A PROJECT REPORT ON

“Emotion Based Music Player”

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF

DIPLOMA IN

Computer Engineering

SUBMITTED TO
MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI

SUBMITTED BY

Vaibhav Yogesh More	1718000022
Durgesh Jagannath Joshi	1718000024
Nikhil Mahadev Mane	1818000013
Athang Chandrakant More	1818000068
Om Sunil Kale	1918000080

GUIDED BY
S.K.Thorat

Gokhale Education Society’s
Sir Dr. M. S. Gosavi Polytechnic Institute, Nashik Road, Nashik

Gokhale Education Society's
Sir Dr. M. S. Gosavi Polytechnic Institute, Nashik Road, Nashik

CERTIFICATE

This is to Certified that the project entitled "**Emotion Based Music Player**" Was successfully completed by Student of sixth semester Diploma in Computer Engineering

Name of students	**Enrollment No.**
Vaibhav Yogesh More	1718000022
Durgesh Jagannath Joshi	1718000024
Nikhil Mahadev Mane	1818000013
Athang Chandrakant More	1818000068
Om Sunil Kale	1918000080

in partial fulfillment of the requirements for the award of the Diploma in Computer Engineering and submitted to the Department of Computer Engineering of Gokhale Education Society's Sir Dr. M. S. Gosavi Polytechnic Institute Nashik work carried out during a period for the academic year 2020-21 as per curriculum .

Name of Guide
Prof. S. K. Thorat

Name of HOD
Prof. S. S. Ajnadkar

External Examiner

Principal

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide Prof. S.K. Thorat for providing their invaluable guidance, comments and suggestions throughout the course of the project, Emotion Based Music Player. I would specially thank Prof. S. S. Ajnadkar of Computer Engineering.

I also thank to Prof. P.P. Jain Principal of Gokhale Education Society's Sir Dr. M.S. Gosavi Polytechnic Institute, Nashik Road, Nashik.

Abstract

The emotion based music player system is computer based software that focus on implementing mood detection.It is a prototype of a new product that merge some separat einter faces:face detection,facial expression Recognition, playlist generation,play music.This project focus on creating an application to suggest songs for user based on their mood by capturing facial expressions.Facil expression is a form of non Verbal communication.This system will take real time input from web-camera which will deter mine user's emotion through facial expressions.Once the emotion is recognized,the system suggests a play-list for that emotion,and will play the music.

There are the different modules in it i.e.

1) Face Detection Module.

2) Emotion Detection Module.

3.Music Player Module

CONTENTS

LIST OF FIGURES

CHAPTER NO 1
INTRODUCTION

Introduction

Smart gadgets people carry everyday with them can obtain lot of data. Data from fitness band, emotion detection from face captured by smartphone, activities detected by the sensors, further can be used for various applications. One of the very precise and straight approach to detect mood is using human facial expressions. Most of the time, emotion is revealed by face itself. Industries have their trained model for the emotion extraction and are now delivering customer services as frameworks providing API with help of cloud based services. We used Microsoft's cognitive servies & Google's own activity recognition API in our approach for faster implementation. Machine learning can be used for classifying music into set of particular emotions. Once, all this data is present, user can be studied about his/her preferences and habits of listening by time, mood, activity, etc. Training on this data can generate better playlist for future listening.Objective behind this work is to let daily factors get considered for better music recom-mendation.

1.1 Scope Of Project:

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. Adding songs
2. Removing songs
3. Updating songs
4. Personalized play-list
5. Capturing face using camera
6. Predetict emotion using camera

1.2 Objectives:

Easy to use
Reduce time.
Provide High Performance system
Provide easy solution to the collage

CHAPTER NO 2
LITERATURE SURVEY

2.1.Literature Survey:

People tend to express their emotions, mainly by their facial expressions. Music has always been known to alter the mood of an individual. Capturing and recognizing the emotion being voiced by a person and displaying appropriate songs matching the one's mood and can increasingly calm the mind of a user and overall end up giving a pleasing effect. The project aims to capture the emotion expressed by a person through facial expressions. A music player is designed to capture human emotion through the web camera interface available on computing systems. The software captures the image of the user and then with the help of image segmentation and image processing techniques extracts features from the face of a target human being and tries to detect the emotion that the person is trying to express. The project aims to lighten the mood of the user, by playing songs that match the requirements of the user by capturing the image of the user. person is trying to express is by facial expression. In some cases, mood alteration may also help in overcoming situations like depression and sadness. With the aid of expression analysis, many health risks can be avoided, and also there can be steps taken that help brings the mood of a user to a better stage.

Currently, there are no dedicated applications to suggest songs based on emotion of music listeners. There are also very few applications that focus on the user preferences and recommendations, and these are not customizable, like AllMusic [1].Other applications suggests predefined (not user-specific)song play-lists. Applicationlike moodfuse [2] include features like manual selection of songs, partly shuffle, play-list. Some popular music applications like Saavn [3] Steremood[4], Spotify provide users definedplay-lists that needs to be created and

updated manually. All of these applications focus on generalcategorization rather than specificity to every user.Many widely used Facial expression categorization techniques like Viola and Jones [5] etc., can be used for initial phase to capture and determine the user's emotion, but these techniques have high computational requirement.

we integrate computer vision and machine learning techniques for connecting facial emotion for music recommendation. For experimental results we used Visual Studio code tool for coding.In this we consider real face input of human using webcam, then image processing techniques are performed on the input acquired image. The features from the input images are extracted using point detection algorithm. The classification algorithm OpenCV is used for training the input images for facial emotion detection. Based on the emotions detected music would be automatically played from the coding folder.

2.2 History:

1.**From ancient live performance to phonograph and records**: **a need to preserve the temporal audio piece**

People's need of enjoying various forms of art started in the very earlier time when people made instruments and did live performances. They also invented notes to communicate with each player as well as preserve the songs. However, the key, rhythm, beats, etc. of the song could be preserved, the performance itself couldn't. For those pure instrumental performance, it might not be a huge problem since people were able to produce many tokens of the original instrument as well as tokens of the original performance so long as they follow the rules recorded on the notes, but what about the singers?At this point, therefore, we see the need of humans to invent a technology, a durable tool that could extend, maintain the skills and thus break the limitation of time and space, they need to maintain the entire succession of accumulated elements for future innovationThe phonograph, to some extent, sees an affordance which is latent in the environment.While phonograph incorporates a desire to preserve the temporal music or broader, sounds, accompanying that desire would be a need to play that audio piece in the future.

2.**Records –> tapes –> Walkman: an increasingly need of portability and convenience.**

If the invention of the phonograph and records are intended to preserve and play music which is still focus on the music per se, the long-term transition from records to MP3, nevertheless, might be more or less deviated. Can we invent a technology that would allow users to carry them so that they could listen to music everywhere? Can we make them smaller, lighter so that users won't be annoyed while carrying them? The idea of new innovation, the transition of an idea to a project, a project to an object, is not only incorporating the people who inhabit it but also they wish to effect: product development team sees the potential of the user's need and imbues such potential into the update of new innovation. On the other hand, user's earlier roles, habits, functions also provide a precondition for such innovatio The music player, therefore, is becoming smaller and smaller and Sony reached one of its annex by selling the Walkman since 1979.

3.**From Walkman to MP3: analog to digital**

Meanwhile, the blossom of digital music allows the musicians to delegate part of the real performance in the recording room to some electrical synthesizing effect the computer is later be programmed to be able to simulate a ton of sounds from different kinds of instrument which gives musicians multiple choices as well as save a lot of money to find the players for each instrument needed in the song But at the same time, modern musicians are required and trained to be more and more adjusted to work with those new electrical technologies and new occupations are created in a macro social level.

4.From MP3, MP4 to Streaming media: not music itself but the market of music player devices is under a revolutionary– mixed affordance: the whole industry:

In the digital age, everything can be integrated and thus be defined as a computer, while MP3 is focusing on digital music which also provides a large potential of storage, why don't we add some new that as

well? The iPod touch is later known for offering anything but communication that a smartphone does.

But as the price is becoming higher and higher as the new music player is becoming more and more function diversified, why not just directly integrate the music listening function onto the smartphone? It's highly integrated, convenient and much cheaper than purchasing an external device. Such need thus accelerates a whole revolution in the streaming media where Apple music, Spotify, Pandora, etc. came out almost overnight and everyone starts to download those application on their phones to listen to music. While these applications substitute the previous music players, they are competing with each other as well as collaborating with the smartphone industry such as Apple store. An economic mediation through time and space is thus emerging as well

2.3 Proposed 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 find and delete songs manually. User's have to manually select songs every time based on interest and mood. User's also have difficulty 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 modified or altered in one click. User's 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.

To design and implement a real time music player which plays music according to the emotion of the user, using convolutional neural network.

2.4 Need Of Proposed System :

To provide a solution for the substantial method of using a music player by incorporating emotion detection.to learn in detail compare emotion detection using different deep learning techniques. To design and implement a reliable CNN classifier to distinguish between different facial expressions. To identify the conditions under which the realization of an application for emotion detection can lead to improvements in subjective and/or objective measures of system usability

CHAPTER NO 3
DEFINE PEOBLEM

3.1 Problem Definition:

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

3.2 Technology Description

Python:

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its no table use of significant indentation. Its language constructs as well as its object oriented approach aim to help programmers write clear, logical code for small and large-scale projects.Python dynamically-typed and garbage collected. It supports multiple programming paradigms, including structuredPython is an interpreted high-level general-purpose programminglanguage. Python's design philosophy emphasizes code readability with its no table use of significant indentation. Its language constructs as well as its object oriented approach aim to help programmers write clear, logical code for small and large-scale projects.Python dynamically-typed and garbage collected. It supports multiple programming paradigms, including structured (particularly,procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 Python consistently ranks as one of the most popular programming languagesRather than having all of its functionality built into its core, Python was designed to be highly extensible (with modules). This compact modularity has made it particularly popular as a means of

adding modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one— and preferably only one —obvious way to do it" design philosophy.Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity.When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C,or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.(particularly,procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 Python consistently ranks as one of the most popular programming languagesRather than having all of its functionality built into its core, Python was designed to be highly extensible (with modules). This compact modularity has made it particularly popular as a means of adding modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one— and preferably only one —obvious way to do it" design philosophy.Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity.When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C,or

use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

2.OpenCv:

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez . The library is cross- platform and free for use under the open-source Apache 2 License.Starting with 2011, OpenCV features GPU acceleration for real-time operations.Officially launched in 1999 the OpenCV project was initially an Intel Research initiative toadvance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls.[4] The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described[5] as:Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months[6] and development is now done by an independent Russian team supported by commercial corporations.

3.Keras:

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the

TensorFlow library.Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML. As of version 2.4, only TensorFlow is supported. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) and its primary author and maintainer is François Chollet, a Google engineer. Chollet is also the author of the XCeption deep neural network model.Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.Keras allows users to productize deep models on smartphones , on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics processing units and tensor processing units.

CHAPTER NO 4
REQUIREMENT ANALYSYS

REQUIREMENT ANALYSYS

4.1 Hardware Requirements :

- Processor: I5 processor
- Ram: 4GB or more
- Hard disk: 40GB
- External camers if required

4.2 Software Requirements :

- Python version 3.0 or above
- Operating system windows 7 or above
- Visual Studio Code 1.35.0 and above
- Note pad ++ 4.2 or above/Sublime editor

4.3 Functional Requirements :

Sr.No	Scenario and Actors	Activities
1	Run your application Actor - User	User will run the application.
2	Seat infront of camera Actor – User	User will seat infront of camera for a particular time(seconds).for detect the user face
3	Detect the face and emotion and play the relevent song as per there emotion Actor - System	After seating an perticulat time in front of camera system detect the user face and emotion and after detecting the emotion system play the song for the user as per there emotion

4.4 Non Functional Requirements :

1) Extensibility:

- The system must be easily extendible.
- The system must be able to cope with regular changes.

2) Reliability:

- The system must be recoverable quickly.
- The system must have 70% accuracy.
- The system must have 70% precisio.

3)Accuracy:

- System have 70% accuracy.
- We can improve the accuracy by taking lots of more images and taking better classifier

4.5 Existing System:

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

- Saavan and Spotify – These application [3] gives good user accessibility features to play songs and recommends user with other songs of similar genre

- Moodfuse - In this application [2], 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 [4] plays music from YouTube

• Musicovery - This application [1] has High quality songs and comprehensive music recommendations. It also suggest predefined play-list for the user.

4.6 Feasibility Study :

4.6.1 Operational Feasibility:

▪ A venture attainability study is an extensive report that inspects in detail the five edges of investigation of a given undertaking. It additionally contemplates its four Ps, its dangers and POVs,and its limitations (schedule, expenses, and standards of value).

• The objective is to decide if the venture ought to proceed, be upgraded, or else deserted through and through The five casings of investigation are: The edge of definition; the edge of relevant dangers; the edge of possibility; the parametric casing; the edge of predominant and possibility methodologies.

• The four Ps are generally characterized as Plan, Processes, People, and Power. The dangers are viewed as outer to the undertaking (e.g., climate conditions) and are partitioned in eight classes.

• (Plan) budgetary and authoritative (e.g., government structure for a private venture);(Processes) ecological and mechanical; (People) showcasing and sociocultural; and (Power) lawful and political. POVs are Points of Vulnerability: they vary from dangers as in they are inside to the extend and can be controlled or else wiped out.

4.6.2 Technically Feasibility:

• This appraisal depends on a blueprint structure of framework prerequisites, to decideif the organization has the specialized aptitude to deal with consummation of the venture.

- The specialized achievability evaluation is centered around increasing a comprehension of the present specialized assets of the association and their pertinence to the normal needs of the proposed framework.

- It is an assessment of the equipment and programming and how it addresses the issue of the proposed framework

4.6.3 Economically Feasibility:

- Operational practicality is the proportion of how well a proposed framework tackles the issues, and makes the most of the open doors distinguished during extension definition and how it fulfills the prerequisites recognized in the necessities examination period of framework improvement.

- To guarantee achievement, wanted operational results must be granted during plan and advancement. These incorporate such structure subordinate parameters as dependability, viability, supportability, ease of use, producibility, superfluity, maintainability, moderateness and others.These parameters are required to be considered at the beginning times of structurewhenever wanted operational practices are to be figured it out.A framework planand improvement requires proper and opportune use of designing and the executivesendeavors to meet the recently referenced parameters

4.6.4 Schedule Feasibility :

- Schedule Feasibility is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then its schedule feasibility is appraised as high.

CHAPTER NO 5
DESIGN OF SYSTEM

DESIGN OF SYSTEM

5.1 System Architecture:

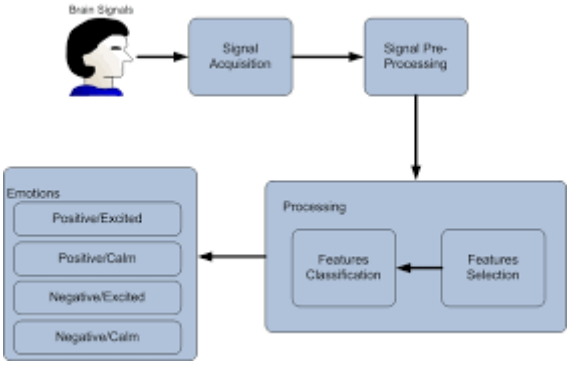


Fig:5.1.1 System Architecture

Detection of face from a given input image or video is facedetection. There are various algorithms for face detection. Viola jones algorithm is used for face detection. The main steps in Viola Jones algorithm are:HAAR feature -

HAAR features represent some characteristics of the face. Haar features are similar to those convolution kernals which are used to detect the presence of the feature in the given image. Each feature result in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle. In the feature, the black region are replaced by plus ones and white region is minus one.

Integral image -

In haar feature calculation, as every time window moves need to sum up all pixels of the black region and those of white region. It is a tedious operation and the solution is integral image. It reduces the computation rather than summing up all pixels under a rectangle with just four corner values of the integral image. To find the value of any pixel just sum the values of pixels to the top and left.

Cascading - It is the basic principle of Viola Jones face detection algorithm to scan the detector many times through that image itself, each time with a new size. Though an image should contain one or more faces it is clear that an excessive large amount of the evaluated sub-windows might still be negatives. He algorithm should hence concentrate on discarding non-faces quickly. Therefore a single strong classifier formed out of linear combination of all the best features is not good to evaluate on each window because of computation cost.

Facial Feature Extraction Module:

For feature extraction, CNN is used. For the emotion recognition module, we have to train the system using datasets containing images of happy, anger, sad and neutral emotions. In order to identify features from dataset images for the model construction, CNN has the special capability of automatic learning. In other words, CNN can learn features by itself. CNN has the ability to develop an internal representation of a two dimensional image. This is represented as a three dimensional matrix and operations are done on this matrix for training and testing. Moreover, in some other neural networks like fully connected networks, all nodes in a layer are connected to every node in the next layer. Associated with each connection, weights are present. This will increase the computational complexity. But in CNN, nodes in a layer are connected to only valid nodes in the next layer. Thus there will be less computational complexity. This includes various layers in order to train and test input images. Final layer is fully connected which have classification task and thus images can be classified according to emotions. The emotions detected should be one these labels: angry, happy, sad, and neutral. Before entering the CNN, entire dataset will be divided into two. 80 percent of this will beused for training and remaining 20 percent will be used for testing

5.2 Data Flow Diagram (Levels)

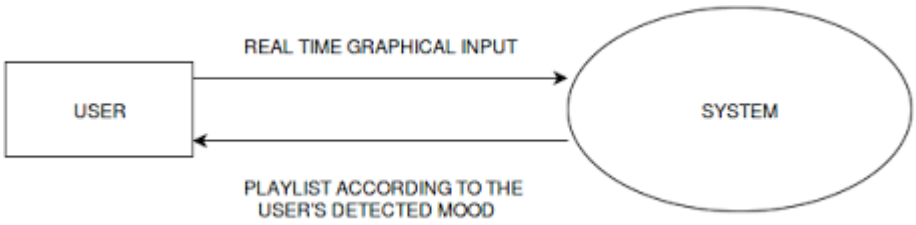


Fig:5.2.1 DFD Level 0

A DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities.

DFD level 1:

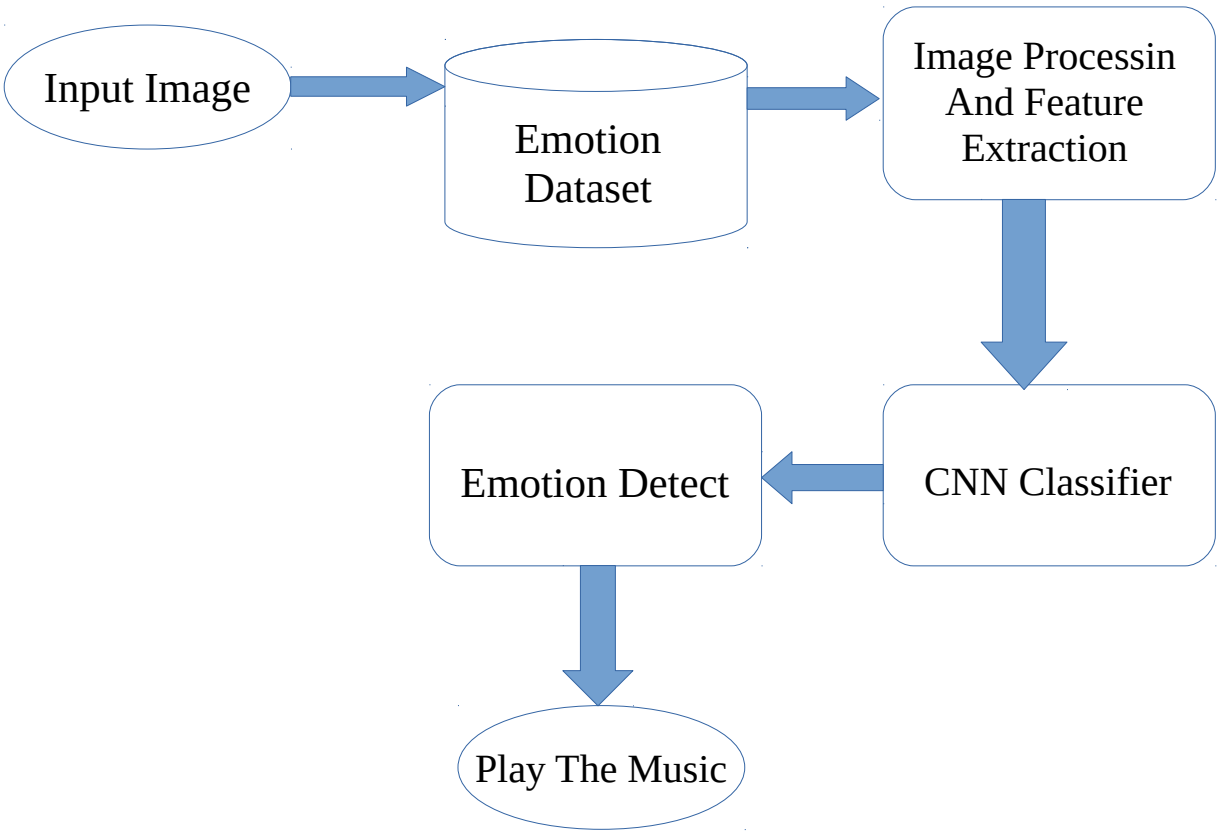


Fig:5.2.2 DFD Level 1

A **level 1 DFD** notates each of the main sub-processes that together form the complete system. We can think of a **level 1 DFD** as an "exploded view" of the context diagram.

5.3 . UML Diagrams :

5.3.1 Class Diagram :

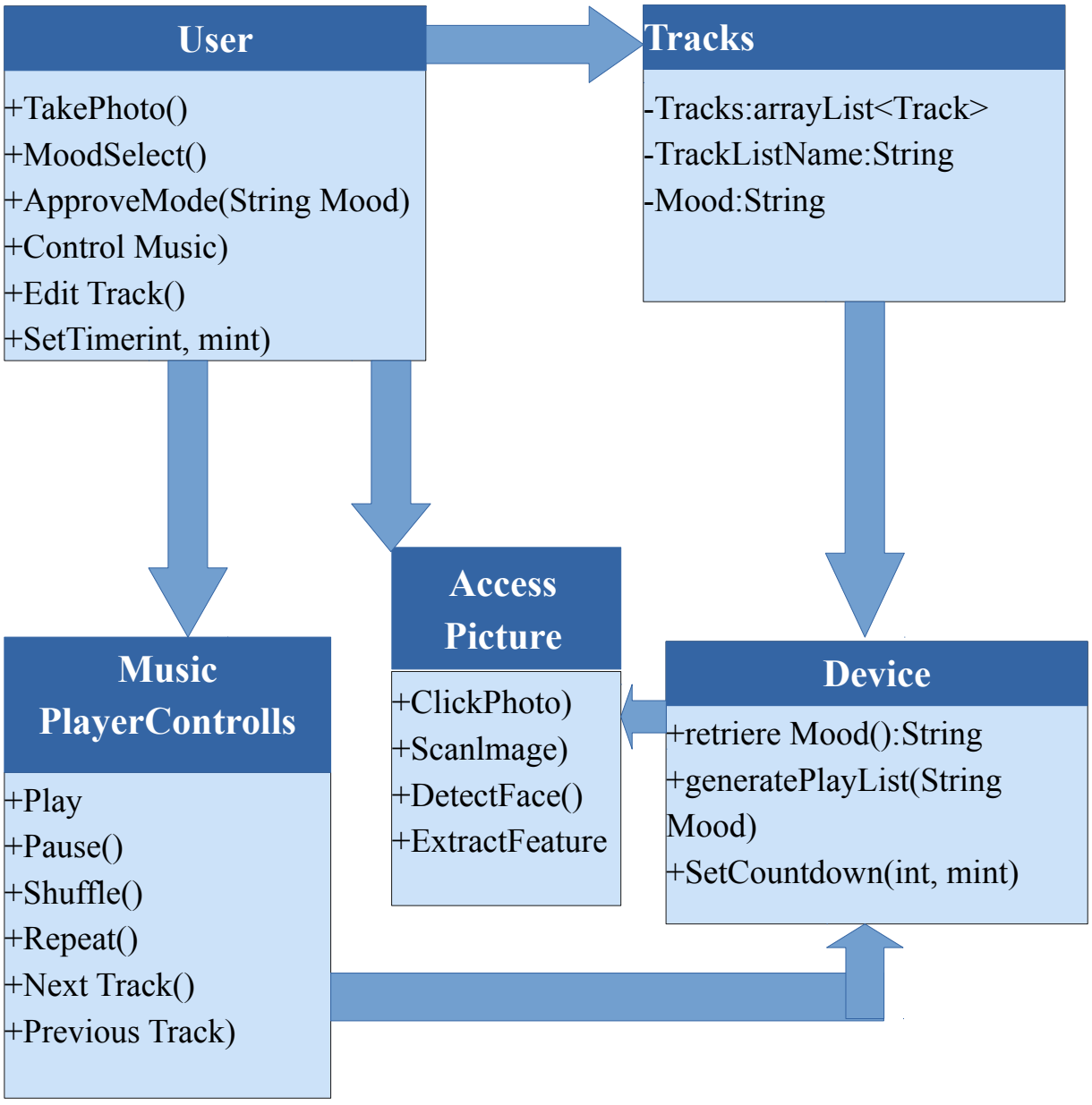


Fig:5.3.1 Class Diagram

A **UML diagram** is a **diagram** based on the **UML** (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

5.3.2 Sequence Diagram:

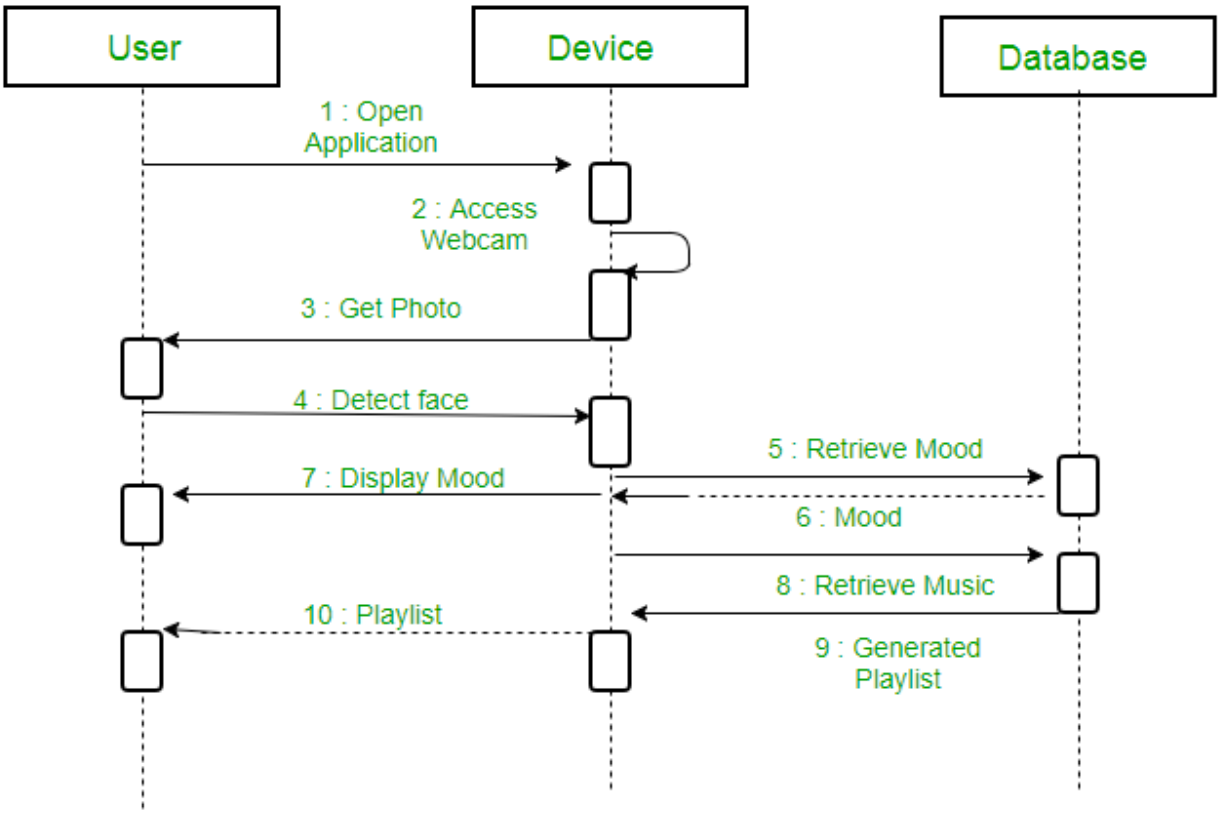


Fig:5.3.2 Sequence Diagram

A sequence diagramshows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines,

different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

5.3.3.Activity Diagram:

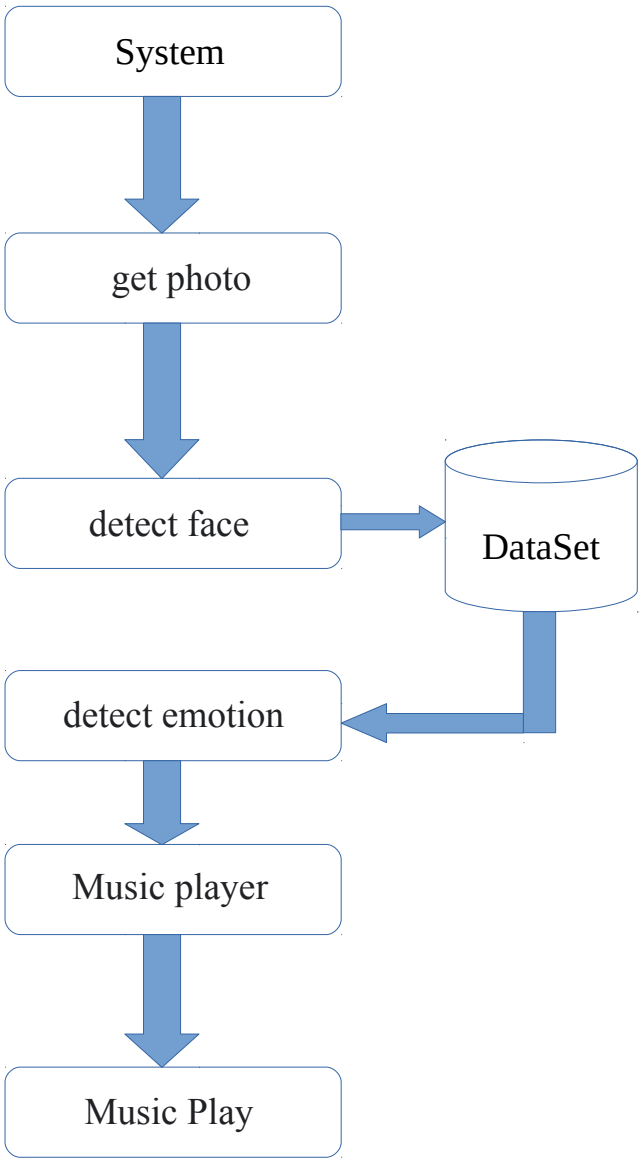


Fig:5.3.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

5.3.4.Usecase Diagram:

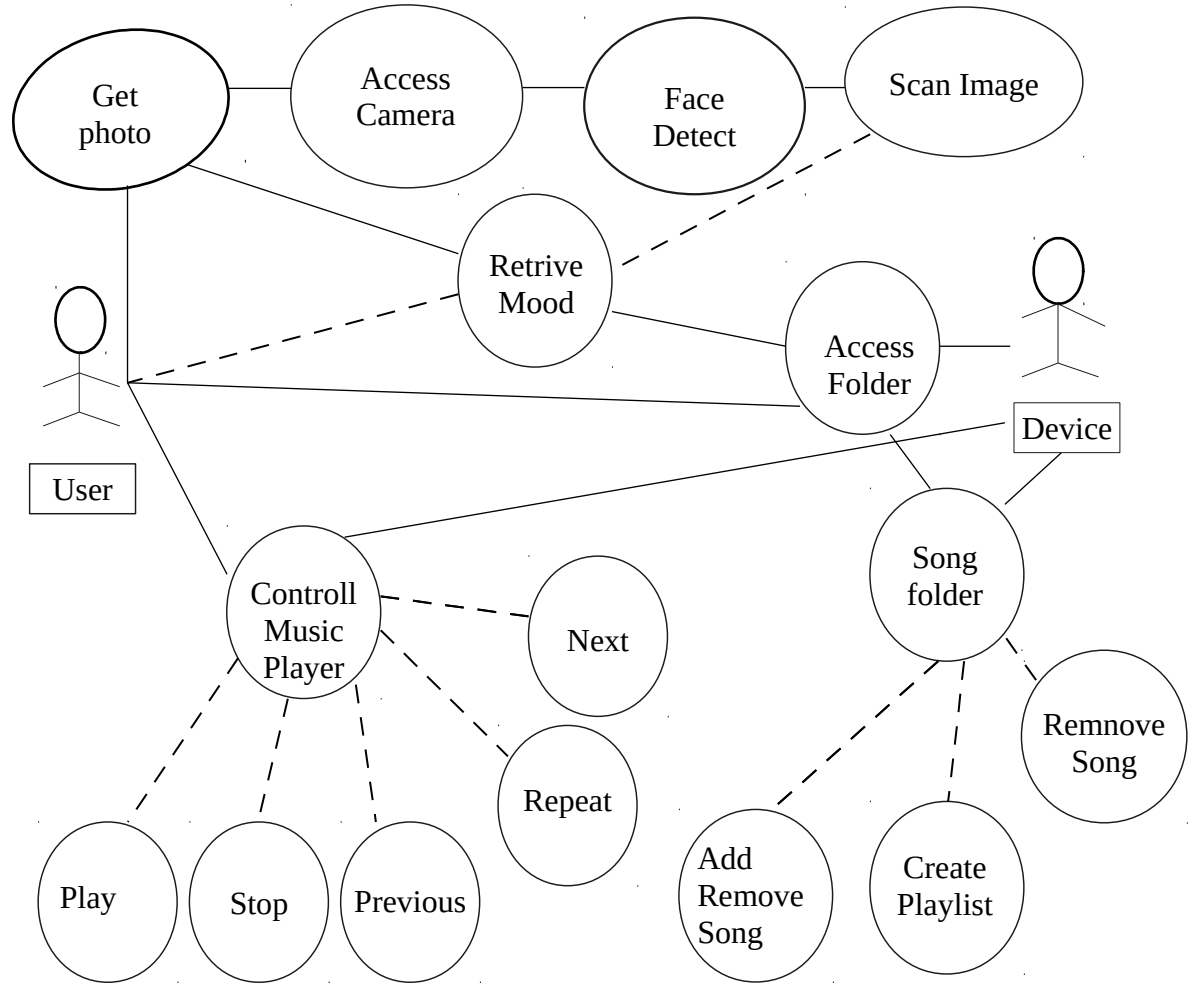


Fig:5.3.4 Usecase Diagram

A **use case** is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language (UML) as an actor) and a system to achieve a goal. The actor can be a human or other external system.

CHAPTER NO 6
IMPLEMENTATION OF SYSTEM

IMPLEMENTATION OF SYSTEM

6.1 Module Design

1. Face Detection Module :

a. Module Name : Face Detection

b. Purpose of Module : To detect users faces

c. Data used : Eyes, nose and lips

d. Algorithm : Haar cascade

e. Code :

```
from cv2 import cv2
faceCascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap=cv2.VideoCapture(0)
while True:
    ret,frame=cap.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray,1.1,4)
    for(x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),4)
        cv2.imshow('demo video',frame)
    if cv2.waitKey(1)& 0xFF== ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

2. Model Training:

```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout,Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.losses import categorical_crossentropy
from keras.optimizers import Adam
from keras.utils import np_utils
df=pd.read_csv('fer2013.csv')
```

```
X_train,train_y,X_test,test_y=[],[],[],[]
for index, row in df.iterrows():
    val=row['pixels'].split(" ")
    try:
        if 'Training' in row['Usage']:
            X_train.append(np.array(val,'float32'))
            train_y.append(row['emotion'])
        elif 'PublicTest' in row['Usage']:
            X_test.append(np.array(val,'float32'))
            test_y.append(row['emotion'])
    except:
        print(f"error occured at index :{index} and row:{row}")

num_features = 64
num_labels = 7
batch_size = 64
epochs = 30
width, height = 48, 48
X_train = np.array(X_train,'float32')
train_y = np.array(train_y,'float32')
X_test = np.array(X_test,'float32')
test_y = np.array(test_y,'float32')
train_y=np_utils.to_categorical(train_y, num_classes=num_labels)
test_y=np_utils.to_categorical(test_y, num_classes=num_labels)
#normalizing data between oand 1
X_train -= np.mean(X_train, axis=0)
X_train /= np.std(X_train, axis=0)
X_test -= np.mean(X_test, axis=0)
X_test /= np.std(X_test, axis=0)
X_train = X_train.reshape(X_train.shape[0], 48, 48, 1)
X_test = X_test.reshape(X_test.shape[0], 48, 48, 1)
# print(f"shape:{X_train.shape}")
##designing the cnn
#1st convolution layer
model = Sequential()
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu',
input_shape=(X_train.shape[1:])))
model.add(Conv2D(64,kernel_size= (3, 3), activation='relu'))
```

```
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Dropout(0.5))

#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Dropout(0.5))

#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
# model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2, 2)))
model.add(Flatten())
#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_labels, activation='softmax'))
# model.summary()
#Compliling the model
model.compile(loss=categorical_crossentropy,
optimizer=Adam(),
metrics=['accuracy'])

#Training the model
model.fit(X_train, train_y,
batch_size=batch_size,
epochs=epochs,
verbose=1,
validation_data=(X_test, test_y),
shuffle=(True)
#Saving the  model to  use it later on
fer_json = model.to_json()
```

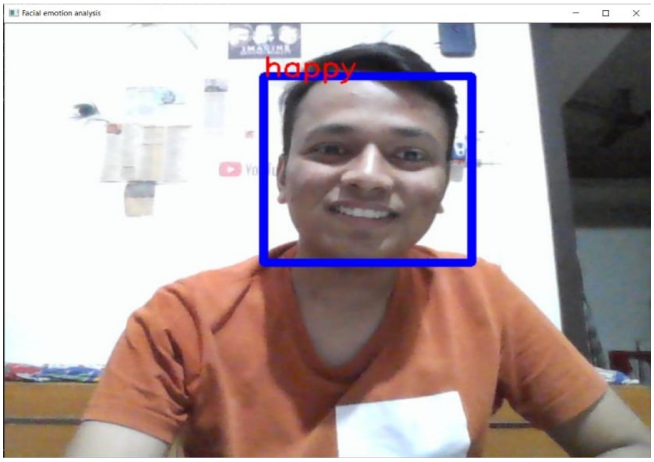
```
with open("fer.json", "w") as json_file:
json_file.write(fer_json)
model.save_weights("fer.h5")
```

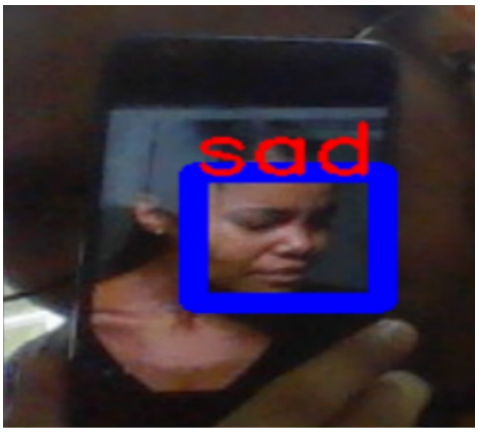
3.Emotion Detection And Playing song :

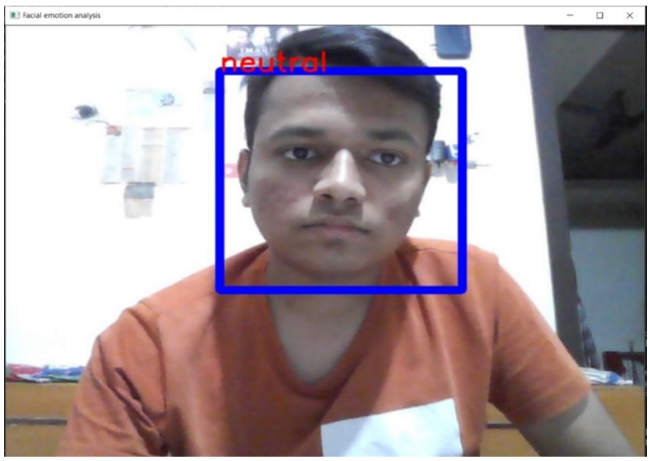
```
from cv2 import cv2
import os
import numpy as np
from keras.models import model_from_json
from keras.preprocessing import image
import time
import random
model = model_from_json(open("fer.json", "r").read())
model.load_weights('fer.h5')
face_haar_cascade=cv2.CascadeClassifier('haarcascade_frontalface_defa
ult.xml')
cap=cv2.VideoCapture(0)
now = time.time()
future = now + 15
while True:
    ret,test_img=cap.read()
    test_img=cv2.flip(test_img,1,0)
    if not ret:
        continue
        gray_img= cv2.cvtColor(test_img, cv2.COLOR_BGR2GRAY)
        faces_detected = face_haar_cascade.detectMultiScale(gray_img,
         1.32, 5)
    for (x,y,w,h) in faces_detected:
        cv2.rectangle(test_img,(x,y),(x+w,y+h),(255,0,0),thickness=7)
        roi_gray=gray_img[y:y+w,x:x+h]
        roi_gray=cv2.resize(roi_gray,(48,48))
        img_pixels = image.img_to_array(roi_gray)
        img_pixels = np.expand_dims(img_pixels, axis = 0)
        img_pixels /= 255
        predictions = model.predict(img_pixels)
        max_index = np.argmax(predictions[0])
        emotions = ('angry', 'disgust', 'fear', 'happy', 'sad','surprise','neutral')
```

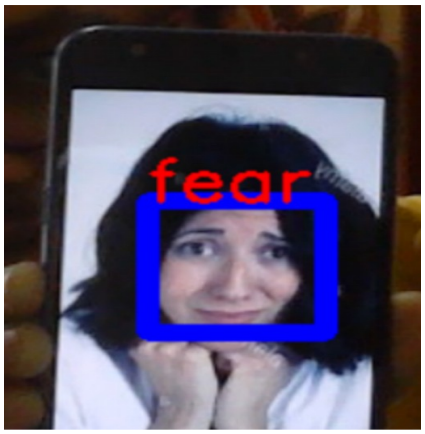
```
        predicted_emotion = emotions[max_index]

        cv2.putText(test_img, predicted_emotion, (int(x), int(y)),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)
        resized_img = cv2.resize(test_img, (1000, 700))
        cv2.imshow('Main Window ',resized_img)
        key = cv2.waitKey(30)& 0xff
        if time.time() > future:
           try:
              cv2.destroyAllWindows()
                   cap.release()
              if predicted_emotion == "happy":mc= ['F:\\Final year project\\
song\\HAPPY1.mp3','F:\\Final year project\\song\\HAPPY2.mp3','F:\\
Final year project\\song\\HAPPY3.mp3','F:\\Final year project\\song\\
HAPPY4.mp3','F:\\Final year project\\song\\HAPPY5.mp3']
               pm=random.choice(mc)
               os.startfile(pm)

              if predicted_emotion == "sad":mc= ['F:\\Final year project\\
song\\sad1.mp3','F:\\Final year project\\song\\sad2.mp3','F:\\Final year
project\\song\\sad3.mp3','F:\\Final year project\\song\\sad4.mp3','F:\\Final
year project\\song\\sad5.mp3','F:\\Final year project\\song\\sad6.mp3','F:\\
Final year project\\song\\sad7.mp3']
                pm=random.choice(mc)
                os.startfile(pm)

                if predicted_emotion == "fear":mc=['F:\\Final year project\\
song\\fear1.mp3']pm=random.choice(mc)
                os.startfile(pm)
                break

           except:
              print('Please stay focus in Camera frame atleast 15 seconds &
run again this program:)')
              break
         if key == 27:
           break
```

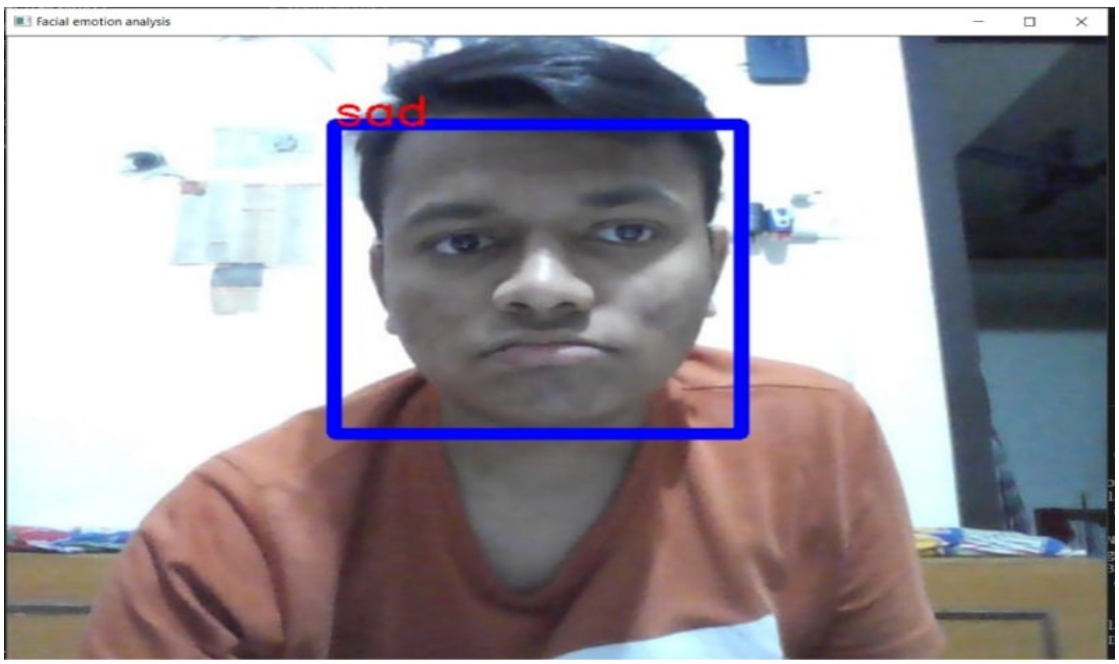
6.2.Module Design:











CHAPTER NO 7

SYSTEM TESTING

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.
System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. Needed The actual result is the behavior produced or observed when a component or system is tested

System testing is performed on the entire system in the context of either functional requirementspecifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification(s).

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholders desire.

7.1 Software testing

7.1.1 Unit Testing :

Unit tests are typicall automated tests written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, or an individual method.By writing tests first for the smallest testable units, then the compound behaviors between those, one can build up comprehensive tests for complex applications.To isolate issues that may arise, each test case should be tested independently. Substitutes such as method stubs,mock_objects,fakes, and test harnesses can be used to assist testing a module in isolation.During development, a software developer may code criteria, or results that are known to be good, into the test to verify the unit's correctness. During test case execution, frameworks log tests that fail any criterion and report them in a summary. For this, the most commonly used approach is test - function - expected value.Writing and maintaining unit tests can be made faster by using parameterized tests. These allow the execution of one test multiple times with different input sets, thus reducing test code duplication. Unlike traditional unit tests, which are usually closed methods and test invariant conditions, parameterized tests take any set of parameters. Parameterized tests are supported by TestNG, Junit and its .Net counterpart, XUnit. Suitable parameters for the unit tests may be supplied manually or in some cases are automatically generated by the test framework. In recent years support was added for writing more powerful (unit) tests, leveraging the concept of theories, test cases that execute the same steps

7.1.2 Integration Testing :

Integration testing(sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. Some different types of integration testing are big-bang, mixed (sandwich), risky-hardest,top-down, and bottom-up. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration. In big-bang, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.The lowest level components are tested first in bottom-up testing. They are then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.The top integrated modules are tested first in top-down testing and the branch of the module is tested step by step until the end of the related module.Sandwich testing combines top-down testing with bottom up testing.

7.1.3 System Testing:

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together . System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behavior produced or observed when a component or system is tested. System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification(s).

7.1.4 Acceptance Testing:

In engineering and its various subdisciplines , acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve checimal tests, physical tests, or performance tests.In systems engineering, it may involve black-box testsing performed on a system (for example: a piece of software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery. Acceptance testing is also known as user acceptance testing (UAT), end-user testing,operational acceptance testsing (OAT), acceptance test-driven development (ATDD) or field (acceptance) testing. Acceptance criteria are the criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. A smoke test may be used as an acceptance test prior to introducing a build of software to the main testing process

7.2 Test cases:

Test Case Id	Descriptio	Expected Result	Actual Result	Status
001	To check whether the camera is open or not	After user execute the program camera will successfully open	Camera is succcessfully open	Pass
002	To check whether the camera is open or not	After user execute the program camera will successfully open	Camera is not opening	Fail
003	To check whether the camera is lagging or not	After opening the camera,camera will open smoothly	Camera is successfully open and runs smoothly	Pass
004	To check whether the camera is lagging or not	After opening the camera,camera wll open smoothly	Camera is successfully open but after opening the camera,camera is to lagging	Fail
005	To check whether the face is detected or not	After opening the camera and after detecting the face there should get square around users face	Camera will open without any lagging and detect face properly with draw the rectangle in user face	Pass
006	Detect face	After opening the camera there will be some obstacles infront of users face the system doesnot recognise the users face	After getting obstacles infront of camera system does not recognise users face	Pass

007	To check whether the face is detected or not	After opening the camera and after detecting the face there sould get square around users face	Camera will open without any lagging but its fail to draw the rectangle into the user face	Fail
008	Passing happy image as a input to system	After successfully detect the face system will detect correct emotion.	System will detect Correct emotion as per giving input	Pass
009	Passing happy image as a input to system	After successfully detect the face system will detect correct emotion.	System willl detect the face but its predict the emotion wrong	Fail
010	Saving songs into the folder	Songs will be adding to the specific folder	Song is added successfully	Pass
011	Close the camera window	Close the camera detection window while the specific time is exhausted	Window will be closed while specific time is exhausted	Pass
012	Close the camera window	Close the camera detection window while the specific time is exhausted	After exhausted the specific time limit camera is still open and try to capture the image	Fail
013	Close the camera window	After detecting the face and and recoganize users emotion just close the window	After detecting face and emotion,still window is open and try to detect user mode	Fail
014	Close the camera window	After detecting the face and recognise users emotion just close the window	After detecting face and emotion camera will automatically get closed	Pass

015	Play the song as per emotion	After detecting emotion song will be play automatically as per the user emotion	Song is playin as per the emotion	Pass
016	Play the song as per emotion	After detecting emotion song will play automatically	After detecting the emotion nothing is going to happen	Fail
017	Play the song as per emotion	After detectin emotion song will be play automatically as per the user emotion	Song is playing but not for the user emotions randomly any song is playing	Fail
018	Close the camera window and play the song	After detecting emotion just close the camera and instantly play the song as per user mode	Camera will close while detecting emotion and instantly song will be played	Pass
019	Close the camera window and play the song	After detecting emotion just close the camera and instantly play the song as per user mode	Camera is still open after detection and song is playing and still user mode is capturing	Fail
020	Close the window while escape or q key is pressed	After pressing q and escape button camera will close	Camera is close after pressing escape or q key	Pass

CHAPTER NO 8

CONCLUSION & FUTURE SCOPE

CONCLUSION AND FUTURE SCOPE

As proposed system, a music player which plays songs according to the user's emotion has been designed. The system has been divided into different modules for implementation which includes face detection, emotion detection and song classification. The proposed system is designed as an emotion aware application which provides a solution to the tangible approach of manual segregation of large playlists. Implementation of static face detection is done using Viola Jones Algorithm and testing of the same was done using images from different facial datasets. Dynamic face detection will be implemented as future work so that users can analyse emotions real time and such an application involves computational complexity and larger amount of dataset for getting higher accuracy level. TheCNN classifier is designed in such a way that perticular amount of emotionlabels can be recognized: happy, anger, sad and neutral and more emotions can be worked for in the future.

REFERENCES

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

In this application first user will set there playlist and after seting the playlist user will able to listen only those song which are mention on the playlist user will not able to edit the playlist after they created it.

[2] Moodfuse, 2017. [Online; accessed 11-July-2017].

In this application, user should manually enter mood and genre first in to the system that wants to be heard and then moodfuse recommends the songs-list as per entering user mood and gener. And then user will select which song they want to listen.

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

These application gives good user accessibility features to play songs and after finishing the song system recommends user with other songs of similar playlist or similar genre.

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

In his application first user should select his mood manually by selecting the moods from the list after selection application will play the music from YouTube

[5] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, pp. 511-518

IEEE, (2001.)

WEB LINKS

[1]Python download link:-

https://www.python.org/downloads/release/python-360/

[2]Keras download link:-
https://pypi.org/project/keras/

[3]TensorFlow download link:-
https://www.tensorflow.org/learn

[4]Dataset download link:-
https://www.kaggle.com/

[5]Opencv download link:-
https://pypi.org/project/opencv-python/

[6]Harcascade frontal face download link:-
http://alereimondo.no-ip.org/OpenCV/34

[7]Songs download link:-
https://www.pagalworld.pw/