

REACTION BASED AUDIO PLAYER

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

PALETI SRIHARI (Reg.No:37110535)

PACHAVA VENKATA GOUTHAM (Reg.No:37110525)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF
COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE
AND TECHNOLOGY
(DEEMED TO BE
UNIVERSITY)**

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved
by AICTE

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

March - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC I 12B Status by UGC I Approved by
AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONA FIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of PALETI SRIHARI (37110535), PACHAVA VENKATA GOUTHAM (37110525) who carried out the project entitled "REACTION BASED AUDIO PLAYER" under my supervision from November 2020 to March 2021.

Internal Guide

Dr. T. PREM JACOB, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

WE PALETI SRIHARI(37110535) and PACHAVA VENKATA GOUTHAM(37110525) hereby declare that the Project Report entitled **REACTION BASED AUDIO PLAYER** done by us under the guidance of **Dr.T.PREM JACOB M.E., Ph.D**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree COMPUTER SCIENCE AND ENGINEERING.

DATE:

PLACE:CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.T.PREM JACOB M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Recent studies confirm that humans respond and react to music and that music has a high impact on person's brain activity. People tend to listen to music based on their mood and interests. This project focuses on creating an application to suggest songs for user based on their mood by capturing facial expressions. Using the device's camera, we can capture the user's facial expression. Facial expression is a form of nonverbal communication. Computer vision is an interdisciplinary field that helps convey a high-level understanding of digital images or videos to computers. In this system, computer vision [18] components are used to determine the user's emotion through facial expressions. Once the emotion is recognized, the system suggests a play-list for that emotion, saving a lot of time for a user over selecting and playing songs manually. Apart from facial expression, our system also provides the flexibility of playing selected songs or play the shuffled playlist.

TABLE OF CONTENTS

CHAPTER.NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	ii
1	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	1
	1.2 OVERVIEW	1
2	AIM AND SCOPE	3
	2.1 AIM	3
	2.2 SCOPE	3
	2.2.1 APPLICATIONS	4
	2.2.2 DATASET	4
	2.2.3 TRAINING PROCESS	4
	2.2.4 FACIAL EXPRESSION DETECTION	4
3	SYSTEM IMPLEMENTATION	5
	3.1 PROPOSED SYSTEM	5
	3.2 SOFTWARE REQUIREMENTS	5
	3.3 HARDWARE REQUIREMENTS	6
	3.4 INTRODUCTION TO PYTHON	6
	3.5 APLICATIONS	6
	3.6 DATASET	7
4	PROBLEM IDENTIFICATION AND OBJECTIVES	9
	4.1 FISCHER FACE ALGORITHM	9
	4.1.1 FORMULA USED BY FISCHERFACE	10
	4.2 RESIZING THE IMAGE	10
	4.3 GRAY SCALING IMAGES	11
	4.4 EMOTION DETECTION	11
	4.5 MODES OF PLAYING MUSIC	11

5	METHODOLOGY	13
	5.1 COLLECTING DATA	13
	5.2 LOADING AND SAVING TRAINED MODEL	14
	5.3 HAARCASCADE MODEL	14
	5.4 RESULT CALCULATION	15
	5.5 PLAYING MUSIC	15
6	CONCLUSION	17
	REFERENCE	18
	APPENDIX	19
	A.. SOURCE CODE	19
	B. SCREENSHOTS	24

CHAPTER -1

INTRODUCTION

1.1 OUTLINE OF PROJECT

Everyone was interested to listen to music. Music is important in everyone's life. Music is a great healer if begin and end your day with music. It's play important role in human life. Recent studies confirm that music is good for health.

Most people are interested to listen to songs related to their current mood. For example, if a person's mood is happy then he tends to listen to happy songs. If the user in a sad mood then he tends to listen to sad songs. Nowadays, humans are interested in listening to music. There are so many music apps in the market like wynk, savaan, etc. These music apps are just the basic needs of the user. The user still faces difficulty in selecting songs to play based on their mood. These music players apps have various features like playback, increasing and decreasing the speed of the song, fast-forward but these are just basic needs of the user. So, that we came with an idea of a reaction-based audio player.

1.2 OVERVIEW

The main purpose of the project is to capture a facial image of the user and to display the emoji of the user and play the songs related to the emotion of the user. We capture the facial image of the user with the use of a system camera. The user's facial image should not be a blur for emotion detection. We use the Haar cascade classifier to detect facial expression. We need the datasets to train the algorithms. We get data from the Cohn Kanade database. we use the fisherface algorithm to detect the emotion of the user. after displaying the emoji automatically playlist related to the emoji will play

The facial recognition method can describe or verify a person using a digital image or a picture frame from a picture document. There are various modes in which facial recognition schemes work, but as a summary, they get by comparing selected facial characteristics from given representation with looks within the information. It is also identified as a Biometric Machine learning-based program that can uniquely describe the person by analysing patterns from the individual's facial textures and body. More frequently than not, a facial recognition application

is utilized for safety purposes. Starting from individuals' handsets' face unlocking feature to governing buildings, businesses, banks, finances, and military alike harness the force for thwarting crime and providing the right services to the trusty ones. However, using the technology for nothing but protection could be silly, as it will do a lot more. Facial recognition technology is quickly spreading. It is starting to follow areas like education, family safety, and personal industries, making the efficiency of recognition more robust and more guaranteed.

We look up to the emotion-based music player; Its concept revolves around identifying a person's facial expressions via analyzing a set of instantaneously taken images and running specific machine learning models over it. Fishface of the CV2 library is the prime machine learning module that is considered to generate the required results. The reason for considering the CV2 module is that the enormous growing global community for python is updating OpenCV's two classifiers: The Haar Classifier and the Local Binary Pattern (LBP) classifier and are continually being fed with more data and being trained vigorously than any other open-sourced facial recognition module. The model is also considered an excellent resource for the instantaneous capturing and analysis of facial images. The entire technique happens in a straightforward and complex-free manner where all the work the user needs to do is turn our music playing application to get it started.

There are three steps in learning our emotion recognition model: feature extraction, subset feature, and classifier of emotion. The Haar Cascades method is used to detect the face from a given input image. Based on the extraction of eyes, mouth, and other minute constituents, the inputs are passed to the Sobel edge detection to obtain the characteristic values. There are Four types of different emotional categories can be obtained through classifier training.

The potential of this technique of facial emotion recognition is a whole, and hence the scope for adding more features to this model is enormous.

CHAPTER -2

AIM AND SCOPE

2.1 AIM

The main purpose of the project is to capture a facial image of the user and to display the emoji of the user and play the songs related to the emotion of the user. We capture the facial image of the user with the use of a system camera. The user's facial image should not be a blur for emotion detection. We use the Haar cascade classifier to detect facial expression.

2.2 SCOPE

The problem of Music Emotion Recognition is a very interesting field of study and poses a lot of application. This gave scope for the related and well known work done by experts in the field of mood recognition. Here are few reputed works.

Nowadays, everyone was interested to listen to music because music plays a key role in reducing stress. Recent studies confirm that listening to music can have psychological benefits such as cognitive performance, reduce stress, eat less, sleep better, improve motivation, etcetera. This usage can be benefited to all ages of people. There are different types of genres in music such as rock music, pop music, art music, funk, Latin music, and so on. Music has a high effect on humans brain activity. People tend to listen to specific music genres on basis of their mood and interests. This project mainly focuses on creating an audio player to recommend songs for the user based on their present mood; sad, happy, neutral, and angry. There are two ways to find the current mood of a person either the user's facial image or the user's heart rate. In this project, we use the facial image to find the current mood of the user. While using already existing algorithms is very slow, low accurate and every so often uses additional hardware. In this project, we are using the Haar cascade method to extract the facial features and then we use the fisherface algorithm to identify the reaction of the user. There are various ways to get datasets for training the algorithm. In this project, we are using the Chon Kanade dataset. If the user's detected reaction is happy then the audio player plays the happy songs.

- Applications
- Data set
- Training process
- Face mask detection

2.2.1 APPLICATIONS

The applications are the library materials used in the project for the python .This includes , Numpy , OpenCV-python , Update Model, eel library, Tkinter, Argparse. These are to be imported or installed into the python inorder to work the program. Command prompt is opened to install all the applications which are needed as applications with a - command

2.2.2 DATA SET

First we collected a dataset form the internet. We are collecting different types of dataset with different types of facial expression dataset. These are the main training mods which help to train the program

2.2.3 TRAINING PROCESS

We are train the collected video frames using Deep leaning techniques. Because, it will train to multiple times and push a very fast results. Deep Learning is a one of the best algorithm for making a project and it is also give an accurate output.

2.2.4 FACIAL EXPRESSION DETECTION

Finally,we capture image using system camera ,we maachine learning techinqe using python.we use haar cascade method to dectect human face in image and use fisherface algorithm to detect facial expression and play song according to user mood.

CHAPTER -3

SYSTEM IMPLEMENTATION

3.1 PROPOSED SYSTEM

we use music player's frequently in our daily life.at present they are so many music players but they have basic features:

1. Selecting songs based on your wish
2. Shuffle songs
3. Classify the songs manually to particular emotions.

At present they are many music players have been developed with many features like forward, local playback, multicast stream. even though these features are available users need an easy way to select a song. Users don't need to select every song based on their current mood. So, we need need to create a music player with additional features like a reaction detection player where the system captures the image of the user and predicts the reaction. Then the music player plays the songs according to user reaction. We propose a reaction-based audio player. which plays songs based on the user's current mood. reaction-based audio player is an autogeneration audio player. this audio player recognizes the reaction of the user using a machine learning algorithm. to find the current mood of the user we need to capture the image of the user using a system camera. after capturing user images, we use the haar cascade method and fisherface algorithm to get the reaction of the user. we need to train datasets for algorithms. we get the training dataset from Cohn Kanade. we built a website for an audio player using javascript, html, and css. this is an audio player using machine learning for detecting reactions based on the user's facial expression of the user. the interface is made up of javascript, html, and css, and the main code in python.

3.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 , 8, 10 (64 bit)
- Software : Python 3.7
- Tools : Comand Promt

3.3 HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above

3.4 INTRODUCTION TO PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

3.5 APPLICATIONS

The applications are the library materials used in the project for the python .This includes cv2,argpars,time,os,Update_Model,glob,random,eel, Numpy , Opencv-python , Matplotlib and Scipy. These are to be imported or installed into the python inorder to work the program. Command prompt is opened to install all the applications which are needed as applications with a command .

cv2

argparse

time

os

Update_Model

glob

random

eel

tkinter as tk

numpy as np

glob

random

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

Fig 3.1 importing applications

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

Fig 3.2 importing applications

3.6 DATA SET

we can get facial expressions with the use of fisherface with the help of a trained model. they are some benefits to allow users to take dataset. we take the dataset according to user uses. we have the option of taking a huge dataset or fewer datasets. but sometimes we get benefits by taking fewer datasets. we use a huge dataset to get higher accuracy of output but in some cases, we use less data for some benefits. for example, suppose we take a huge dataset of 20- 25k it will give high accuracy. in some situations like that the user of the device area few people. now in such cases, we take 500- 600 images as input related to the user that we will get high accuracy and with some added benefits such as less amount of dataset required and less amount of memory is used to store the data. using input related to user benefits not only less amount of memory stored but also small

memory of data give output quickly which result in rapid response time. we use the Cohn-Kanade dataset to train the models. we made some classification based on our needs in the training model.



Fig 3.3 cohn kanede datasets

CHAPTER -4

There are two main problems to solve, the first one being the identification of a face and its emotional status to get ahead with the classification. The second one is to provide a minimal and simplistic approach to use our music player by any individual. By looking into the camera, we gaze at our phones to open our screens, tag friends in photos on social media, even to validate our identities at the ATMs, powerful AI are being used. Since majority of them are owned by private organisation and tech giants, they are not accessible to general public. Henceforth making use of the available Open ML modules is the key as they are equally capable. CV2 is our considered choice. As its Fishface module is a highly performing resource. Firstly, as we access the user's camera, we will be collecting a set of instantaneous images that will be later used for emotion classification. The reason for taking multiple images is that python will be able to use the best possible images as per the requirements it need and then pick the most relevant ones while also following a mixed analysis of the entire set to compare the results. This helps in delivering the most accurate results. To address the other question of why we do not collect a video footage and run algorithms on it is that, a video computation takes extensive load over the GPU (Graphics Processing Unit) and the general computers with minimal range of computational power might end up failing in getting the job done. Also, the modules we tend to use are more picture oriented and susceptible to errors if used for a video. For training data model the collected images stored in the folders will be used and will be mapped with the emotion it is most inclined towards. These data we collect from the user (facial images) will be stored at an instance in a dictionary and then use it for further training the model.

4.1 : Fischer Face Algorithm

The FischerFace or the FishFace algorithm is used for high end purposes of facial recognition like face identification. It uses Discriminant Analysis to find the facial features to discriminate between the persons. But since we focus on identifying a person's facial expression instead of identification of a person, the process gets

comparatively fast and easy. We can expect a nice reconstruction of the original image if it is taken in a well illuminated area as it helps the algorithm to understand the expressions better. The Fisherface is especially useful when facial images have large variations in illumination and facial expression.

4.1.1 : Formula used by Fischerface:

Let X be a random vector with samples drawn from c classes:

$$X = \{X_1, X_2, \dots, X_c\} \quad X_i = \{x_1, x_2, \dots, x_n\}$$

The scatter matrices S_B and S_W are calculated as:

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^c \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

where μ is the total mean:

$$\mu = \frac{1}{N} \sum_{i=1}^c x_i \quad \text{And } \mu_i \text{ is the mean of class } i \in \{1, \dots, c\} :$$

$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$ 10 Fisher's classic algorithm now looks for a projection W , that maximizes the class separability criterion:

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

4.2 : Resizing the image:

It mainly relates to the size that can give an accurate output, regardless of the image we have chosen for the dataset. The size is selected so that the model can be easily differentiated by haarcascade model from face to image. And the size that we get from the real-time scan is not always the same as the data, so we resize it to the exact size of the model data. We have picked 350*350 in our scenario.

4.3 : Gray scaling images:

Though colour pictures sound much appealing than the grayscale images, it is the vision of the machine that is important for us as we give the input data to it. Since the module better understands an image based on the variation in brightness and contrast by comparing individual pixels, Gray scaling can help generate better results. Hence we use certain modules to convert the image and give it a Gray scale tone.

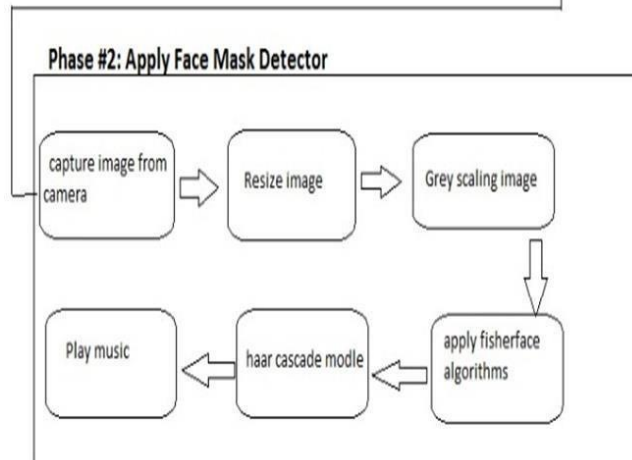
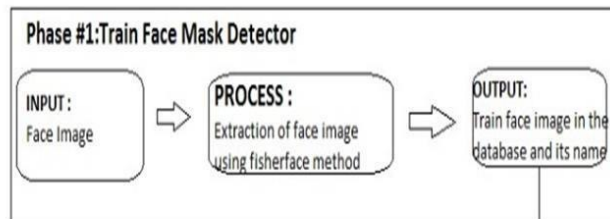
4.4 : Emotion Detection:

The relation between python and javascript via the eel library has been introduced here. That gives us the privilege of JavaScript and python methods. Since the library is implemented in python, the flow will be in python language, and post its process completion the control is then leveraged to the html and JavaScript. As a result, we get to view emoticons in accordance with the outcome.

4.5 : Modes of playing music:

Since music is a great source to get change in mindset and relax ourselves, our mood may not be complimenting a goal we look forward to. Hence the user gets to see three different modes in which of which one can be chosen.

- Emotion based music
- Queue
- Random music



CHAPTER -5

METHODOLOGY

This application requires laptop with camera. In order to get accurate output, we need to collect data and train the model with the data collected. For this to be done we used Fisherman algorithm.

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.

The steps we took to implement the model are:

5.1 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.

5.2 Loading and saving trained model:

For training, We used Fisherface method of cv2 library. For training data model we wrote a python code to grab all the classified images from folders and map it with it's emotion. This data at this instance is stored in dictionary and then used it for training the model.

5.3 Haarcascade model:

Haarcascade model is precise face detection trained model which is provided by Open-cv. It returns the co-ordinates in terms of (x, y) at (left, bottom) of face frame and width, height from these co-ordinates. We made use of `.detectMultiScale()` method it is capable of detecting multiple expressions and it returns an array of all the face co-ordinates as an element.

Haar cascade model is a face detection model, which is used to identifies objects in the image. this method was proposed by paul and Michael they are four stages in the haar cascade:

1. calculating haar feature
2. creating internal images
3. AdaBoost
4. cascading classifier.

To train the classifier we need positive and negative images. positive images -the images which contain the image which the classifier wants to identify. negative images -the images which don't contain the image which the classifier wants to identify. we need to train the classifier so we need positive images with human faces and negative images without human faces. They are three features in haar cascade

1. edge feature
2. line feature
3. four rectangle feature.

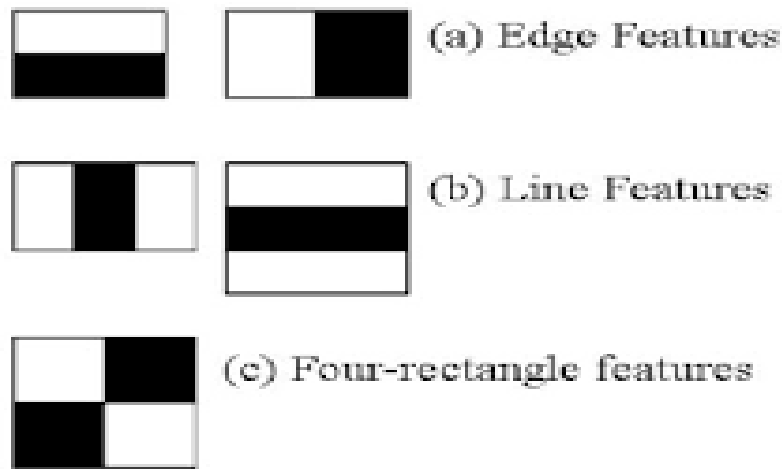


Fig 2: Haar Features

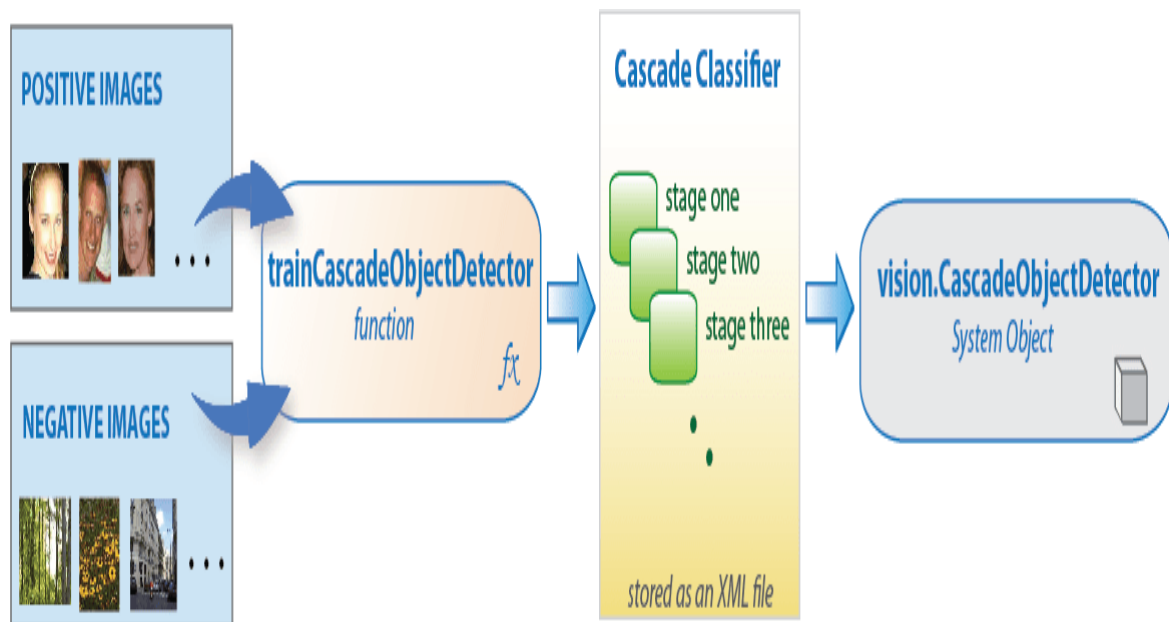


Fig 3:Working of Haar Cascade

5.4 Result Calculation:

In our model we did not stick with one image for testing, When the code run's it take around 10 images in a short time(1-2 sec) and for all these images it will compute result and according to the average value it will give result.

5.5 Playing music:

We linked 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 emotions(considered 4 emotions in this model).

• • • •

Sad happy angry neutral

Methods for playing songs

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

Queue

Based on Emotion

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 song will be played. In third one we can directly use random function and all the methods used in this mode 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.

CHAPTER -6

CONCLUSION

The project is about predicting human facial expressions. Based on the resulted face, It plays music. To achieve that, we used Machine Learning and Open-CV in this project.

The music player's interface comprises HTML, CSS, JS, and main code in Python. The player will contain three modes:

- Queue Mode
- Emotion Based
- Random Mode

In the first mode as queue works it has been implemented. The Python code gets emotion from the user's facial expression using the above-implemented algorithm in the Emotion mode. In the random mode, we directly used the random function, and all the methods are dynamic. It can handle a change in the number of songs accordingly.

REFERENCES

- [1] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, "Theano: a CPU and GPU math expression compiler", Proceedings of the Python for Scientific Computing Conference (SciPy), Jun. 2010.
- [2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, "Scikit -learn: Machine learning in Python", Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
- [3] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project", ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pp. 108-122, 2013.
- [4]<https://www.youtube.com/watch?v=ellfb5D3HO8>
- [5]<https://www.youtube.com/watch?v=4W5M-YaJtIA>
- [6]<https://thecodacus.com/opencv-face-recognitionpython-part1/>
- [7]<http://www.paulvangent.com/2016/04/01/emotionrecognition-with-python-opencv-and-a-face-dataset/>
- [8]<https://pythonprogramming.net/haar-cascade-face-eyedetection-python-opencv-tutorial>

APPENDIX

A.SOURCE CODE:

Capture.py

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

frequency=2500
duration=1000

eel.init('WD_INNOVATIVE')
emotions=["angry", "happy", "sad", "neutral"]
fishface = cv2.face.FisherFaceRecognizer_create()
font = cv2.FONT_HERSHEY_SIMPLEX
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()
```

```

#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("model.xml")
            return identify_emotions()
            break

```

```
#eel.start('main.html', options=web_app_options)
#options={'host':'file', 'port': '///'}

eel.start('main.html')##WD_INNOVATIVE//main.html")
#, options)
```

van Gent, P. (2016). Emotion Recognition With Python, OpenCV and a Face Dataset. A tech blog about fun things with Python and embedded electronics. Retrieved from: <http://www.paulvangent.com/2016/06/30/making-an-emotion-aware-music-player/>

'''

Display.py

```
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()
```

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("model.xml")
    print("Model saved!!")
```

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

```
    for emotion in emotions:
        training_data+=sorted(glob.glob("dataset/%s/*" %emotion))
        for item in training_data:
            image=cv2.imread(item)
            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))
```

B.SCREENSHOTS



face1



face2



face3



face4



face5



face6



face7



face8



face9



face10



main0



