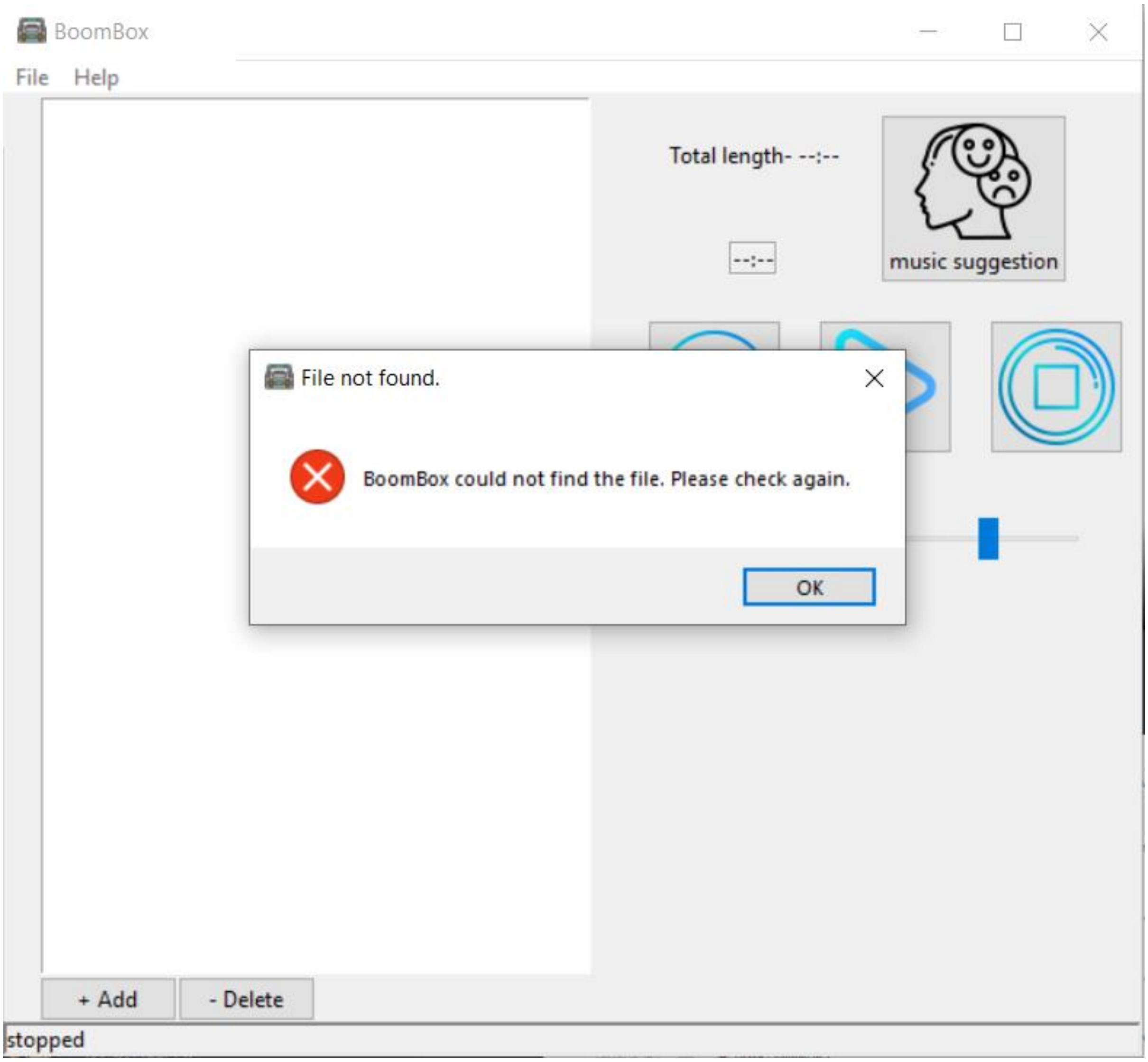


FIG 10.3 – ERROR IN PLAYING MUSIC WITHOUT OPENING A FILE

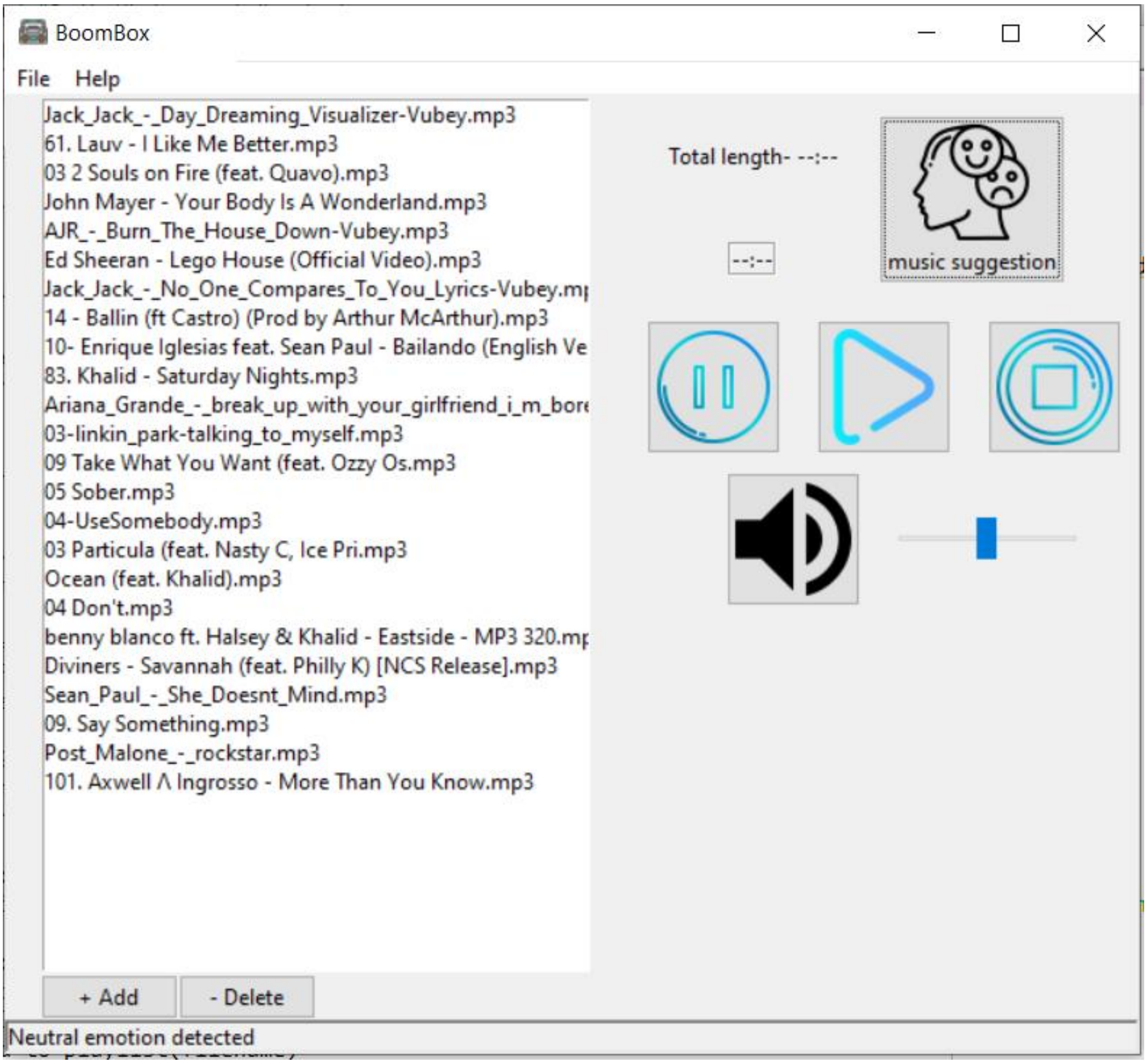


FIG 10.4– PLAYLIST BASED ON DETECTION OF NEUTRAL EMOTION

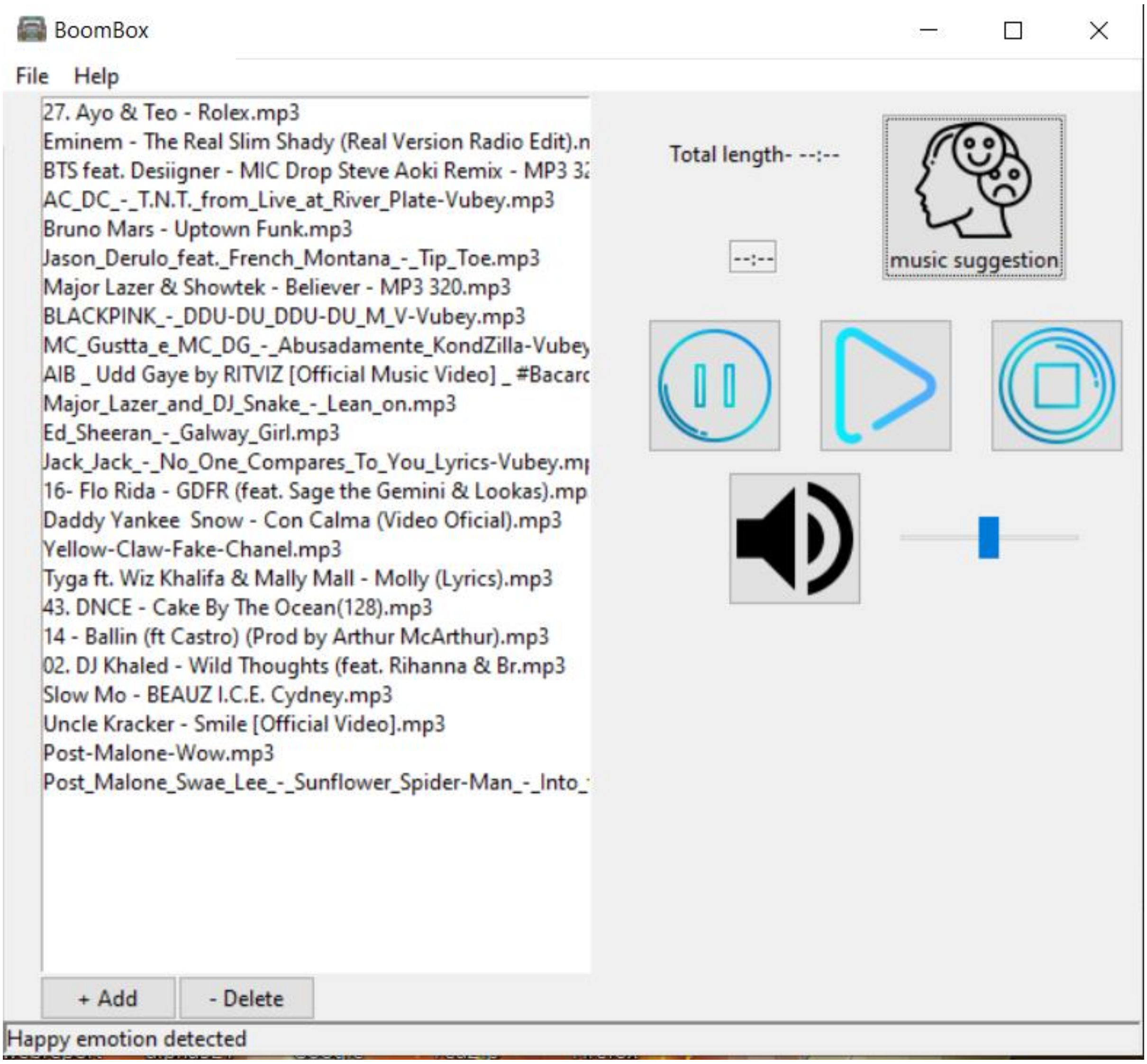


FIG 10.5 –PLAYLIST BASED ON DETECTION OF HAPPY EMOTION

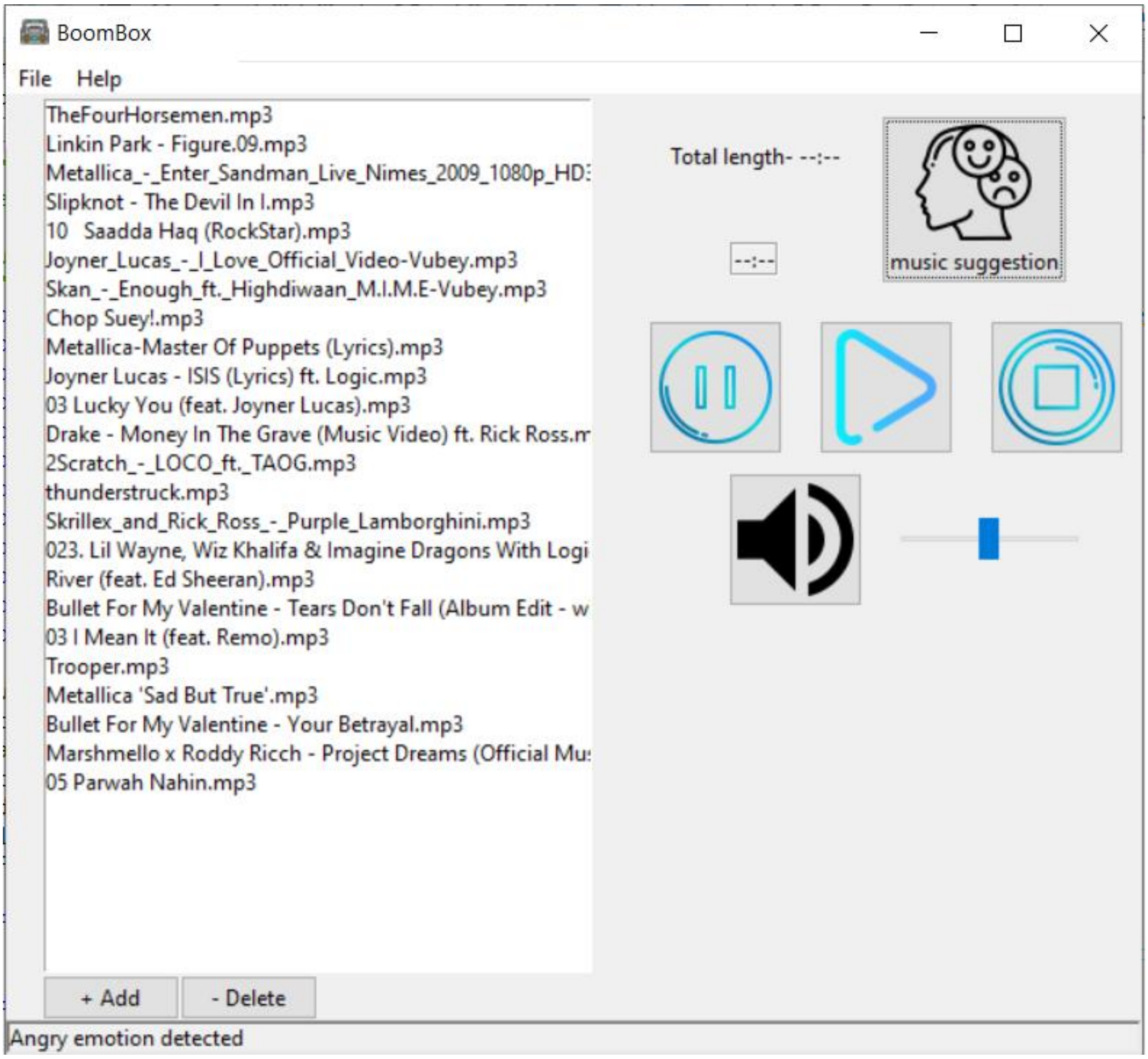


FIG 10.6 – PLAYLIST BASED ON DETECTION OF ANGRY EMOTION

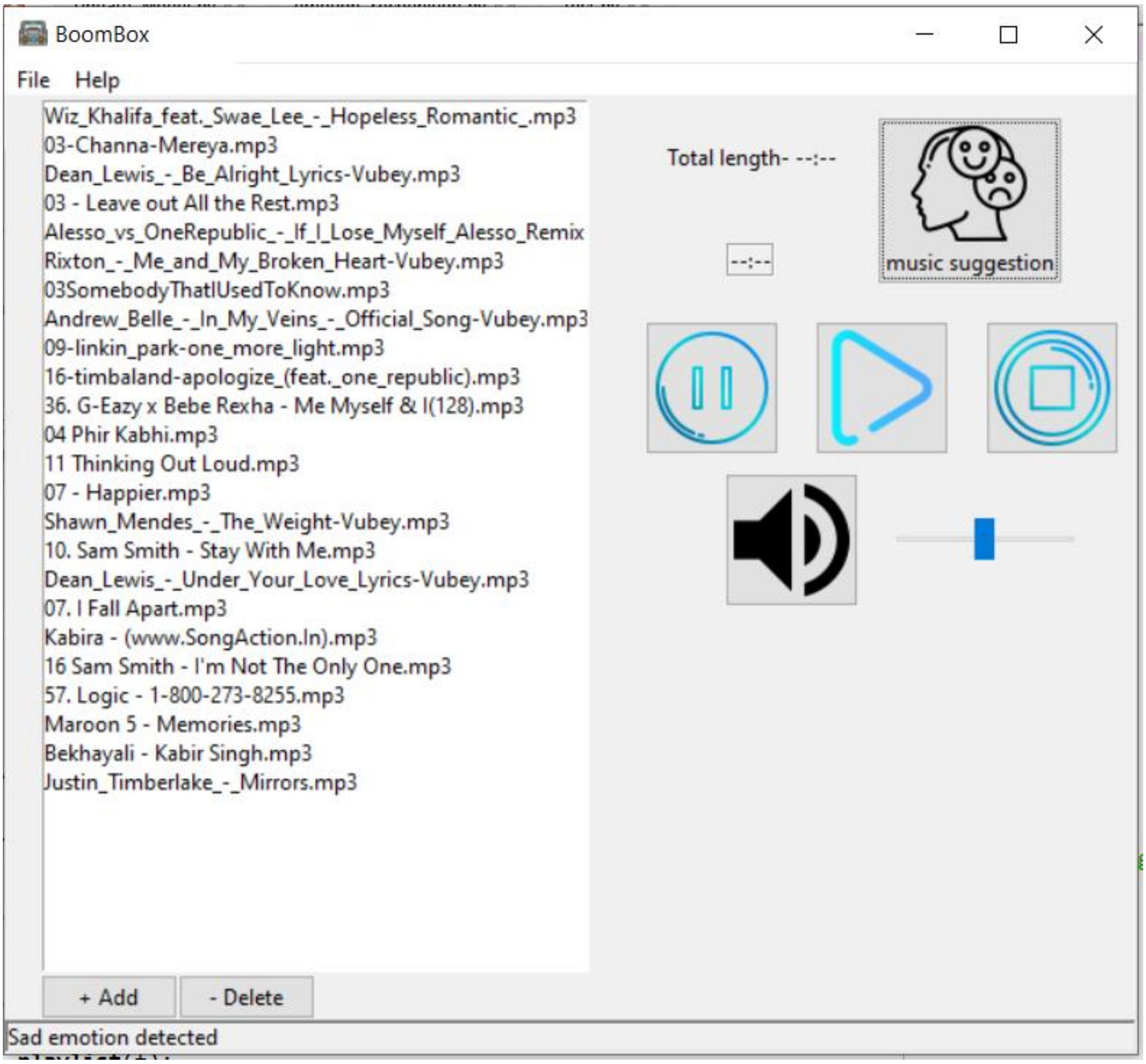


FIG 10.7 – PLAYLIST BASED ON DETECTION OF SAD EMOTION

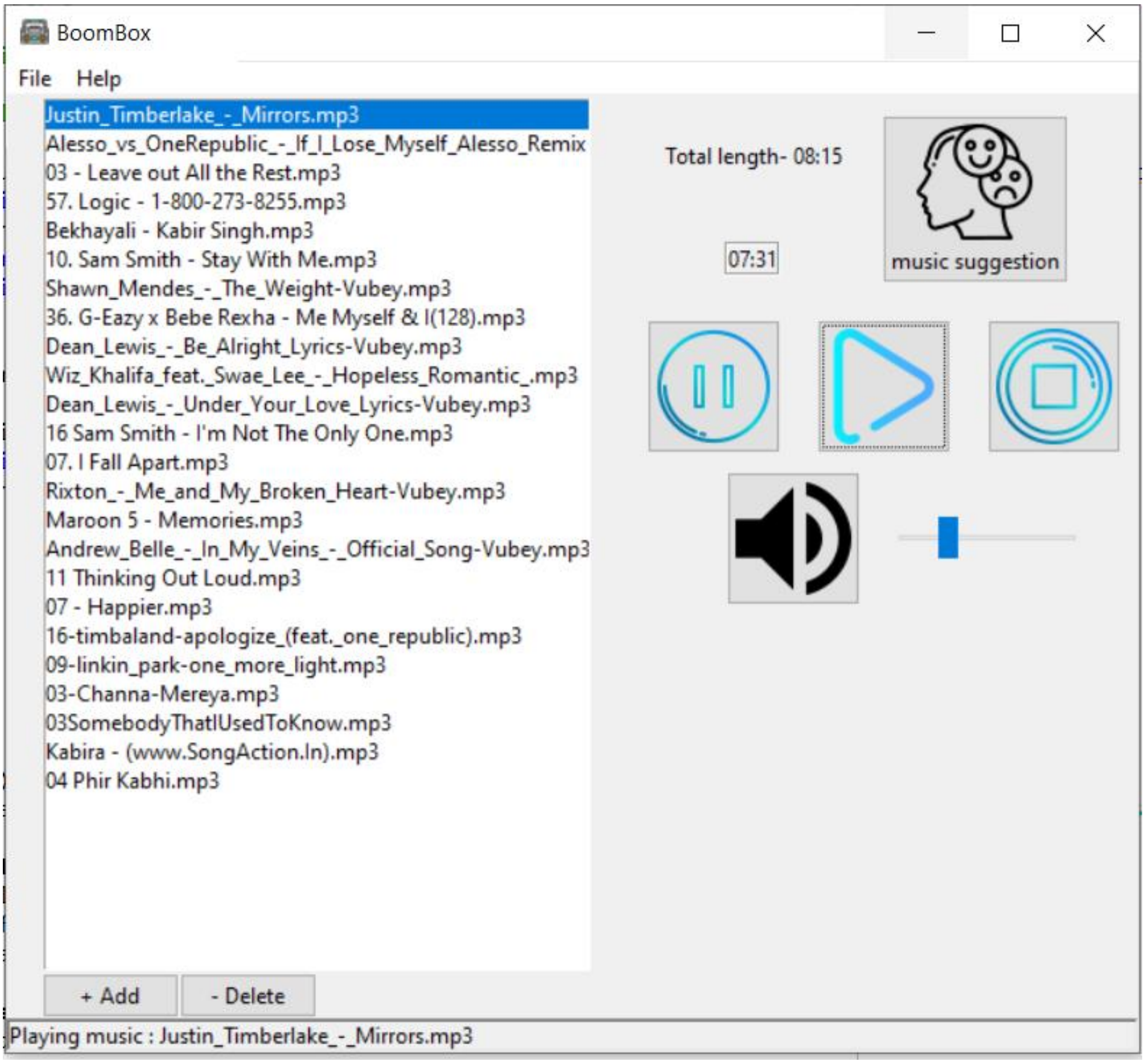


FIG 10.8–PLAYING MUSIC FROM THE CREATED PLAYLIST

CHAPTER 11

CONCLUSION AND SCOPE

The Emotion-Based Music Player is used to automate and give a better music player experience for the end user. The application solves the basic needs of music listeners without troubling them as existing applications do, it uses technology to increase the interaction of the system with the user in many ways. It eases the work of the end-user by capturing the image using a camera, determining their emotion, and suggesting a customized play-list through a more advanced and interactive system.

The system can improve in the following ways:

- Enable the classifier to identify a wide range of emotions.
- Personalized classification of songs based on the preference of the user.
- Can detect when a person is sleepy during driving.
- Addition of an Audio detection component to increase the accuracy of the classifier by sensing the tone of the voice.
- Integration of the classifier into a web/mobile application.

CHAPTER 12

REFERENCES

12.1 References

1. Vijaykumar R. Ghule, Abhijeet B. Benke, Shubham S. Jadhav, Swapnil A. Joshi," Emotion Based Music Player Using Facial Recognition", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 2, February 2017.
2. W. Swinkels, L. Claesen, F. Xiao and H. Shen, "SVM point-based real-time emotion detection", 2017 IEEE Conference on Dependable and Secure Computing, Taipei, 2017.
3. H. Ebine, Y. Shiga, M. Ikeda and O. Nakamura, "The recognition of facial expressions with automatic detection of the reference face," 2000 Canadian Conference on Electrical and Computer Engineering. Conference Proceedings. Navigating to a New Era (Cat.No.00TH8492), Halifax, NS, 2000, pp. 1091-1099 vol.2.
4. Trung-Thanh Dang, Kiyoaki Shirai, "Machine Learning Approaches for Mood Classification of Songs towards Music Search Engine", 2009 International Conference on Knowledge and Systems Engineering, Hanoi, Vietnam
5. E. Sariyanidi, H. Gunes, and A. Cavallaro. Automatic analysis of facial affect: A survey of registration, representation, and recognition. IEEE transactions on pattern analysis and machine intelligence, 37(6):1113–1133, 2015.
6. Behzad Hasani and Mohammad H. Mahoor, "Facial Expression Recognition Using Enhanced Deep 3D Convolutional Neural Networks", IEEE conference on CVPR, 2017.
7. Ratna Astuti Nugrahaeni, Kusprasapta Mutijarsa, "Comparative Analysis of Machine Learning KNN, SVM, and Random Forests Algorithm for Facial Expression Classification", ieee.org, 2016.
8. Keyu Yan, Wenming Zheng, Zhen Cui, Yuan Zonga, Tong Zhang, Chuangao Tang, "An extended dictionary representation approach with deep subspace learning for facial expression recognition", Elsevier.com,2018.

9. D. Yang, Abeer Alsadoon, P.W.C. Prasad, A. K. Singh, A. Elchouemi, “An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment”, Elsevier, 2018
10. Shlok Gilda, Husain Zafar, Chintan Soni , Kshitija Waghurdekar,”Smart Music player integrating facial emotion recognition and music mood recommendation”, IEEE 2017