CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

People tend to express their emotions, mainly by their facial expressions. Music has always been known to alter the mood of an individual. Capturing and recognizing the emotion being voiced by a person and displaying appropriate songs matching the one's mood can increasingly calm the mind of a user and overall end up giving a pleasing effect. The project aims to capture the emotion expressed by a person through facial expressions. A music player is designed to capture human emotion through the web camera interface available on computing systems. The software captures the image of the user and then with the help of image segmentation and image processing techniques extracts features from the face of a target human being and tries to detect the emotion that the person is trying to express. The project aims to lighten the mood of the user, by playing songs that match the requirements of the user by capturing the image of the user. Since ancient times the best form of expression analysis known to humankind is facial expression recognition. The best possible way in which people tend to analyze or conclude the emotion or the feeling or the thoughts that another person is trying to express is by facial expression. In some cases, mood alteration may also help in overcoming situations like depression and sadness. With the aid of expression analysis, many health risks can be avoided, and also there can be steps taken that help bring the mood of a user to a better stage.

1. **FIRST LEVEL OF HEADING:**

One innovative approach to overcoming these obstacles is the incorporation of facial recognition technology into music recommendation systems. With the help of face recognition technology—especially its capacity to decipher facial expressions—it is possible to ascertain a user's emotional condition in real time. Through the analysis of facial expressions, the system is able to deduce the user's present emotional state and select music that is more in line with that state, resulting in a more tailored and fulfilling listening experience.   
  
The project's problem statement draws attention to the shortcomings of the existing music recommendation systems, which mostly rely on user feedback expressed explicitly and historical data. These systems might find it difficult to adjust to users' sudden and erratic emotional states. The integration of facial recognition technology also presents technological difficulties, such as guaranteeing precise emotion detection and sustaining user privacy.

**Second Level of Heading:**

The primary issue this project attempts to solve is the inability of current music recommendation algorithms to adjust to users' changing emotional states. Conventional methods frequently fall short in explaining the instantaneous emotions and mood swings that shape musical preferences. In addition, there are ethical and technical issues with integrating facial recognition into these systems, like protecting user privacy and guaranteeing precise emotion detection.  
  
The goal of this project is to create a sophisticated music recommendation system that makes use of facial recognition technology to evaluate users' emotional states in real-time and adapt accordingly. This entails creating algorithms that can deduce emotions from facial expressions and incorporating this knowledge into the process of recommending music. The creation of a working prototype and testing are included in the task scope.

* + 1. *Third Level of Heading*

This work's scope includes the suggested system's design, implementation, and assessment. With the use of a webcam, the system will be able to record users' facial expressions, analyze them to deduce their emotional states, and then use this data to suggest music that corresponds with the emotions that are identified. Achieving high accuracy in emotion identification, protecting user privacy, and easily integrating the system with current music recommendation frameworks are some of the limits and constraints.

Progress has been achieved in hybrid models that incorporate several modalities. A multi-modal recommendation system that integrates audio signal processing with facial expression analysis was presented by Yang et al. (2019), offering a more thorough comprehension of user preferences. This dual technique makes recommendations that are more accurate and enjoyable by taking into account both the user's emotional state and the acoustic qualities of music songs.  
  
Among the major challenges that need to be overcome are accurate emotion detection in real-time and in a variety of lighting conditions, protecting user privacy, and easily combining facial recognition with the current recommendation algorithms. Though there is still opportunity for improvement, studies like those by Haque et al. (2018) employing generative adversarial networks (GANs) have demonstrated success in increasing facial recognition accuracy under various scenarios.

1. **figures and Tables**

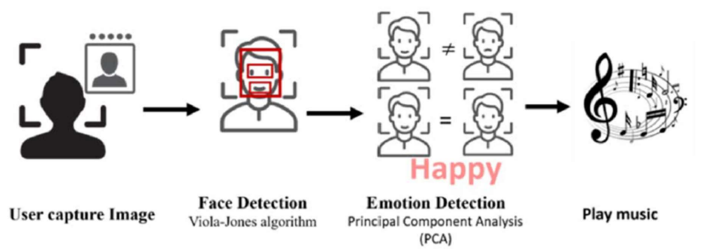
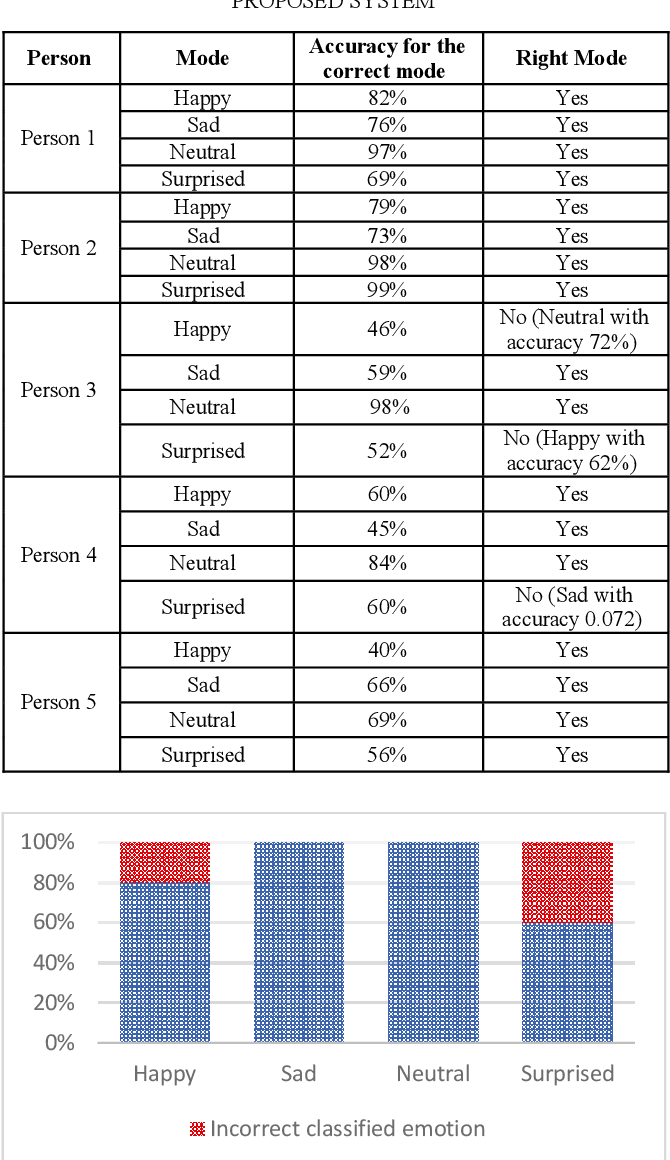


Figure 1.1 The proposed system architecture

Table 1.1: A sample table

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No.** | **Mode** | **Accuracy** | **Right Mode** |
| 1 | Happy | 82% | Yes |
| Sad | 76% | Yes |
| Neutral | 97% | Yes |
| Surprised | 69% | Yes |
| 2 | Happy | 46% | No |
| Sad | 59% | Yes |
| Neutral | 98% | Yes |
| Surprised | 52% | No |



1. **Scope**

The project's goal is to create a facial recognition-powered music recommendation system that customizes the user's listening experience. A powerful facial recognition module that can reliably identify and categorize emotions, facial expressions, and other pertinent facial traits will be the system's central component. The system will be able to associate the identified emotions with particular musical genres, performers, or playlists by integrating this data with user profiles and preferences. In order to provide individualized music recommendations based on the user's listening history and mood, the recommendation engine will integrate this data with a carefully selected music library, comprehensive metadata, and cutting-edge algorithms. The facial recognition capabilities will be seamlessly integrated into the user interface, allowing for easy interaction and control over the music recommendations.

**Social Impact:**

*Personalized User Experience*: The music recommendation system may offer a highly personalized listening experience by utilizing facial recognition and emotion analysis. This can strengthen the bond between the user and the music by raising user pleasure, engagement, and contentment with the music discovery experience.  
*Inclusive Music Discovery*: People with disabilities, such as limited mobility or vision problems, may be able to interact with music more successfully thanks to the facial recognition-based technology. The system can promote a more inclusive and approachable music landscape by lowering obstacles to music discovery.

**Environmental Impact:**

*Energy Efficiency*: Compared to standalone physical media players or local music libraries, the use of a cloud-based or centralized music recommendation system may result in a more energy-efficient use of resources.   
*Data Center Sustainability*: By utilizing renewable energy, effective cooling techniques, and other green initiatives, the data centers and underlying infrastructure that enable the music recommendation system can be planned and run with sustainability in mind.

**CHAPTER 2**

**PROBLEM DEFINITION**

**CHAPTER 2 PROBLEM DEFINITION**

Stress and exhaustion have grown commonplace in today's fast-paced environment, having a substantial negative influence on both physical and mental health. Stress hormones like cortisol are known to be elevated by high amounts of stress, which can result in a number of health problems like anxiety and melancholy as well as physical symptoms like headaches and muscular soreness. These illnesses show how critical it is to develop efficient stress-reduction techniques in order to support general wellbeing.   
  
Music therapy is a potentially effective method of stress management as it has been demonstrated to lower stress hormones and lessen depressive symptoms. Dopamine and other feel-good chemicals are released into the brain when listening to music, which can have a tremendous effect on emotional responses. Music has the ability to treat bodily ailments as well, such as aches and pains brought on by stress.

But studies show that listening to music that doesn't fit with how you're feeling might actually make tension worse instead of better. For instance, cheerful music may startle someone who is depressed, while melancholy music may make someone who is already depressed feel even more depressed. This emphasizes how crucial customized music selection is for regulating emotions and reducing stress.

Especially after a hard day, listening to music that matches your mood can greatly improve your health and relaxation. Personalized music recommendations that are in line with the listener's emotional condition can produce a calming and restorative atmosphere that aids in relaxation and lessens the negative effects of stress. In light of this, creating systems that can reliably identify and address an individual's emotional needs through customized music choices may be essential to enhancing both mental and physical health in our high-stress culture.   
  
  
In conclusion, using customized music therapy to treat the issue of stress and exhaustion has a lot of promise. Through the utilization of technologies such as facial recognition and emotion detection, it is feasible to develop music recommendation systems that offer a tailored and efficient method of relieving stress, consequently enhancing general health and wellness.

**CHAPTER 3**

**LITERATURE REVIEW**

**CHAPTER 3 LITERATURE REVIEW**

Li et al. (2018) explore how users' emotional moods and musical preferences can be deduced from their facial expressions. Through the use of deep learning techniques, they were able to show that it is possible to reliably identify emotions from facial clues and then modify music recommendations accordingly [1].

Furthermore, there has been a great deal of research done on the convergence of sentiment analysis and facial recognition. In order to assess user comments and facial expressions simultaneously, Sari et al. (2020) combined face recognition with natural language processing (NLP). Their approach efficiently gathers user sentiment across various modalities, providing a more comprehensive dataset for making tailored music suggestions [2].  
  
Wang et al. (2020) made another noteworthy contribution by investigating how to improve music suggestions by combining user demographic data with facial cues like expressions and landmarks. Their method produced more individualized music recommendations by using machine learning algorithms to evaluate facial data and extract pertinent user preferences [3].

In order to increase the accuracy of music recommendations, Chen et al. (2019) concentrated on integrating facial recognition technology with collaborative filtering techniques. Their algorithm outperformed conventional collaborative filtering techniques in recommendation performance by adding facial features as extra input variables [4].

Mollahosseini et al. (2016) conducted a groundbreaking study that showcased the application of convolutional neural networks (CNNs) in real-time facial expression recognition. Their technology, AffectNet, offers a strong framework for identifying subtle facial emotions that are associated with musical tastes. This approach demonstrates how deep learning may be used to improve the adaptability and responsiveness of music recommendation systems [5].   
  
On the basis of this framework, Yang et al. (2019) investigated a hybrid model that incorporated audio signal processing and facial recognition. Their goal was to develop a multi-modal recommendation system that takes into account the acoustic characteristics of music recordings in addition to face expression analysis. This two-pronged strategy guarantees a more comprehensive comprehension of user preferences, leading to more accurate and fulfilling music recommendations [6].

In a different direction, Pantic et al. (2017) suggested a system that enhances music suggestions by combining facial recognition with social and environmental data. Their method provides highly contextualized and timely music recommendations by evaluating users' facial expressions coupled with their location, time of day, and social contacts. This method emphasizes how crucial it is to take into account more extensive contextual aspects when making recommendations [7].   
  
Furthermore, Haque et al. (2018) conducted a novel study that looked into the application of generative adversarial networks (GANs) to improve the accuracy of facial emotion detection in a variety of lighting and environmental settings. According to their research, especially in dynamic real-world environments, higher facial recognition accuracy directly boosts the accuracy of emotion-based music recommendations [8].

**CHAPTER 4**

**PROJECT DESCRIPTION**

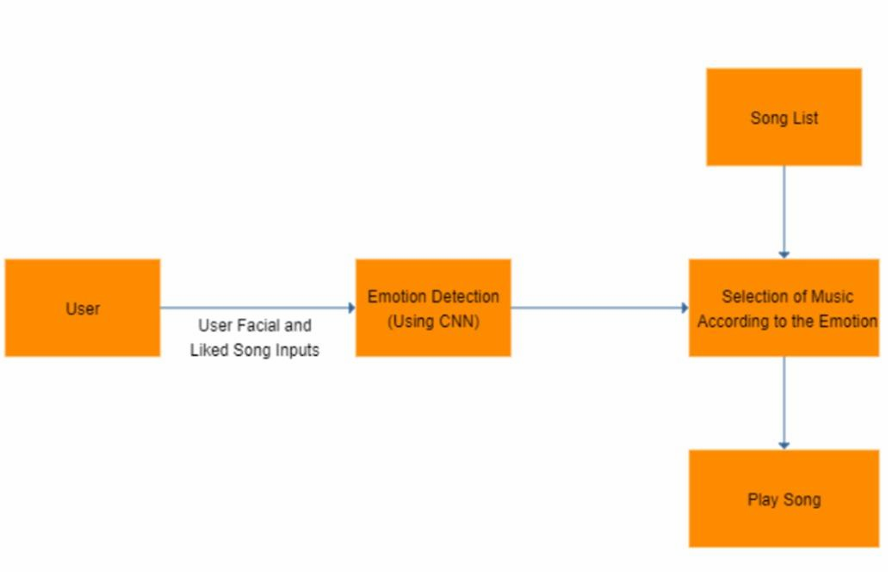
**CHAPTER 4 PROJECT DESCRIPTION**

● We first process the image of the user taken as an input with the help of a python library for Computer Vision called 'OpenCV'. This captured image is then made available for the CNN in combination with DNN to make a prediction whether the current mood of the user is 'Happy' or 'Sad'.

● The second part is the usage of Unsupervised Machine Learning techniques for clustering songs. The songs are clustered as either of the two classes-'VERY ENTERTAINING'(class 0) and 'RELAXED'(class 1) using the popular K-means algorithm. Then the recommendation is made in order of the current popularity of the respective songs.

● We have an unique story in the way we recommend the songs for each mood, for example when other sites recommend sad songs when a person is sad or feeling bad, we recommend users with songs which will cheer them up('VERY ENTERTAINING') and 'RELAXING' songs when they are 'HAPPY'.

**4.1.PROPOSED DESIGN**

****

**4.2 REQUIREMENTS**

● PC

● PYCHARM

● SPOTIFY API

● WEBCAM

● DATA FROM KAGGLE

● JUPYTER NOTEBOOK

**4.3 FUNCTIONAL REQUIREMENTS LIBRARIES USED:**

● OpenCV.

● Tensorflow and Keras.

● Sklearn.

● LightGBM.

● Spotipy.

● Tkinter.

● Pillow.

**CHAPTER 5**

**METHODOLOGY**

**CHAPTER 5 METHODOLOGY**

**Emotion Extraction Module-**The image of the user is captured with the help of a camera/webcam. Once the picture is captured, the frame of the captured image from webcam feed is converted to a grayscale image to improve the performance of the classifier, which is used to identify the face present in the picture. Once the conversion is complete, the image is sent to the classifier algorithm which, with the help of feature extraction techniques can extract the face from the frame of the web camera feed. From the extracted face, individual features are obtained and are sent to the trained network to detect the emotion expressed by the user. These images will be used to train the classifier so that when a completely new and unknown set of images is presented to the classifier, it is able to extract the position of facial landmarks from those images based on the knowledge that it had already acquired from the training set and return the coordinates of the new facial landmarks that it detected. The network is trained with the help of CK extensive data set. This is used to identify the emotion being voiced by the user

**Audio** **Extraction Module**- After the emotion of the user is extracted the music/audio based on the emotion voiced by the user is displayed to the user, a list of songs based on the emotion is displayed, and the user can listen to any song he/she would like to. Based on the regularity that the user would listen to the songs are displayed in that order. This module is developed using web technologies like HTML

**Emotion - Audio Integration Module**- The emotions which are extracted for the songs are stored, and the songs based on the emotion are displayed on the web page built using PHP and MySQL. For example, if the emotion or the facial feature is categorized under happy, then songs from the happy database are displayed to the user.

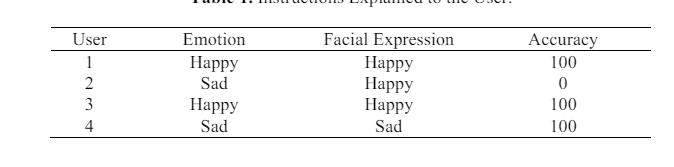
This study proposes a music recommendation system which extracts the image of the user, which is captured with the help of a camera attached to the computing platform. Once the picture has been captured, the captured frame of the image from the webcam feed is then converted to a grayscale image to improve the performance of the classifier that is used to identify the face present in the picture. Once the conversion is complete, the image is sent to the classifier algorithm which, with the help of feature extraction techniques is able to extract the face from the frame of the web camera feed. Once the face is extracted individual features from the face are extracted and is sent to the trained network to detect the emotion expressed by the user. A classifier that is used to detect or obtain the facial landmarks from the face of the user is trained on HELEN dataset. The HELEN dataset contains more than 2000 images. These images will be used to train the classifier so that when a completely new and unknown set of images is presented to the classifier, it is able to extract the position of facial landmarks from those images based on the knowledge that it had already acquired from the training set and return the coordinates of the new facial landmarks that it detected. The network is trained with the help of CK extensive data set. This is used to identify the emotion being voiced by the user. Once this has been detected, an appropriate song is selected by the music player that would best match the mood of the user. The overall idea behind making the system is to enhance the experience of the user and ultimately relieve some stress or lighten the mood of the user. The user does not have to waste any time in searching or to look up for songs and the best track matching the user’s mood is detected and played automatically by the music player. The image of the user is captured with the help of a webcam. The user’s picture is taken and then as per the mood/emotion of the user an appropriate song from the playlist of the user is played matching the user’s requirement. The system has successfully been able to capture the emotion of a user. It has been tested in a real- time environment for this predicate. It has to be, however, tested in different lighting conditions to determine the robustness of the developed system. The system has also been able to grab the new images of the user and appropriately update its classifier and training dataset. It is seen that the classifier has an accuracy of more than 80 percent for most of the test cases, which is pretty good accuracy in terms of emotion classification. It can also be seen that the classifier can accurately predict the expression of the user in a real-time scenario when tested live for a user.

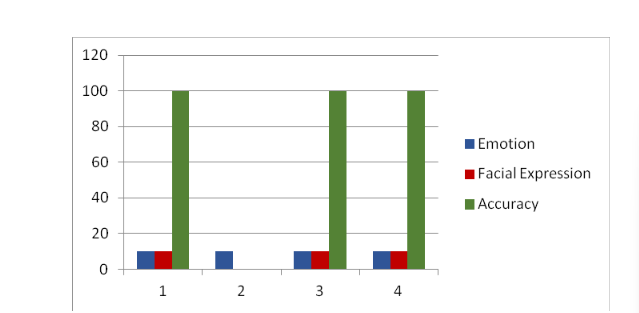
**CHAPTER 6**

**RESULTS AND ANALYSIS**

**CHAPTER 6 RESULTS AND ANALYSIS**

Experiment Results- Instructions Explained to the User. In this scenario the users were given instructions as to what is to be done to perform the prediction of the emotion expressed which provided the following results. Sometimes in cases where the inner emotion is sad and facial expression is happy it resulted in a fail case. The values are given in Table 1 and the result is shown

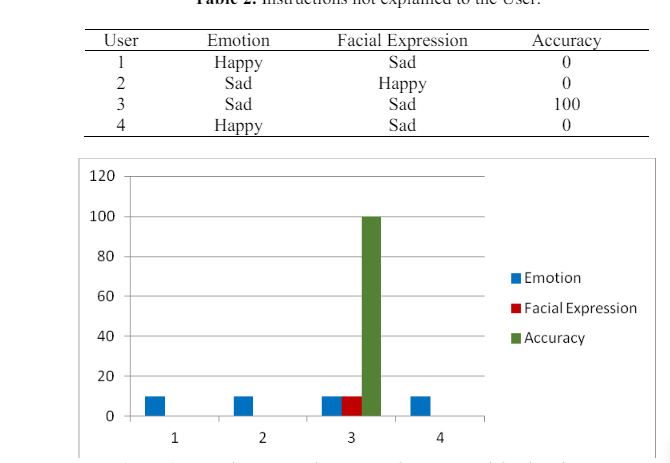
**** *Table6.1: Instructions explained to the user*

****

*Figure6.1.1:Experiment Results- Instructions Explained to the User.*

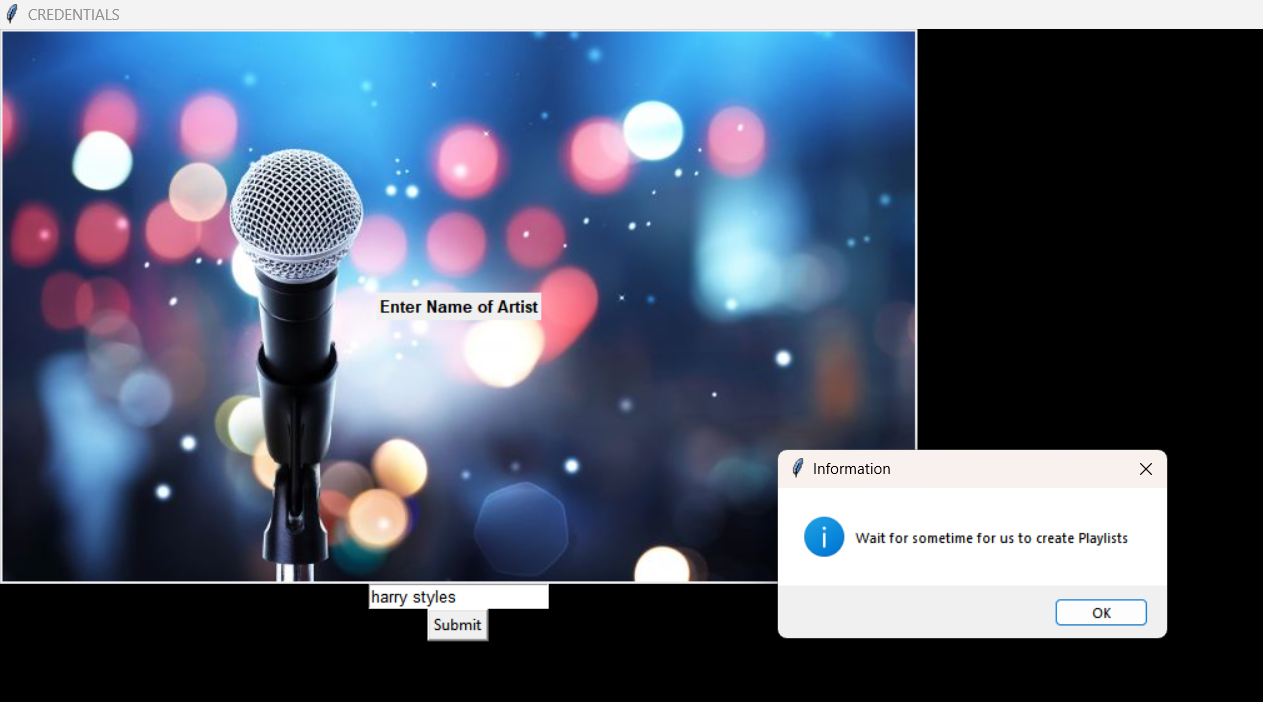
In this scenario the users were not given any instructions as to what is to be done and thus the inner emotions or the emotions recognized failed, there were also cases where in the emotion matched with the facial expressions of the user. The values are given in Table 2 and the result is shown

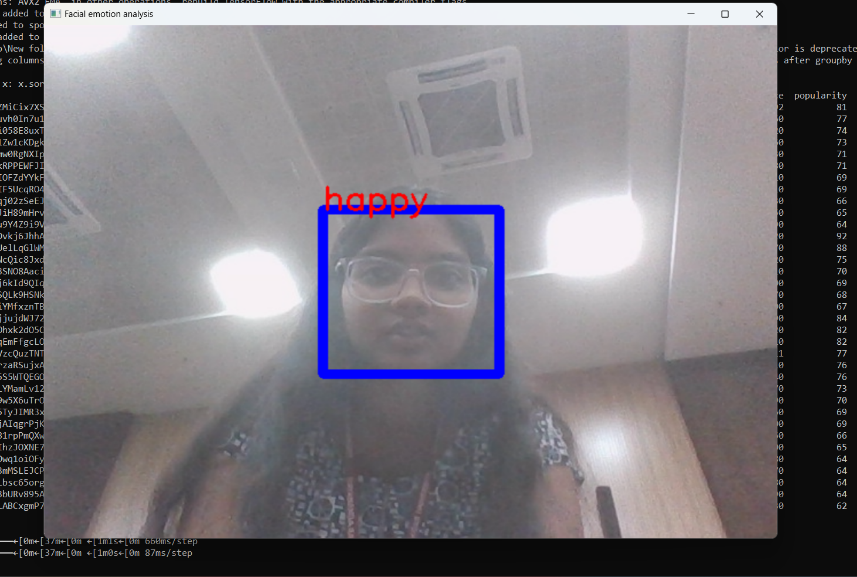
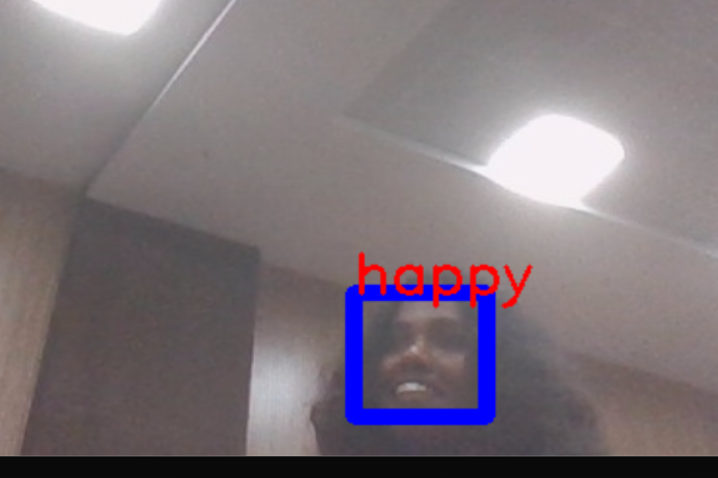
*Table 6.2. Instructions not explained to the User.*

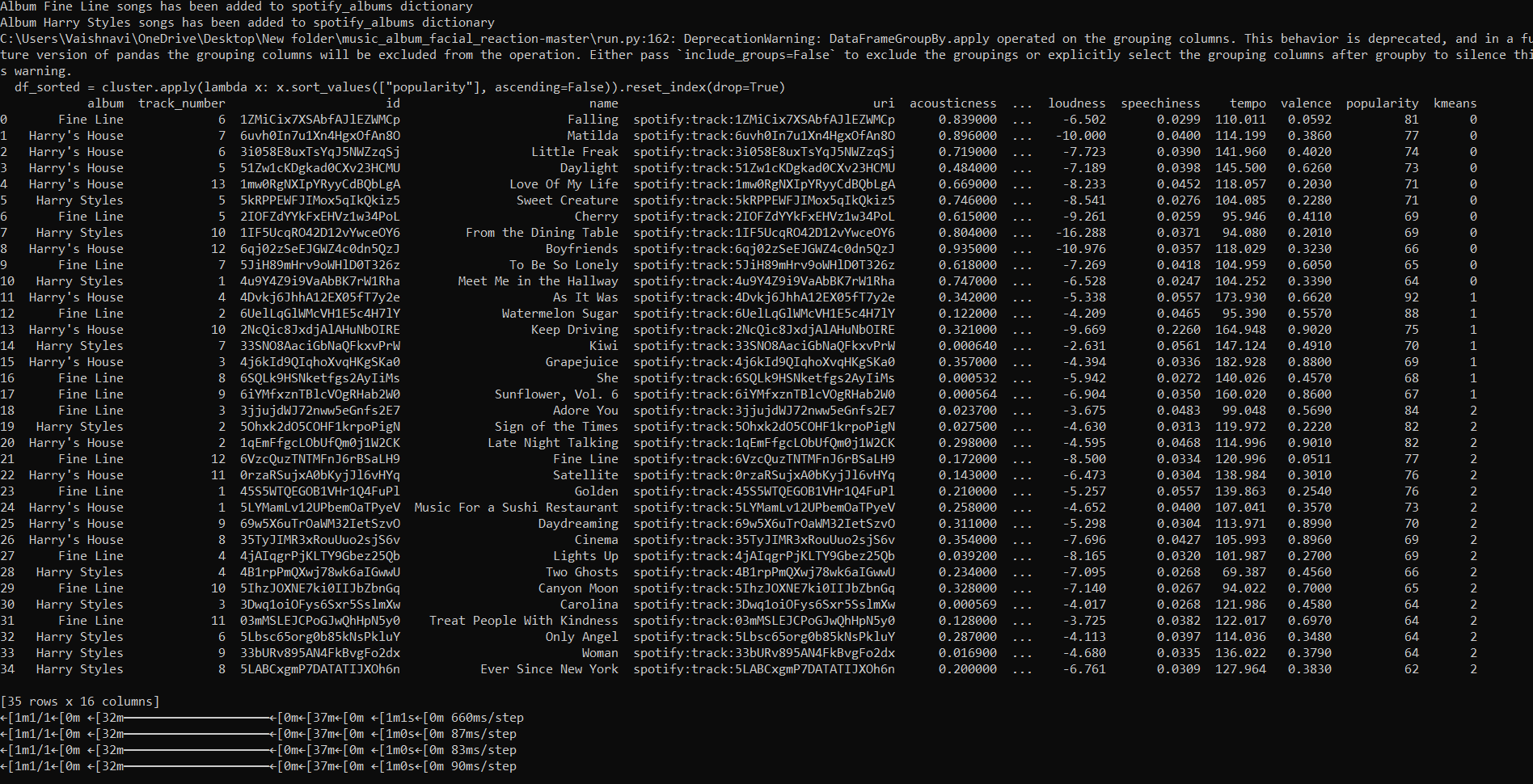
****

*Figure 6.2.1 Experiment Results- Instructions not explained to the User.*

**OUTPUT:**

****

****

****

**CONCLUSION AND FUTURE WORK**

Facial expression-based emotion identification is a significant area of research that has received a lot of attention in the past. It is evident that the difficulty of recognizing emotions using image processing algorithms has been growing daily. Scholars are consistently investigating solutions to address this through the application of various feature types and image processing techniques. Algorithms for image processing have several applications in the fields of human and medical sciences.

The use of image processing algorithms to extract user emotion and then utilize that feeling to treat the user is a constantly evolving field of research and development. The ability to recognize emotions has become increasingly important in all facets of life, and if a strong algorithm is developed that can correctly categorize an individual's feelings, then this will help the industry advance significantly. The user's emotion has been effectively captured by the system. For this predicate, it has undergone real-time testing.

To ascertain how robust the developed system is, it must be tested under various illumination scenarios. Additionally, the system was able to obtain the user's updated photos and update its classifier and training dataset accordingly. The facial landmarks scheme was used in the system's design, and it was tested in a variety of settings to determine the desired outcome. In terms of emotion categorization, it can be observed that the classifier has an accuracy of more than 80% for the majority of the test instances. When the classifier is tested on a real user, it is also evident that it is capable of effectively predicting the user's expression in a real-time scenario.

**Future Work**

● Reduce the time required to train the classifier

● Use of EEG signals to make the software even more optimized and to detect the exact mood /emotion of the user.

**REFERENCES**

[1] Londhe RR and Pawar DV 2012 Analysis of facial expression and recognition based on statistical approach International Journal of Soft Computing and Engineering 2

[2] Kabani H, Khan S, Khan O and Tadvi S 2015 Emotion based music player International Journal of Engineering Research and General Science 3 750-6

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

[4] Hadid A, Pietikäinen M and Li SZ 2007 Learning personal specific facial dynamics for face recognition from videos International Workshop on Analysis and Modeling of Faces and Gestures pp1-15 Springer Berlin Heidelberg

[5] 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.

**CODE/PROGRAM**

from tkinter import ttk

from PIL import Image, ImageTk

from PIL.ImageTk import PhotoImage

from sklearn.metrics import silhouette\_score

import config

import spotipy

import pandas as pd

from spotipy.oauth2 import SpotifyClientCredentials

from emotion\_video\_classifier import emotion\_testing

import tkinter as tk

from tkinter import messagebox

client\_credentials\_manager = SpotifyClientCredentials(client\_id=config.cid, client\_secret=config.secret)

sp = spotipy.Spotify(client\_credentials\_manager=client\_credentials\_manager)

root = tk.Tk()

root.title('CREDENTIALS')

root.geometry("600x400")

root.configure(bg='black')

name1 = tk.StringVar()

photo = PhotoImage(file="musicback.jpg")

l = tk.Label(root, image=photo)

l.image = photo  # just keeping a reference

l.grid()

def submit():

    global name

    name = name\_entry.get()

    messagebox.showinfo("Information", "Wait for sometime for us to create Playlists")

    root.destroy()

name\_label = tk.Label(root, text='Enter Name of Artist',

                      font=('calibre',

                            10, 'bold'))

name\_entry = tk.Entry(root,

                      textvariable=name1, font=('calibre', 10, 'normal'))

sub\_btn = tk.Button(root, text='Submit',

                    command=submit)

name\_label.grid(row=0, column=0)

name\_entry.grid(row=3, column=0)

sub\_btn.grid(row=5, column=0)

root.mainloop()

result = sp.search(name)  # search query

artist\_uri = result['tracks']['items'][0]['artists'][0]['uri']

# Pull all of the artist's albums

sp\_albums = sp.artist\_albums(artist\_uri, album\_type='album')

# Store artist's albums' names' and uris in separate lists

album\_names = []

album\_uris = []

for i in range(len(sp\_albums['items'])):

    album\_names.append(sp\_albums['items'][i]['name'])

    album\_uris.append(sp\_albums['items'][i]['uri'])

def albumSongs(uri):

    album = uri  # assign album uri to a\_name

    spotify\_albums[album] = {}  # Creates dictionary for that specific album

    # Create keys-values of empty lists inside nested dictionary for album

    spotify\_albums[album]['album'] = []  # create empty list

    spotify\_albums[album]['track\_number'] = []

    spotify\_albums[album]['id'] = []

    spotify\_albums[album]['name'] = []

    spotify\_albums[album]['uri'] = []

    tracks = sp.album\_tracks(album)  # pull data on album tracks

    for n in range(len(tracks['items'])):  # for each song track

        spotify\_albums[album]['album'].append(album\_names[album\_count])  # append album name tracked via album\_count

        spotify\_albums[album]['track\_number'].append(tracks['items'][n]['track\_number'])

        spotify\_albums[album]['id'].append(tracks['items'][n]['id'])

        spotify\_albums[album]['name'].append(tracks['items'][n]['name'])

        spotify\_albums[album]['uri'].append(tracks['items'][n]['uri'])

spotify\_albums = {}

album\_count = 0

for i in album\_uris:  # each album

    albumSongs(i)

    print("Album " + str(album\_names[album\_count]) + " songs has been added to spotify\_albums dictionary")

    album\_count += 1  # Updates album count once all tracks have been added

def audio\_features(album):

    # Add new key-values to store audio features

    spotify\_albums[album]['acousticness'] = []

    spotify\_albums[album]['danceability'] = []

    spotify\_albums[album]['energy'] = []

    spotify\_albums[album]['instrumentalness'] = []

    spotify\_albums[album]['liveness'] = []

    spotify\_albums[album]['loudness'] = []

    spotify\_albums[album]['speechiness'] = []

    spotify\_albums[album]['tempo'] = []

    spotify\_albums[album]['valence'] = []

    spotify\_albums[album]['popularity'] = []

    # create a track counter

    track\_count = 0

    for track in spotify\_albums[album]['uri']:

        # pull audio features per track

        features = sp.audio\_features(track)

        # Append to relevant key-value

        spotify\_albums[album]['acousticness'].append(features[0]['acousticness'])

        spotify\_albums[album]['danceability'].append(features[0]['danceability'])

        spotify\_albums[album]['energy'].append(features[0]['energy'])

        spotify\_albums[album]['instrumentalness'].append(features[0]['instrumentalness'])

        spotify\_albums[album]['liveness'].append(features[0]['liveness'])

        spotify\_albums[album]['loudness'].append(features[0]['loudness'])

        spotify\_albums[album]['speechiness'].append(features[0]['speechiness'])

        spotify\_albums[album]['tempo'].append(features[0]['tempo'])

        spotify\_albums[album]['valence'].append(features[0]['valence'])

        # popularity is stored elsewhere

        pop = sp.track(track)

        spotify\_albums[album]['popularity'].append(pop['popularity'])

        track\_count += 1

import time

import numpy as np

sleep\_min = 2

sleep\_max = 5

start\_time = time.time()

request\_count = 0

for i in spotify\_albums:

    audio\_features(i)

    request\_count += 1

    if request\_count % 5 == 0:

        print(str(request\_count) + " playlists completed")

        time.sleep(np.random.uniform(sleep\_min, sleep\_max))

        print('Loop : {}'.format(request\_count))

        print('Elapsed Time: {} seconds'.format(time.time() - start\_time))

dic\_df = {}

dic\_df['album'] = []

dic\_df['track\_number'] = []

dic\_df['id'] = []

dic\_df['name'] = []

dic\_df['uri'] = []

dic\_df['acousticness'] = []

dic\_df['danceability'] = []

dic\_df['energy'] = []

dic\_df['instrumentalness'] = []

dic\_df['liveness'] = []

dic\_df['loudness'] = []

dic\_df['speechiness'] = []

dic\_df['tempo'] = []

dic\_df['valence'] = []

dic\_df['popularity'] = []

for album in spotify\_albums:

    for feature in spotify\_albums[album]:

        dic\_df[feature].extend(spotify\_albums[album][feature])

length = len(dic\_df['album'])

data = pd.DataFrame.from\_dict(dic\_df)

data.drop\_duplicates(inplace=True, subset=['name'])

name = data['name']

df = pd.read\_csv('Spotify Dataset Analysis/data.csv.zip', compression='zip')

df.drop\_duplicates(inplace=True, subset=['name'])

name = df['name']

data1 = data.append(df)

name = data1['name']

from sklearn.cluster import KMeans

from sklearn.preprocessing import MinMaxScaler

col\_features = ['danceability', 'energy', 'valence', 'loudness']

X = MinMaxScaler().fit\_transform(data1[col\_features])

kmeans = KMeans(init="k-means++",

                n\_clusters=2,

                random\_state=15).fit(X)

data1['kmeans'] = kmeans.labels\_

# print(silhouette\_score(X, data1['kmeans'], metric='euclidean'))

data2 = data1[:data.shape[0]]

cluster = data2.groupby(by=data2['kmeans'])

data2.pop('kmeans')

df1 = cluster.apply(lambda x: x.sort\_values(["popularity"], ascending=False))

df1.reset\_index(level=0, inplace=True)

def get\_results(emotion\_code):

    NUM\_RECOMMEND = 10

    happy\_set = []

    sad\_set = []

    if emotion\_code == 0:

        happy\_set.append(df1[df1['kmeans'] == 0]['name'].head(NUM\_RECOMMEND))

        return pd.DataFrame(happy\_set).T

    else:

        sad\_set.append(df1[df1['kmeans'] == 1]['name'].head(NUM\_RECOMMEND))

        return pd.DataFrame(sad\_set).T

def final():

    root1 = tk.Tk()

    root1.title("Your Playlist")

    root1.configure(bg='black')

    df = get\_results(emotion\_code)

    cols = list(df.columns)

    tree = ttk.Treeview(root1)

    tree.pack(side=tk.TOP, fill=tk.X)

    tree["columns"] = cols

    for k in cols:

        tree.column(k, anchor="w")

        tree.heading(k, text=k, anchor='w')

    for index, row in df.iterrows():

        tree.insert("", 0, text=index, values=list(row))

    root1.mainloop()

    if emotion\_word == 'sad':

        print('emotion detected is SAD')

    else:

        print('emotion detected is HAPPY')

emotion\_word = (emotion\_testing())

if emotion\_word == 'sad':

    emotion\_code = 0

else:

    emotion\_code = 1

window = tk.Tk()

window.title("Music Recommender System")

window.configure(background='black')

window.grid\_rowconfigure(0, weight=1)

window.grid\_columnconfigure(0, weight=1)

message = tk.Label(

    window, text="Music Recommender System",

    bg="yellow", fg="black", width=50,

    height=3, font=('times', 30, 'bold'))

message.place(x=200, y=20)

pred = tk.Button(window, text="print",

                 command=final, fg="white", bg="black",

                 width=20, height=3, activebackground="Red",

                 font=('times', 15, ' bold '))

pred.place(x=200, y=500)

quitWindow = tk.Button(window, text="Quit",

                       command=window.destroy, fg="white", bg="black",

                       width=20, height=3, activebackground="Red",

                       font=('times', 15, ' bold '))

quitWindow.place(x=1100, y=500)

image1 = Image.open("musicimg (1).jpg")

test = ImageTk.PhotoImage(image1)

label1 = tk.Label(image=test)

label1.image = test

label1.place(x=470, y=150)

root.mainloop()

window.mainloop()