A Report

on

Emotion Based Music Player

Submitted in partial fulfillment of the
requirement for the award of the degree of

School of Computing Science and Engineering

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Gautam Kumar

Submitted By

UJJWAL KUMAR JHA (19021011880)
SWATSOM SHARMA (19021011224)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
MONTH, YEAR

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **" Emotion Based Music Player "** in partial fulfillment of the requirements for the award of the Bachelors of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of **Mr. Gautam Kumar** , Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

UJJWAL KUMAR JHA (19021011880)
SWATSOM SHARMA (19021011224)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

CERTIFICATE

The Final Project Viva-Voce examination of **Ujjwal Kumar Jha and Swatsom Sharma** has been held on December 2021 and his work is recommended for the award of **Bachelor of Technology (B. Tech – CSE).**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: November, 2013

Place: Greater Noida

Abstract

In this project, we introduce an affective cross-platform music player, EMP (Emotion Based music player) [1], which recommends music based on the user's real situation. Our music player consists of three modules:

1. Queue module
2. Emotion module
3. Random module

In Queue mode you can add a song to your queue list and play your song. In random mode the song will play randomly. Emotion mode takes a picture of the user's face as input and uses in-depth reading algorithms to identify their emotions with 90.23% accuracy. The Music Split Module uses audio features to get an amazing result of 97.69% while splitting songs into 4 different mood lessons. These feelings are sad, happy, angry, and neutral. This is a project that uses machine learning to get emotions based on user expression. The interface is made with HTML (Hyper Text Markup language), CSS (Cascading Style Sheets) and JS (java Script), and the main code is Python. A window will open in a chrome browser with a player interface. Select the Emotion mode from the bottom right corner. This will start the webcam. The face will be scanned at the end of the song currently playing. You can manually move the song controller near the end to start the task.

The Recommendation Module raises songs for the user by marking their feelings on the emotional type of song, taking into account the user's preferences. We use the Fisher face Machine learning algorithm [2].

Face-to-face detection in Fisherface works with the help of trained models.

Table of Contents

Title		Page No.
Candidates Declaration		**I**
Acknowledgement		**II**
Abstract		**III**
Contents		**IV**
List of Table		**V**
List of Figures		**VI**
Acronyms		**VII**
Chapter 1	**Introduction**	**1**
	1.1 Introduction	**2**
	1.2 Formulation of Problem	**3**
	1.2.1 Tool and Technology Used	
Chapter 2	**Literature Survey/Project Design**	**5**
Chapter 3	**Functionality/Working of Project**	**9**
Chapter 4	**Results and Discussion**	**11**
Chapter 5	**Conclusion and Future Scope**	**41**
	5.1 Conclusion	**41**
	5.2 Future Scope	**42**
	Reference	**43**
	Publication/Copyright/Product	**45**

Acronyms

ML	Machine Learning
SML	Supervised Machine Learning
UML	Unsupervised Machine Learning
ER	Entity – Relationship Diagram
LR	Logistic Regression
AD	Activity Diagram
UCD	Use – Case Diagram

List of Figures

S.No.	Title	Page No.
1	Supervised Machine learning	
2	Iterative WaterFall Model	
3	Use Case Diagram	
4	Flow Chart Diagram	
5	Angry face diagram	
6	Sad face diagram	
7	Smile face diagram	
8	Neutral face diagram	
9		
10	Website interface	

CHAPTER-1

Introduction

Face expression recognition is used to recognize the essential human emotions. The facial expressions convey essential emotional facts and details. In the next generation of computer vision systems, programs and systems focused on the interaction of images may play an important role. Face emotion can be used in the areas of human machine interface (HMI) defense, entertainment and. A human being can convey his / her emotions through his / her mouth, eyes, etc. People usually have a great number of songs in their collection or playlists. So, to avoid difficulty choosing a song, most people only pick a song from their playlist at random and some of the songs might disappoint the user. As a result, some of the songs are not matching to the user’s current emotion. In addition, there's no widely used program that can play songs based on the user's current emotions. Music plays a crucial role in enhancing or improving an individual’s life as it is an important medium of entertainment and relaxation for music listeners and has even proved to have a therapeutic weightage.

Songs, as a medium of expression, have always been a popular choice to depict and understand human emotions. Reliable emotion-based classification systems can go a long way in helping us parse their meaning. However, research in the field of emotionbased music classification has not yielded optimal results.

In this project, we present an affective cross-platform music player, EMP, which recommends music based on the real-time mood of the user. EMP provides smart mood-based music recommendation by incorporating the capabilities of emotion context reasoning within our adaptive music recommendation system. We, in our paper, aim to remove this manual method of selecting songs, as it's a tedious and time-consuming job.

Face detection and facial feature extraction from images is the first step in emotion-based music player. For the face detection to work effectively, we need to provide an input image which should not be blurred and tilted. Once proper facial images are retrieved, linking them with music recommendations will be done.

CHAPTER-1.2

Formulation of Problem

Connecting music and emotion in one single app is not applicable in most areas because of complexity of the project and the implementation, and even if they solved it in a research paper no one and there is no app is applying this program, even the very well-known apps Spotify, Gaana , and Hungama . All of them are simple in the way of offering their service, they offer the music itself with some features like pausing, stopping, change the song back and front, shuffle, and creating playlists. These are what they really offer to users, so starting by studying Emotions and the construction of them, and developing the algorithm that can help me solve this problem is the main focus and the main problem, and as I said before this is a really hard and tough problem but in simple way it is already solved. Starting by design a very simple app that is easy to use, this the first step. And then starting to build the back-end of the functionality of the app by building a Convolutional Neural Network to classify the images, to try getting the most out of this algorithm and get accuracy rate that can tell that the app solved the problem. These are the main points of the problem statement of this project.

1.3 Tool and Technology Used

In this project we use limited tools and technology use like as: -

1. Jupyter
2. Dataset
3. Python
4. Linear Discriminant Analysis (LDA)
5. Principal Component Analysis (PCA)

1 – JUPYTER The Jupyter Notebook is an open-source web application that permits you to make and share records that contain live code, conditions, perceptions and story text. Utilizations include: information cleaning and change, mathematical reenactment, factual displaying, information representation, AI, and significantly more.

2 - DATASET A Dataset is an assortment of information. As such, a dataset compares to the substance of a data set table, or a single statistical data matrix, where each segment of the table addresses a specific variable, and each line relates to a given individual from the informational index being referred to.

3- PYTHON Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

4-Linear Discriminant Analysis or **Normal Discriminant Analysis** or **Discriminant Function Analysis** is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

For example, we have two classes and we need to separate them efficiently. Classes can have multiple features. Using only a single feature to classify them may result in some overlapping as shown in the below figure. So, we will keep on increasing the number of features for proper classification.

5- Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation that converts a set of correlated variables to a set of uncorrelated variables. PCA is the most widely used tool in exploratory data analysis and in machine learning for predictive models. Moreover, PCA is an unsupervised statistical technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit.

In order to successfully build this model, we use the Machine Learning Algorithms. The main algorithm is the Supervised Machine Learning Algorithm in which we train the model using the well labeled dataset. The dataset includes the values ranging from a starting value to the highest value, the symptoms and the type of the disease. We will divide the dataset into three parts namely, train, test and validation sets. We will train the model using the training dataset, and then validate it using validation dataset, and finally test the model using the test dataset to check whether the model is giving the accurate results or not. This model basically follows the classification approach as it predicts the class of the disease. It tells from which disease, the person is suffering

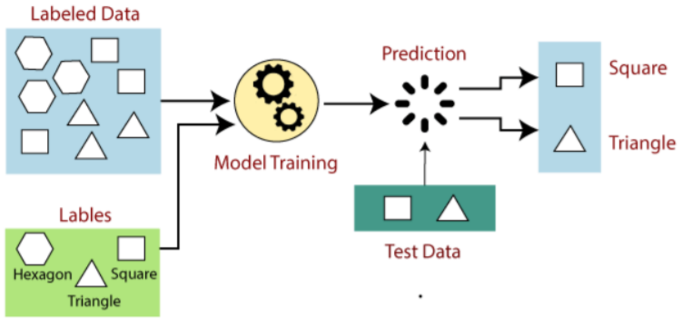


Figure-1—Supervised Machine Learning.

Requirements—

In order to successfully build this model, we use the Machine Learning Algorithms. The main algorithm is the Supervised Machine Learning Algorithm in which we train the model using the well labeled dataset. The dataset includes the values ranging from a starting value to the highest value, and the type of Mood. We will divide the dataset into three parts namely, train, test and validation sets. We will train the model using the training dataset, and then validate it using validation dataset, and finally test the model using the test dataset to check whether the model is giving the accurate results or not. This model basically follows the classification approach as it predicts the type of Mood.

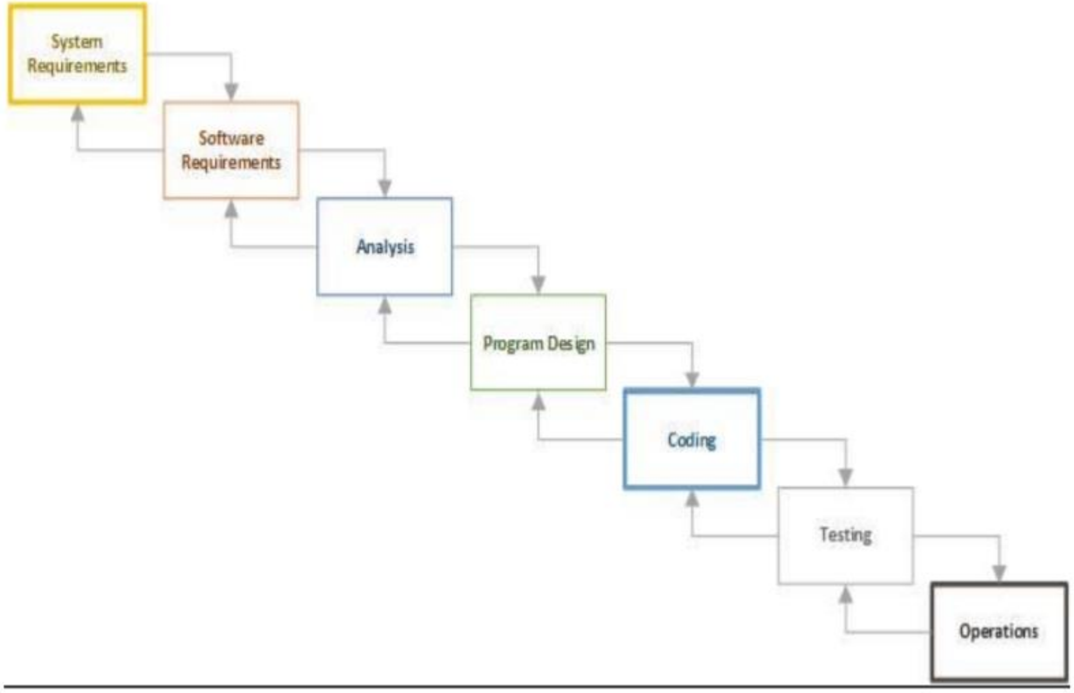


Figure-2:-- Iterative Waterfall Model.

Chapter 2

Literature Survey

Emotion based music players are the need of the hour and will provide a boon to the fields of Emotion Intelligence, Medical Science and Psychology. In recent times, techniques such as Neural Networks (NN), Support Vector Machines (SVM) have been used. We analyze these techniques which are in association with our application.

Anagha S. Dhavalikar and Dr. R. K. Kulkarni Proposed Automatic Facial Expression recognition system. In this method of theirs, there were three phases or steps.

- Face detection.
- Feature Extraction and
- Expression recognition.

The First Phase, Face Detection consists of the analysis done by YCbCr Color model which includes the ISO illumination processing for getting face and operations for retaining required face i.e eyes and mouth of the face. The proposed algorithm also uses the AAM i.e. Active Appearance Model. This is a method which is used for facial properties extraction. In this algorithm the several points and features on the face (also known as Action Units), AU, like eyes, eyebrows, mouth or lips are identified and then a file is made which gives properties about action points detected. The face emotions are given as input and the AAM Model changes according to expression

Advantages –

- Considers all detailed aspects of the face for analysis.
- This model is based on supervised learning.

Disadvantages

- Shows lower efficiency for blur images.
- Each AU is needed for the best results.

Yong-Hwan Lee, Woori Han and Young Kim proposed a system based on Bezier curve fitting. This algorithm uses a multiple phase process for face expression and emotions. The first one is detection and processing of facial area from the input original image. The second step is the verification of emotion in the region of interest. The first phase of face detection uses color still image based on skin color pixel by initialized spatial filtering,based on the result of lighting compassion. Later it uses the Feature Map To measure eyes and lips face location and facial shape. After extracting regions of interest, this method extracts points from the feature map in order to apply Bezier curve to the eye and mouth techniques to study The difference between the Hausdorff distance and the Bezier curve between the image from the database and the face picture entered.

Advantages –

- The size of the dataset isn't restricted. The algorithm shows high efficiency even on large datasets.
- The algorithm can also be used for 3D pictures. Thus a 3D picture can also be exploited for information.

Disadvantages

- Each point of study in the data has a global influence. There are no outliers. This can cause over fitting.

Hence the efficiency can decrease, if the data set is distorted due to over fitting.

Feasibility Study

Feasibility study is disbursed when there's a fancy problem or opportunity. it's considered because the primary investigation which emphasises on “Look before 18 You Work” approach to any project .A Feasibility study is undertaken to see the likelihood of either improving the present system or developing a very new system.

We are visiting develop the new system which is possible as our application is incredibly user friendly and simple to apply/understand.

Technical Feasibility—

In this kind of study, the present technology in employed in a corporation is checked like the prevailing software, hardware, and private staff to work out whether it'll work for the proposed system. The technology that was important in developing a brand-new system like development tools, back-end database system was available from within the organization. The proposed system is capable of handling large storage of information. The back-end and front-end technology has greater important for providing an accurate, error-free, frequencies of knowledge to be used.

• This project is technically feasible in all aspects. This project will provide latest platform like android technology.

Economical Feasibility—

For proving that system developed is economical, the economical feasibility study takes place to test the value of developing a system against the advantages that it provides. If the price are less and benefits are over we will define our system to be economically developed. User saves time in looking for a specific product to be purchased by simply few clicks. The registration process is very much faster than the manual registration. The all data is stored computerized so it saves paper. The record is of freed from human errors as there's less chance of mistakes. The above benefits are in terms of saving time, minimize errors and supply efficiency in work done.

- In terms of economical feasibility our application is incredibly reasonable in cost. So application is economically feasible.

Operational Feasibility—

The operational feasibility is anxious with the operability of the system after it's been installed. That is, some programmer might not like changes in their routine 19 method of labor or has fear that they'll lose their contemporaries. The following areas will have the operational feasibility within the proposed project.

- The organization has approved this method as their working system.
- The user of the system has accepted the proposed system as their new working system and realized the advantages of it.
- The system will add a correct way after it's been installed and therefore the installation process is straightforward to use.

In Short—

Feasibility Analysis—

This app is very useful for all persons in every aspect. “Feasibility Study is the evaluation of the proposed system regarding its workability, effectiveness, and its usefulness for any person / organization.”

(i) Technical Feasibility—

a. This software can be implemented with the existing technical resources.

b. Required Hardware and software tools are easily available and easy to use.

c. Hence this software is Technically Feasible.

(ii) Economic Feasibility –

This proposed system is financially feasible, because of the following reasons—

a. Installation of the system can be made with very low cost.

b. The hardware requirements for this app is minimum and can be easily affordable, making it economically feasible.

(iii) Operational Feasibility—

a. The operational feasibility of this app is user friendly with good GUI (Graphical User Interface).

b. Information updating is done easily by the administrator.

c. As in present day situation technology is known for making changes and robustness in performing tasks.

(iv) Behavioral Feasibility— People often do not change the devices they are using. So, a prior information has to be collected on how strong a reaction the user having regarding this app.

a. User friendly.

b. Knowledge of programming is required.

c. This will save a lot of time and reduce the workload. Hence this website is behaviorally feasible in the organization.

(v) Schedule feasibility—

The time schedule required for the development of the project is very important since more development time effects machine time, costs and delay time in the development of other systems.

Chapter-3

Working of project

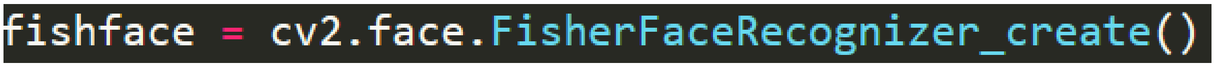
Detecting emotions

Collecting data

Facial expression detection in Fisherface works with the help of trained models. Reason behind this is to allow user to take dataset according to their use. Suppose if we take a huge amount of dataset of around 25-30k it will give nice accuracy no doubt but if the situation is like that the user of the devices are a few people. Now in such condition if we take some precise dataset with around 400-450 images as input releted to the user then it will also give good accuracy with the benefit of less amount of dataset and less storage on memory to operate. As well as small memory of data give output fast which result in quick response time. Here we first tried with Cohn-Kanade dataset then we made some classification in the as our need make it to train our model.

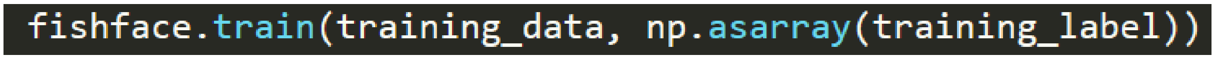
Loading and saving trained model

For training, We have used Fisherface method of cv2 library.



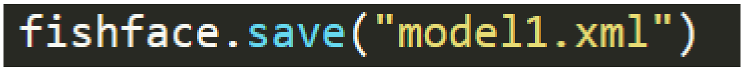
```
fishface = cv2.face.FisherFaceRecognizer_create()
```

For training data model we have make a python code which grab all the classified images from folders and map it with it’s emotion. These data we at an instance stored in dictionary and then use .train method to train model.



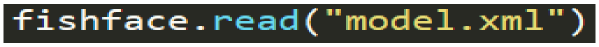
```
fishface.train(training_data, np.asarray(training_label))
```

To save the model for later use we have implemented .save method.



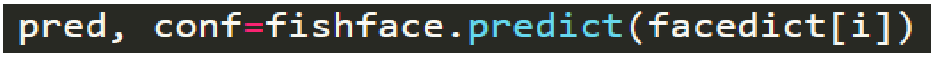
```
fishface.save("model1.xml")
```

Now at the detection time first we have load model in memory using .read method.



```
fishface.read("model.xml")
```

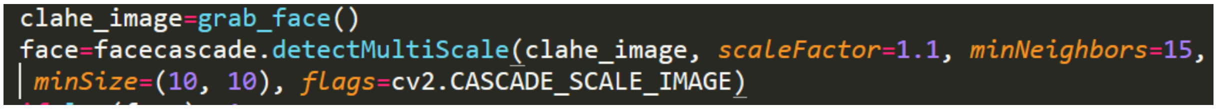
Prediction of result is based on the prediction and confidence value which .predict method return.



```
pred, conf=fishface.predict(facedict[i])
```

Haarcascade model

Haarcascade model is precise face detection trained model which is provided by Open-cv. It return the co-ordinates in terms of (x, y) at (left, bottom) of face frame and it's width and height from those co-ordinates.



```
clahe_image=grab_face()
face=facecascade.detectMultiScale(clahe_image, scaleFactor=1.1, minNeighbors=15,
  minSize=(10, 10), flags=cv2.CASCADE_SCALE_IMAGE)
```

As here in the .detectMultiScale() method it is capable of detect multiple faces and it return an array of all the faces(co-ordinates) as an element.

The arguments has set according to the threshold what we need for our checking purpose. We have set it such like it doesn't affect our model accuracy.

Result Calculation

In our model we have not stick on one image for testing, While the code will run it will take around 10 images in a short time(1-2 sec) and for all those images it will compute result and according to the average value of that it will give result. Apart from that we have make two codes one work on single face at a time while another work with multiple faces in the image.

Machine Learning

Fisherface ML algorithm

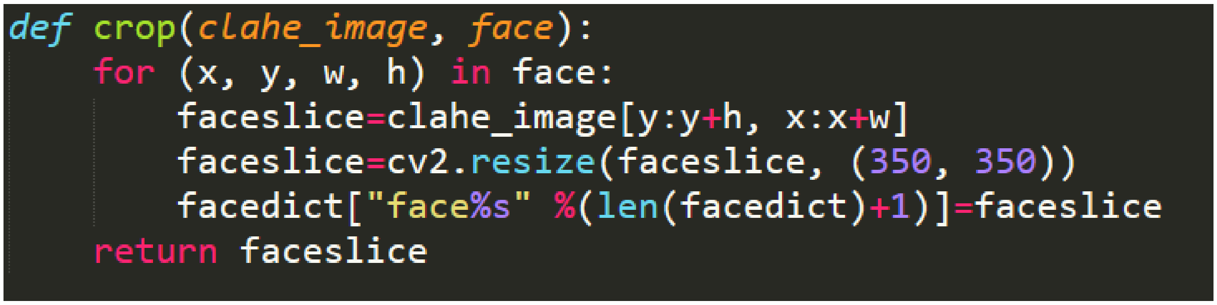
Fisherface algorithm is an algorithm which work on the basis of LDA and PCA concepts. Linear discriminant analysis (**LDA**) is a supervised Learning method of machine learning. Now supervised Learning is that where we use such data whose answer is also given to the model to learn it. It work on the concept of dimensionality reduction. Which reduce the execution time among classification.

Principal Component Analysis (**PCA**) is a one kind of conversion from correlated variables to uncorrelated in the form of mathematical values.

It is mostly used for the observing data and from that by some probabilistic calculation generate models. The flow of Fisherface is like it takes classified images then it will reduce the dimension of the data and by calculating it's statistical value according the given categories it stores numeric values in .xml file. While prediction it also calculate the same for given image and compare the value with the computed dataset values and give according result with confidence value.

Resizing images

Whatever the image we have chosen for dataset it mostly related to the size which can give an precise output. The size is chosen such like the model can able to easily distinguish face from image by haarcascade model. And the size what we get from real time scan is not always same as data (very less difference) so, We resize it to the exact model data size. In our case we have chosen 350*350.



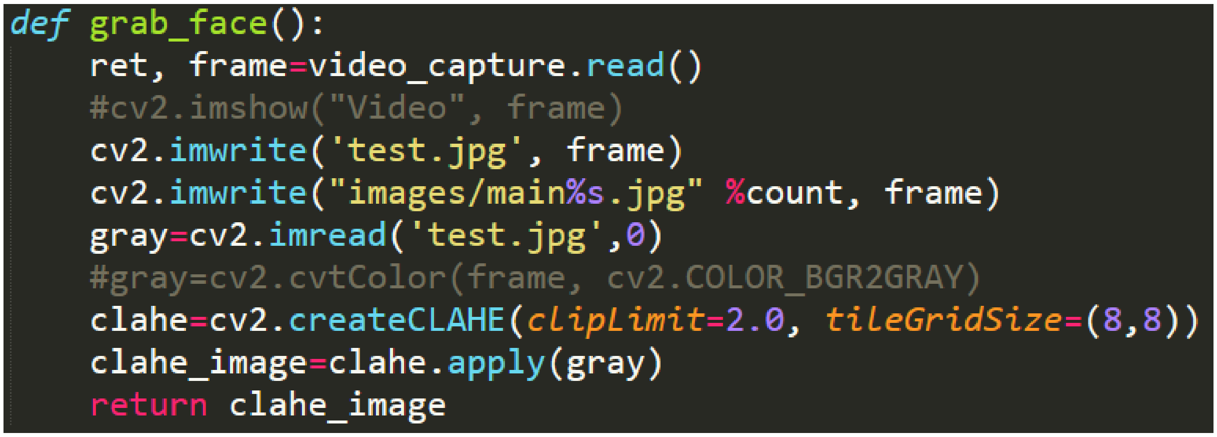
```
def crop(clahe_image, face):
    for (x, y, w, h) in face:
        faceslice=clahe_image[y:y+h, x:x+w]
        faceslice=cv2.resize(faceslice, (350, 350))
        facedict["face%s" %(len(facedict)+1)]=faceslice
    return faceslice
```

Here In this method, we have implemented the cropping of image by given parameters of haarcascade by clahe_image[] and use of cv2's method .resize() to the given size. Finally, We have stored those images in dictionary and after some count(=10) take it to check result.

Gray scaling images

It was the need for the method and because of it's contrast and shaded face, it result in benefit for algorithm to get output.

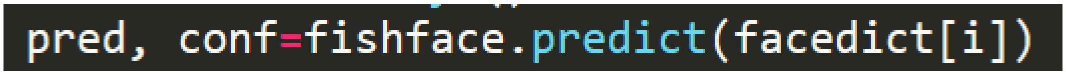
Face detection



```
def grab_face():
    ret, frame=video_capture.read()
    #cv2.imshow("Video", frame)
    cv2.imwrite('test.jpg', frame)
    cv2.imwrite("images/main%s.jpg" %count, frame)
    gray=cv2.imread('test.jpg',0)
    #gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    clahe=cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    clahe_image=clahe.apply(gray)
    return clahe_image
```

As the given in the code grab_face() methods uses to get the images and do all operation and finally return cropped ,grayed face value in dictionary.

Train and predict methods



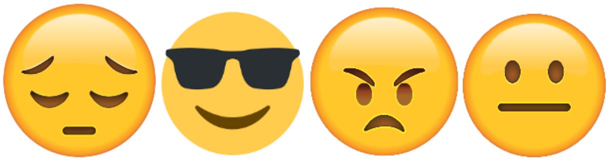
```
pred, conf=fishface.predict(facedict[i])
```

This code is use to get prediction and confidence value for given amount of image. Then get the max function with obtained output and final result is shown to the user.

Playing music

Detected emotions

We have implemented the linking of python with javascript through eel library. Which provide us the privilege to access python methods from js as well as vice versa. Here the striating flow will be in python code as the library is implemented in python then it transfer the control to html, JS. And according to the result we show emoticons.



Sad happy angry neutral

According to which we can classify emotion directory for playing song we have chosen this 4 emotions.

Methods for playing songs

In JavaScript file we have implemented too much methods for the switching of song.

1. Queue

2. Based on Emotion
3. Random

In the first one as queue works it has been implemented. In second one we call python code to get emotion from user's facial expression and according to that chosen next song which is also randomly and played it. In third one we directly used random function and all the methods are dynamic it can handle as change in number of songs accordingly.

HTML, CSS and JS concepts for online music player.

As we know the css give a great look to communicate and through JS we can interact with user and not look like complicated program run at console and it also give user privilege to choose any song to play.

Project Related Diagram Models—

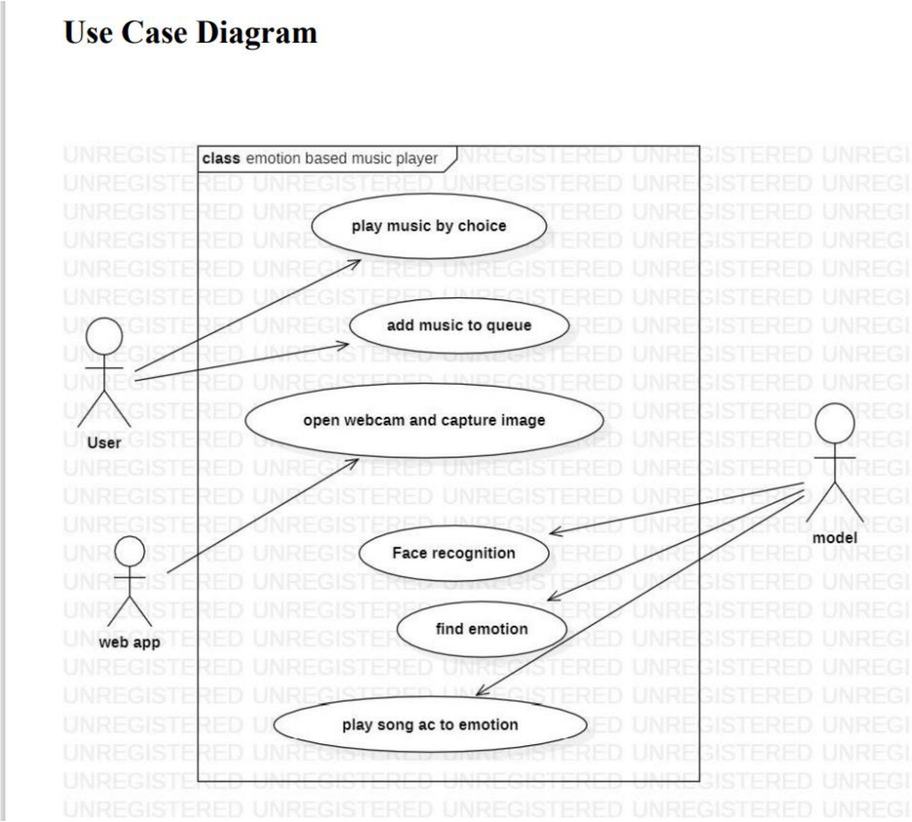


Fig. 3 Use Case Diagram

Flow Chart

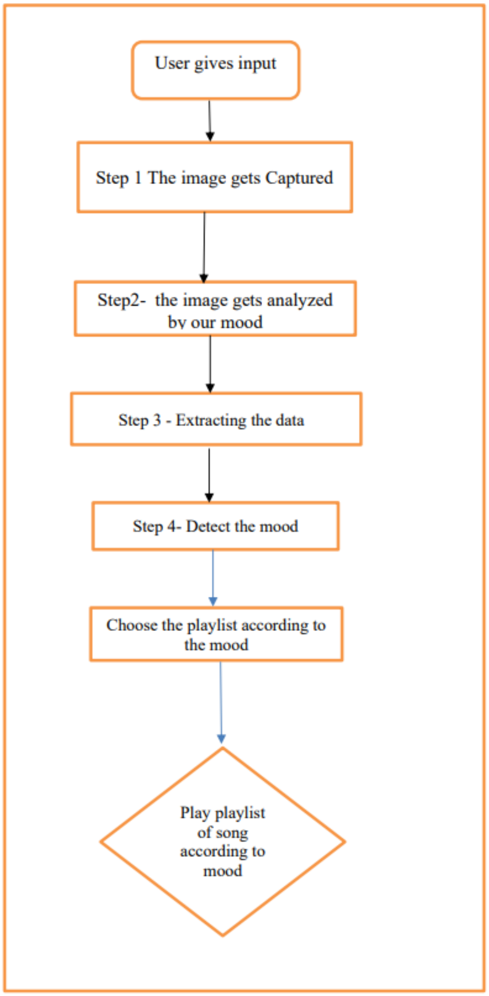

User gives input
Step 1 The image gets Captured
Step2- the image gets analyzed by our mood
Step 3 - Extracting the data
Step 4- Detect the mood
Choose the playlist according to the mood
Play playlist of song according to mood

Fig.4 Flow Chart Diagram

Chapter-4

Implementation and Testing

The static image was used for facial reaction or response perception. The captured photo is taken from photos stored for testing. Cohn-Kanade database has a total number of 890 photos and as for JAFEE database, it has a total number of 213 photos out of which database is distinct in both the parts: first is the training set and the other one is evaluation set.

For training and evaluation, we arrange the database in an 80/20 relation manner. Both the set carries seven expression assertion.

- Cohn-Kanade Database: For the evaluation model we gave one sole captured photo to the system. Then system first recognize the position of exclusive mark and the then it finds those marks.
- For training and evaluation, the mode, the database was given to the model. After evaluation, the confusion matrix and grouping report or the arrangement in a specific order is been generated.

Some of the mind's states of an individual studied by the system are depicted below:

1) ANGRY FACE

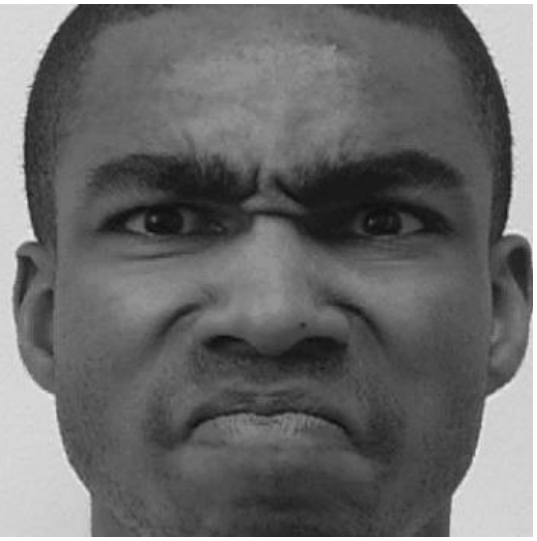


Fig-5. angry face

In this picture, the user's state of mind is angry.

2) SAD FACE

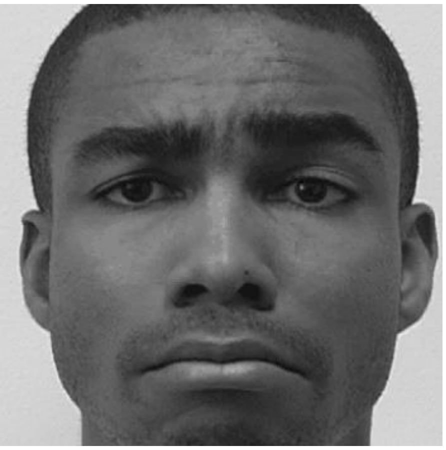


fig-6. sad face

In this image, the user's state of mind is sad or depressed.

2) **HAPPY FACE:**

Fig-7. happy face

In this image, user's state of mind is happy or joyful.

4) NEUTRAL FACE:

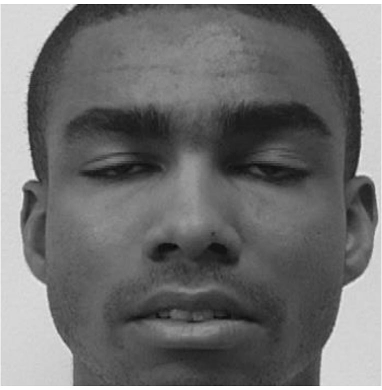


Fig8. Neutral face

In this image captured, the user’s state of mind is neutral.

Table1: Classification Report of CK+ Database

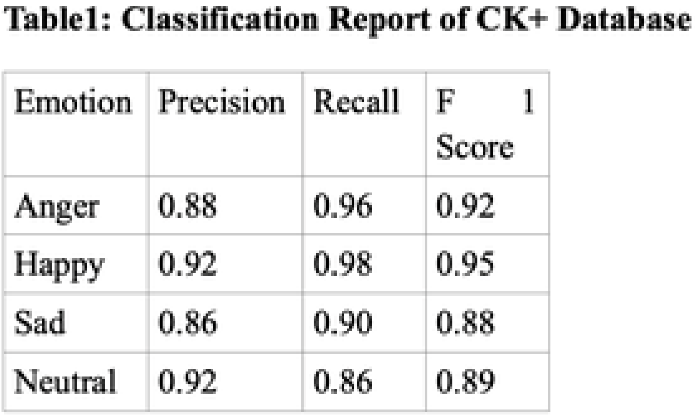


Table1: Classification Report of CK+ Database

Emotion	Precision	Recall	F 1 Score
Anger	0.88	0.96	0.92
Happy	0.92	0.98	0.95
Sad	0.86	0.90	0.88
Neutral	0.92	0.86	0.89

Table2: Accuracy Report of CK+ Database

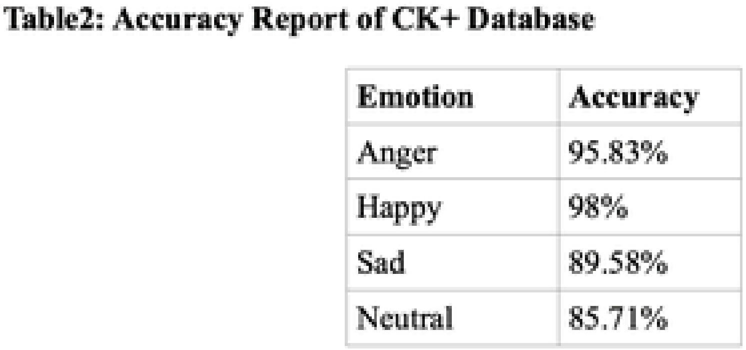


Table2: Accuracy Report of CK+ Database

Emotion	**Accuracy**
Anger	95.83%
Happy	98%
Sad	89.58%
Neutral	85.71%

Testing: --

Software Testing is the most important part for the development of any type of software. It involves checking the system, whether the software is responding well to the requests or not. It checks whether the outputs for the corresponding inputs are well or not. It also tells whether the system is responding well to the requests or not.

In my this model. Testing is done by checking the Login credentials whether it is responding well or not. Testing is also done by checking the user details and predictions generated, whether the model is able to make successful predictions or not. Other than that there are also various types of testing's done to check the functionalities of the model.

The model is tested for validation, its implementation and its navigation.

1. **Validation Testing: --**

   The user must login to the system with his/her unique login name and password and must enter all the mandatory fields. If the same does not happens, then a warning message is displayed.

   The user accesses his / her account with the credentials. If the credentials are correct, then, the login is successful else login is unsuccessful.

   We check this in the Validation Testing.

2. **Functional Testing: --**

   The entire system is divided into sub - modules. Here, in this testing, addition or updating of the user data / information in the database is performed.

   In this type of testing, we try to add something new data / information or we try to update the prior information. If all the things goes well, then this testing gives positive result else not.

3. Navigational Testing:--

The system is tested so that all the pages of the model are properly accessible with their respective links and the data.

We check whether all the navigations in the model are proper or not, i.e., whether we are successfully navigating to other screens of the model well or not.

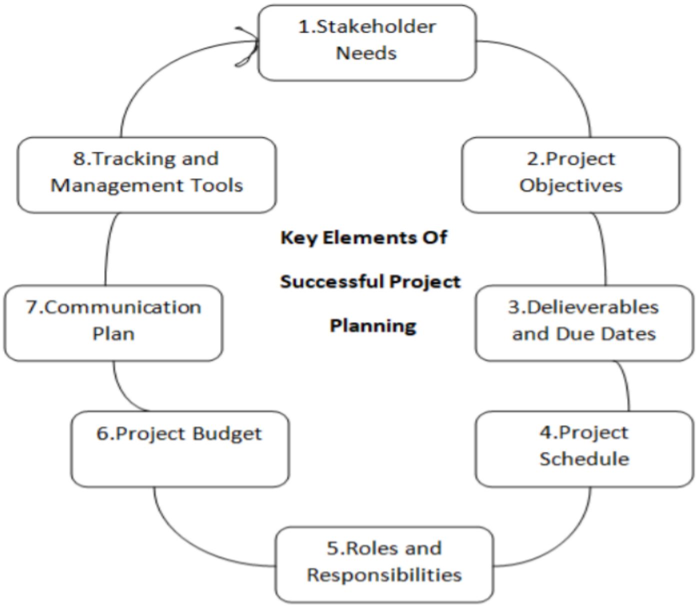

1.Stakeholder Needs
2.Project Objectives
3.Delieverables and Due Dates
4.Project Schedule
5.Roles and Responsibilities
6.Project Budget
7.Communication Plan
8.Tracking and Management Tools
Key Elements Of Successful Project Planning

Fig9.

To uncover the errors in the system, I have done testing as follows:--

1. Input Checking
2. Condition Testing
3. Loop Testing
4. Output Testing
5. Acceptance Testing

Testing Approaches—

There are three types of software testing approaches which are as follows—

1. White Box Testing
2. Black Box Testing
3. Grey Box Testing

Testing Levels—

There are three types of software testing levels which are as follows—

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

Types of Black Box Testing—

1. Functionality Testing
2. Non-functionality Testing.

Software Debugging—

Software Debugging is the process of removing bugs, and errors from the software in order to make it work efficiently.

It may possible that software may contain some errors and bugs, so we need to remove them.

After removing these errors, software works efficiently

Source Code :

Interfacing of this project using HTML, CSS and JavaScript

HTML Code

```
<!DOCTYPE html>
<html>
<head>
     <meta charset="UTF-8">
    <title>musical Arena</title>
    <script type="text/javascript" src="/eel.js"></script>
    <link rel="stylesheet" type="text/css" href="header.css">
    <link rel="stylesheet" type="text/css" href="background.css">
    <link rel="stylesheet" type="text/css" href="player.css">
    <style>
    body{
        margin: 0;
    }
    </style>
</head>


<body>
    <br><br><br><br><br>
    <div id="first">
        <h1 font-family="verdana" align="center" style="font-size: 40px;color: white;margin: 0;">Music Player<h1>
        </div>
    <!--<div id="second" class="second">

        <button onclick="myFunction()">
    </div>
    -->

    <script src="code.js"></script>

        <div id="queue"><br>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Queue<input type="button" id="next" onclick="nextsong()"><hr></div>

    <div id="third">
        <div id="emoji"></div>
        <div id="xyz">
        &nbsp;&nbsp;&nbsp;Playing :&nbsp;&nbsp;
        <label id="sname" align="center">none</label></div>
        <div id="mod">mode : Queue-mode <input type="radio" name="mode"
checked="checked" onclick="setmod(this)" value="1"> &nbsp;&nbsp;Emotion-mode <input
type="radio" name="mode" onclick="setmod(this)" value="2"> &nbsp;&nbsp;Random-mode
            <input type="radio" name="mode" onclick="setmod(this)" value="3"></div>
        <audio controls id="main_slider">
        <source id="sel"  type="audio/mpeg">
Your browser does not support the audio element.
        </audio>

        <script>
    document.getElementById("main_slider").onended=function(){
        if(mod==1)
            next_in_Q();
        else if(mod==2){
            getTime();
        }
        else
            rand_play();

    };

        </script>
        </div>
        </body>
</html>
```

CSS Code

```
#third{
            width: 100%;
            height: 80px;
            padding: 20px;
            padding-bottom: 0px;
            padding-top: 10px;
            float: left;
            position: fixed;
            bottom: 0;
            left:0;
            background-color: rgba(1,1,1,0.8);
            color:rgb(250,250,250);
            font-family: "Segoe Script";
        }
         #mod{
            display: inline;
            position: absolute;
            right: 50px;
        }
audio{
        position: fixed;
        margin-top: 28px;
        right: 20px;
        width: 90%;
        }
label{

    font-family: "Segoe Script";
    width: 500px;
    height: 10px;
}
#emoji{
    width: 75px;
    height: 75px;
    display: inline-grid;
    border-color: rgba(1,1,1,0.5);
    background:transparent url("next.png");
    /*mix-blend-mode: multiply;*/
    background-size: cover;
    border-bottom-left-radius: 50%;
    border-top-left-radius: 50%;
    border-bottom-right-radius: 50%;
    border-top-right-radius: 50%;
}
#xyz{
            display: inline;
            position: absolute;

        }


body{
    padding: 0px;
    background-color:orange;
/*  background-image: url("background.jpg");*/
    background-attachment: fixed;
    height:100%;
}
    .song{
        margin: 25px;
        float: left;
        width: 194px;
        height: 200px;
        transition: width 1s;
        background-color: rgba(1, 3, 5,0.8);
        padding: 15px;
        border-bottom-right-radius: 10%;
        border-top-left-radius: 10%;
        border-bottom-left-radius: 10%;
        border-top-right-radius: 10%;
         word-wrap: break-word;
         overflow: hidden;
         z-index: 2;

    }
    .song:hover{
        width: 400px;
    }
    #data{
        color: white;
        float: left;
        width: 160px;
        height: 200px;
        margin-top: 0px;
        padding-top: 0px;
        padding-left: 30px;
        z-index: 2;
        font-size: 16px;
        font-family: "Monotype Corsiva Italic";

    }
    #pic{
        float: left;
        background-image: url("E:\\Imagere\\2.jpg");
            background-size: cover;
            height: 200px;
            width: 190px;
            border-bottom-right-radius: 10%;
            border-top-left-radius: 10%;
            border-bottom-left-radius: 10%;
            border-top-right-radius: 10%;
            z-index: 2;
    }
.play{
        border-color: rgba(1,1,1,0.5);
        background:transparent url("play.png");
        mix-blend-mode: multiply;
    background-size: cover;
    border-bottom-left-radius: 50%;
    border-top-left-radius: 50%;
    border-bottom-right-radius: 50%;
    border-top-right-radius: 50%;
    height: 50px;
    width: 50px;
    margin-left: 16%;
    margin-top: 55%;
    z-index: 2;
}
.play:focus{
    outline: none;
}
.play:hover{
    box-shadow: 0px 0px 5px 5px black;
    border-color: black;
    background-color: rgba(1,1,1);
    border-bottom-left-radius: 50%;
    border-top-left-radius: 50%;
    border-bottom-right-radius: 50%;
    border-top-right-radius: 50%;
    z-index: 2;

}

.add{
    border-color: rgba(1,1,1,0.5);
    background:transparent url("add.png");
    mix-blend-mode: multiply;
    background-size: cover;
    border-bottom-left-radius: 50%;
    border-top-left-radius: 50%;
    border-bottom-right-radius: 50%;
    border-top-right-radius: 50%;
    height: 50px;
    width: 50px;
    margin-left: 13%;
    margin-top: 55%;
    z-index: 2;
}
.add:focus{
    outline: none;
}
.add:hover{
    box-shadow: 0px 0px 5px 5px black;
```

```
        border-color: black;
        background-color: rgb(1,1,1);
        border-bottom-left-radius: 50%;
        border-top-left-radius: 50%;
        border-bottom-right-radius: 50%;
        border-top-right-radius: 50%;

    }
```

Python Code for Capturing Image

```
import cv2
import argparse
import time
import os
#import Update_Model
import glob
import random
import eel
import light
#import winsound

frequency=2500
duration=1000

eel.init('WD')
emotions=["angry", "happy", "sad", "neutral"]
fishface = cv2.face.FisherFaceRecognizer_create()
font = cv2.FONT_HERSHEY_SIMPLEX
'''try:
    fishface.load("model.xml")
except:
    print("No trained model found... --update will create one.")'''

parser=argparse.ArgumentParser(description="Options for emotions based music
player(Updating the model)")
parser.add_argument("--update", help="Call for taking new images and retraining the model.",
action="store_true")
args=parser.parse_args()
facedict={}
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, (350, 350))
        facedict["face%s" %(len(facedict)+1)]=faceslice
    return faceslice

def grab_face():
    #ret, frame=video_capture.read()
    ret, frame=light.nolight()
    #cv2.imshow("Video", frame)
    cv2.imwrite('test.jpg', frame)
    cv2.imwrite("images/main%s.jpg" %count, frame)
    gray=cv2.imread('test.jpg',0)
    #gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    clahe=cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    clahe_image=clahe.apply(gray)
    return clahe_image

def detect_face():
    clahe_image=grab_face()
    face=facecascade.detectMultiScale(clahe_image, scaleFactor=1.1, minNeighbors=15,
minSize=(10, 10), flags=cv2.CASCADE_SCALE_IMAGE)
    if len(face)>=1:
        faceslice=crop(clahe_image, face)
        #return faceslice
    else:
        print("No/Multiple faces detected!!, passing over the frame")

def save_face(emotion):
    print("\n\nLook "+emotion+" untill the timer expires and keep the same emotion for some
time.")
    #winsound.Beep(frequency, duration)
    print('\a')


    for i in range(0, 5):
        print(5-i)
        time.sleep(1)

    while len(facedict.keys())<16:
        detect_face()

    for i in facedict.keys():
        path, dirs, files = next(os.walk("dataset/%s" %emotion))
        file_count = len(files)+1
        cv2.imwrite("dataset/%s/%s.jpg" %(emotion, (file_count)), facedict[i])
    facedict.clear()

def update_model(emotions):
    print("Update mode for model is ready")
    checkForFolders(emotions)

    for i in range(0, len(emotions)):
        save_face(emotions[i])
    print("Collected the images, looking nice! Now updating the model...")
    Update_Model.update(emotions)
    print("Model train successful!!")

def checkForFolders(emotions):
    for emotion in emotions:
        if os.path.exists("dataset/%s" %emotion):
            pass
        else:
            os.makedirs("dataset/%s" %emotion)

def identify_emotions():
    prediction=[]
    confidence=[]

    for i in facedict.keys():
        pred, conf=fishface.predict(facedict[i])
        cv2.imwrite("images/%s.jpg" %i, facedict[i])
        prediction.append(pred)
        confidence.append(conf)
    output=emotions[max(set(prediction), key=prediction.count)]
    print("You seem to be %s" %output)
    facedict.clear()
    return output;
    #songlist=[]
    #songlist=sorted(glob.glob("songs/%s/*" %output))
    #random.shuffle(songlist)
    #os.startfile(songlist[0])
count=0
@eel.expose
def getEmotion():

    count=0
    while True:
        count=count+1
        detect_face()
        if args.update:
            update_model(emotions)
            break
        elif count==10:
            fishface.read("model2.xml")
            return identify_emotions()
            break

eel.start('main.html')
```

Python Code for display

```
import tkinter as tk
import cv2
from PIL import Image, ImageTk

width, height = 800, 600
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)

root = Music_player.Tk()
root.bind('<Escape>', lambda e: root.quit())
lmain = Music_player.Label(root)
lmain.pack()

def show_frame():
    _, frame = cap.read()
    frame = cv2.flip(frame, 1)
    cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
    img = Image.fromarray(cv2image)
    imgtk = ImageTk.PhotoImage(image=img)
    lmain.imgtk = imgtk
    lmain.configure(image=imgtk)
    lmain.after(10, show_frame)

show_frame()
root.mainloop()
```

Python Code for Update Model

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

fishface=cv2.face.FisherFaceRecognizer_create()
data={}

def update(emotions):
    run_recognizer(emotions)
    print("Saving model...")
    fishface.save("model2.xml")
    print("Model saved!!")

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

    for emotion in emotions:
        training=sorted(glob.glob("dataset/%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

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))
```

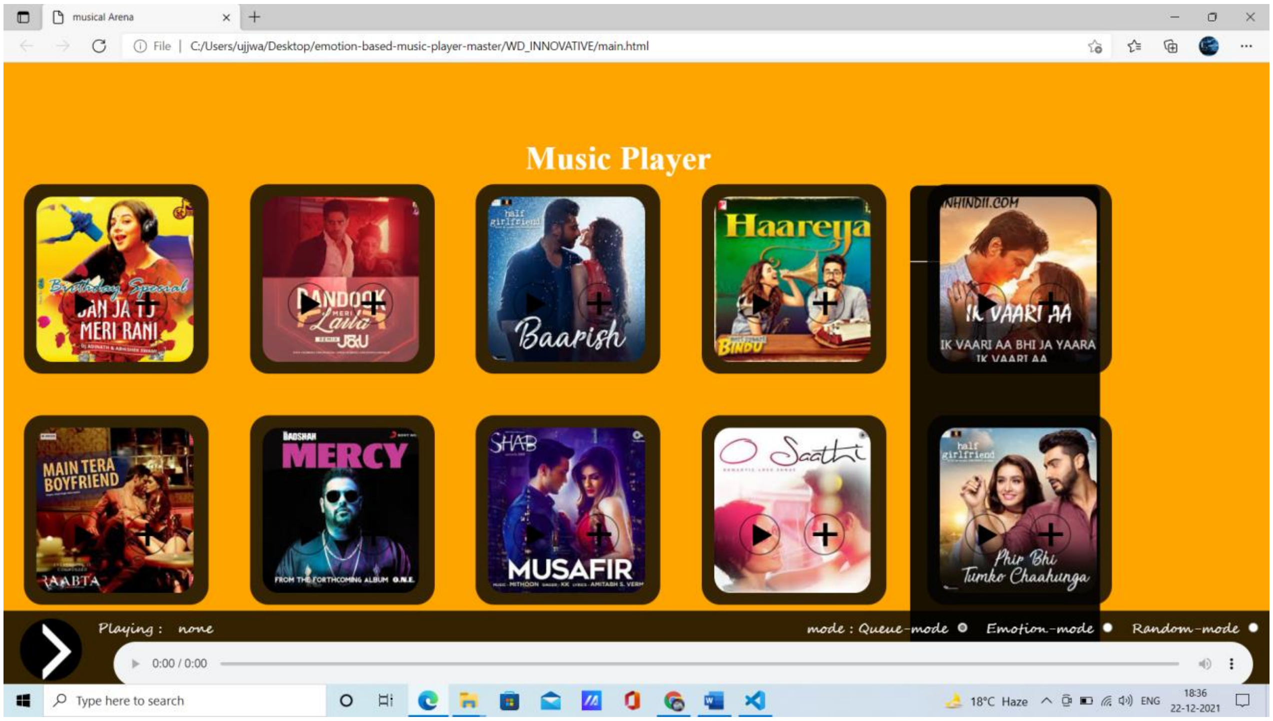


fig-10. Website Interface

Chapter—5

Conclusion and Future Works

So, in a way, this model is very useful and is of great use to all the person who want to listen a song according to mood. In our present state of work, we have proceeded with the music from the database that is available online. In the near future we can proceed with music online but with many more options like selecting a particular language, latest songs, not very new songs and old songs like 80's or 90's and many more. The future scope of work aims at holding music therapy sessions that will definitely help many who are suffering from mental illness, who are sad but want to be happy, etc. The mobile application-based emotion detection-based music player has to be user-friendly with more and more options. We can also create apps and websites with this feature and include speech recognition along with facial emotion detection

The following features can make this model very effective –

In future, the main focus is to add new features to the model. Rather than predicting the type of disease, it would give solutions to that particular disease as well which will solve the problem of physical meet with the doctors.

The model will also be implemented in Android App also. The Android app will work same as this model which will have many additional features.

REFERENCES

Websites—

[1]. https://wikipedia.org

[2]. https://www.softwaredevelopment.in

[3]. https://mlappdevelop.com

[4]. https://MLAppworld.com

[5]. https://en.wikipedia.org/wiki/

[6]. https://developer.org 52

[7]. http://diseaseprediction.com

[8]. www.help.com

[9]. www.justdial.com

[10]. https://patents.google.com.

Books—

[1]. Software Engineering – Beginners to Advanced.

[2]. Machine Learning Model Development

[3]. PL/SQL- by P. Steven.

[4]. Machine Learning Engineering.

Other Reference

[1] Journal Ilia Cursor, Volume 6, Issue 3, 2012

[2] Nikhil Aware, Teas Arguer, Amy Beading, D.D. Sakpal"Emotion Based Music Player" International Journal of Innovative Research & Development, Volume 3,Issue 3, 2014

[3] James, H. Immanuel, J. James Anto Arnold, J. Maria MasillaRuban, M. Tamilarasan, and R. Saranya. "EMOTION BASED MUSIC RECOMMENDATION SYSTEM." EMOTION 6, no. 03 (2019)