Module 3: - Emotion Audio Integration

Emotions extracted for the songs are saved as a meta-information within the database. Mapping is 18 executed by way of querying the meta-statistics database. The emotion extraction module and audio feature extraction module is eventually mapped and mixed the use of an Emotion-Audio integration module.

4.3.1 Audio Emotion Recognition

In music Emotion popularity block, the playlist of a consumer bureaucracy the input. An audio sign first of all undergoes sure quantity of pre-processing. for the reason that, song documents obtained from the net are usually stereo alerts: all stereo alerts are transformed to 16-bit PCM mono sign at a sampling rate of forty-four.1 kHz, is known as Audacity. Con model of a stereo signal into mono sign is critical to lessen the mathematical complexity of processing the similar content material of both the channels of a stereo sign. Is then extracted from the complete audio signal Audio documents are typically very huge in size. they are computationally expensive in terms of reminiscence. consequently, to reduce the reminiscence complexity of an audio file incurred throughout audio function extraction, an eighty sec window is extracted from each audio report. This pre-processed signal then undergoes audio function extraction, where centroid, spectral flux, spectral roll off, kurtosis, zero-crossing rate and thirteen MFCC coefficients are extracted. Toolboxes used for audio feature extraction consists of MIR 1.5. Auditory Toolbox and Chroma toolbox. track based emotion popularity is then done the use of CNN. Audio indicators are labelled below 7 categories. even as various mood models had been proposed inside the literature, they failed in capturing the perceptions of a person in actual time. they're figments of theoretical aspects that researchers have related to one-of-a-kind au dio indicators. The temper version and the instructions considered in this paper don't forget, how a user may additionally companion extraordinary moods with an audio sign and a tune. even as composing a tune, an artist might not hold the uniformity of a mood across the whole excerpt of a song.

Songs are generally based on a theme or a scenario. A track may boast sadness in first half even as the second one 1/2 can also become cheerful. as a result, the model followed on this paper takes into con- sideration these kinds of factors and generates paired classes for such songs, aside from the individual training.

The model adopted is as follows. All the ones songs which can be cheerful. active and playful are classified beneath pleasure. Songs which can be very miserable are categorized below the magnificence sad. Songs that replicate mindset, anger related to patriotism, and are revengeful are classified beneath anger.

The class pleasure Anger is associated with songs that own anger in a playful mode. sad-anger category is composed of all those songs that revolve across the subject of being extremely depressed and angry. All other songs other than these fashionable categories falls under the opposite class and whilst a person is detected with feelings including surprise and worry, the songs from the opposite class are cautioned.

4.4 Conclusion

Depending on the project subject, the design phase products include diagrams, flow-charts, sketches, site trees, HTML screen designs, and UML schemas. The project team use these designs to choose the definitive design that you can produce in the project. The Design Document is developed by the team, identifying the steps used in the design of the application/system.

5. IMPLEMENTATION AND RESULTS

5.1 Introduction

The programming language which we used for developing our project in Python. Python is a general-purpose language, which makes it easy to collaborate across the project because it has a simple syntax. Python has very extensive inbuilt libraries for data science and various Image classification algorithms used in Machine Learning, Deep Learning, and Artificial Intelligence Since importing and implementing the Deep Learning libraries is very easy in Python.So, we opted for Python for this Project. We used Open CV in our project Open CV library is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces. CNN and Haar Casade Classifier is used in our project.

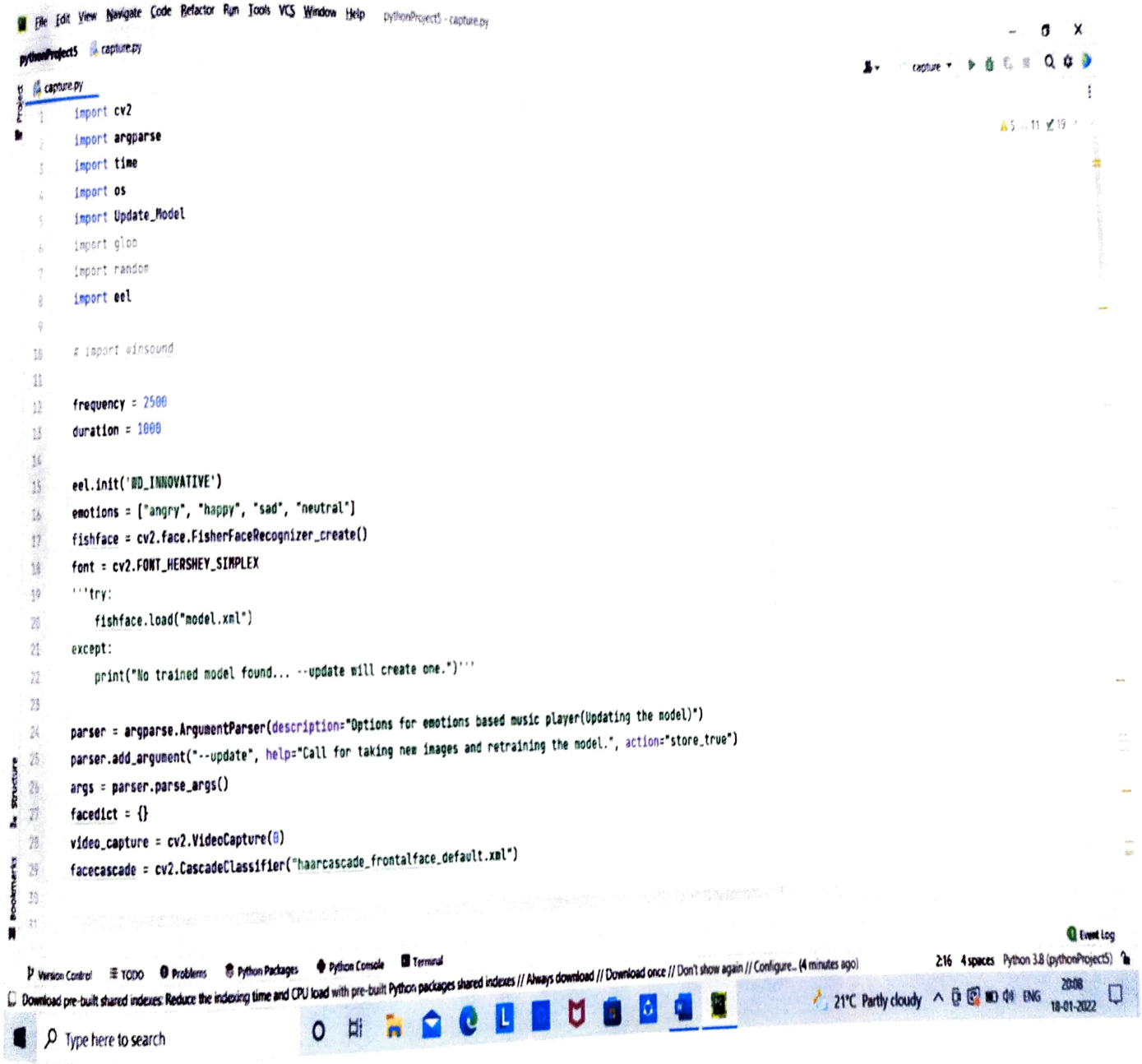
5.2 How does the Feature Extractor work?

Feature Extractor is the class which is used to get all the features required for the emotion detection. First, an array is initialized to store all the classifier xml files and the dimensions of the images of the features. The start method is where the class starts doing its stuff. The test images used in this project are stored in the location specified in the String testImagesLoc. While the features are extracted, they're also temporarily cached, and these are stored in the location specified in cacheLoc.

First of all, the OpenCV library is loaded in the start method. Then a test image is loaded, and the features are extracted from this image. Here, first the face is extracted from the image, and then the features are extracted from that facial image. The detect Features method is used to obtain all the features that are detected based on the classifier used. It also temporarily saves the features extracted in the cache directory. The extract Feature method is used to actually extract the features based on all the detections. It is a recursive method, and the integer counter is used to differentiate between the code which deals with extraction of face, and extraction of features from a single face. So, basically, a face is extracted from an image, and the different classifiers are used on that particular face to get the various features out of it.

5.3 Input and Output

5.3.1 Input Design



5.3.1: **Input Design of project**

5.3.2 Output Design

5.3.2: **Output Design of project**

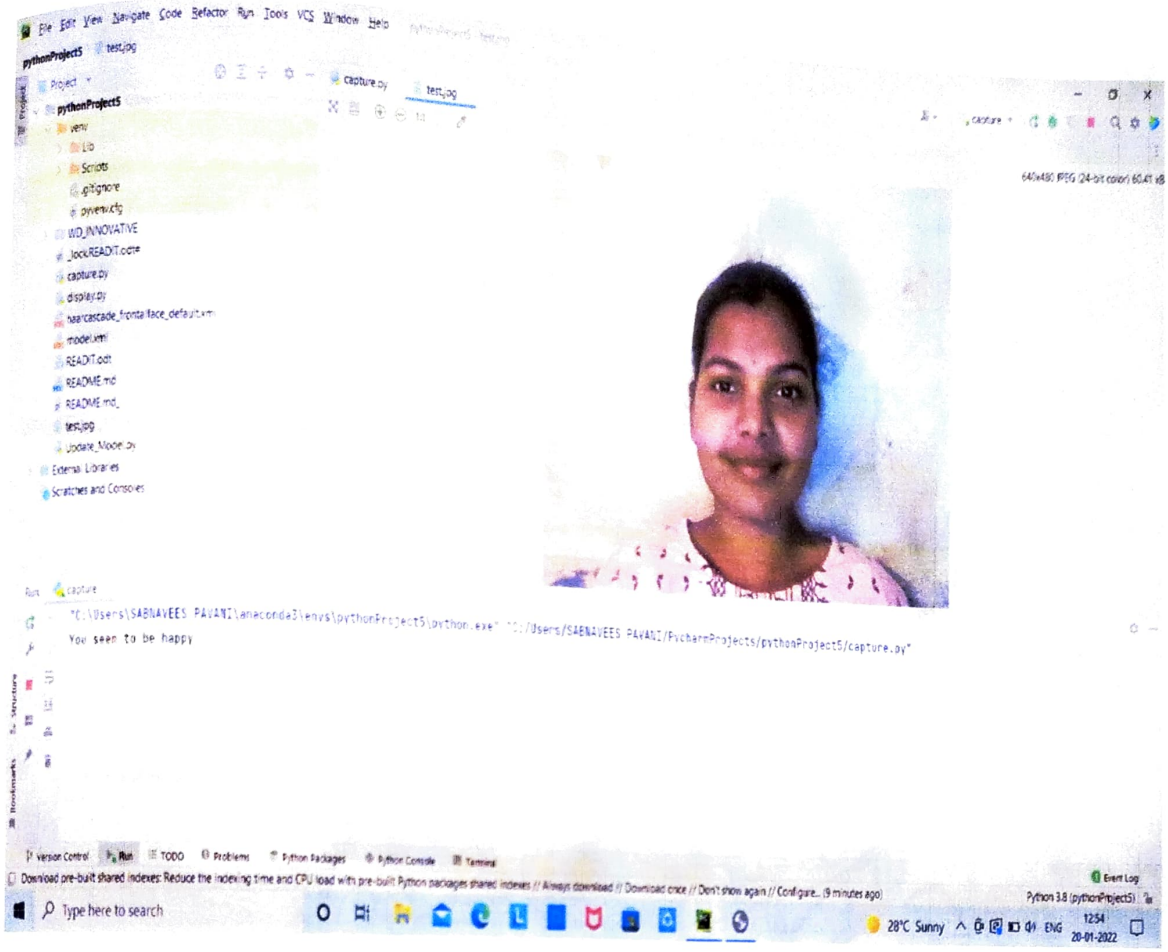
5.4 Code Implementation

5.4.1 Sample Code

```
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
'''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()
    # 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 + " until 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!!")
```

5.4.2 Output Screens



5.4.2.1: **Output of Detected Camera**

The image of the user is saved in the test.png file

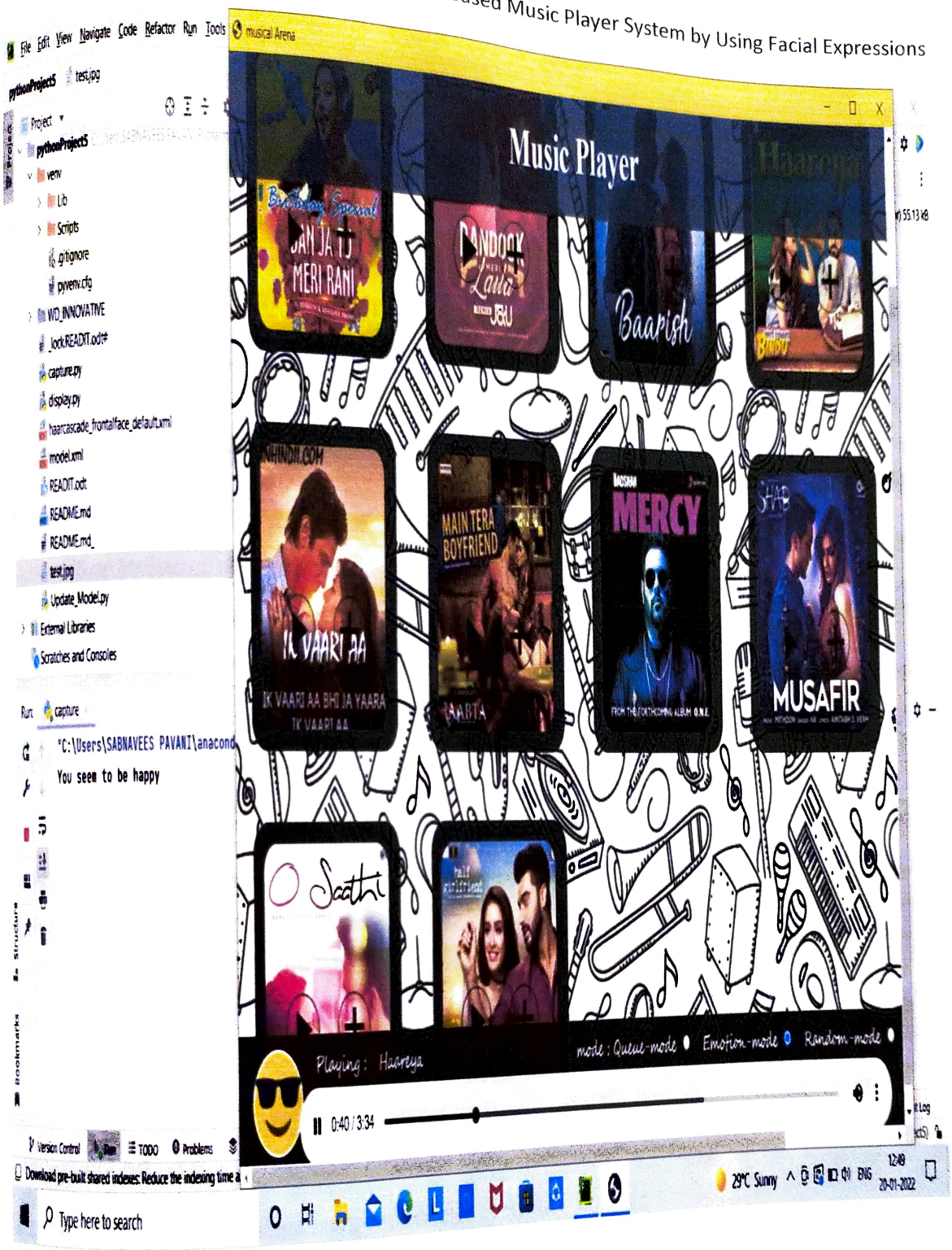


Figure 5.4.2.2: **Output of Music**

5.5 Conclusion

The principle idea of this challenge is to, robotically play songs based on the emotions of the consumer. Its pursuits to provide consumer preferred song with emotion attention. In current machine per- son need to manually choose the songs, randomly played songs may not healthy to the temper of the person, consumer has to classify the songs into diverse emotions and then for playing the songs user has to manually pick a selected emotion. those difficulties can be averted with the aid of using Emo participant (Emotion based track player). The feelings are recognized using a neural community studying approach CNN set of rules. CNN may be used for identifying and regain the photos. in keeping with the emotion, the song could be performed from the predefined directories. Comparison of Existing and Proposed System.

For Existing system, even though those functions satisfy the users primary necessities. but the user has to face the project of manually browsing through the playlist of songs and select songs based totally on his contemporary mood and behaviour that is the requirements of an individual. A user sporadically suffered through the need and choice of browsing via his playlist, consistent with his mood and emotions.

For Proposed system, the music participant itself selects a track in line with the current mood of the user. the usage of a software to reduce the efforts of coping with playlists. We said a manner to mechanically detect the temper of the consumer and generate playlist of songs which is appropriate for the modern temper. The photo is captured using webcam and that photograph is handed below exceptional degrees to locate the mood or emotion of the user. The software is for this reason developed in one of these ways that it could manipulate content material accessed by person, examine the picture homes and determine the mood of the consumer. Advantages of the Proposed System

1.Users don't want to select song manually.

2.No need of playlist.

3.Users don't want to classify the songs based on the emotions

6.TESTING AND VALIDATION

6.1 Introduction

The testing is finished to discover that there are any mistakes or mistakes after of entirety of the challenge. So, it'll no longer motive any trouble to the people the usage of our project.

6.2 Types of Testing

6.2.1 Unit testing

For Unit testing, we are going to studying taking pictures of pics had been we can gather it from webcam and ship to the trained statistics that is achieved. Haar Casade set of rules detects the value of every pixel of the photograph and undergoes numerous functions to extract the emotion kingdom of the face. Unit checking out is a technique by way of which character devices of supply code—units of one or extra computer program modules collectively with related manipulate Information, regression methods are examined to determine whether they're in shape to be used.

Input

```
bottle==0.12.19
bottle-websocket==0.2.9
cffi==1.14.5
Eel==0.14.0
future==0.18.2
gevent==21.1.2
gevent-websocket==0.10.1
greenlet==1.0.0
numpy==1.20.1
opencv-contrib-python==4.5.1.48
opencv-python==4.5.1.48
pycparser==2.20
pyparsing==2.4.7
whichcraft==0.6.1
zope.event==4.5.0
zope.interface==5.3.0
```

6.2.2 Integration testing

Integration checking out is conducted to assess the compliance of a system or element with distinctive functional necessities and are in- tended to check integrated programming parts to decide whether they definitely run as one software.

Here we accomplished the mixing checking out, wherein we re- moved an important bundle in our application and carried out the execution and then there is an error. Mixture testing is explicitly planned for unproductive issues that emerge from the mixture of elements.

Input

```
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
'''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")
```

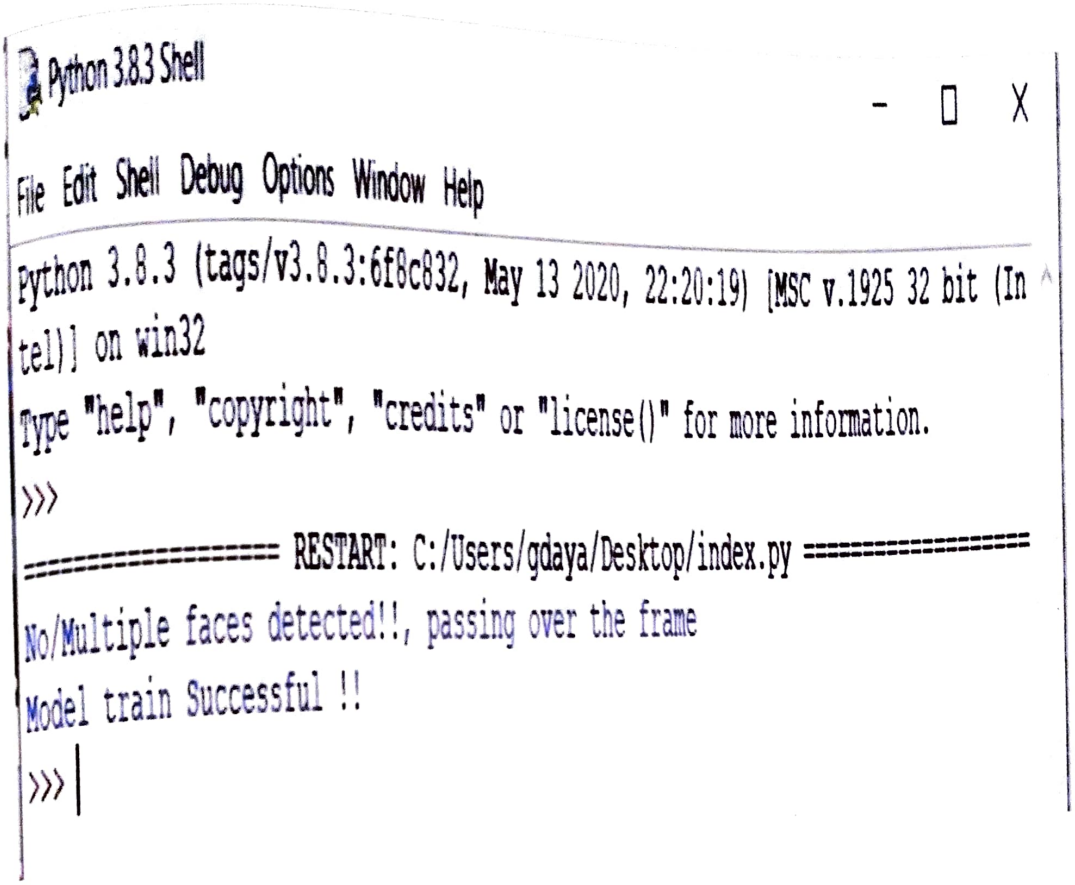
6.2.3 Functional testing

Input

```
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!!")
```

Test Result



6.2.3: **Functional Testing**

6.2.4 White Box Testing

white box testing is a software program checking out technique in which the internal shape of the item is understood to the tester. The tester chooses enter attributes which make a contribution extra in case of prediction price and expel the values and determines the anticipated charge appropriately. its miles utilized to test areas which can't be reached from a discovery stage.

```
import cv2
video_capture=cv2.VideoCapture(0)
def nolight():
        return video_capture.read()
```

6.2.5 Black Box Testing

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. Black box testing can be used to test for functional and non-functional system requirements, though the majority of black box testing focuses on functional requirements.

```
import Update_Model

emotions=["angry", "happy", "sad", "neutral"]
Update_Model.update(emotions);
```

6.2.6 Test Result

S.NO	TESTING	STATUS
1	UNIT TESTING	PASS
2	FUNCTIONALITY TESTING	PASS
3	INTEGRATION TESTING	PASS
4	WHITEBOX TESTING	PASS
5	BLACKBOX TESTING	PASS

Table 6.2.6: **Test Results**

6.3 Testing Strategy

A way for structure trying out fuses device exams and shape frame- works into an interior and depiction. The testing system ought to collaborate check organizing, check arrangement, take a look at execution, and the consequent statistics aggregate and assessment .A system for programming testing need to in shape low-stage checks which are important to watch that a touch source code segment has been exactly found out similarly as raised degree checks that choose massive machine limits towards customer necessities.

6.4 Conclusion

Set of images for each emotion (neutral, sad, angry and happy) are saved in the proposed model for the comparison purposes. The newly load images will be compared with the saved dataset in order to detect the emotion of the users. Task Executing Software for multiple users. Free Trial Project Testing Phase means a group of activities designated for investigating and examining progress of a given project to provide stakeholders with information about actual levels of performance and quality of the project.

7. CONCLUSION

7.1 Project Conclusion

Experimental results have shown that the time required for audio feature extraction is negligible and songs are stored pre-handed the total estimation time of the proposed system is proportional to the time required for extraction of facial features. Also, the various classes of emotion yield a better accuracy rate as compared to previous existing systems. The computational time taken is 1.000sec which is very less thus helping in achieving a better real time performance and efficiency.

The system thus aims at providing the Windows operating system users with a cheaper, additional hardware free and accurate emotion-based music system. The Emotion Based Music System will be of great advantage to users looking for music based on their mood and emotional behaviour. It will help reduce the searching time for music thereby reducing the unnecessary computational time and thereby in- creasing the overall accuracy and efficiency of the system. Also, with its additional features mentioned above, it will be a complete system for music lovers and listeners.

7.2 Future Enhancements

The proposed system might have many functions and it may be user friendly but the proposed system can have further advancement in future. The future scope in this system will be to create a mechanism that will be helpful in music therapy treatment and will provide the music therapist needed to treat patients suffering from disorders such as mental stress, anxiety, acute depression, and trauma. The proposed system is currently available on windows operating system, In future it will be available for the user using different operating system such as ios,ubuntu,etc. and mobile phone platform as well.

The proposed system tries to avoid unforeseen results generated in the future in extremely poor lighting conditions and very poor camera resolution. In the proposed work only one emotion is detected at a time so that it can be further enhanced to detect mixed emotion.

8. REFERENCES

8.1 Referred Books

- SOFTWARE ENGINEERING – Sommerville

8.2 Referred Sites

- https://en.wikipedia.org
- https://www.javatpoint.com
- https://www.tutorialspoint.com

8.3 References

[1] S. Gilda, H. Zafar, C. Soni and K. Waghurdekar," Smart music player integrating facial emotion recognition and music mood recommendation," 2019 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2019.

[2] Abat, F., Maaoui, C., and A. Prusk. Human-computer interaction using emotion recognition from facial expression. 2017 UKSim 5th European Symposium on Computer Modeling and Simulation (2017).

[3] Zeng Z, Pantic M, Roisman GI and Huang TS 2008 A survey of affect recognition methods Audio, visual, and spontaneous expressions IEEE transactions on pattern analysis and machine intelligence 31 39-58.

[4] S. G. Kamble and A. H. Kulkarni, "Facial expression-based music player," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016. Rahul Hirve1, Shrigurudev Jagdale2, Rushabh Banthia3, Hilesh Kalal4 K.R. Pathak5 1, "EmoPlayer - An Emotion Based Music Player" 2016 IJIR Department of Computer Engineering, Vol-2, Issue-5, 2016 ISSN: 2454-1362.

[5] Gupte A, Naganarayanan A and Krishnan M Emotion Based Mu- sic Player XBeats International Journal of Advanced Engineering Research and Science 3 236854.

[6] 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

[7] ParulTambe, YashBagadia, Taher Khalil and Noor UlAin Shaikh 2015 Advanced Music Player with Integrated Face Recognition Mechanism International Journal of Advanced Research in Computer Science and Software Engineering 5

[8] Kanade T, Cohn JF and Tian Y 2000 Comprehensive database for facial expression analysis InProceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition 46-53 IEEE