iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VII **Month of publication:** July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36356

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

ESPOTIFY (An Emotion Based Music Player)

Pushkal Kumar Shukla[1], Radha Singh[2], Ritika Singh[3], Ishita Sharma[4], Prachi[5]
[1]*Assistant Professor, Department of CSE, NIET, Gr. Noida, India*
[2, 3, 4, 5]*Student, Department of CSE, NIET, Gr. Noida, India*

Abstract: ***Human emotion plays an essential role in social relationships. Emotions are reflected from verbalization, hand gestures of a body, through outward appearances and facial expressions. Music is an art form that soothes and calms the human brain and body. To analyze the mood of an individual, we first need to examine its emotions. If we detect an individual's emotions, then we can also detect an individual's mood. Taking the above two aspects and blending them, our system deals with detecting the emotion of a person through facial expression and playing music according to the emotion detected that will alleviate the mood or calm the individual and can also get quicker songs according to the emotion, saving time from looking up different songs. Different expressions of the face could be angry, happy, sad, and neutral. Facial emotions can be captured and detected through an inbuilt camera or a webcam. In our project, the Fisherface Algorithm is used for the detection of human emotions. After detecting an individual's emotion, our system will play the music automatically based on the emotion of an individual.***

Keywords: Feature extraction, emotion recognition, haar cascade technique, expressions.

I. INTRODUCTION

According to recent studies, it is confirmed that human beings tend to respond and react to songs or music and that music has a high impact on human's brain activity. On average, an Indian listens to up to four hours of music daily.

In today's era, music is a source of entertainment, and People tend to listen to songs according to their mood. Our project focuses on creating a system that will play songs according to the user's mood, and for detecting the mood, we will use Fisher face Algorithm. Human beings convey nonverbal information through facial expressions, though many other animals' species show facial expressions. Human emotions that the face can display can be categorized into four basic expressions: happiness, neutral, anger, and sad. Our project uses Fisherface and cv2 to determine the user's emotion through facial expressions. The webcam captures the image of the person, i.e. the user. It then extracts all the facial features of the user. After detecting the emotion, our suggested system will play the song according to the user's mood to alleviate the mood or calm the individual and get a quicker song according to their mood, saving much time from looking up different songs. Manually sorting and selecting music would consume a lot of time and effort. Many projects already do the same job to match the mood and classify the songs according to them. In one of the projects, they asked the user to enter the mood manually and suggest the song list. In another project, the user needs to pick the mood from the mood tab and then the application plays the song from YouTube.

Figure 1 Music player system

Some music players have the features like adding music into the queue or adjusting the song's speed, etc. all these features are good, but they only satisfy the basic need of the music lover. They fail to arrange the music according to mood.

There are different approaches to determine the mood of a person or user and accordingly play songs, but the thing is that they require some additional devices such as sensors to detect the mood, and sometimes they fail to produce accurate results.

Our project, espotify, determines the user's mood and then automatically plays the songs according to their mood. It provides adequate and more accurate results than the existing systems available also, and it does not require additional devices such as sensors.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

II. LITERATURE SURVEY

Here are a few research papers which are associated with the current system which includes the description of the respective system.
Many projects already do the same job to match the mood and classify the songs according to them. In one of the projects, they asked the user to enter the mood manually and suggest the song list. In another project, the user needs to pick the mood from the mood tab, and then the application plays the song from YouTube.
Some music players have the features like adding music into the queue or adjusting the song's speed, etc. all these features are good, but they only satisfy the basic need of the music lover. They fail to arrange the music according to mood.
There are different approaches to determine the mood of a person or user and accordingly play songs, but the thing is that they require some additional devices such as sensors to detect the mood, and sometimes they fail to produce accurate results.
Our project, espotify, determines the user's mood and then automatically plays the songs according to their mood[4,5]. It provides adequate and more accurate results than the existing systems available also, and it does not require additional devices such as sensors.
Emophony [6] This application takes the emotions and interests of the user as input and then suggests songs accordingly. For detecting emotions, it uses an AI system. After detecting the user's emotions, it automatically plays the song, and a playlist based on mood is generated, which contains different genres of songs. In this system, the FER- 2013 dataset was used for the training purpose. Different emotions present in the dataset are disgust, fear, surprise, etc.
Emotion based music recommendation system, in this system; they took the point that for determining a person's mood, its face plays a vital role. In this webcam was used for capturing the face of the user and later to detect the emotion. Histogram of Oriented gradients, image pyramid, and linear Classifier techniques was used to detect the emotions.
Emotify [11], this system uses Microsoft Face API for recognizing human faces in a photograph. In this, they used Microsoft technology which uses different cloud database-based face algorithms for detection and recognition of the user's emotion. The Face API was used to detect different emotions such as disgust, neutral, fear, etc. The Face API in this system generates JSON format responses.
Emotion based music playing device [12], In this project, the Support Vector Machine (SVM) algorithm was used to detect the user's emotions. This system detects the user's emotions through facial expression, and then it automatically plays the song according to the user's mood. In this, the dataset contains emotions such as disgust, fear, happiness, etc. In this project, the songs were sorted according to the emotions and saved in the folder.
Face Emotion based Music Player System [7], In this project, they used a Convolutional Neural Network Algorithm for detection and recognition of users' emotions. After detecting the user's emotions, it automatically generates a playlist according to the user's mood.

III. PROPOSED SYSTEM

Our project Espotify is a music player based on emotion that detects and recognizes the user's emotions and then plays songs according to the user's mood [1,2,3]. Our system has a dataset that contains different emotions such as sad, neutral, angry, and happy. We have used the front-End for a music player with chrome.
We have a real-time image capturing via webcam or an inbuilt camera. After capturing the user's image, it then compares it with the provided dataset, which is saved in the system locally and predicts the emotion with the help of the ML algorithm [8,9,10]. After recognizing the emotion, it automatically plays the songs based on the mood of the user. For detecting emotions, we have used the Fisherface algorithm.
Fisherface is regarded as one of the best algorithms for face detection, and it is also believed to be better than other available techniques. An XML file is generated as output after training the model. All the information about the dataset, i.e., expression and emotions, is stored in a model XML file. Also, it stores the size of the training model. for integrating front-end and back-end, we have used the EEL library. EEL library helps to access the JavaScript, HTML code from the python code and vice versa.
Our project consists of three modes: emotion mode, queue mode, and random mode. Buttons are created in HTML code, and their functionalities are in JavaScript code. In queue mode, songs are added to the queue, and songs are played according to the queue. In order to use emotion mode, the user needs to select the emotion mode, and then python code will detect and recognize the user's emotions and play the songs according to the user's emotions. Furthermore, in the last mode, the random mode will randomly play the song without detecting and recognizing the user's emotions.

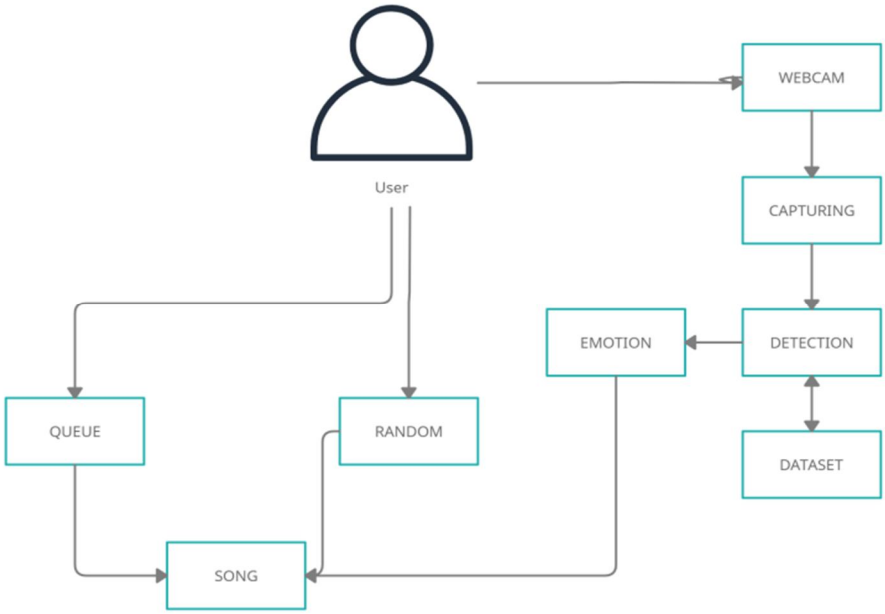


Figure 2 Architecture of our proposed system.

A. Algorithm

1) Open the application.

2) Select the mode (Queue, Emotion, and Random).

3) If the selected mode is emotion, then initiate the camera capture the user's image, analyze it and play songs according to the mood.

4) If the selected mode is the queue, play the songs added to the queue.

5) If the selected mode is random, songs will be played randomly from the songs list.

IV. METHODOLOGIES

A. Detecting Emotions

1) Collecting Data: Facial expression detection in Fisherface works with the help of trained models. The reason behind this is to allow users to take a dataset according to their use. Suppose if we take a massive amount of dataset of around 25-30k, it will give excellent accuracy no doubt, but if the situation is like that the users of the devices are a few people. In such conditions, if we take some specific dataset with around 400-450 images as input related to the user, it will also give good accuracy with the benefit of less amount of dataset and less storage on memory to operate. As well as a small memory of data gives output fast, which results in quick response time. We first tried with the Cohn-Kanade dataset; then, we made some classification as needed to train our model.

2) Data Loading and saving trained model

For training, we have used the Fisherface method of the cv2 library.

Fishface = cv2.face.FisherFaseRecognizer_create()

We have made a python code for the training data model, which grabs all the classified images from folders and maps them with their emotion. These data are stored in a dictionary and then use the .train method to train models.

Fishface.train(training_data, np.asarray(training_label))

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

Fishface.save("model1.xml)

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

Fishface.read("model.xml")

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

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

3) Haar Cascade model

The Haar Cascade model is a precise face detection trained model which Open-cv provides. It returns the coordinates in terms of (x, y) at (left, bottom) of the face frame and its width and height from those coordinates.

clahe_image=grab_face()

facefacecascade.detectMultiscale(clahe_image,scaleFactor1.1,minNeighbors15,minsize(10,10),flagscv2.CASCADE_SCALE_IMAGE) As here in the .detectMultiScale() method it is capable of detecting multiple faces and it returns an array of all the faces(co-ordinates) as an element. The arguments have been set according to the threshold that we need for our checking purpose. We have set it such that it doesn't affect our model accuracy.

B. Machine Learning

1) Fisherface ML Algorithm: Fisherface algorithm is an algorithm that works based on LDA and PCA concepts. Linear discriminant analysis (LDA) is a supervised Learning method of machine learning. It works on the concept of dimensionality reduction, which reduces the execution time among classification. Principal Component Analysis (PCA) is a one kind of conversion from correlated variables to uncorrelated mathematical values. It is mainly used for observing data and from that by some probabilistic calculation generate models. Now supervised Learning is where we use such data whose answer is also given to the model to learn it. The flow of Fisherface is like it takes classified images, then it will reduce the dimension of the data, and by calculating its statistical value according to the given categories, it stores numeric values in a .xml file. While prediction also calculates the same for a given image, compares the value with the computed dataset values, and gives the result a confidence value.

2) Resizing Images: Whatever image we have chosen for the dataset is mainly related to the size, giving a precise output. The size is chosen such that the model can easily distinguish a face from the image by haar cascade model. Furthermore, the size of what we get from a real-time scan is not always the same as data (significantly less difference) so, We resize it to the exact model data size. In our case, we have chosen 350*350.

```
defcrop (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 the image by given parameters of the haar cascade by clahe_image[] and use of cv2's method .resize() to the given size. Finally, we have stored those images in the dictionary and, after some counting (=10), take them to check the result.

3) Gray Scaling Images: It was the need for the method, and because of its contrast and shaded face, it resulted in a benefit for the algorithm to get output.

4) Face Detection: The code grab_face() methods are used to get the images, do all operations, and return cropped, grayed face values in the dictionary.

```
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

5) Train and Predict Method

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

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

C. Playing Music

1) Detected Emotions: We have implemented the linking of python with JavaScript through the eel library, which provides us the privilege to access python methods from js and vice versa. Here the striating flow will be in python code as the library is implemented in python, then it transfers the control to HTML, JS. Furthermore, according to the result, we show emoticons

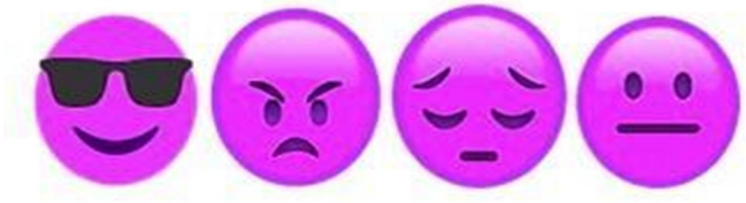


Figure 3 Emoticons showing emotions (Happy, Angry, Sad, Neutral)

According to which we can classify the emotion directory for playing songs we have chosen these 4 emotions.

2) Methods for playing songs

In the JavaScript file we have implemented too many methods for the switching of songs.

a) Queue
b) Based on Emotion
c) Random

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

3) HTML, CSS and JS Concepts for Online Music Player: The CSS gives a great look to communicate, and through JS, we can interact with the user and not look like a complicated program run at the console, and it also gives the user privilege to choose any song to play.

V. RESULTS AND SNAPSHOTS

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

Given below are the snapshots of three modes, i.e., queue mode, emotion mode, and random mode.

1) Queue mode

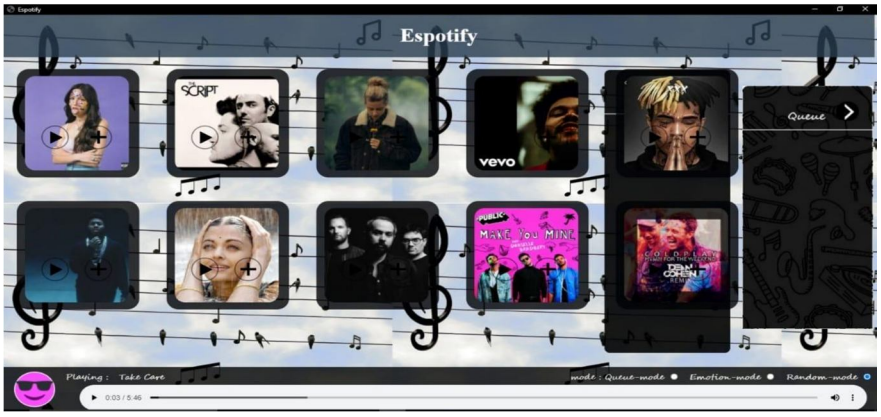


Figure. 4 Implementation of queue mode

In this mode, songs are added to the song queue, and then according to the queue, songs are played.

2) *Emotion Mode*: In emotion mode, we have different modes in the dataset, such as sad, angry, happy, and neutral. First image of the user is captured by webcam or an inbuilt camera, then the image will be analyzed, and their emotion is detected; then, according to emotion, the song will be played

a) Happy

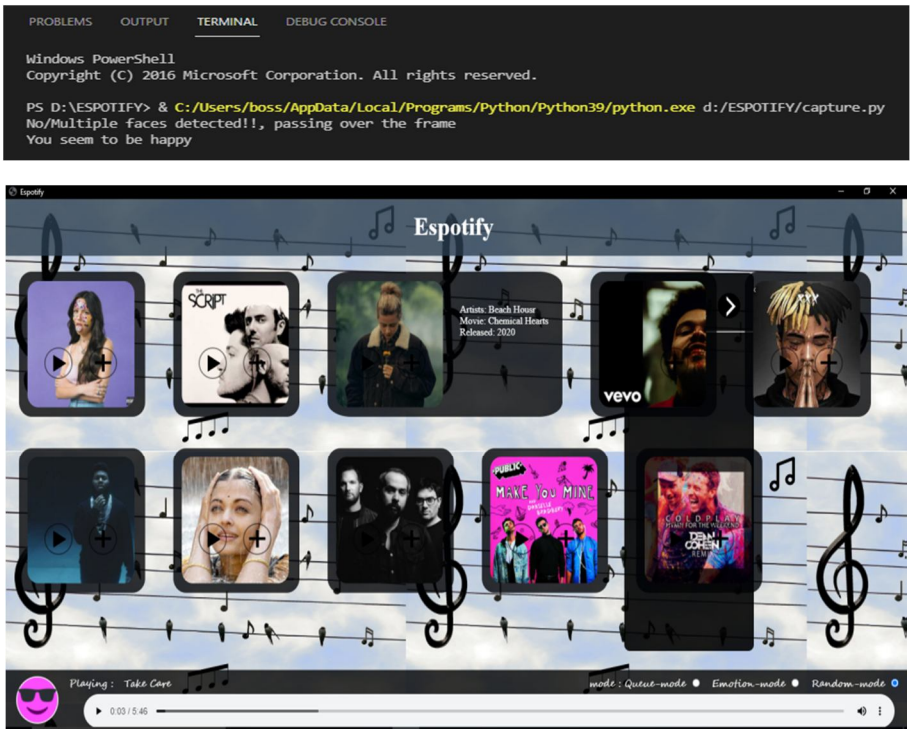


Figure. 5 Detection of happy emotion

b) Sad

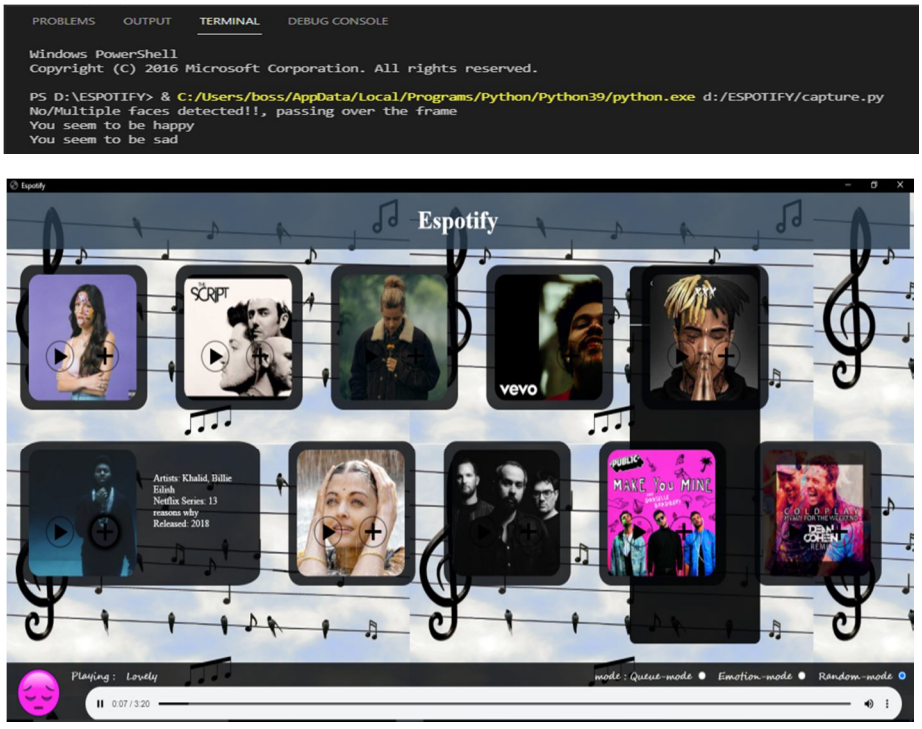


Figure. 6 Detection of sad emotion

c) Angry

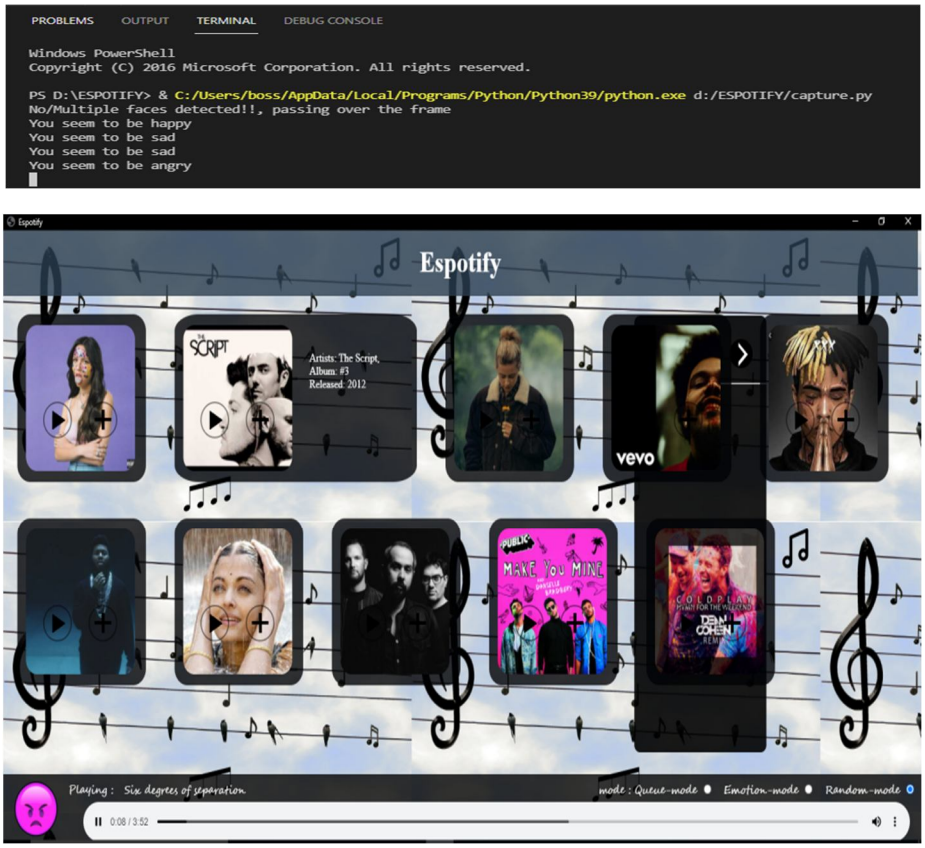


Figure. 7 Detection of angry emotion

d) Neutral

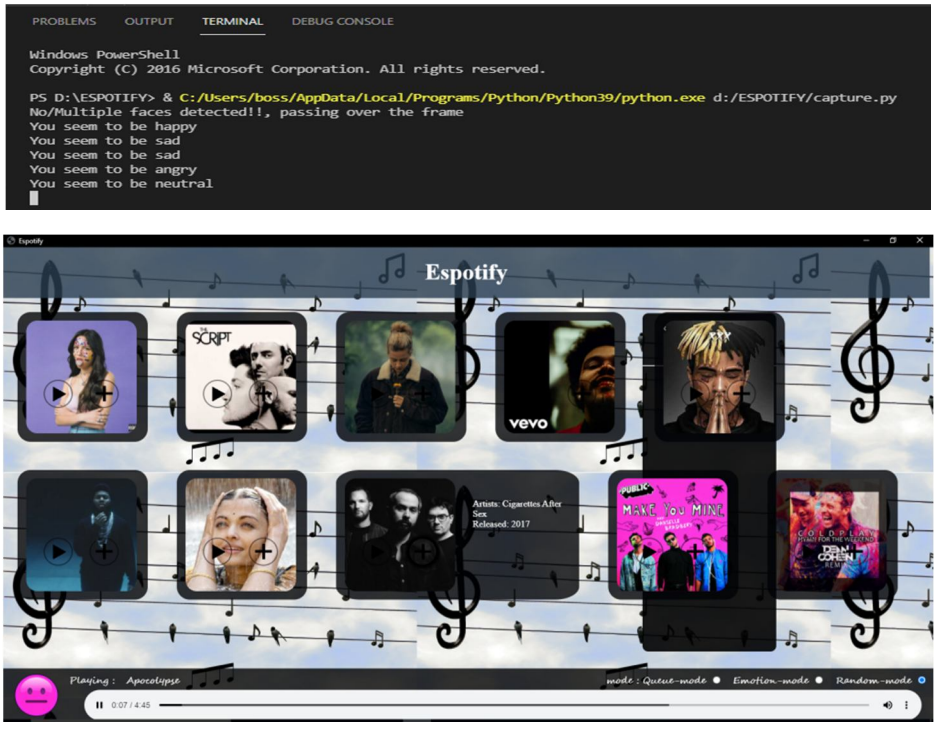


Figure. 8 Detection of neutral emotion

3) Random Mode: In random mode, songs will be played randomly. No queue is maintained in this mode.

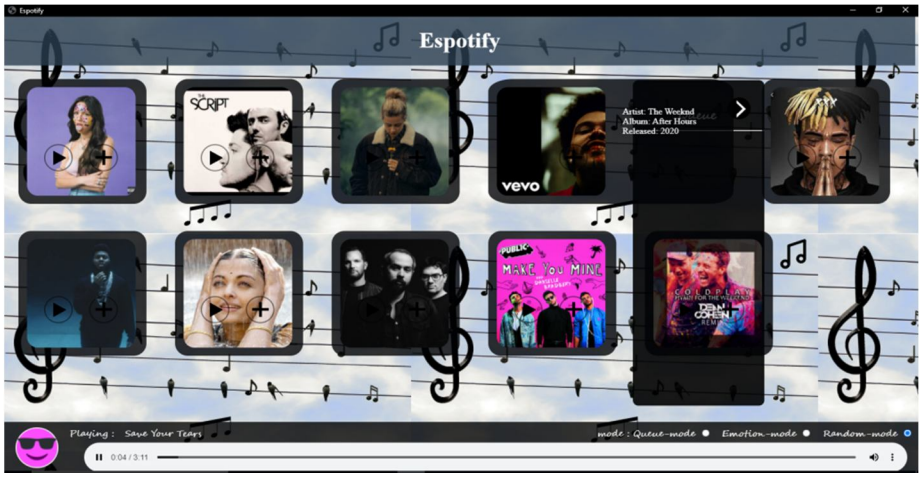


Figure. 9 Implementation of random mode

V. CONCLUSION

Our project avoids manual work and gives a better music player experience for the end-user. It solves the basic needs of music lovers without troubling them as existing applications do: it uses technology to increase the interaction of the system with the user in many ways. It eases the end-user's work by capturing the image using a camera, determining their emotion, and suggesting a customized play-list through a more advanced and interactive system.

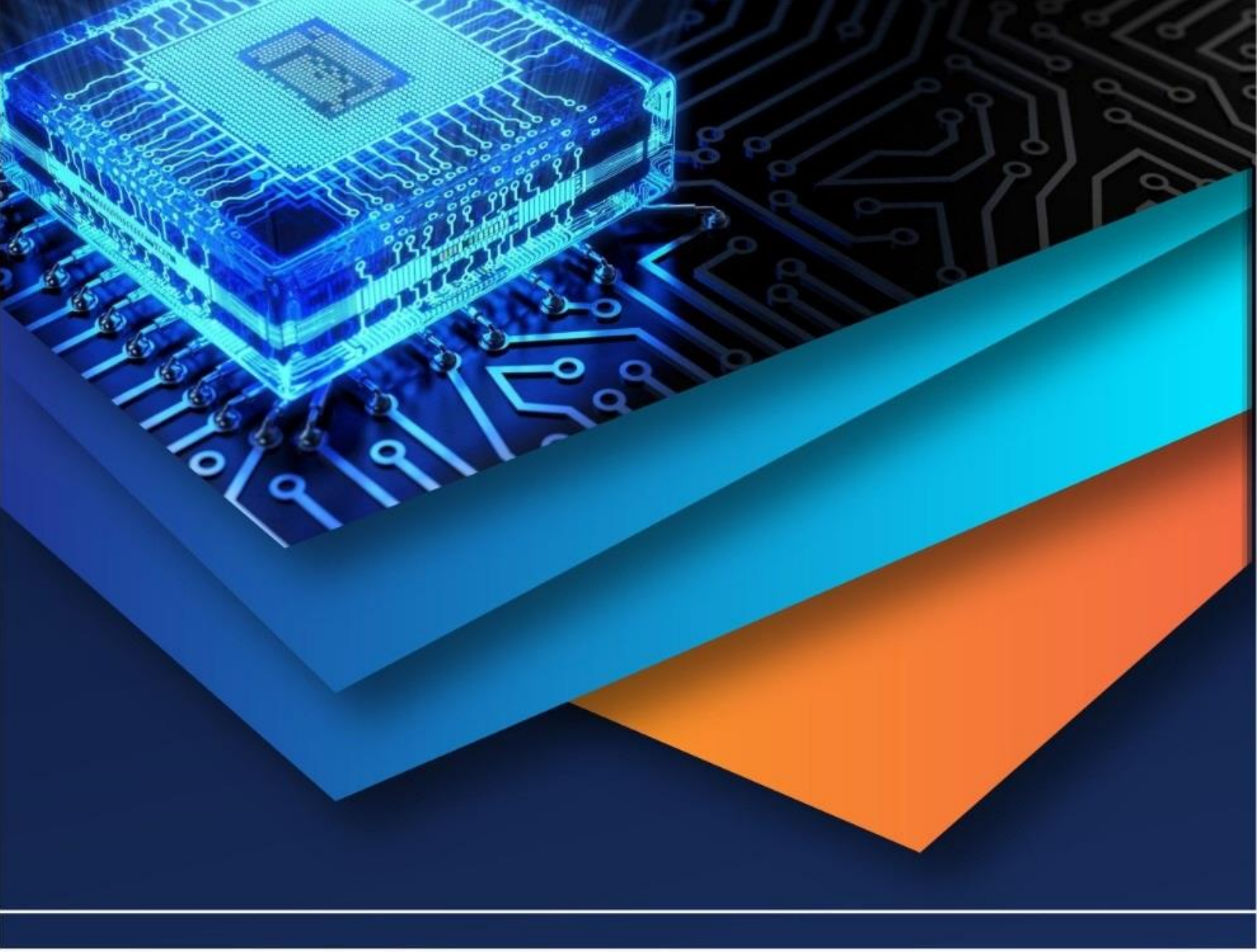
VI. FUTURE SCOPE

Our project can be further improved by modifying and adding few functionality.

A. Including other emotion
B. Optimizing the algorithm by including additional features which helps the system to categorize users based on many other factors like location and suggesting the user to travel to that location and play songs accordingly.
C. Detecting human faces from a video.
D. Android Development

REFERENCES

[1] Seangliet, Y.Lee, B.S., & Yeo, C.K. "Mood Prediction from Facial Video with Music Therapy On a Smartphone", 2016 Wireless Telecommunications Symposium (WTS)
[2] Seungjae Lee, Jung Hyun Kim, Sung Min Kim, & Won Young Yoo. (2011). Smoodi: Mood-based music recommendation player, 2011 IEEE International Conference on Multimedia and Expo.
[3] Patil, J. V., & Bailke, P. (2016). Real time facial expression recognition using RealSense camera and ANN, 2016 International Conference on Inventive Computation Technologies (ICICT).
[4] Byeong-jun Han, Seungmin Rho, Roger B. Dannenberg and Eenjun Hwang, ―SMERS: music emotion Recognition using support vector regression, 10thISMIR, 2009.
[5] Viola, P., &Jones, M.(n.d.), Rapid object Detection using a boosted cascade of simple features. Proceedings of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.
[6] Emophony – Face Emotion Based Music Player Banpreet Singh Chhabra – IRJET (JUNE 2020)
[7] Face Emotion based Music Player System Babita Sonare, Sarvesh Thube, Sandesh V Sirsat, Pranita M Ghode, and Akshay Shimpi – IRJET (2020)
[8] Arushi Raghuvanshi and Vivek Choksi, "Facial Expression Recognition with Convolutional Neural Networks", CS231n Course Projects, Winter (2016)
[9] Real Time Facial Emotion Recognition based on Image Processing and Machine Learning,Rituparna Halder ESL, Sushmit Sengupta ESL, International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, April 2016.
[10] Building Emotional Machines: Recognizing Image Emotions through Deep Neural Networks Hye-Rin Kim, YeongSeok Kim, Seon Joo Kim, In-Kwon Lee,IEEE TRANSACTIONS ON MULTIMEDIA Citation information:DOI 10.1109/TMM.2018.2827782, year:2016.
[11] Hafeez Kabani, Sharik Khan , Omar Khan , Shabana Tadvi"Emotion Based Music Player"International Journal of Engineering Research and General ScienceVolume3,Issue1, January-February, 2015.
[12] Dhruvisha Bansal, Pinkal Bhatt, Megha Dusane, Avneet Saluja, Kushal Patel. EMOTION BASED MUSIC PLAYING DEVICE. International Research Journal of Engineering and Technology. Volume 07 Issue 06 (2020).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY