A
Project Report
On

FACIAL SENTIMENTS BASED MUSIC PLAYER

BTech-sem VI

Developed By:
DIRGH PATEL (IT-093)
PRANAY RAYCHA (IT-108)

Guided By:
Prof. Archana N. Vyas
Department of Information Technology
Faculty of Technology
DDU University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad – 387001

April-2021

A
Project Report
On

FACIAL SENTIMENTS BASED MUSIC PLAYER

B.Tech-Sem VI

In partial fulfillment of requirements for

Bachelor of Technology

in

Information Technology

Submitted By:

DIRGH PATEL (IT-093)

PRANAY RAYCHA (IT-108)

Under the Guidance of

Prof. Archana N. Vyas

DEPARTMENT OF INFORMATION TECHNOLOGY

FACULTY OF TECHNOLOGY, DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001

CANDIDATE'S DECLARATION

We declare that pre-final semester report entitled " **FACIAL SENTIMENTS BASED MUSIC PLAYER**" is our own work conducted under the supervision of the guide **Prof. Archana N. Vyas**.

We further declare that to the best of our knowledge the report for B.Tech. VI semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Name: **DIRGH PATEL**
Student ID: **18ITUOS123**

Candidate's Name: **PRANAY RAYCHA**
Student ID: **18ITUOS038**

Submitted To:
Prof. Archana N. Vyas
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Gujarat.

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project carried out in the subject of Software Design Project ,entitled **"FACIAL SENTIMENTS BASED MUSIC PLAYER"** and recorded in this report is a bona fide report of work of

1) **Dirgh Patel** Roll No**. IT093** ID No: **18ITUOS123**
2) **Pranay Raycha** Roll No. **IT108** ID No: **18ITUOS038**

of the Department of Information Technology, semester VI. They were involved in Project work during the academic year 2020-2021.

Prof.Archana N. Vyas
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGMENT

We owe our deep gratitude to our project guide **Prof.Archana N. Vyas**, who took a keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We would also like to express our gratitude to our Head, Mr. Vipul Dabhi along with Dharmsinh Desai University for giving us this opportunity.

With Sincere Regards,

Dirgh Patel
Pranay Raycha

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

We know the importance of music in our daily life. We humans respond and react to music and that music has a high impact on a person's brain activity. People tend to listen to music based on their mood and interests. We have the ability to communicate in different ways, non-verbal communication is one of them. It is a very powerful way of communication in which humans don’t speak but they express their mood through facial expressions. We want to create a project which focuses on creating an application to suggest songs for the user based on their mood by capturing facial expressions. In this system, computer vision components are used to determine the user's emotion through facial expressions. Once the emotion is recognized, the system suggests a list of songs for that emotion, saving a lot of time for a user over selecting and playing songs manually.

1. INTRODUCTION

1.1 Project Details

Title: **FACIAL SENTIMENTS BASED MUSIC PLAYER**

Features:

This system contains all the functionalities that a music player provides us. In addition to that we have added a feature which provides a personalised playlist according to the user's mood.

1.2 Purpose

The main purpose of this system is to provide users a better and faster way to suggest music. We aim to build an application that focuses on reducing human efforts by generating a playlist based on facial sentiments.

Advantages of proposed system:

- Users don't want to select songs manually.
- No need for a playlist.
- Users don't want to classify the songs based on the emotion.

1.3 Scope

- In this project, there is a real-time camera app that is capturing an image and then the application puts this for image testing, and then it plays appropriate music after that, the music of the app is built-in music so the user can not add their music to the app at the time being.
- In the future we can expand the scope to adding assistants like google assistant which will hear the needs of the user and show the appropriate results.

1.4 Objective

The Objectives of the system are:

- Build a Face Detection System.
- Develop Emotion Recognition System.
- Build an easy-to-use app using cross-platform open-source library "Tkinter".

There are some limitations:

- Users can not add music by him/herself.

1.5 Technology And Literature Review

We tried to build a new app with much more efficiency and also a cost effective app. We started by classifying emotions into four major emotions, (Happy, Sad, Angry, and Neutral) and also using Convolutional Neural Network and we got over 90% as accuracy and we used CNN because it plays a major role in simulating the human brain and analysing visuals.

The dataset is (48 x 48) pixels grayscale images and they are classified to seven emotions (Happy, Sad, Angry, Neutral, Disgust, Fear, and Sur-prise), these seven were too huge and hard to get good results while analysing them all together so we chose 5 main emotions to analyse (Happy, Sad, Angry, Surprise and Neutral), also there are huge number of images in this dataset that been used (24282 image), (7164 for Happy, 4938 for Sad, 4982 for Neutral, and 3993 for angry, 3205 for surprise).

The Convolutional Neural Network used to classify these images to detect emotions from facial expressions, the CNN in this experiment broke down into many layers, and Layer 1 is the Input Layer this layer one of the most important parts of whole process, this layer is used to detect the face from the image using OpenCV. OpenCV is a library of functions that has been used to help in the process of face detection, and it's a very well-known library.

Layer 2 is the convolutional layer part and this part the core of any convolutional neu-ral network, convolution as a concept and a step in the process is step that is used to extract features from an image. The convolutional saves the relationship between each and every pix-els in the image by learning the features of this image using Small Square from each image. Let's get into it in detail, first let's consider an image with a 5*5 matrix and a 3*3 matrix as shown below:

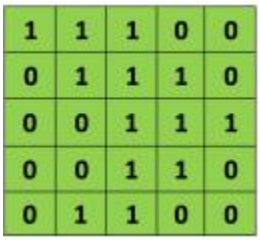


Figure 1.1 Image matrix (5*5)

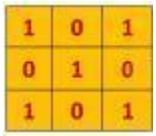


Figure 1.2 Filter Matrix

The first matrix as we said above is the 5*5 image matrix this matrix represents the image by 1s and 0s, and second matrix is the 3*3 matrix which called the filter, this matrix will grow through every point and in the 5*5 matrix to generate the final three by three con-volved featured matrix, the convolved matrix is generated from giving the 3*3 filtered matrix get in into the 5*5 matrix through multiplication to generate the last matrix as shown below (EXAMPLE ONLY). The final convolved matrix or the feature map which has all needed fea-tures in an image that could be useful to use and detect emotions:

Figure 1.3 Final Convolved Matrix

The second step or the second layer is the ReLU (Rectified Linear Unit) it's used in every convolution step in the model and the output of this operation as shown below:

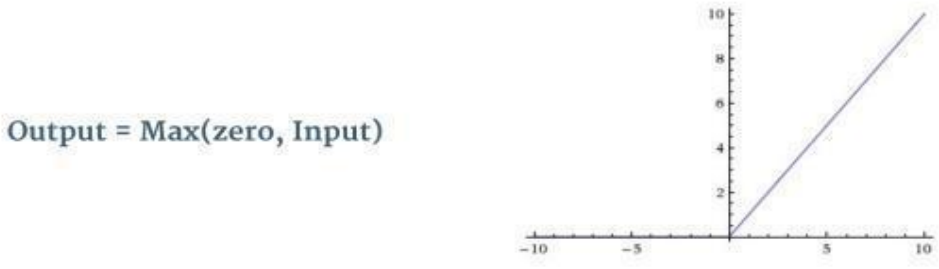


Figure 1.4 Output of the ReLU

In detail to explain the Rectified Linear Unit, the ReLU is an operation to replace every negative value in an image by 0 (DETECT BLACK AREAS). The goal of using the Rectified Linear Unit function is proof of the concept of Non-Linearity and every real time and new Convolutional Neural Network must Non-Linear based to learn in a more efficient way. The Rectified Linear Unit in action is shown below :

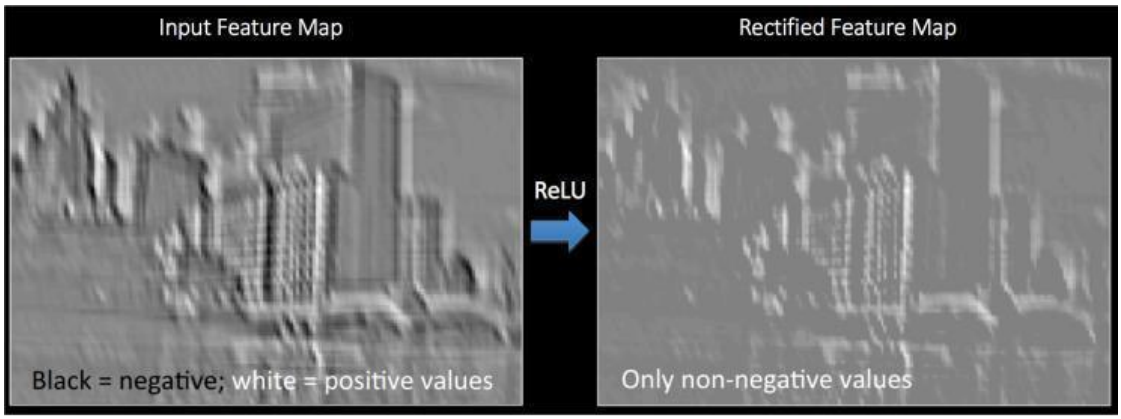


Figure 1.5 ReLU in Action

One more layer and one of the most important ones is the Pooling Layer, this step is called Spatial Pooling this step is to reduce the dimensionality of each feature map (OUTPUT OF THE CONVOLUTIONAL LAYER) and again the feature map is the output image

with more well-known feature (FINAL 3*3 MAP).

There are many types of Spatial Pooling like Max, Average, Sum and more. Below in Figure 9 will be a sample of a Max Pooling for an image (MATRIX)

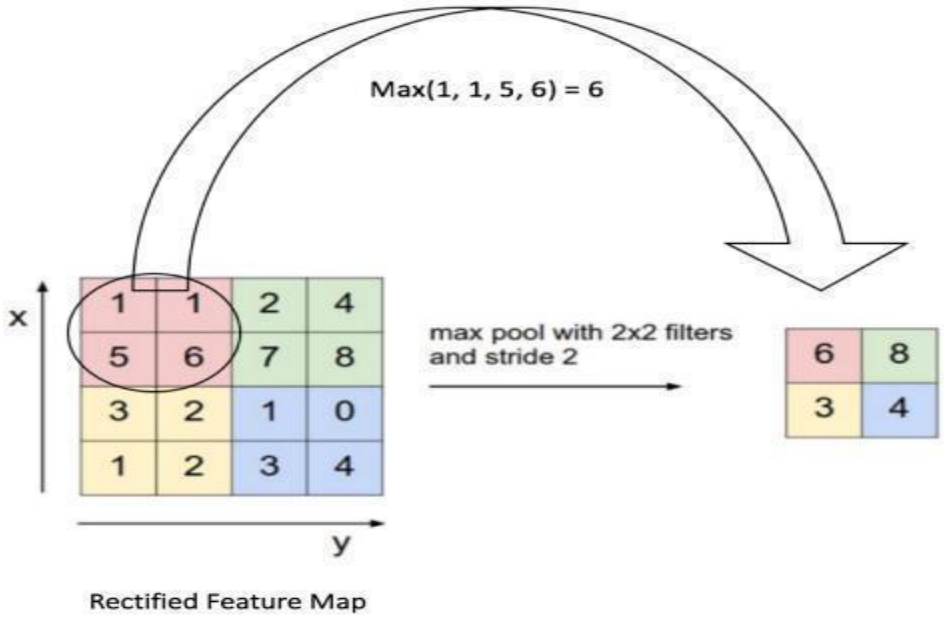


Figure 1.6 Example of a Max Pooling

This is when it comes to a matrix but to be more efficient and reliable let's discuss it with an image, so here's below a sample pooling for a ReLU sample image to reduce dimen-sionality of an extracted feature map:

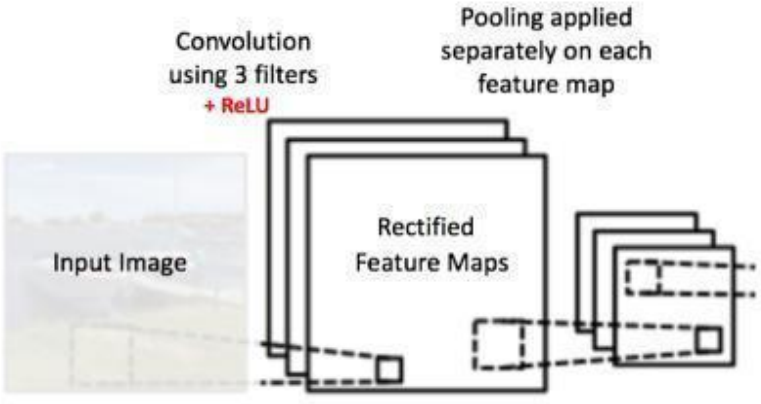


Figure 1.7 Sample Pooling for an Image

To finalize the Pooling Layer let's say what the Pooling is doing as a core of the net-work:

1. Reduce input representations to be smaller and manageable.
2. Reduce number of parameters and reduce also any computations in the network.
3. Help in making the network more stable regarding the small transformations and distortions.
4. Help us to reach most stability in the scale representation of an image.

Let's summarize all studied layers above, first, the input Layer which is the layer responsible for face detection from the image through any library that can be used and in our case it is the (OpenCV) library. Second, the convolutional Layer which is getting into a big image with a big matrix like (5*5) matrix image to reduce the size of the matrix with increasing the value which is the goal of this step, the goal was to create a more efficient smaller matrix which is the feature map matrix, which has all needed features that could be extracted from a bigger image or a matrix. The last thing is the Pooling Layer which decreases the dimensionality of a feature map matrix to a more efficient one and example of it could be Max, Avg, Sum and so on, below in figure 11 a summary of the process the first three layers and the core of any net-work:

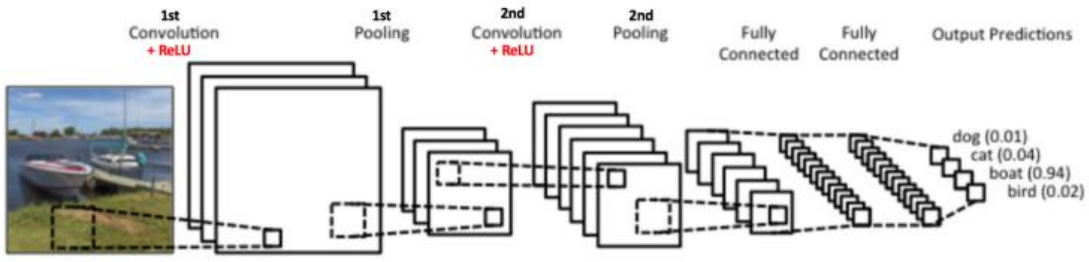


Figure 1.8 Summary of first 3 layers

One more layer is the classification layer (Fully Connected Layer), as discussed above there are two main steps which are the Convolutional and the Pooling Layers and the purpose of these layers is to extract high level and high intense features from an image, by adding Fully Connected Layer it will be classifying every generated feature into useful classes depends on the needed classes from the dataset like in this case (Happy, Sad, Angry, and Neutral). Adding a Fully Connected Layer is very useful to let the network learn to be Non-Linear, and also using the output of the Convolution Layer alone or the Pooling alone will be useful as a fea-ture selection by combine them all together will be very useful and more effective and better classification phase.

2. PROJECT MANAGEMENT

2.1 Feasibility Analysis

2.1.1 Technical Feasibility

Viewing our project from a technical point of view (thinking about various tools and technologies being used in developing the system) we have decided that following technologies will be more than enough to develop a complete working system (including tech. & tools used for project tracking, monitoring etc. along with development).

For Tools and Technologies: Tkinter, OpenCV, keras, CNN.
For Version Control/Tracking: Git and GitHub

Convolutional Neural Network is a Neural Network that proved high efficiency in dealing with image recognition and classification. It proved also efficient in detecting objects, faces, and many advanced features like detecting traffic signs and sure dealing with robots and their work. For front-end we have used Tkinter which is the standard GUI library for Python. Python when combined with **Tkinter** provides a fast and easy way to create GUI applications. It provides a powerful object-oriented interface to the Tk GUI toolkit.

2.1.2 Time Schedule Feasibility

We have planned the steps for completion of our project in given duration. Firstly we will prepare SRS document and the GUI design tentatively by 15th January, 2021.The diagrams required for the design as well as the dataset required will be tentatively completed/collected by 23th January 2021.For coding and unit testing 4-5 weeks and for system and integration testing another 2-3 weeks will be required. Hence tentatively by the last week of March 2021 we will be able to complete the project and ready for its demonstration. Being a 2 member team we will be able to complete our project in the estimated time.

2.1.3 Operational Feasibility

In the current COVID pandemic situation, we are forced to manage our project remotely. So, operability and management of a project is going to be somewhat difficult but it will be quite feasible to develop a project remotely using help of tools like Git and GitHub as well as good social media platforms to communicate ideas and work regarding the project. from an organizational point of view, it is sufficient to maintain proper teamwork even remotely using such tools. Hence our project is operationally feasible

2.2 Project Planning

2.2.1 Project Development Approach And Justification

For software development an Iterative waterfall model is used.

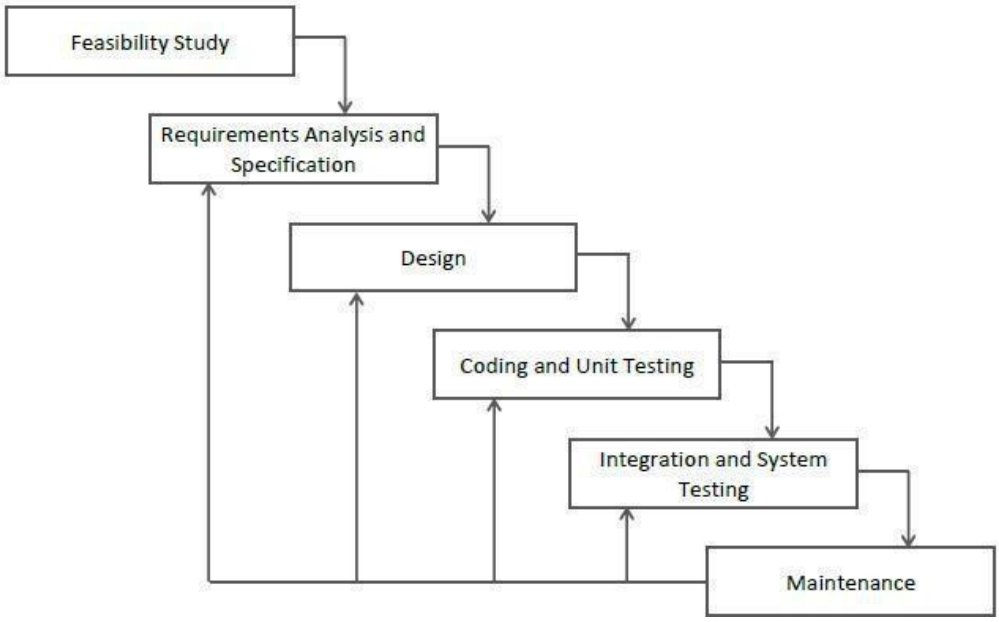


Fig 2.2.1 Iterative Waterfall Model

Iterative Waterfall Model is the extension of the Waterfall model. This model is almost the same as the waterfall model except some modifications are made to improve the performance of the software development. The iterative waterfall model provides customer's feedback paths from each phase to its previous phases.

We have used this model to allow us to go back to the previous phase and change the requirements and some modification can be done if necessary. Also it reduces efforts and time required to detect and correct the errors.

Phases of Iterative Waterfall Model

1. Requirement Analysis
2. Feasibility Study
3. Software Design
4. Coding/Implementation
5. Software Testing
6. Software Deployment
7. Software Maintenance

Advantages of Iterative Waterfall Model

1. Iterative waterfall model is very easy to understand and use.
2. Every phase contains a feedback path to its previous phase.
3. This is simple to make changes or any modifications at any phase.
4. By using this model, developers can complete projects earlier.
5. Customer involvement is not required during the software development.
6. This model is suitable for large and complex projects.

Disadvantages of Iterative Waterfall Model

1. There is no feedback path for the feasibility study phase.
2. This model is not suitable if requirements are not clear.
3. It can be more costly.
4. There is no process for risk handling.
5. Customers can view the final project, no prototype for taking customer review.
6. This model does not work well for short projects.
7. If modifications are required repeatedly then it can be more complex projects.

2.2.2 Roles And Responsibilities

We have divided different responsibilities among two of us regarding projects as per following.

1) Feasibility Analysis, Req. Gathering and Analysis – Dirgh, Pranay

2) System Design(Overall Architecture): Dirgh, Pranay

3) Dataset collection of images: Dirgh

4) GUI Design: Dirgh

5) Frontend Implementation: Designing – Dirgh

6) Programming – Dirgh, Pranay

7) Backend Implementation: Dirgh, Pranay

8)Testing Activities:

Unit Testing - Will be carried out for different modules individually by person associated with respective module.

Integration/System Testing : Dirgh, Pranay

2.2.3 Group Dependencies:

The dependencies among the tasks include the following:

- Analysis or System Requirement Study (SRS) is independent of all, yet will be started after completion of feasibility study and project planning.
- Designing of prototypes can be done simultaneously with system analysis.
- Development of the project is preceded by the designing of prototype and system analysis.
- Testing can be only done once the development of some major functionalities are completed and are ready to be tested.
- Documentation is independent of all the tasks and can be done as the other tasks proceed.

2.3 Project Scheduling

Scheduling the project tasks is an important project planning activity. It involves deciding which task should be taken up and when in order to schedule the project activities; a software project manager needs to be the following :

1.Identify all the tasks needed to complete the project.

2.Break down large tasks into smaller activities.

3.Determine dependencies among different activities.

4.Establish most likely estimates for the time durations necessary to complete the activities.

ID	Task Name	Start Date	Finish Date	Duration
1	Feasibility Study	10/1/21	12/1/21	2 days
2	Requirement Gathering	14/1/21	21/1/21	6 days
3	Analysis	22/1/21	29/1/21	6 days
4	Design	2/2/21	16/2/212	12 days
5	Coding	17/2/21	27/3/21	40 days
6	Testing	28/3/21	30/3/21	3 days

Table 2.3 Project Scheduling Chart

3. SYSTEM REQUIREMENT STUDY

3.1 Current System Study

Using a traditional music player, a user has to manually look through his playlist and select songs according to his mood and emotional experience. Various music players have been developed with features like fast forward, reverse, variable playback, local playback, streaming playback with multicast streams and including volume modulation, genre classification etc.

Although these features satisfy the user's basic requirement, yet the user has to face the task of manually browsing through the playlist of songs and selecting songs based on his current mood and behavior. That is the requirement of an individual, a user sporadically suffered through the need and desire of browsing through his playlist, according to his mood and emotion.

3.2 Limitations of Current System

I. Current system provides a limited amount of songs. But users can manually add songs to folders created when the system is not in use.
II. It is hard to detect mood in extreme light conditions for eg. detection in dark rooms, or using low resolution web-cams.

3.3 Minimum Hardware/Software Requirements

There are no such specific hardware requirements other than basic requirements such as a computer, and a good quality web-cam.

3.3.1 Software Requirements:

- Python version 2.7 or above
- OpenCV
- Keras, Tensorflow, numpy

3.3.2 Hardware Requirements:

- Microsoft® Windows® 7/8/10.
- 2 GB RAM minimum, 8 GB RAM recommended
- Web-cam.

4. SYSTEM ANALYSIS

4.1 Requirements of system(SRS)

4.1.1 Functional Requirements:

R1: Camera permission phase

Description: Users have to allow the system to use the web-cam to capture images.
Input: Click “Yes” or “No”.
Output: If users click on the system will exit else continue to capture the image.

R2: Emotion Capture phase

Description: As soon as the permission phase is done, the application will capture the user’s image through a web-cam and start detecting the user's emotion.
Input: User face.
Processing: Scanning of face.
Output: Face detected.

R3: Image Processing Phase

Description: After the image is captured, the application sends the captured image for processing and after the captured image is processed the image feedback is sent to the application.
Input: Scanned face from Emotion Capture Phase.
Processing: Image processing.
Output: Image feedback is sent to the application.

R4. Emotion Detection Phase

Description: In this phase, the application receives the image information and recognizes the emotion based on the defined threshold. This emotion is then used to fetch the emotion’s play-list.
Input: Input from image processing phase.
Processing: Detection of emotion.
Output: Emotion generated.

R5: Confirmation Phase

Description: Here the user will be prompted by a pop up confirming the detected emotion. If the user is not satisfied with the emotion detected, they can re-detect it.
Input: Confirmation pop-up: “Confirm Mood”
Processing: If “Yes” the system starts playing song, if “No” re-scan face.

R6: Display phase

Description: Here the songs are provided to the user and the user can play any song from the list displayed. The user has the option to play, pause, skip forward, skip backward and also pause a song in the application.

4.1.2 Non-Functional Requirements:

N1: Security Requirements

It is the primary requirement of any system. The system shall ensure all data is protected from unauthorized access. The central repository should be platform independent so that it can be accessible and store application data via the web application and the mobile app.

N2: Operational Requirements

- The system will draw information from main songs databases, which contain basic information about songs availability, songs details.
- The system will store information by the user when making the update of songs in the playlist.

N3: Performance

It is very important to know how well the system performs certain functions under specific conditions. Examples are speed of response, throughput, execution time and storage capacity. This platform should be designed in such a way that its performance is smooth for users.

N4: Availability

System must be available to all the users when it is required at any hour of the day. In any abnormal situation like a power cut user's data should not be lost.

System should provide confidentiality for user data using database encryption and local encryption to protect data in the event of device theft(laptop/handheld device).

5. SYSTEM DESIGN

5.1 Use-Case Diagram

(Admin & User both)

Figure 5.1 Use case Diagram

5.2 Class Diagram

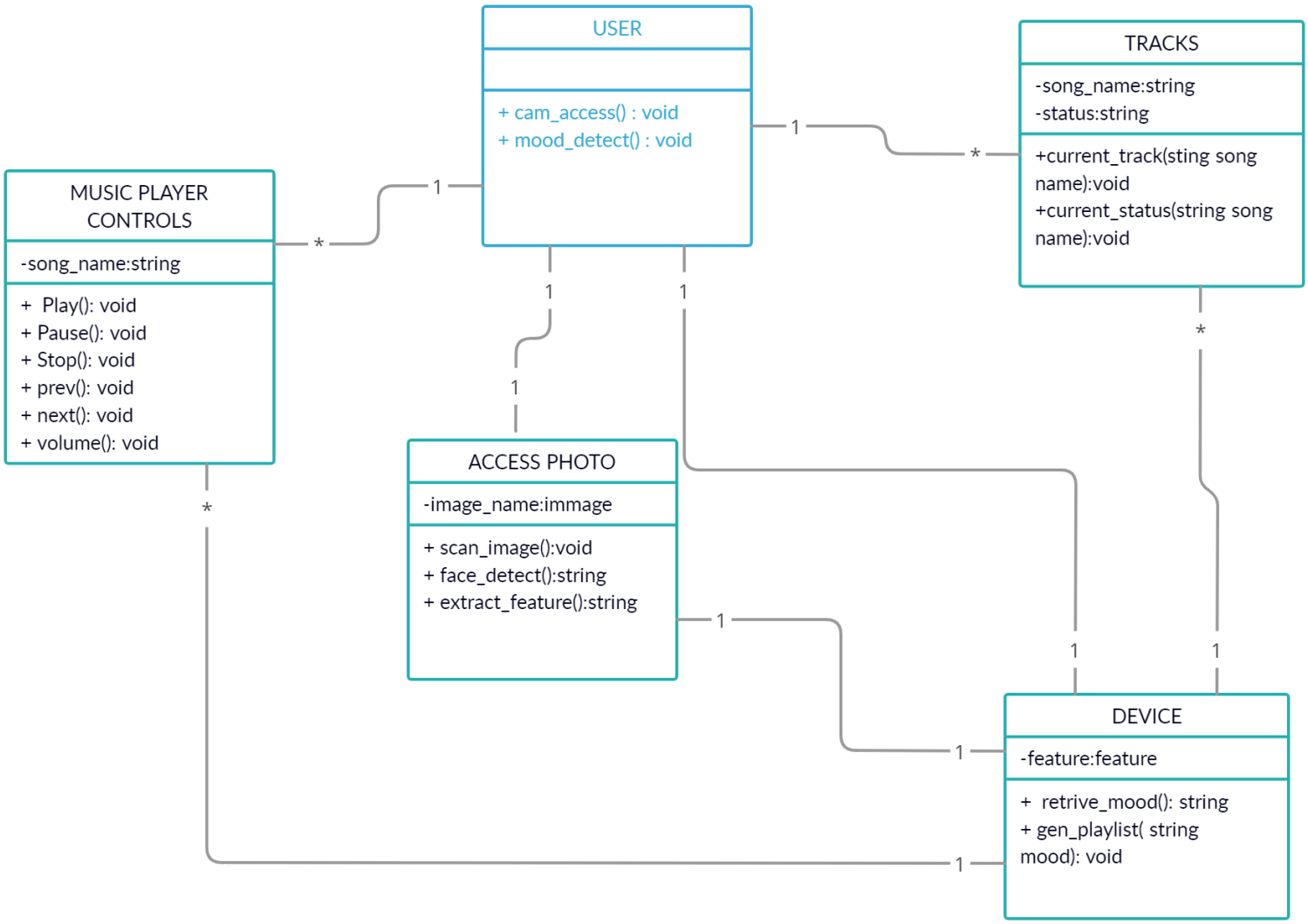


Figure 5.2 Class Diagram

5.3 Sequence Diagram

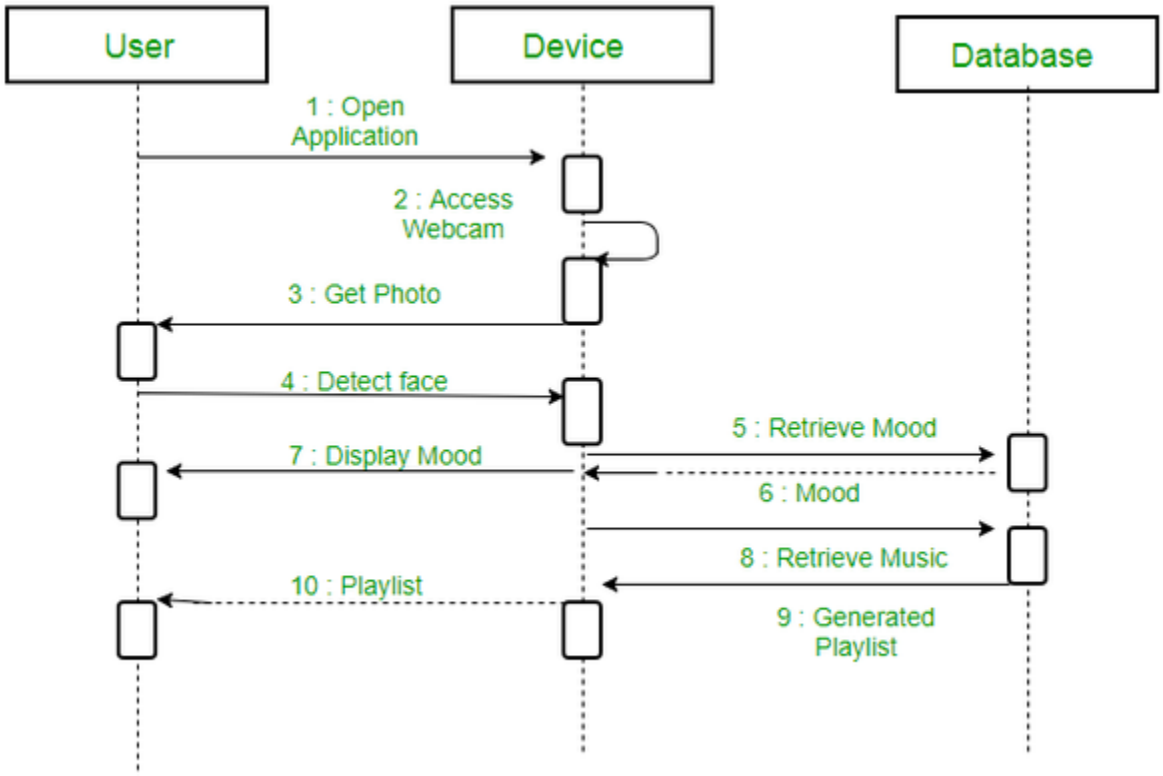


Figure 5.3 Sequence diagram

5.4 Activity Diagram

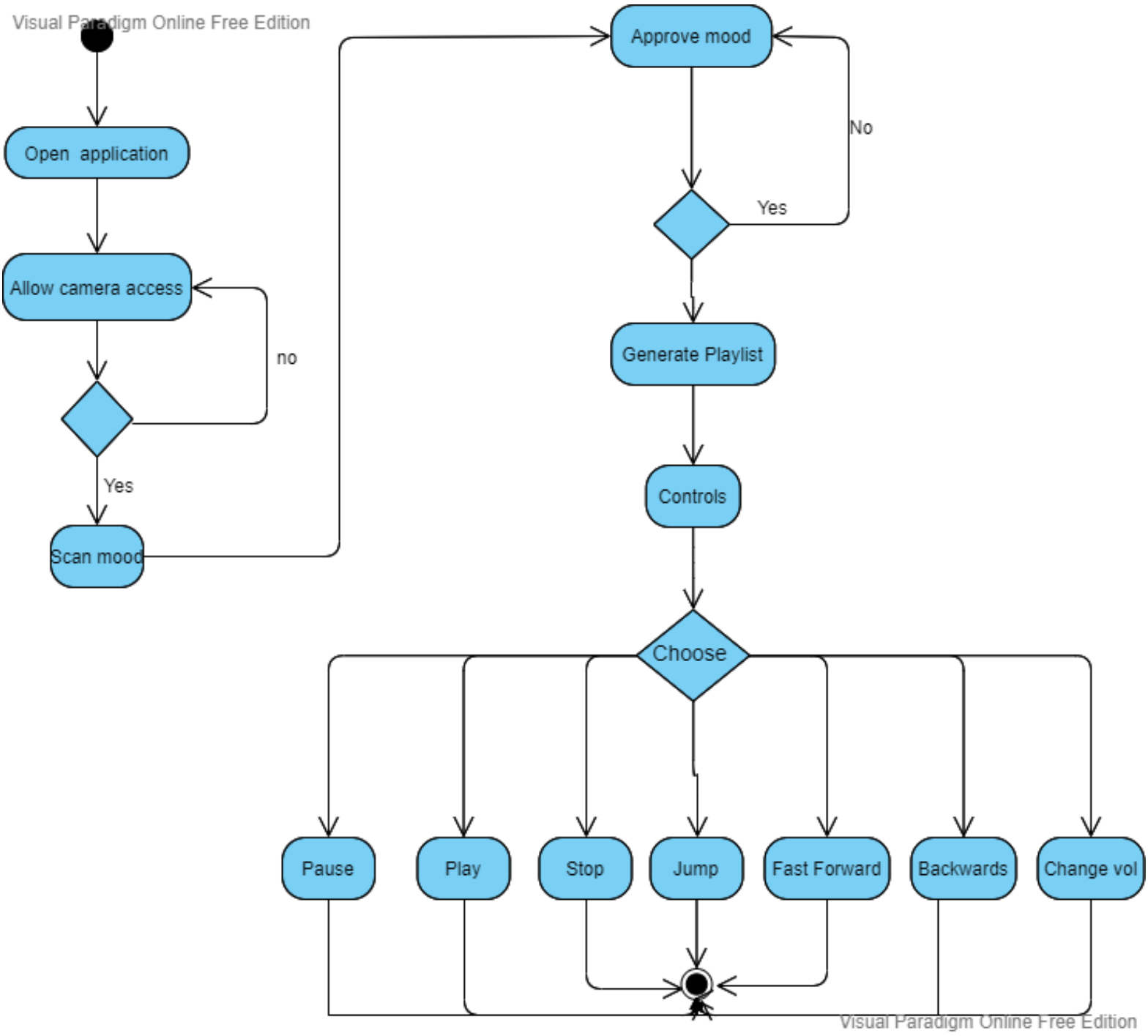


Figure 5.4 Activity Diagram

5.5 State Diagram

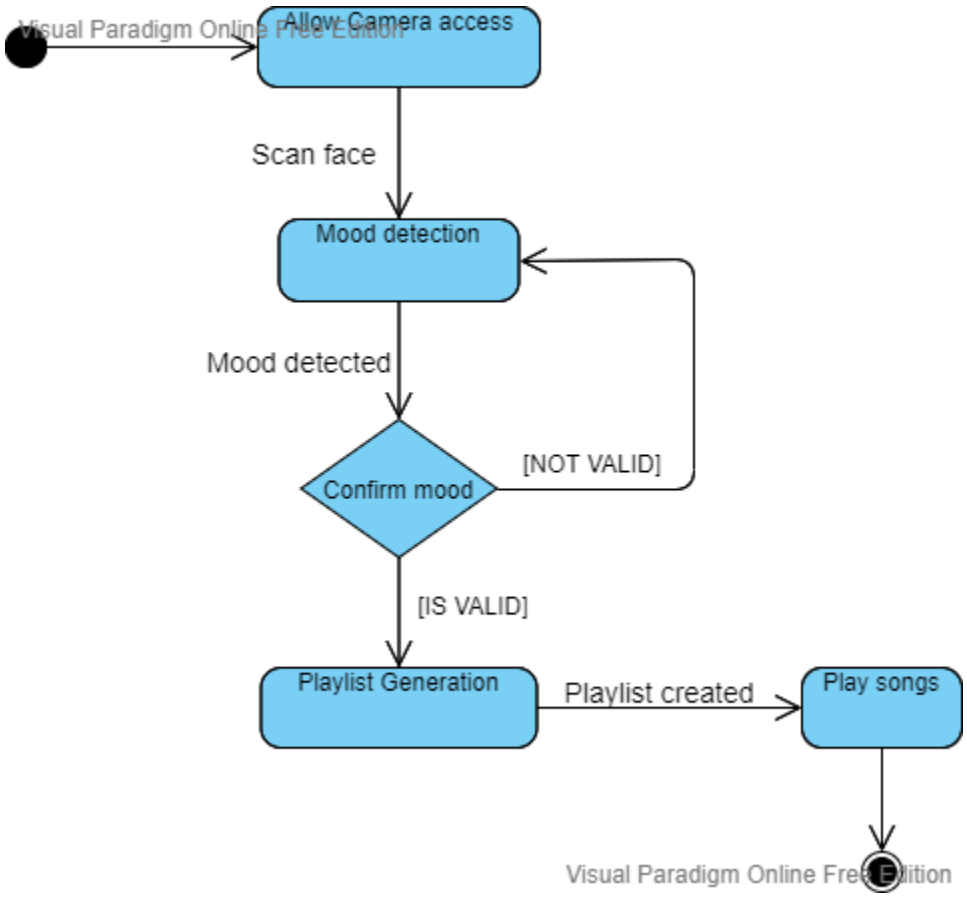


Figure 5.5 State Diagram

6. IMPLEMENTATION PLANNING

6.1 Implement environment

For implementation our project we have used:

1. VS Code(Code Editor)
2. Tkinter(for GUI)

6.2 Programs/Modules specification

Our application is based on MVC architecture so we have created different repository which provides different methods to access data.The controller is responsible for responding to the user input and perform interactions on the data model objects.The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.And pass it to the view which renders and update the user's view.In this way model,view and classes together form as system which separates all three of them.

6.3 Coding standards

To make the system coding easy, easy to remember and reducing the chances of errors some techniques are used at the time of coding of the application which is called coding standard.The coding standard which we adopted during the coding is explained as follow

Some examples are given below:

- Each nested block should be properly indented and spaced.
- The code should be properly commented for understanding easily.
  Comments regarding the statements increase the understandability of the code.
- Better to avoid use of digits in variable names.
- The names of the function should be written in camel case starting with small letters.
- The name of the function must describe the reason for using the function clearly and briefly

7. TESTING

7.1 Testing plan

To test the app there are some basic features that should be tested, start by running the normally without any errors, to check the camera is working properly, and the most important part is to test the emotion detection accuracy functionalities, starting from capturing an image that can be used in the testing and detect the emotion and select the right song.

Here is some steps with figures for the software testing:

1. Run the app

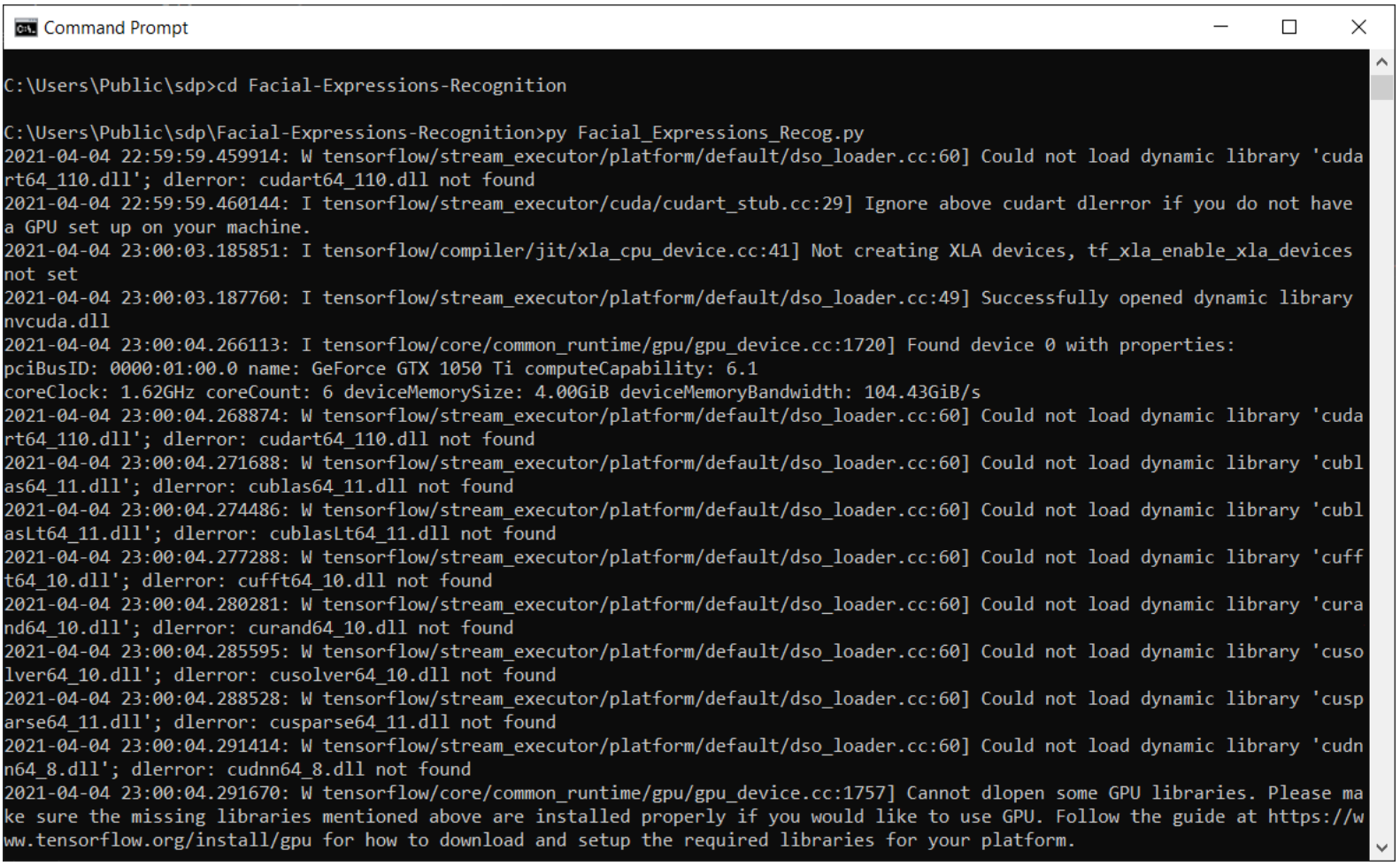


Fig 7.1 File execution

2. **Capture emotion. Once detected press ‘Q’ to exit**

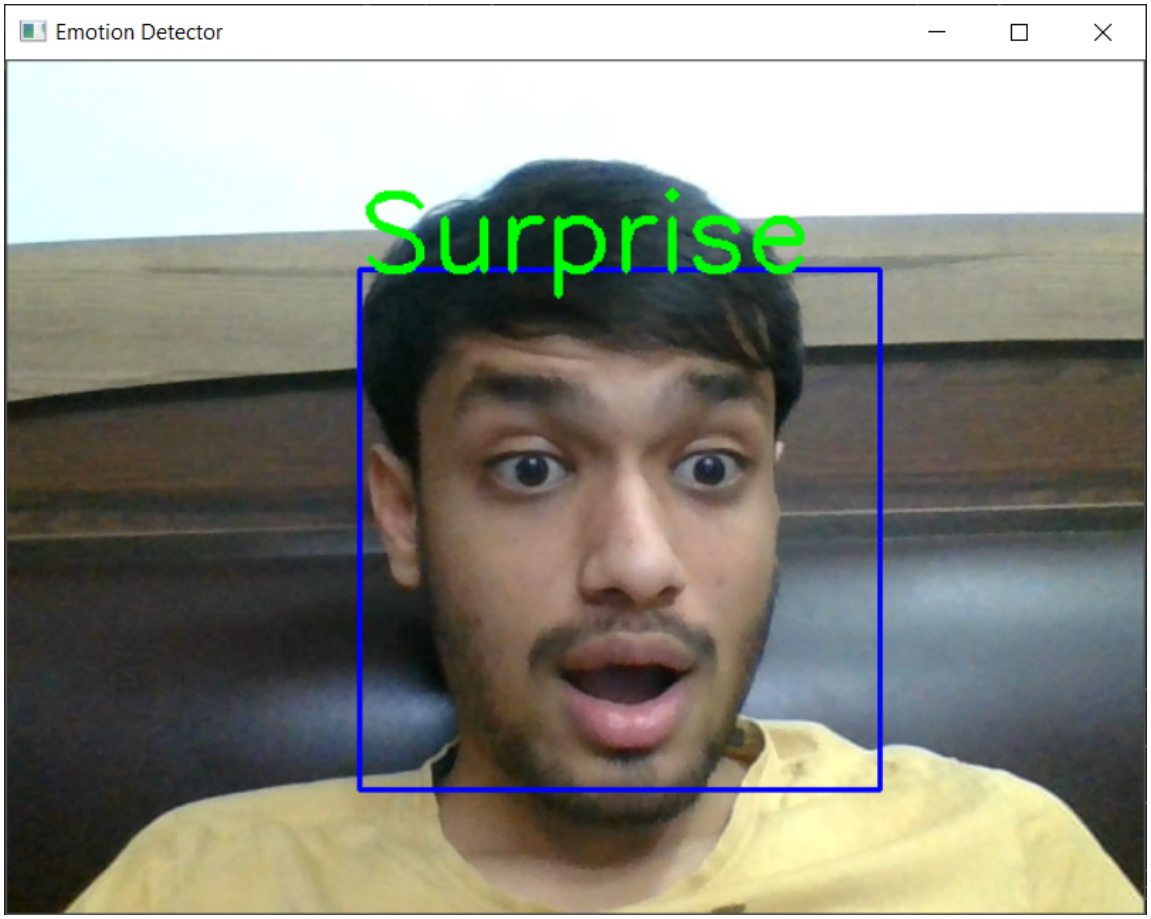


Figure 7.2 Emotion Captured

7.2 Results and Discussions

In the experiment, the images dataset was very huge with almost 27000 images or even more and there are 5 for each emotion. The accuracy rate for the app was 72% which is somehow good but it needs more training to be more accurate. The app is working properly with almost 90% of the running trials, which is also a good result, but when it comes to detecting the face in real time it's working properly.

7.3 Test Cases

Purpose	Input	Test case	Expected output	Actual output	Results
Giving camera access	Click “Yes”	Permission	Open camera	Open camera	PASS
Giving camera access	Click “No”	Permission	Close app.	Close app	PASS
Detecting mood	Human face	Validation	Correct mood	Correct mood	PASS
Confirm mood	Detected Mood	Confirmation	Correct mood	Correct mood	PASS
Confirm mood	Detected Mood	Confirmation	Correct mood	Not correct mood	FAIL
Suggesting song	Confirmed Mood	Search	Display songs according to the mood	Display songs according to the mood	PASS
Add song	New song	Updation	Playlist updated	Playlist updated	PASS
Add song	New song	Updation	Playlist updated	Playlist not updated when local disk is full	FAIL
Delete song	Song name	Updation	Playlist updated	Playlist updated	PASS
Play song	Select song from playlist	Play	Song starts Playing	Song starts Playing	PASS
Pause song	Select song to pause	Pause	Song is paused	Song is paused	PASS
Next/ Previous song	Click “Next” or “Previous”	Song changed	Song change	Song changed	PASS
Jump to specific time	Drag Status bar	Updation	Song plays from desired time	Song plays from desired time	PASS

Table 7.3 Test cases

8. USER MANUAL

User Manuals are manuals that enable the user of a system or application to understand the working of the system and help them to use them efficiently.
It is usually written by a technical writer, although user guides are written by programmers, productor project managers, or other technical staff, particularly in smaller companies.
Our user guides contain both a written guide and the associated images. In the case of our application, it is usual to include screenshots of how the program should look. The language use is dispatched to the intended audience.

8.1 Home(welcome) Page

To use our application, the user needs to first allow web-cam access, so click on “YES”

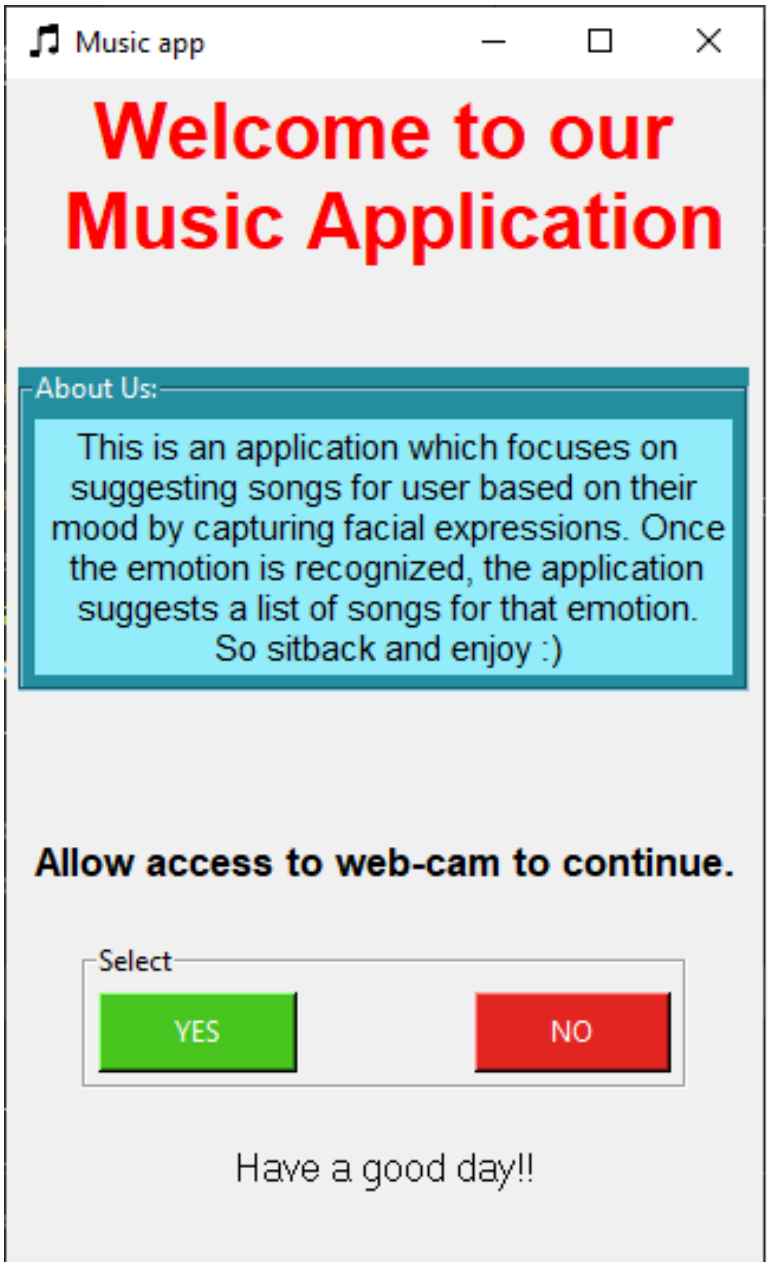


Figure 8.1 Home Page

8.2 [Emotion Detection] will now detect the user's face, scan it.

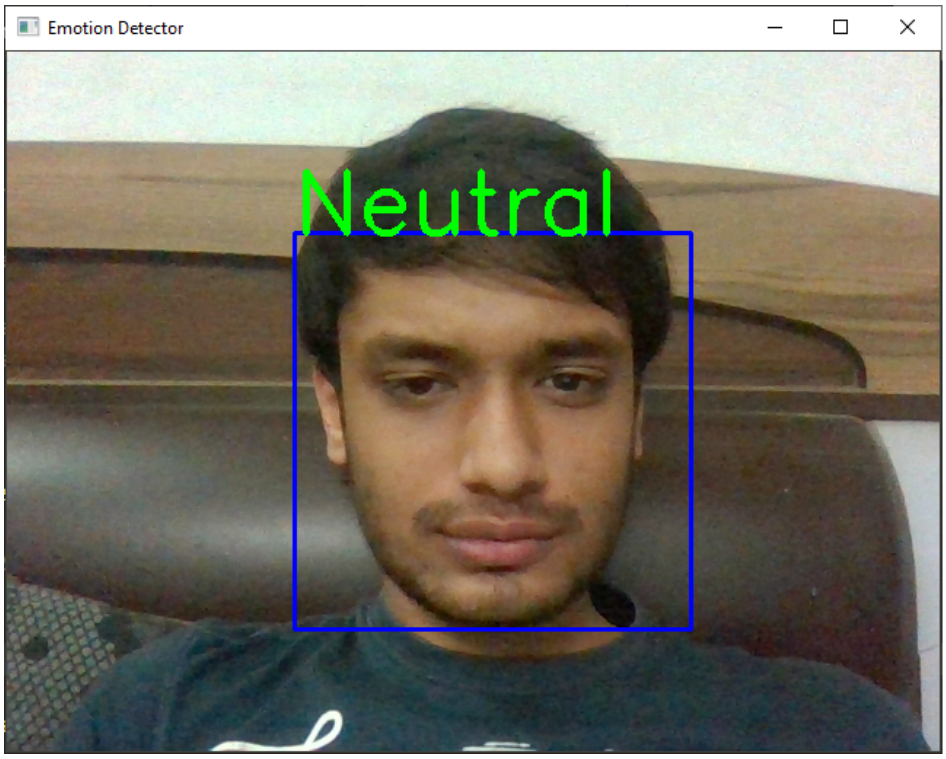
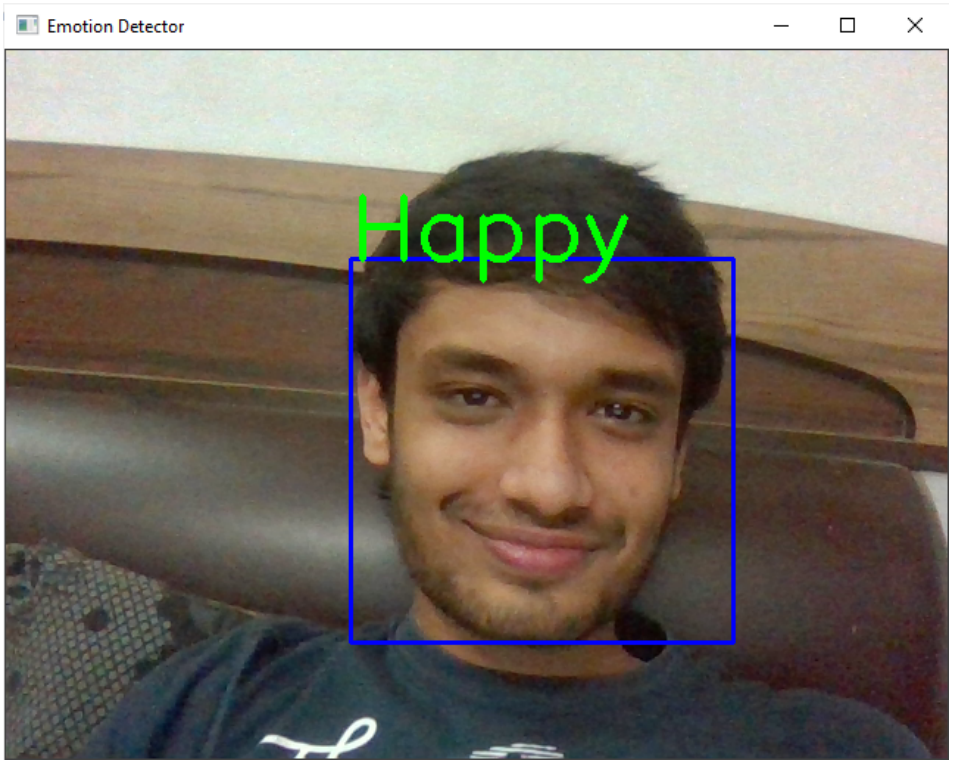


Figure 8.2 Emotion Detection Window



8.3 Now after detecting the mood, the mood is confirmed from the user.

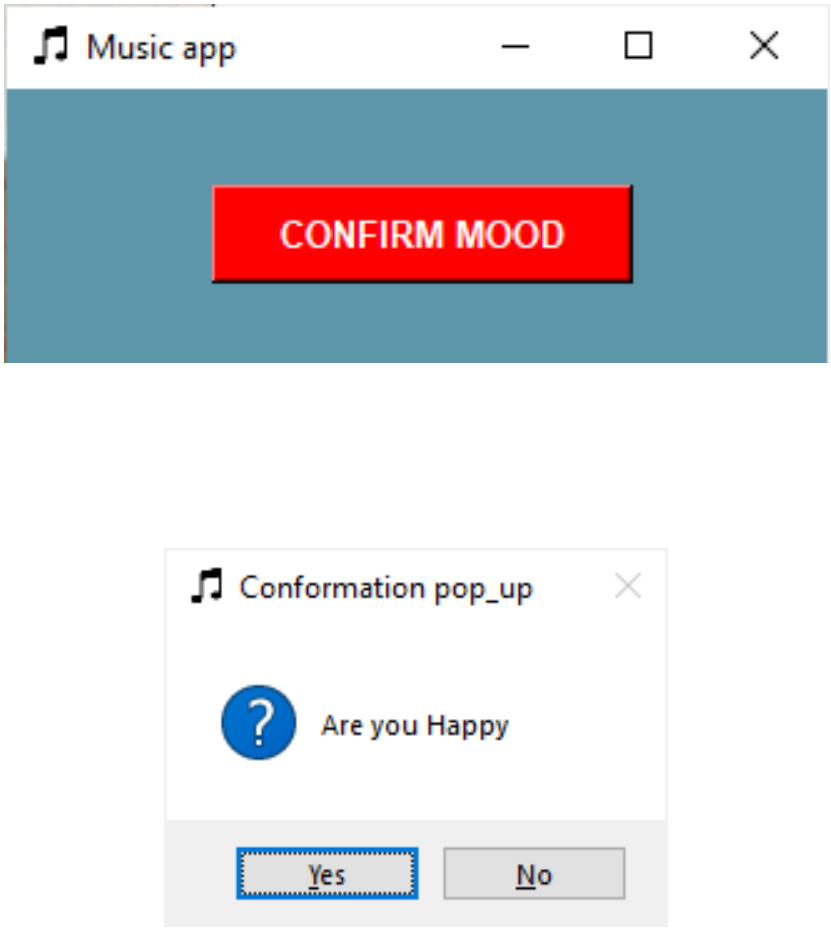


Figure 8.4 Confirmation pop-up.

8.4 Depending on the mood now, this system will suggest appropriate songs to the user. And the user can adjust various controls.

Playing:

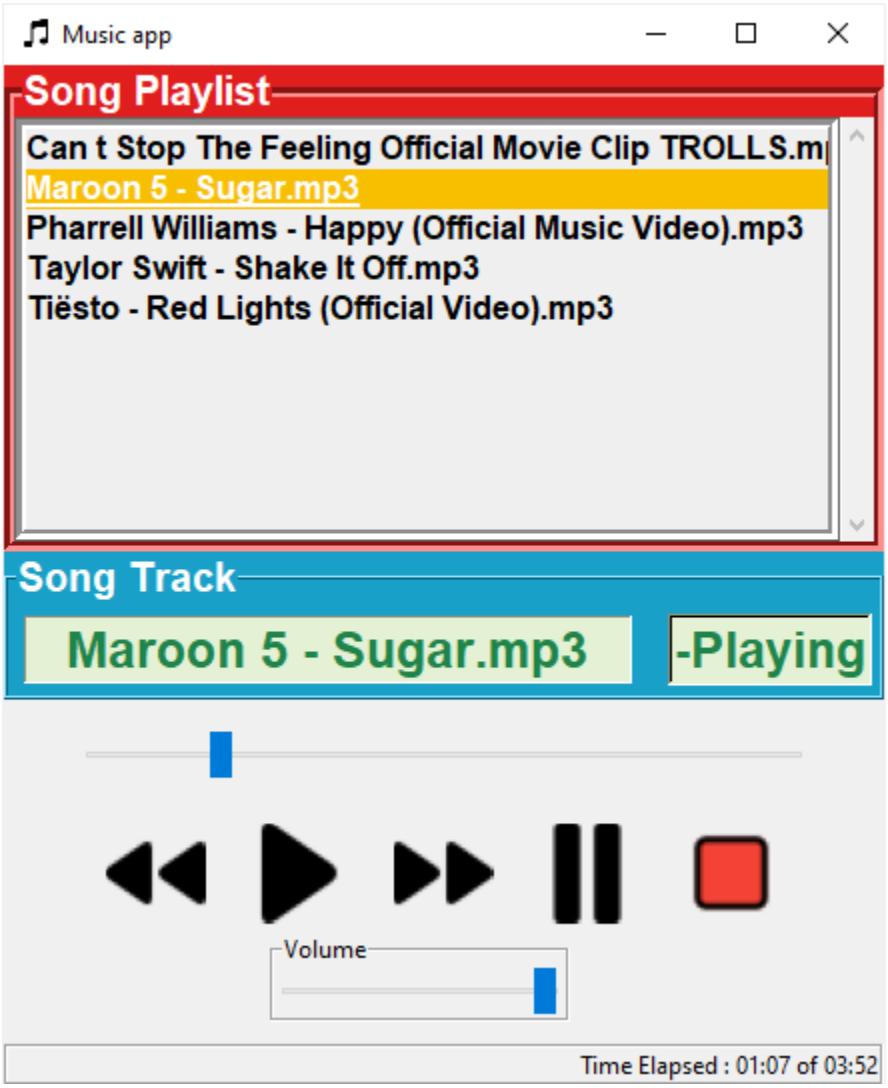


Figure 8.5 Playing

Paused:

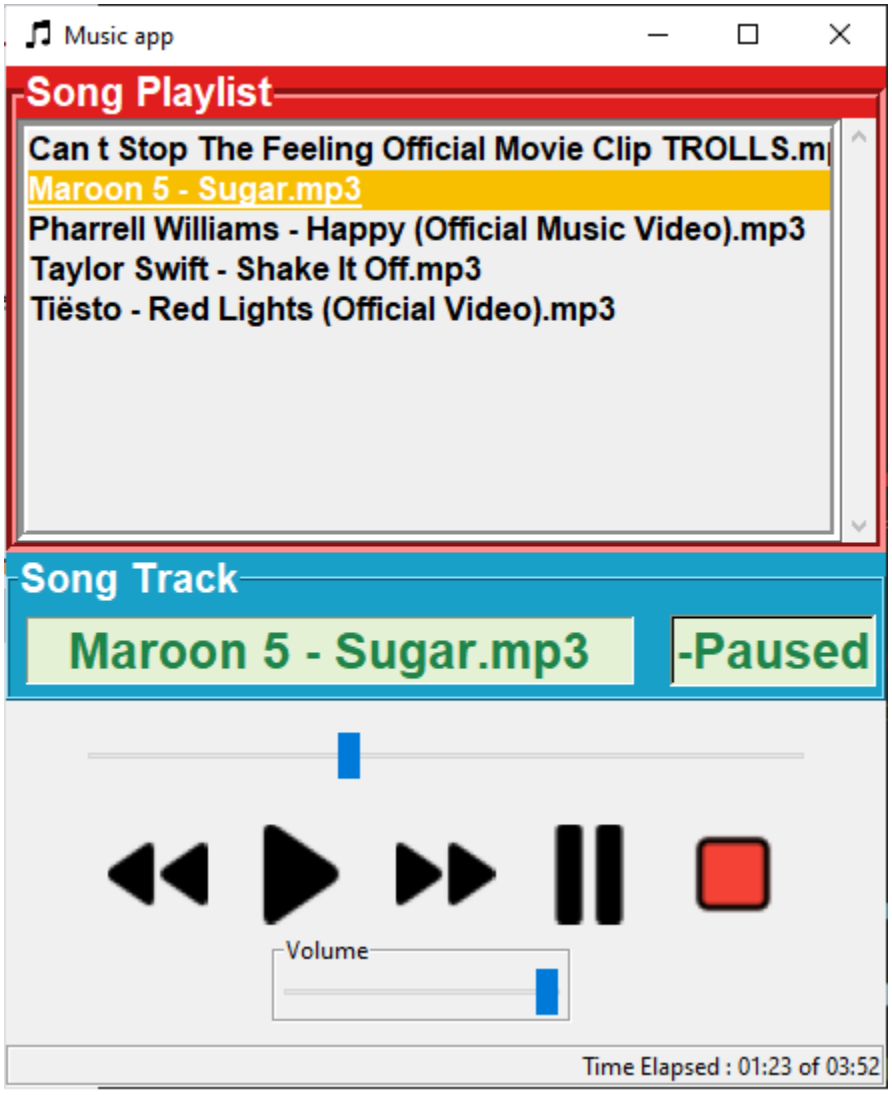


Figure 8.6 Paused

Stopped:

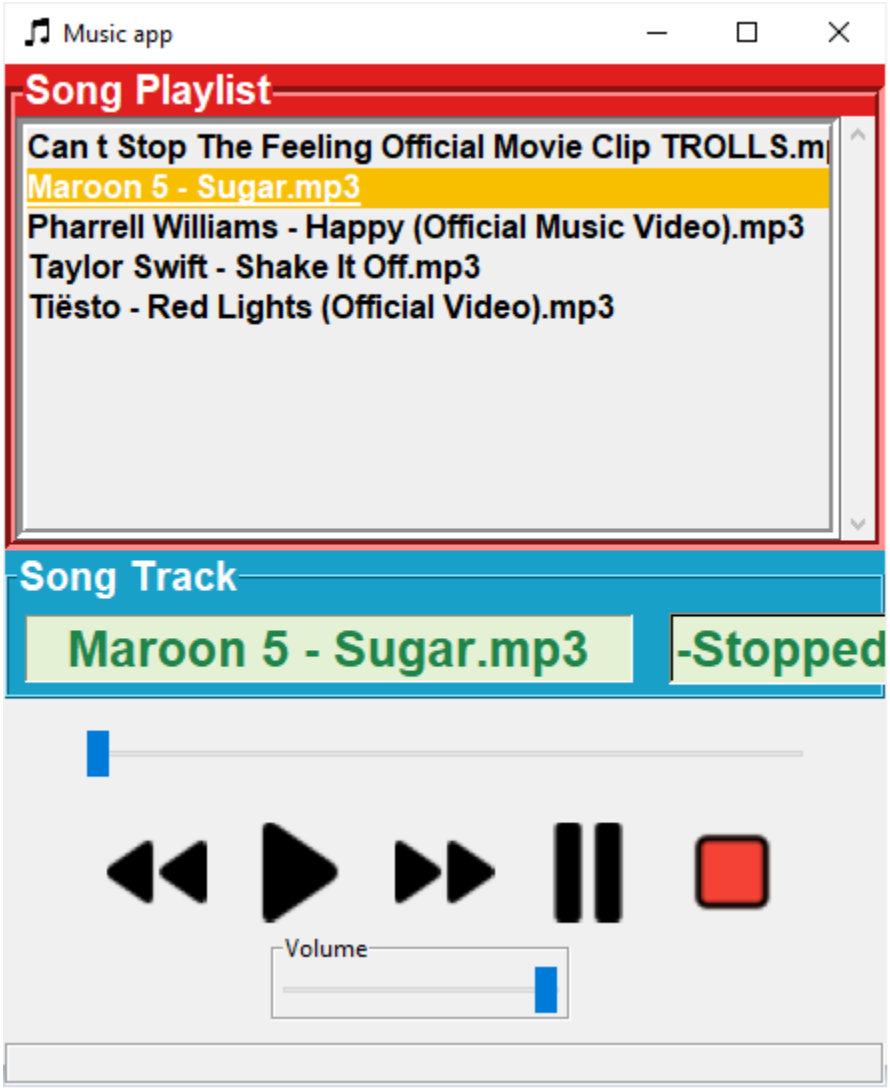


Figure 8.7 Stopped

9. LIMITATION AND FUTURE ENHANCEMENT

9.1 Limitation

- Unable to detect mood in extreme bright light conditions.
- The system comes with a limited amount of songs and the system exe needs to be downloaded before using the system.

9.2 Future Enhancements

- We try to overcome our limitations. The proposed system will also tend to avoid the unpredictable results produced in extreme bad light conditions and very poor camera resolution.
- Users can manually add their own playlist to the application. And with the help of reading the metadata of an MP3 the songs can be segregated accordingly.
- In the future we can expand the scope to adding assistants like google assistant which will hear the needs of the user and show the appropriate results.

10. CONCLUSION AND DISCUSSION

10.1 Conclusion

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.

10.2 DISCUSSION

10.2.1 Self-Analysis of Project Viabilities

According to us, this project is absolutely a good start for gaining hands-on experience on projects. It is useful if it is managed according to the goal for which it is made.

10.2.2 Problems Encountered and Possible Solution

There are a couple of problems encountered during this project.

1. First problem we faced was to gain good accuracy , in our project we used CNN technique and first we kept no of epochs 25 ,that time we were getting 60 % accuracy then we increased no of epochs to 40 and we got 72% accuracy.
2. In the GUI part we faced many problems like, if we paused and then unpaused any song the song would restart instead of continuing from where we paused it, for that we declared “paused” as a global variable and changed its state whenever required.
3. While importing one file into another file just for the use of the emotion which was detected we discovered that the system went into a loop of detecting emotion. For that we created a simple .txt file that contains the emotion detected which could be used further.
4. At low light and low resolution our app is not so accurate, we are working on that.

10.2.3 Summary of Project Work

It is a great achievement to successfully complete the project.The knowledge of software engineering has helped immensely in overcoming the various road blocks.We have done work with pre-planned scheduling related with time constraints and result oriented progress in project development.

11. BIBLIOGRAPHY

- **Reference docs:**
    - **For GUI :**
      https://www.geeksforgeeks.org/python-tkinter-tutorial/
    - **For OpenCV:**
      https://www.tutorialspoint.com/opencv/index.htm
      https://youtube.com/playlist?list=PLQVvvaa0QuDdttJXlLtAJxJetJcqmqlQq
    - **For Deeping Learning:**
      https://youtube.com/playlist?list=PLZoTAELRMXVPkl7oRvzyNnyj1HS4wt2K-

- **Reference sites for debugging:**

  Stackoverflow, Reddit, Github.