Major Project Report

on

EMO PLAYER
an emotion based music player

submitted by

Gurudatt Kumar (12170030)
Kumar Gaurav (12170039)
Manish Kumar Chaudhary (12170041)

In partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

APRIL 2020

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

Certified that this is a bonafide record of the Major Project titled

EMO PLAYER
an emotion based music player

done by
Gurudatt Kumar (12170030)
Kumar Gaurav (12170039)
Manish Kumar Chaudhary(12170041)

of VIIIth Semester, Computer Science and Engineering in the year 2019-20 in partial fulfillment requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology.

Dr.Latha R Nair	**Dr.Sheena Mathew**	**Dr.Sudheep Elayidom**
Head of Division	Batch Coordinator	Project Guide

Acknowledgement

We take this opportunity to thank the supreme being, the source of all knowledge whose blessings are our guiding light in any venture we take up. We're in short of words to express our gratitude to Dr. Sudheep Elayidom, our project guide who guided us and helped us constantly with his inputs and suggestion without which we couldn't have implemented this project the way it is working today. We are highly indebted to Dr.Sheena Mathew and Mrs. Ancy Zachariah, our batch coordinator for their constant supervision and support in completing this project. We also express our heartfelt thanks to Mrs. Minu Poulose who was in charge of the project lab during the semester along with Dr. Sheena Mathew, for helping us by providing all the necessary amenities for completing the project. A bouquet of gratitude to Dr. Latha R Nair, head of the division of Computer Science and Engineering for all kinds of encouragement extended to us.

Thank You,

Gurudatt Kumar(12170030)
Kumar Gaurav(12170039)
Manish Kumar Chaudhary(12170041)

Declaration

We, Mr. Gurudatt Kumar, Mr. Kumar Gaurav, & Mr. Manish Kumar Chaudhary hereby declare that this project is the record of authentic work carried out by us during the academic year 2019 - 2020 and has not been submitted to any other University or Institute towards the award of any degree.

ABSTRACT

Recent studies confirm that humans respond and react to music and that music has a high impact on person's brain activity. The average Indian listens up to two hours of music every day.People tend to listen to music based on their mood and interests. This project focuses on creating an application to suggest songs for user based on their mood by capturing facial expressions. Facial expression is a form of nonverbal communication. Computer vision is an interdisciplinary field that helps convey a high-level understanding of digital images or videos to computers. 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 play-list for that emotion, saving a lot of time for a user over selecting and playing songs manually. Emotion-Based Music Player also keeps track of user's details like number of plays for each song, sorts songs based on category and interest level, and reorganizes the play-list every time. The system also notifies user about the songs that are never played so that they can be deleted or modified.

Contents

List of Figures

Chapter 1

Introduction

Music is an important entertainment medium.With advancement of technology, the optimization of manual work has gained a lot of attention. Currently, there are many traditional music players that require songs to be manually selected and organized. Users have to create and update play-list for each mood, which is time consuming. Some of the music players have advanced features like providing lyrics and recommending similar songs based on the singer or genre .Although some of these features are enjoyable for user, there is room to improve in the field of automation when it comes to music players. Selecting songs automatically and organizing these based on the user's mood gives users a better experience.This can be accomplished through the system reacting to the user's emotion, saving time that would have been spent entering information manually.Emotions can be expressed through gestures, speech, facial expressions, etc. For the system to understand a user's mood, we use facial expression.Using the mobile device's camera, we can capture the user's facial expression.There are many emotion recognition systems which take captured image as input and determine the emotion.For this application, we are using Aectiva SDK for recognition of emotion. The system includes a novel algorithm [EMO-algorithm] that organizes songs based on the user's emotions and preferences.This algorithm suggests user's songs to play based on their emotion.

Chapter 2

Literature Review

Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.
The name machine learning was coined in 1959 by Arthur Samuel. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank, and computer vision.
Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.
Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction;

in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data. Effective machine learning is difficult because finding patterns is hard and often not enough training data are available; as a result, machine-learning programs often fail to deliver.
Our work is based and worked and it is run on python runtime library so we refer
[1] this reference gives python contribution for scientific area, and our player playes the music according to the type of emotion and it is based on support vector machine learning technique so we refer which introduce some concepts of machine learning We refer this to understand the language for data mining and machine learning.[4]This video link gives brief idea for emotion recognition in python.[5]This video link gives out brief idea of facial recognition in python. We need a dataset for our player to detect type of emotion , we need to generate it so we refer for data set generation, for emotion recognition with python and open CV.

Steps in Machine Learning

Machine learning is a field of computer science that gives computers the ability to learn without being programmed explicitly. The power of machine learning is that you can determine how to differentiate using models, rather than using human judgment. The basic steps that lead to machine learning and will teach you how it works are described

1. Gathering data
2. Preparing that data
3. Choosing a model
4. Training
5. Evaluation
6. Hyper parameter tuning
7. Prediction.

- Gathering Data:
  Once you know exactly what you want and the equipment's are in hand, it takes you to the first real step of machine learning- Gathering Data. This step is very crucial as the quality and quantity of data gathered will directly determine how good the predictive model will turn out to be. The data collected is then tabulated and called as Training Data.

- Data Preparation:
  After the training data is gathered, you move on to the next step of machine learning: Data preparation, where the data is loaded into a suitable place and then prepared for use in machine learning training. Here, the data is first put all together and then the order is randomized as the order of data should not affect what is learned.

  This is also a good enough time to do any visualizations of the data, as that will help you see if there are any relevant relationships between the different variables, how you can take their advantage and as well as show you if there are any data imbalances present. Also, the data now has to be split into two parts. The first part that is used in training our model, will be the majority of the dataset and the second will be used for the evaluation of the trained model's performance. The other forms of adjusting and manipulation like normalization, error correction, and more take place at this step.

- Choosing a model:
  The next step that follows in the workflow is choosing a model among the many that researchers and data scientists have created over the years. Make the choice of the right one that should get the job done.

- Training:
  After the before steps are completed, you then move onto what is often considered the bulk of machine learning called training where the data is used to incrementally improve the model's ability to predict.

  The training process involves initializing some random values for say A

and B of our model, predict the output with those values, then compare it with the model's prediction and then adjust the values so that they match the predictions that were made previously.

This process then repeats and each cycle of updating is called one training step.

- Evaluation:
  Once training is complete, you now check if it is good enough using this step. This is where that dataset you set aside earlier comes into play. Evaluation allows the testing of the model against data that has never been seen and used for training and is meant to be representative of how the model might perform when in the real world.

- Parameter Tuning:
  Once the evaluation is over, any further improvement in your training can be possible by tuning the parameters. There were a few parameters that were implicitly assumed when the training was done. Another parameter included is the learning rate that defines how far the line is shifted during each step, based on the information from the previous training step. These values all play a role in the accuracy of the training model, and how long the training will take.

  For models that are more complex, initial conditions play a significant role in the determination of the outcome of training. Differences can be seen depending on whether a model starts off training with values initialized to zeroes versus some distribution of values, which then leads to the question of which distribution is to be used. Since there are many considerations at this phase of training, it's important that you define what makes a model good. These parameters are referred to as Hyper parameters. The adjustment or tuning of these parameters depends on the dataset, model, and the training process. Once you are done with these parameters and are satisfied you can move on to the last step.

- Prediction:
  Machine learning is basically using data to answer questions. So this is the final step where you get to answer few questions. This is the point where the value of machine learning is realized. Here you can Finally use your model to predict the outcome of what you want.

  The above-mentioned steps take you from where you create a model to where you Predict its output and thus acts as a learning path.

Chapter 3

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 predened play-list for the user.

Chapter 4

System Analysis

4.1 Preliminary Analysis

As a music lover, I've always felt that music players should do far more things than just playing songs and allowing users to create play-lists. A music player should be intelligent and act according to user's preferences. A music player should help users organize and play the songs automatically without putting much effort into selection and re-organization of songs. The Emotion-Based Music Player provides a better platform to all the music listeners, and ensures automation of song selection and periodic updating of play-lists. This helps users organize and play songs based on their moods. The player should also give recommendation for users to change songs on-the-go. It calculates song-weight based on EMO-algorithm to help users have more customized and organized play-lists.

4.2 Feasibility Study and Testing

The aim of the system testing process was to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not . Our Project went through two levels of testing

1. Unit Testing
2. Integration Testing

Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit.

Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

4.3 Problem with Existing System

Music listeners have tough time creating and segregating the play-list manually when they have hundreds of songs. It is also difficult to keep track of all the songs.Sometimes songs that are added and never used, wasting a lot of device memory and forcing the user to nd and delete songs manually. Users have to manually select songs every time based on interest and mood. Users also have diculty to re-organize and playing music when play-style varies. Currently in existing application, music is organized using play-list, and play-list songs cannot be modied or altered in one click. Users have to manually change or update each song in their play-list every time. The sequence of songs in a play-list might not be the same every time, and songs that a user wants to listen frequently might not be given priority or might be left out from the list. Currently, there are no applications that allows users to play songs on-the-go without selecting songs manually or from a play-list.

4.4 Proposed System

The Emotion Based Music player requires the user to have a prole to access the application. The user needs to grant permissions for the application to access the device's camera and media. The application allows users to upload songs and give feedback on the song. Emotion-Based Music Player saves the user prole on the device and keeps the prole logged-in until user logs out of the device manually. As soon as the user opens the application, the device's camera opens and begin capturing images. The system will determine emotions and create play-lists for the user based on emotion captured. The application also allows user's to easily customize the play lists. It recommends songs for the user that may t their current emotion, helping the user automate the initial song selection. The recommendations are based on the previous information about the user's preferences and usage.

The Block diagram of proposed system is shown below

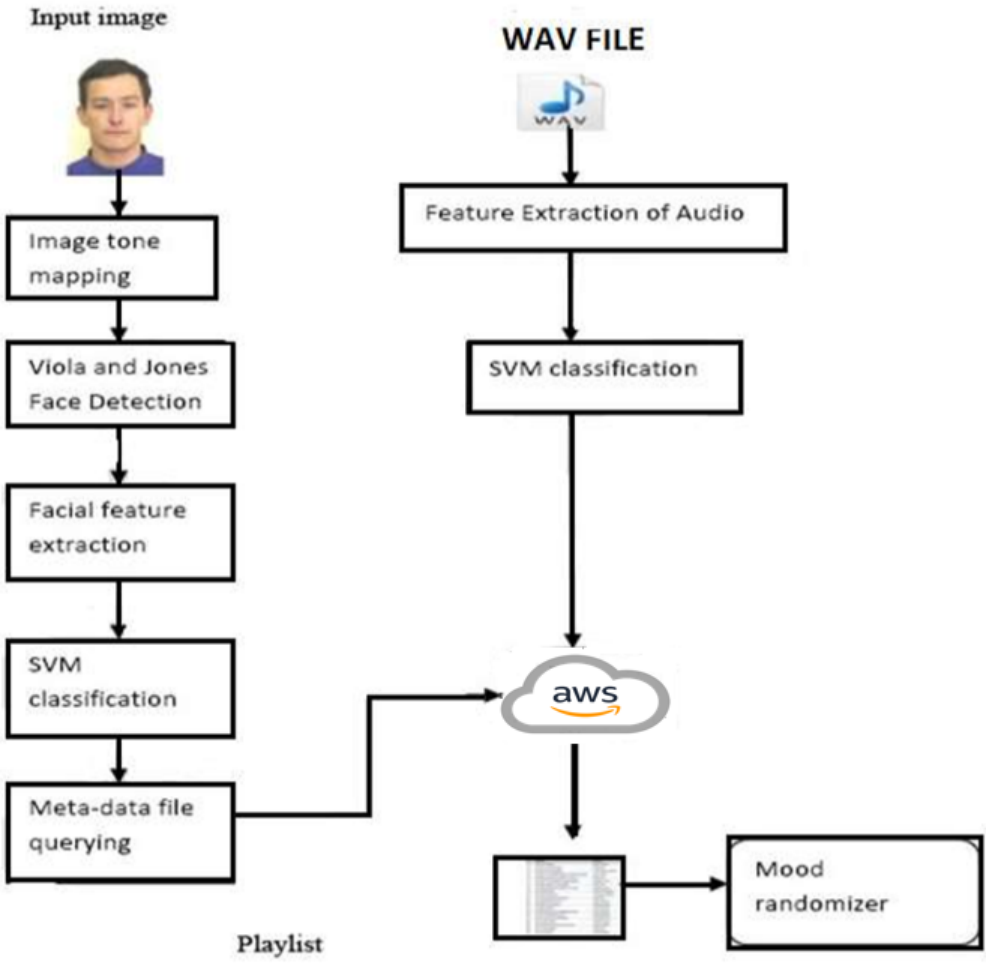

Figure 4.1: Block Diagram of Proposed Algorithm

Chapter 5

System Study

In this section, we are going to present the SRS, system objectives and hardware and software tool requirements.

5.1 Software Requirements Specification

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

Purpose

The main concept of this project is to automatically play songs based on the emotions of the user.
It aims to provide user-preferred music with emotion awareness. In existing system user want to manually select the songs, randomly played songs may not match to the mood of the user, user has to classify the songs into various emotions and then for playing the songs user has to manually select a particular emotion. These difficulties can be avoided by using Emo Player (Emotion based music player). The emotions are recognized using a machine learning method Support Vector Machine (SVM)algorithm.SVM can be used for classification or regression problems.According to the emotion, the music will be played from the predefined directories.

Scope

Emotion Based Music player is a useful application for music listeners with a smart phone and an Internet connection. The application is accessible by anyone who creates a profile on the system. The application is designed to meet the following needs of the users as described below

1. Creating an account or signing up, signing in
2. Adding songs
3. Removing songs
4. Updating songs
5. Personalized play-list
6. Recommendations
7. Capturing emotion using camera

Functional Requirements

The main functionality of this application is to give users more features to automate the selection and organization of the songs based on user preferences.Initially,the user uploads each song and gives details about the song, like category and interest level. Songs are listed and categorized based on user's interest. The Emotion-Based Music Player gives users feedback about their usage. It also notifies users of songs that do not t in the categories listed earlier.

Non Functional Requirements

1. Performance Requirements

   Login should be verified for each user. Test cases corresponding to each problem must be present in the database.

2. Safety Requirements

   The system must not be damaged or manipulated by unauthorized access to the database.

3. Security Requirements

   The user information should not be accessed by other users and the password needs to be encrypted in the database.

5.2 Hardware and Software Requirements

Hardware Requirement

- Processor: 2 GHz
- Browser: Chrome version 50.x and higher
- RAM: 1 GB

Software Requirements

- Cloud Service: AWS S3
- Database: Firebase,NoSQL
- Operating System: Ubuntu, Windows, MAC
- API: Affective Emotion Recognition API

Chapter 6

System Design

6.1 Introduction

Designing requires a careful planning and thinking on the part of the system designer. Designing a system means to plan how the various parts of it are going to achieve the desired goal. After the software requirements have been analysed and specified, design is the first of the three technical activities. Designing, coding and testing are required to build and verify the website.

6.2 Data Flow Diagrams

Data Flow Diagram is a pictorial way of showing the flow of data into/within the system, around the system and out of the system. It is a graphical representation of flow of data within a system. Unlike flowcharts, DFDs do not give detailed descriptions of modules but graphically describe data and how the data interact with the system. The DFD enable us to visualize how the system operates, its final output and the implementation of the system as a whole including modification if any. The purpose of data flow diagram is to provide a semantic bridge between users and system developers.

LEVEL 0 DFD

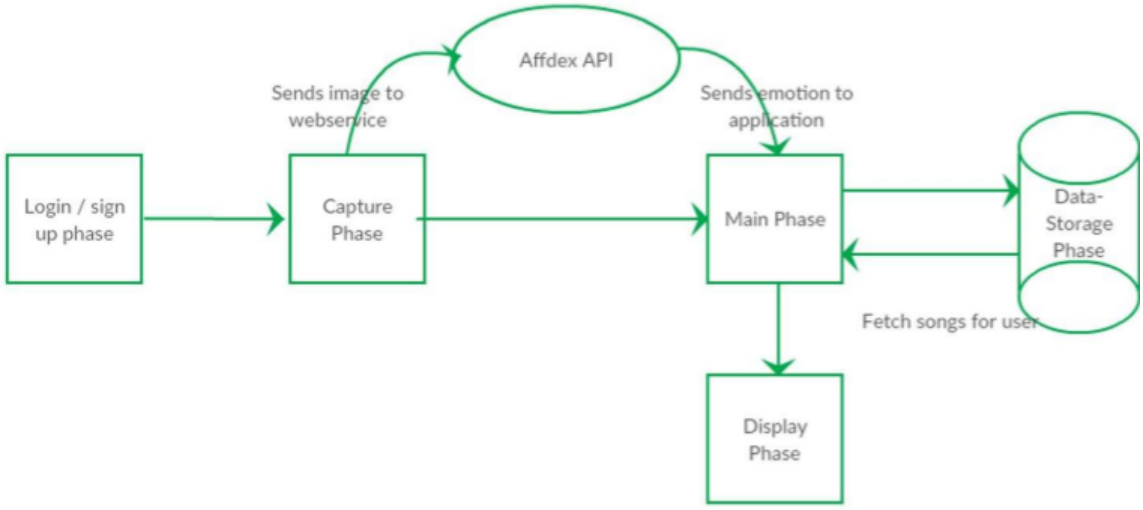

Figure 6.1: Zero Level DFD

Level 1 DFD

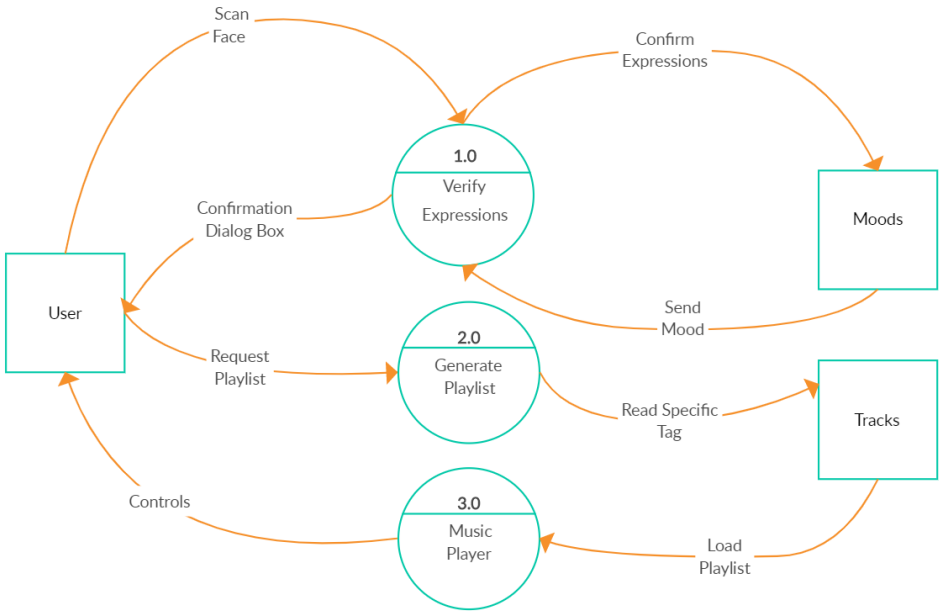

Figure 6.2: Level 1 DFD

6.3 System Architecture

The system architecture of Emotion-Based Music player is shown in Fig below. The application is built using the architectural pattern of Model-View-Controller [13]. It is also widely used architecture. Here, the application is divided into three main logical components: the model, the view and the controller.

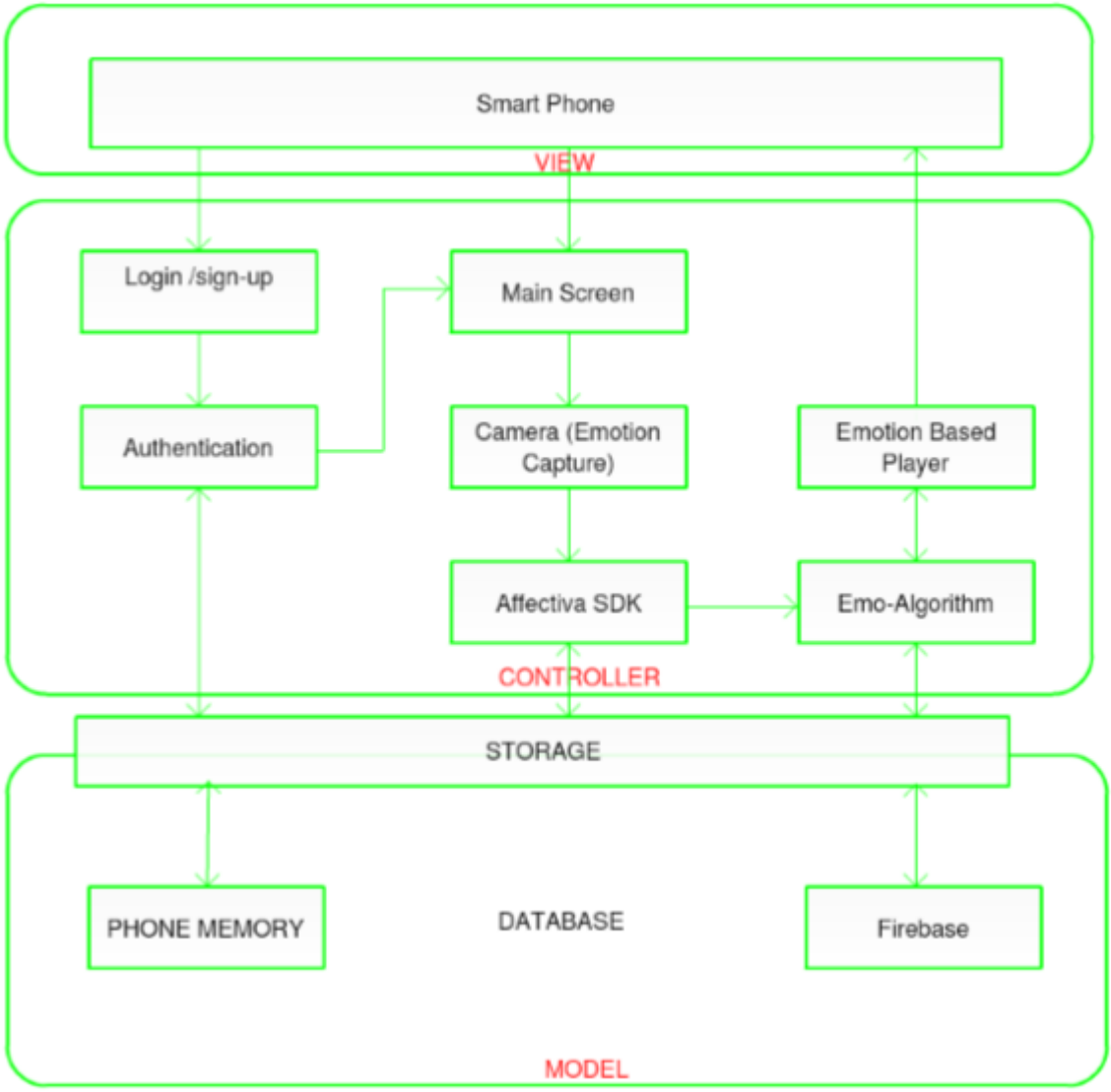

Figure 6.3: System Architecture of Emotion-Based Music Player

- View: The top layer is where the end-user communicates with the application through clicking buttons, typing details, accessing camera, selecting radio but ton, uploading songs, etc. This layer is responsible for displaying all data or a portion of data to user based on the requirement of the application. This layer also acts as a bridge between the user and application itself. Angular 4 [4] is used in this application for displaying the output or response of the system to the user.

- Controller: This middle layer of the application contains the business logic, and the main functionality of the application. As soon as the user interacts with the application, the response is processed in this layer. From log-in to displaying play-list, all the functions that run in background belong to this layer. This mainly consists of all the functions and EMO-algorithm(discussed in later sections) which helps in segregating songs and sending output to view layer.

- Model: This layer is responsible for maintaining the user's data. Emotion-Based Music Player uses Google Firebase [8] for storing user data. Firebase is very useful to maintain user's proles and preferences. The application also stores some temporary data on the device.

Chapter 7

System Implementation

The system uses machine learning model and JavaScript to develop Emotion-Based Music Player, Emo Player. Machine learning model has been used to detect frontal face and detect the emotion while JavaScript has been used to provide User interface and integrate with machine learning model.

The implementation can be broadly classified into three phases:-

1. Face Detection
2. Facial Emotion Detection
3. Integrating Emotion and music layer

7.1 Face Detection

The input image to the system can be captured using a web cam or can be acquired from the hard disk. This image undergoes image enhancement, where tone mapping is applied to images with low contrast to restore the original contrast of the image.The python library opencv2 is used to capture the image through webcam and in image enhancement.On getting image, Haar Cascades model is used to get frontal face.

OpenCV-Python

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.It is a Python wrapper for the original OpenCV C++ implementation.It makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

Haar Cascades

Haar Cascades is an effective approach to do frontal facial detection.We already have its trained model available. We just need to download its xml file and feed image captured by webcam to it.The below piece of code represents the steps needs to do

```
import cv2
video_capture=cv2.VideoCapture(0)
facecascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
def crop(clahe_image, face):
    for (x, y, w, h) in face:
        faceslice=clahe_image[y:y+h, x:x+w]
        faceslice=cv2.resize(faceslice, (48, 48))
        facedict["face%s" %(len(facedict)+1)]=faceslice
    return faceslice
```

Figure 7.1: Haar cascade code

The output of this model will be cropped frontal face image which can be later used by emotion detection model to detect facial emotion.The output of this model will look something like below

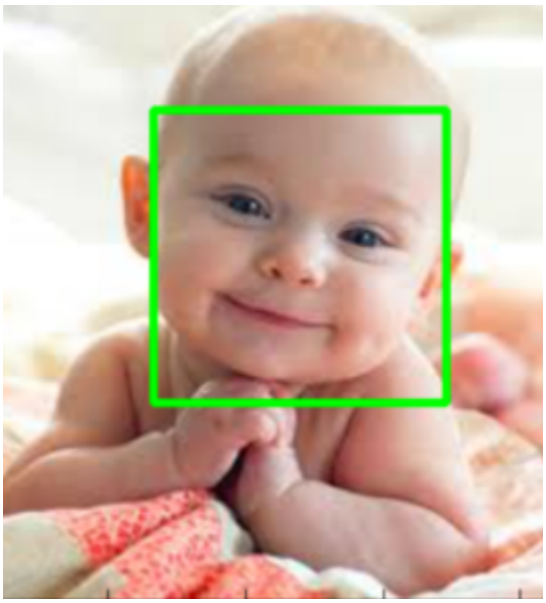

Figure 7.2: Haar cascade output

7.2 Facial Emotion Detection

This is the most important and crucial phase of the system implementation.This phase takes output provided by frontal face detection phase as input and gives output as one emotion

out of seven classified emotions such as angry,happy,sad,neutral,fearful and surprised. Here we have used the child class of OpenCV, FisherFaceRecognizer. The inheritance diagram of FisherFaceRecognizer is shown below:

```
#include <opencv2/face/facerec.hpp>
```

Inheritance diagram for cv::face::FisherFaceRecognizer:

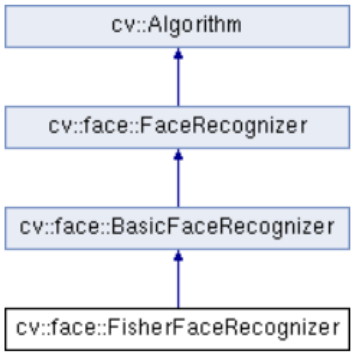

Figure 7.3: Inheritance diagram of FisherFaceRecognizer

Member Function Documentation

retval = cv.face.FisherFaceRecognizer_create([, num_components[, threshold]])

Parameters

- **num_components:** The number of components kept for this Linear Discriminant Analysis with the Fisherface criterion. It's useful to keep all components, that means the number of your classes c (read: subjects, persons you want to recognize). If you leave this at the default (0) or set it to a value less-equal 0 or greater (c-1), it will be set to the correct number (c-1) automatically.
- **threshold:** The threshold applied in the prediction. If the distance to the nearest neighbor is larger than the threshold, this method returns -1.

Notes:

- Training and prediction must be done on grayscale images, use cvtColor to convert between the color spaces.
- The FISHERFACES method makes the assumption that the training and test images are of equal size. (caps-lock, because I got so many mails asking for this). You have to make sure your input data has the correct shape, else a meaningful exception is thrown. Use resize to resize the images.
- This model does not support updating.

Model internal data:

- num_components see FisherFaceRecognizer::create.
- threshold see FisherFaceRecognizer::create.
- eigenvalues The eigenvalues for this Linear Discriminant Analysis (ordered descending).
- eigenvectors The eigenvectors for this Linear Discriminant Analysis (ordered by their eigenvalue).
- mean The sample mean calculated from the training data.
- projections The projections of the training data.
- labels The labels corresponding to the projections.

Model Training

The Emo-player FisherFace model has been trained with the 30,824 data set. The data set has been collected from 2013 february challange on kaggle with the name "Challenges in Representation Learning: Facial Expression Recognition Challenge".The data set contains 30,824 training data set while 2,533 testing data.Training as well as testing data includes seven different groups with names angry,fearful,happy,neutral,sad and surprised.The data set includes facial image of all age group and with different possible expressions.A small glimpse of training data is shown below:

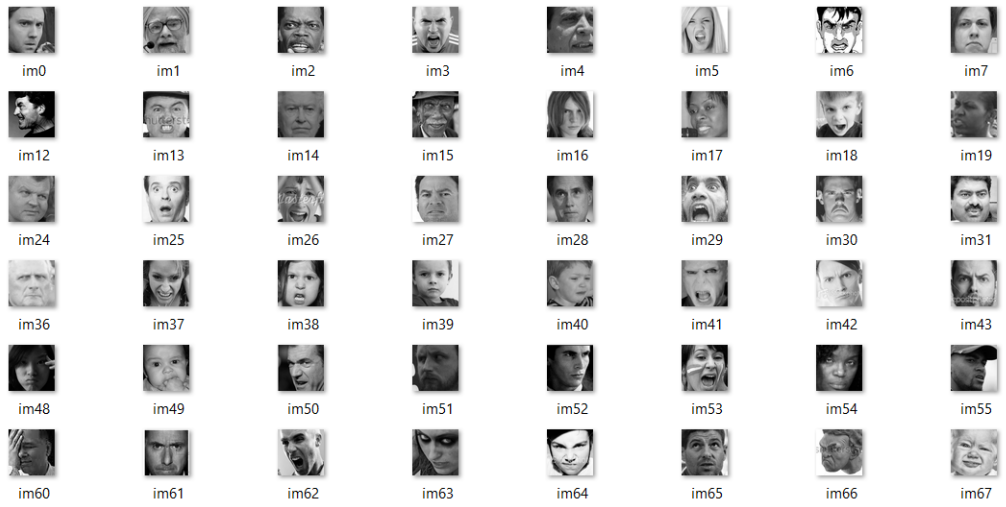

Figure 7.4: Dataset

The training model includes two phases Fetching emotion and Updating model.

Fetching Emotion

The phase includes extracting image from training data set and sorting images based on emotion. This phase mainly includes labelling data set into one out of seven emotion group.This phase has been implemented with below block of code

```
def make_sets(emotions):
    training_data=[]
    training_label=[]

    for emotion in emotions:
        training=sorted(glob.glob("data/train/%s/*" %emotion))
        for item in training:
            gray=cv2.imread(item,0)
            #gray=cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            training_data.append(gray)
            training_label.append(emotions.index(emotion))
    return training_data, training_label
```

Figure 7.5: Fetching Emotion

Updating Model

This phase takes the output of previous phase and save the model having seven facial emotion output as xml file with the help of fishface.save("Model name") as shown below:

```
def update(emotions):
    run_recognizer(emotions)
    print("Saving model...")
    fishface.save("model5.xml")
    print("Model saved!!")
def run_recognizer(emotions):
    training_data, training_label=make_sets(emotions)
    print("Training model...")
    print("The size of the dataset is "+str(len(training_data))+" images")
    fishface.train(training_data, np.asarray(training_label))
```

Figure 7.6: Updating Model Code

7.3 Integrating Emotion and music layer

Java script has been used to integrate the output provided by machine learning model and music player. Based on emotion class detected by the model, java script code fetch song from the specific class of music.There are seven categorised list of song where user can store their song based on their preferences.Within the specific class of song, Java Script code select a

song randomly.

7.4 Application Use case

The Application provides three different modes of operation.Use needs to select one out of three modes manually to work in a specific mode

1.Queue Mode - In this mode user can select a list of songs to be played for a long duration manually from a list of songs available on the server.

2.Random Mode - In this mode Emo player will play the song from server randomly once user access the application and set mode as Random mode.

3.Emotion Mode - In this mode,User needs to be present near webcam to give his/her facial expression to the Emo player.Based on emotion recognised, the system will play a song for you.

7.5 Android User Interface Screenshot

Android Studio and Java has been used to develop the User interface for Android application.

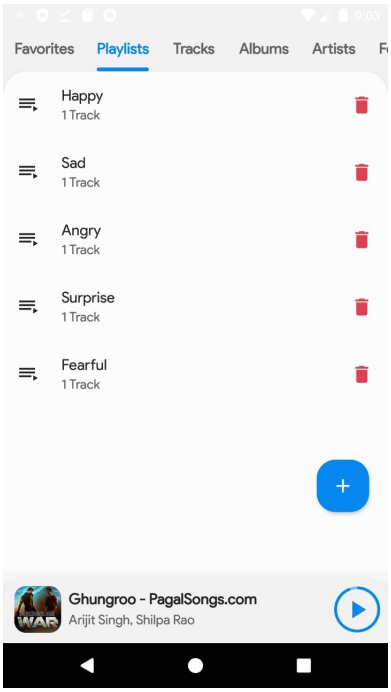

Figure 7.7: Emotion List

Song Track List

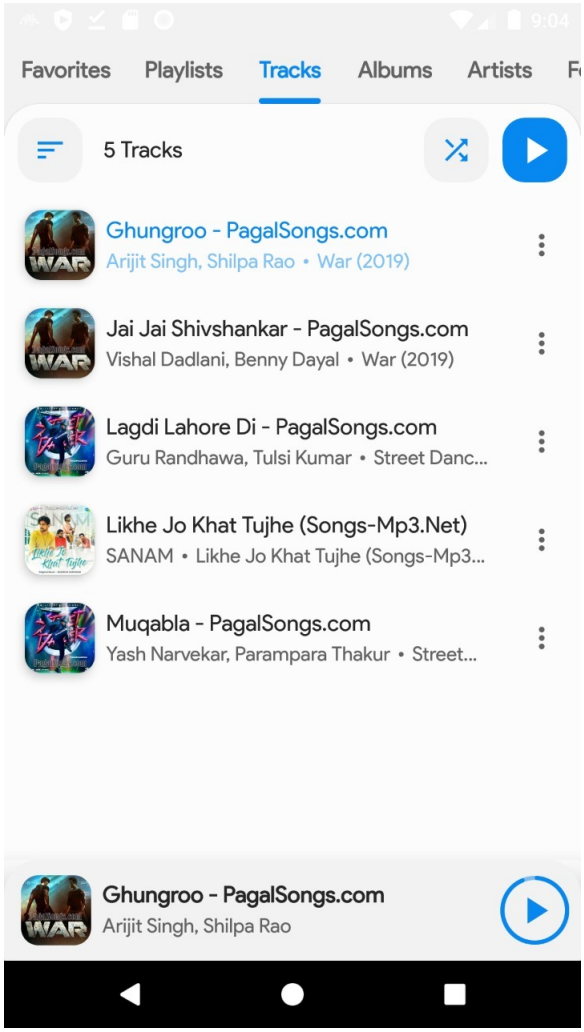

Figure 7.8: Song Track List

Album List

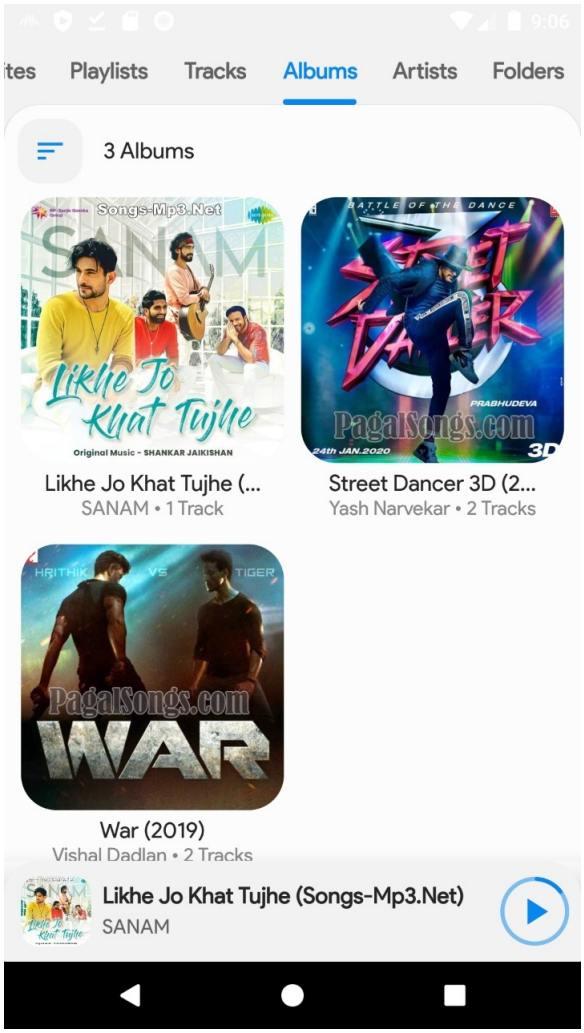

Figure 7.9: Album List

7.6 Web User Interface

Java Script, HTML, Bootstrap has been used to develop the User interface.The User interface of Emo-Player application is shown below:

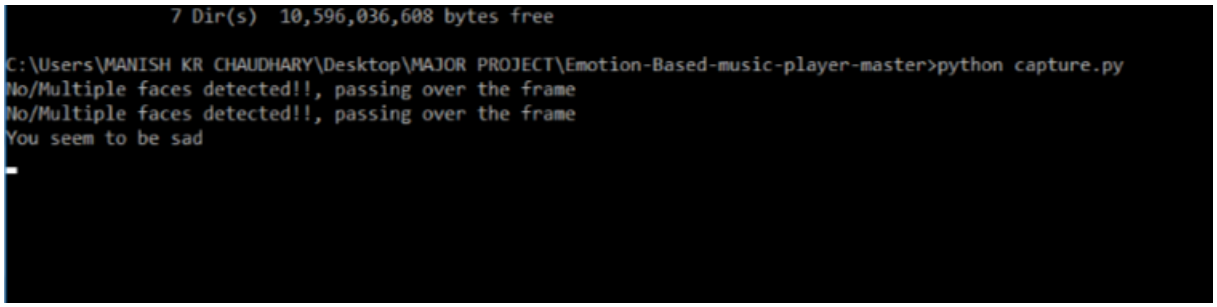

Figure 7.10: Running Model

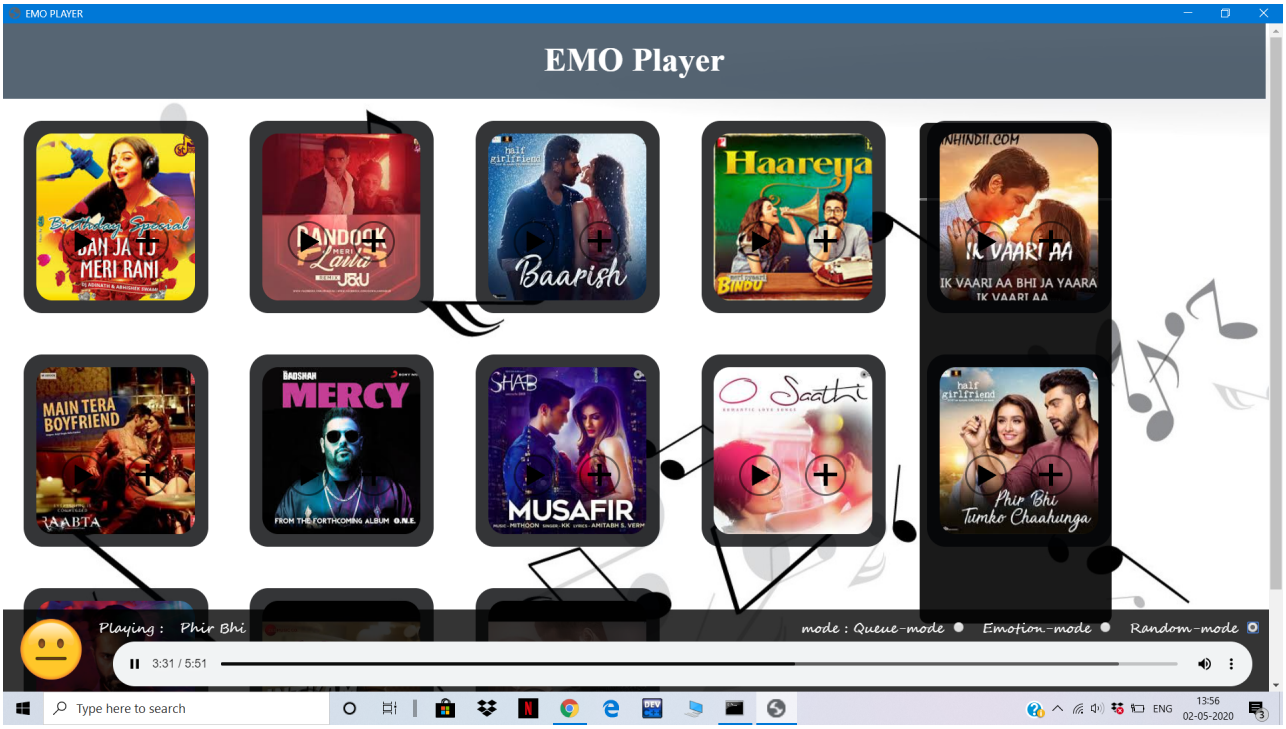

Figure 7.11: User Interface

7.7 Comparison of EmoPlayer with Spotify

Spotify

These application gives good user accessibility features to play songs and recommends user with other songs of similar genre. It gives feature for user to manually select mood and play the music from its playlist. This application doesn't support automatically capture our mood and play the song accordingly.

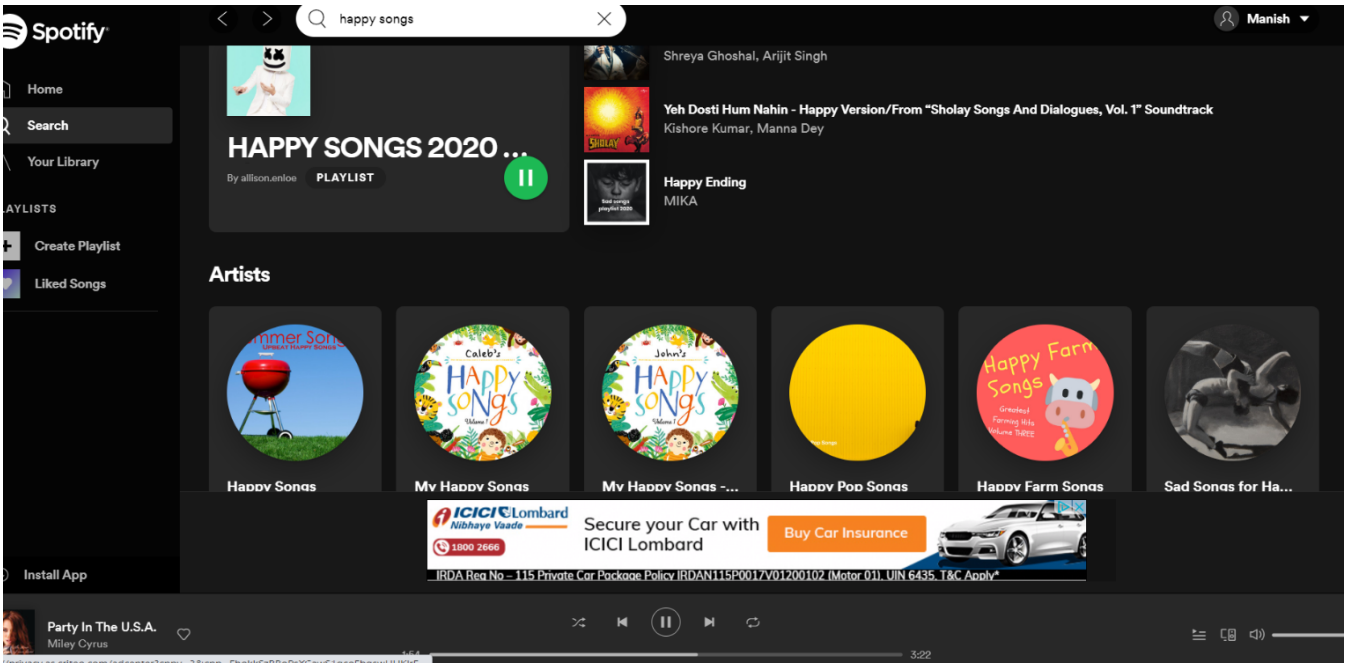

Figure 7.12: Spotify

EmoPlayer

This application have three modes of operation as stated earlier, Queue-mode,Emotion-mode and Random-mode as shown in figure marked by red, yellow and blue circles.The Queue modes enables to create your temporary playlist and play the song in the order you added the song in queue.The Emotion-mode enables to analyse the facial emotion of user with the help of webcam and play the song based on the mood.The Random-mode enables user to randomly play the song from the song database.This application doesn't support the recommendation feature as supported by the Spotify.

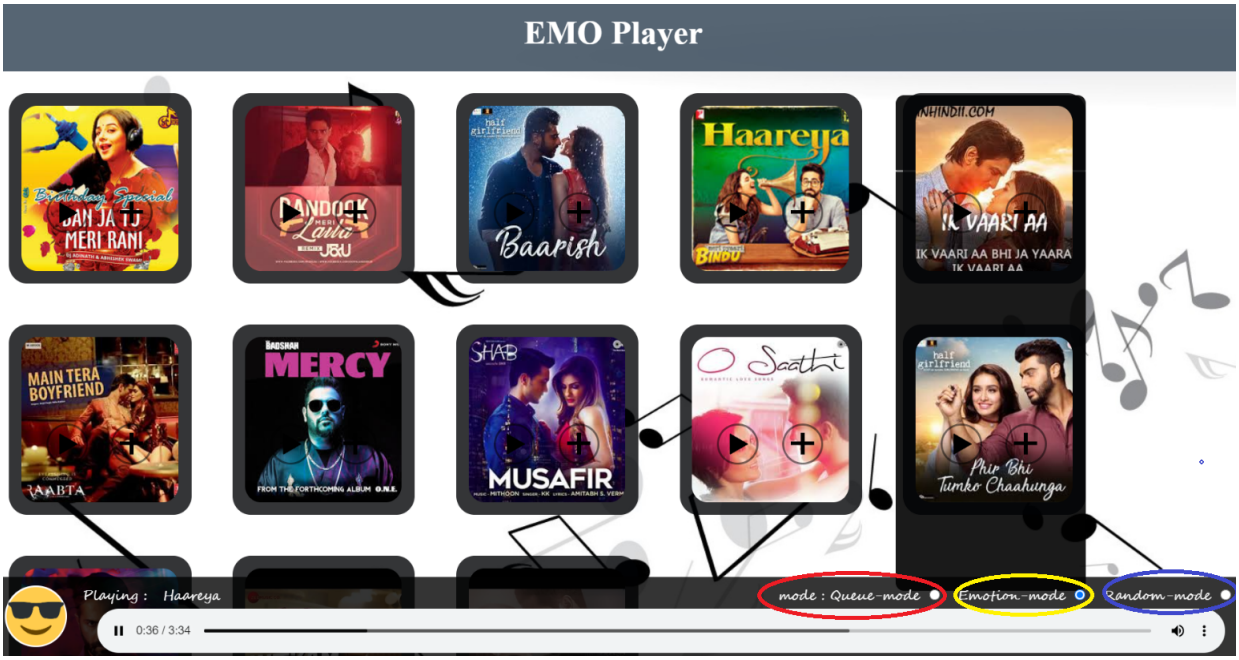

Figure 7.13: Emoplayer

Chapter 8

Future Scope

The application can be improved by modifying and adding few functionality.

• Making the application run without needing an internet connection.

• Can detect sleepy mood while driving.

• Can be used to determine mood of physically challenged mentally challenged people.

• Including other emotions

• Playing songs automatically

• Optimizing the EMO-algorithm by including additional features which helps system to categorize user based on many other factors like location and suggesting the user to travel to that location and play songs accordingly

Chapter 9

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. The user will also be notified of songs that are not being played, to help them free up storage space.

Chapter 10

References

[1] Science about facial expression, 2011. [Online; accessed 11-July-2017].

[2] Average time spent by a person, 2014. [Online; accessed 11-July-2017].

[3] All music, 2017. [Online; accessed 11-July-2017].

[4] Angularjs- framework, 2017. [Online; accessed 11-July-2017].

[5] Cordova - framework, 2017. [Online; accessed 11-July-2017].

[6] Css - style sheet, 2017. [Online; accessed 11-July-2017].

[7] Emotinal recognition - aectiva, 2017. [Online; accessed 11-July-2017].

[8] Firebase- real time database, 2017. [Online; accessed 11-July-2017].

[9] Java script, 2017. [Online; accessed 11-July-2017].

[10] Mean- software bundle, 2017. [Online; accessed 11-July-2017].