International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Moodify: Smart Music Player based on Facial Emotions

M. Srinayani[1], P. Naga Jahnavi[2], S.Kavishree[3]

[1]*Student, Dept. Of Computer Science and Engineering, Sri Chandrasekharendra Saraswati Viswa Mahavidyalaya, Kanchipuram, Tamil Nadu, India*
[2]*Student, Dept. Of Computer Science and Engineering, Sri Chandrasekharendra Saraswati Viswa Mahavidyalaya, Kanchipuram, Tamil Nadu, India*
[3]*Assistant Professor, Dept. of Computer Science and Engineering, Sri Chandrasekharendra Saraswati Viswa Mahavidyalaya, Kanchipuram, Tamil Nadu, India*

Abstract - *Human emotions play a crucial role in analyzing the mood of an individual. Emotions of a person could be understood through various features of the face such as eyes, cheeks, or even through the smile on the face. Music is an art form that helps an individual to calm their body and brain. Our project deals with the blending of emotions and music. The proposed system detects the emotions of the individual through facial expressions and plays the music according to the mood detected. The expressions of the face could be happy, sad, angry, neutral, etc. The emotions on the face can be detected and captured through a webcam or an inbuilt camera. The emotions of an individual from the captured image could be detected using Fisherface Algorithm. Once the emotion is recognized, the system suggests a playlist for that emotion and plays the music automatically.*

***Key Words*: Emotion Recognition; music recommendation; Facial Extraction; Real-time capturing images**.

1. INTRODUCTION

In today's world, music acts as a major source of entertainment. People prefer to listen to songs according to their mood. Selecting and sorting the songs manually would take a lot of effort and time. With the advent of technology, there are many music players which provide facilities such as adding songs to the queue, adjusting the playback speed, etc. Although these features satisfy the basic needs of the user, it fails to sort the songs according to the mood of the user.

There are several techniques that determine the mood of the user and play the songs automatically but, they require few additional hardware devices such as physiological sensors, EEG. These devices and techniques often fail to provide accurate results. Moodify is an immersive, sophisticated, and creative website that determines the mood of an individual and automatically plays the songs according to their mood. It also provides efficient and accurate results than the existing systems as it does not require any additional hardware.

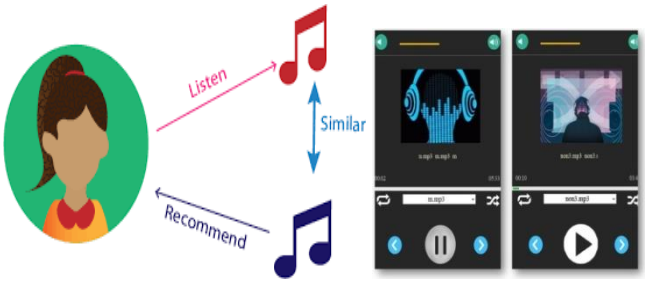

Figure -1: Various Music Player Systems

2. LITERATURE REVIEW

Here are few research papers which are associated with current system which includes the description of the respective system.

Banpreet singh Chhabra (2020), Emophony is an application that suggests songs according to the emotions and interests of the individual. It utilizes an artificially intelligent system to recognize emotions through facial expressions. Once, the emotion of the individual is detected, the system automatically plays music accordingly and suggests a playlist of different genres based on the mood of the individual. Here, FER- 2013 data set is utilized for training the model. The data set consists of emotions like fear, disgust, surprised, etc.

H.Immanuel James, J.James Anto Arnold, J.Maria Masilla Ruban, M.Tamilrasan, R.Saranya (2020), the human face plays a vital role in determining the mood of an individual. The system uses a webcam to capture the face of an individual to later detect the emotion of the user. It uses techniques such as image

pyramid, Histogram of Oriented gradients, and linear Classifier to detect the emotions of the user.

Akshobhya Rao BV, Fathima Rameesha Asokan, Husna Firdous, GP Prerana, Dr. Gopal Krishna Shyam (2020), In this system the microsoft Face API is utilized to recognize the human faces in photographs. It is a technology provided by Microsoft that uses different face algorithms based on the cloud database to detect and recognize the emotion of the individual. In this system, the ace API will detect emotions such as disgust, fear, neutral, etc. The Face API used in this system responds in a JSON format.

Dhruvisha Bansal, Pinkal Bhatt, Megha Dusane, Avneet Saluja, Kushal Patel (2020), This project deals with the detecting emotion of an individual through facial expressions and automatically playing songs according to the mood detected. In this system, the emotions are recognized using Support vector Machine (SVM) algorithm. The dataset taken in the system consists of emotions such as fear, disgust, happy, etc. In this system, the songs are stored in a folder and sorted according to the emotions stated in the dataset.

Babita Sonare, sarvesh Thube, Sandesh V sirsat, Pranita M Ghode, Akshay Shimpi (2020). In this system a playlist of songs would be generated automatically according to the facial expressions of the individual. This system uses the Convolutional Neural Network algorithm to detect and identify emotions. After detecting the emotion, the application returns songs according to the mood of the user.

3. PROPOSED SYSTEM

Moodify is an emotion-based music player that detects the emotions of an individual and plays the songs according to their mood. This system consists of a data set containing emotions such as angry, sad, happy, and neutral. It is a music player with chrome as a front-End to detect the emotions of the user. In this system, the image of a user is captured using a real-time device such as a webcam or an inbuilt camera. It then compares the captured image with the existing datasets that are saved in the local disk of the computer and predicts the emotion of the user with the help of machine learning algorithms. After detecting the emotion, songs are automatically listed and played accordingly.

Fisherface algorithm is used to detect emotions. It is one of the popular algorithms used in face detection and is widely believed to be superior to other techniques. The training model will generate a model XML file as an output. The model XML file stores the information about the emotions and expressions of the dataset. It also stores the size of the training model which would be later used to detect the emotions while testing the model.

EEL library is used to integrate python, JavaScript, and HTML code. It helps to access the JavaScript, HTML code from the python code and vice versa. There are three different modes of playing songs as queue mode, emotion mode, and random mode defined in the HTML code using the radio buttons. Their functionalities are defined in the JavaScript code. Queue mode is used to add songs in a queue and play songs according to the queue. When the user selects emotion mode then the python code is used to detect the emotion of the user and play any song randomly according to the emotion. In the random mode, songs will be played automatically without detecting the emotion of the user.

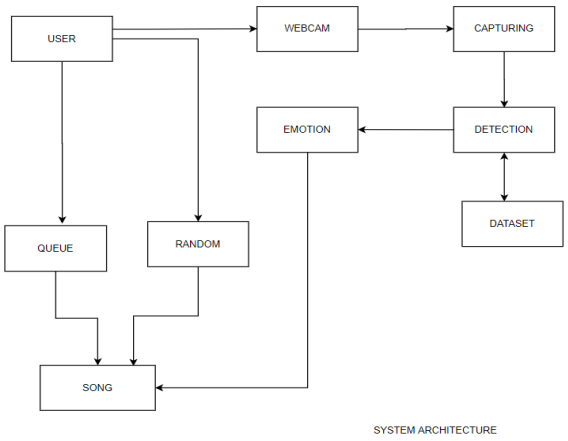

Figure -2: General Architecture of the proposed System

3.1 Algorithm for Proposed System

1) Start the application.
2) Select the mode.
3) If the mode is emotion, capture the face of the user and play song according to the mood.
4) If the mode is queue, play the songs which are added in queue.
5) If the mode is random, the songs will be played randomly from the songs list.

6) Stop the application.

3.2 Data Collection

In the Fisherface algorithm, the detection of facial emotions is done by the trained models. Here we gathered the facial expression dataset from the official website of Kaggle. The dataset includes emotions such as angry, sad, happy, and neutral. A large amount of dataset will result in good accuracy. In the case of a less amount dataset also we will get the same results as previous. The less amount of dataset will occupy less amount of memory which helps in giving the output quickly as the response time will be less. Here we took less amount of dataset to train the model.

3.3 Loading and Saving Trained Model

Fisherface Algorithm is used to train the model. To train the dataset which we collected the classified images are stored in a dictionary by mapping to their corresponding emotion. The dataset stored in a dictionary is used to train a model by the .train method.

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

For saving the trained model .save method is used.

```
fishface.save("model2.xml")
```

The training model will generate a model XML file. The model XML file stores the information about the emotions and expressions of the dataset. It also stores the size of the training model which would be later used to detect the emotions while testing the model.

```
    for emotion in emotions:
        training=sorted(glob.glob("data/%s/*" %emotion))
        for item in training:
            gray=cv2.imread(item,0)
            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))
update(emotions)
```

```
Training model...
The size of the dataset is 800 images
Saving model...
Model saved!!
```

Figure 3: Training the Model

3.4 Haarcascade classifier

Haarcascade classifier is a machine learning algorithm that is used to extract and identify the features of a captured image. There are different types of haarcascade classifiers which are used to identify various features of the face of an individual such as eye cascade and face cascade. It is a machine learning algorithm where a lot of positive and negative images are utilized to train the classifier. In our project, we have utilized haarcascade frontal face _default.xml. It is used to identify and detect objects such as the nose, eyes, and lip in the face. It is designed by the OpenCV to detect the frontal face.

Here, the .detectMultiScale() method is used, it has the capability to detect multiple faces. It will return the faces as an element in the form of an array. The arguments have been passed based on the threshold that we need for results without affecting the model accuracy.

3.5 Fisherface 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 where we use such data whose answer is also given to the model to learn it. LDA works on the concept of dimensionality reduction, which reduces the execution time in classification. Principal Component Analysis (PCA) is one kind of conversion from correlated variables to uncorrelated in the form of mathematical values. It is mostly used for observing data and from that by some probabilistic calculation generates models. 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 an .xml file. While prediction it also calculates the same for a given image and compares the value with the computed dataset values and give according to result with a confidence value.

3.6 Face Capturing and Emotion Detection

The initial process when the execution starts is to access the webcam by selecting the emotion mode in the application. The camera will capture about 10 images of the user. The size of the face captured through the webcam must be of the same size as the dataset that we have chosen. Here the size is chosen so that the haarcascade model detects the face easily from the image captured. So, the image will be resized exactly to the size of image of the dataset model.

```
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, (48, 48))
        facedict["face%s" %(len(facedict)+1)]=faceslice
    return faceslice
```

Figure 4: Cropping the Face

Here, the clahe_image[ ] is utilized to crop the image and .resize () is used to resize the image to the required size. Finally, the images are stored in a dictionary.

The algorithm will need the gray scaling images for getting efficient results because of its contrast and shaded face. The grab_face() method is to get all the captured images and then it will do all operations and finally return cropped, gray scaling face value in a dictionary.

```
def grab_face():
    while True:
        ret, frame = video_capture.read()
        cv2.imshow("Window", frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        cv2.imwrite('test.jpg', frame)
        cv2.imwrite("images/main%s.png" %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
```

Figure 6: Face Detection

The following code is used to get prediction and confidence value for a given amount of image. Then by using the max function will obtain the output and the final result is shown to the user.

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

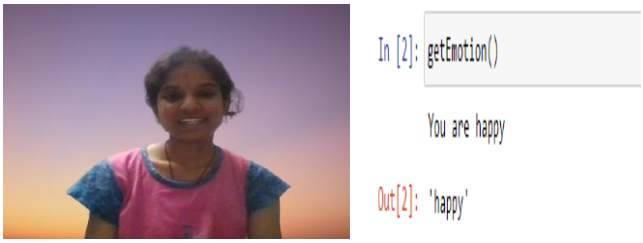

Figure 7: Emotion Detection-Happy

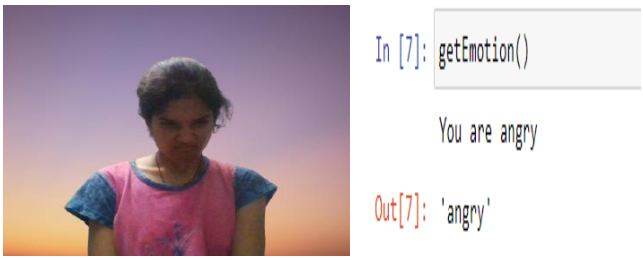

Figure 8: Emotion Detection-Angry

3.7 Music Recommendation

The system has implemented the linking of python with JavaScript, HTML through the EEL library. EEL library provide us the privilege to access python methods from JavaScript and vice versa. The implementation of the system will be in python code as the library is implemented in python then it transfers the control to HTML, JavaScript. Finally, according to the result the emoticons will display.

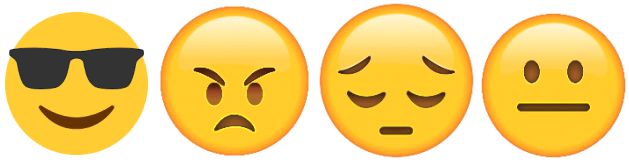

Figure 9: Emoticons (Happy, Angry, Sad, and Neutral) -According to the Classification of Emotions for Playing Songs

The JavaScript code is implemented by adding modes that will be selected by the user.

- Queue-Mode
- Emotion-Mode
- Random-Mode

When the user selects Queue-Mode has selected the songs will be added to the queue and then the songs will play accordingly as in the queue which we added. As the Emotion-Mode is selected the webcam is initiated to capture the face of the user, then the song will be played according to the mood of the user. The Random-Mode is used to play the songs randomly from the songs list.

Figure 10: Selection of Queue-Mode

```
    facedict.clear()
    return output

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')
```

```
Requirement already satisfied: import-ipynb in c:\users\dell\anaconda3\lib\site-packages (0.1.3)
You are happy
```

Figure 11, 12: Selection of Emotion-Mode(Happy)

```
    facedict.clear()
    return output

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')
```

```
Requirement already satisfied: import-ipynb in c:\users\dell\anaconda3\lib\site-packages (0.1.3)
You are angry
```

Figure 13, 14: Selection of Emotion-Mode(Angry)

4. CONCLUSION

The goal of this project is to play songs based on the mood of the person. This project is mainly designed to increase the interaction between the user and the music system. In today's world, people often tend to take more stress and in such situation listening to music would help them to reduce their stress level. This project would serve more beneficial to such people as our project would play songs automatically according to their mood. This system would help to reduce the manual work of searching and playing the songs.

5. FUTURE SCOPE

The project could be further developed by detecting the mood of the user from the video. The system can detect multiple moods of the user for music recommendation.

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] EMOTION BASED MUSIC PLAYING DEVICE sDhruvisha Bansal, Pinka Bhatt, Megha Dusane, Avneet Saluja, Kushal Patel. – IRJET (JUNE 2020)

[8] Face Emotion based Music Player System Babita Sonare, Sarvesh Thube, Sandesh V Sirsat, Pranita M Ghode, and Akshay Shimpi – IRJET (2020)

[9] EMOTION BASED MUSIC RECOMMENDATION SYSTEM H. Immanuel James, J. James Anto Arnold, J. Maria Masilla Ruban, M. Tamilarasan, R. Saranya IRJET (2019)

[10] Arushi Raghuvanshi and Vivek Choksi, "Facial Expression Recognition with Convolutional Neural Networks", CS231n Course Projects, Winter (2016)