# MUSIC RECOMMENDER USING FACIAL EXPRESSION

**A Project Work**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**CSR489::BE-CSE**

**AIML**

**Submitted by:**

**ADITI GUPTA (20BCS6471)**

**Under the Supervision of:**

**GURPREET KAUR**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI-140413,**

**PUNJAB**

**April-2024**

# DECLARATION

I, **'ADITI GUPTA**, student of **'Bachelor of Engineering in CSR-489::BE-CSE AIML'**, **session:2020-2024**, Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, hereby declare that the work presented in this Project Work entitled '**Music Recommender using facial expression'** is the outcome of our own bona fide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**Aditi Gupta 20BCS6471**

# ABSTRACT

Human emotion plays a vital role in recent times. Emotion is based on human feelings which can be both expressed or not. Emotion expresses the human's individual behavior which can be in different forms. Extraction of the emotion states humans' individual state of behavior. The objective of this project is to extract features from the human face and detect emotion, and to play music according to the emotion detected. However, many existing techniques use previous data to suggest music and the other algorithms used are normally slow, usually they are less accurate and it even requires additional hardware like EEG or physiological sensors. Facial expressions are captured by a local capturing device or an inbuilt camera. Here we use an algorithm for the recognition of the feature from the captured image. Thus, the proposed system is based on the facial expression captured and music will be played automatically.

Keywords—Recognition, Python, OpenCV Application.

# ACKNOWLEDGEMENT

The search work presented in this thesis has been performed at the Department of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Punjab, (India). While conducting this research, I received support from many people in one way or another, without whose support, this thesis would not have been completed in its present forms, 1 extend my unlimited thanks to all of them.

My first experience of the project has been successful thanks to the support of staff, my friends & colleagues with gratitude. I wish to acknowledge all of them. However, I wish to make special mention of the following.

The satisfaction that accompanies the successful completion of any task would be incomplete without the people who made it possible and whose constant encouragement and guidance has been a source of inspiration throughout the course of the project.

I take this opportunity to express my sincere thanks to my guide and respected **GURPREET KAUR Mam** for their support and encouragement throughout the completion of this project

Finally, I would like to thank all the teaching and non-teaching faculty members and lab staff of the Department of Computer Science and Engineering for their encouragement. I also extend our thanks to all those who have helped us directly or indirectly in the completion of this project

# *List of Figures*

# Table of Contents

# 1. INTRODUCTION

In this era of artificial intelligence with recent developments in the technology of digital streaming and music services, it is essential to modify and develop a personalized music recommendation system which recommends suitable music to its users based on certain criteria. Though it is a complex problem and a challenge to provide the most accurate recommendations for the users from the big data available on the internet, the algorithm made by a tech giant like Google makes it easy to scrape through the web to provide the most suitable recommendation.

Moreover, E-commerce giants like Amazon, provide personalized recommendations to users based on their most searched preferences, old-user activities, and history, while platforms like iTunes, Spotify, and Wynk use artificial intelligence technologies like machine learning and deep learning to filter out suitable recommendations for their users.

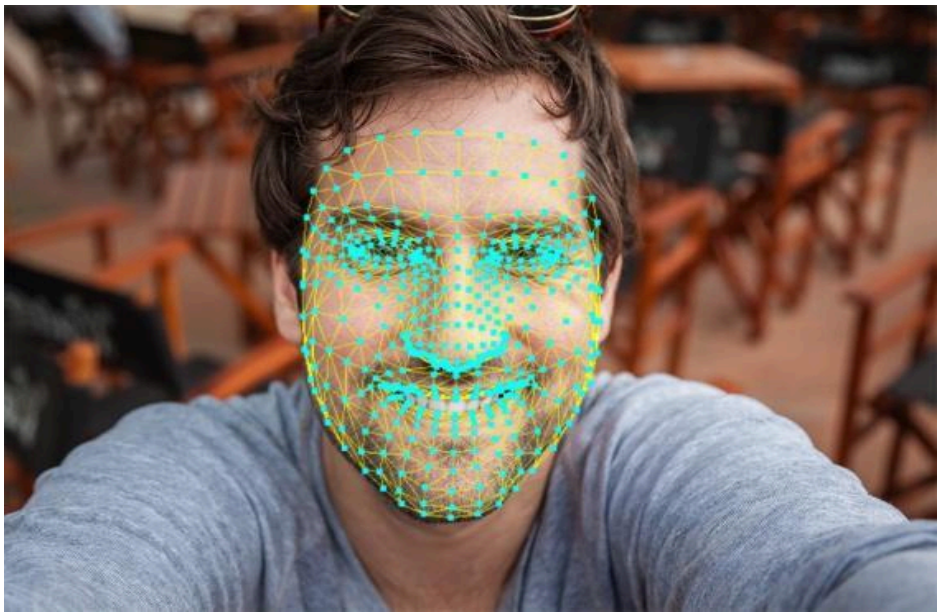


Figure 1.1 : Facial Detection

The emotional behavior of humans is generally classified as follows: happy, angry, surprised, sad, and neutral. Facial lacerations or convexity are complex but rare, making spotting such small differences a challenge because even a small difference leads to very different manifestations. Furthermore, the expression of different people or even the same person can differ for the same emotion, since emotions are non-linguistic and context-specific. If the focus of the face scanner can only be relied on the areas of the face such as the mouth and eye contours, which helps us to show maximum emotional context, how to extract, recognize and classify these gestures is still an important question.

Deep Learning, Neural networks and machine learning have been used for these tasks. The results we have fetched so far in the developed technologies have been promising. Machine learning algorithms have proven to be very useful in fields of pattern recognition and classification in any given set of data or an image be it real-time or stored, and thus can also be used to detect emotions of the interested users.

So the main goal of this project is to implement a program or a system capable of detecting the user's emotions, be it happy, sad, or rocking. Once the emotion is identified, the recommendation system redirects the user to YouTube and provides a music playlist containing video songs of certain types of music that we believe will uplift the user's mood. If the user's expression is found to be cheerful, the most relevant playlist will be activated, namely a soothing playlist among different music videos

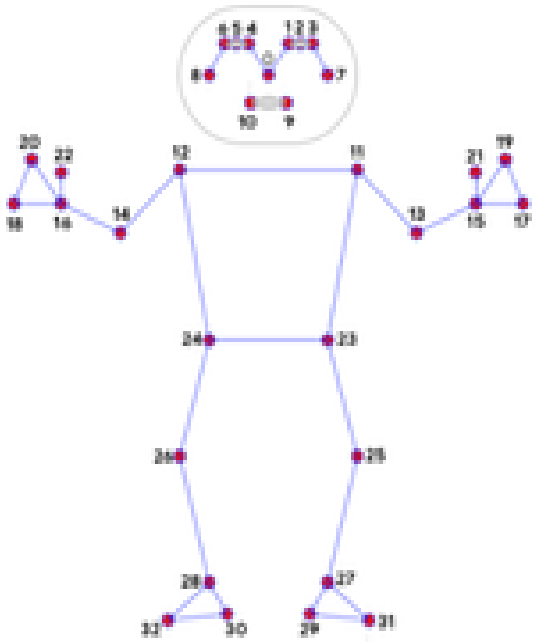
## 1.1 PRELIMINARIES

### A. MediaPipe Holistic

Object detection is one of the largest and most common use cases in computer vision. Several models for detecting objects are used worldwide for specific applications. Many of these models have been used as standalone solutions to a single computer vision problem, with their own fixed applications. Combining these multiple tasks into one comprehensive real-time solution is what MediaPipe does.

MediaPipe is a platform for building machine learning pipelines for processing time-series data such as video, audio, etc. This cross-platform runs on desktop/server, Android, ios and embedded devices such as Raspberry Pi and Jetson Nano. It can be used to import advanced machine learning models such as face detection, multi-hand tracking, object detection and tracking, and more.

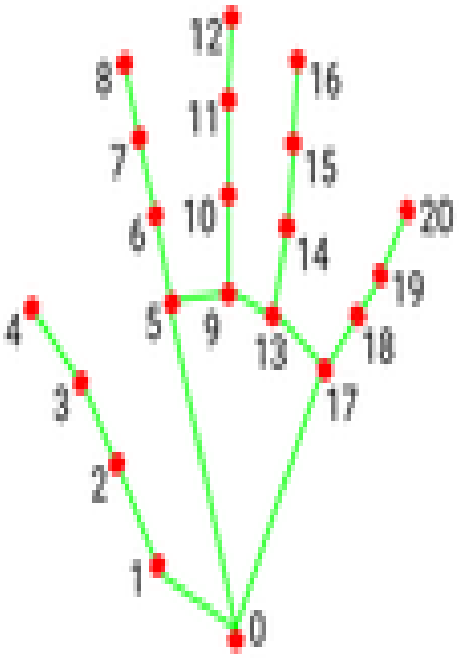
## 1. Pose landmarks



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Figure 1.1.1 : Pose landmarks

## 2. Hand Landmarks




0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 1.1.2: Hands landmarks

### 3. Face Landmarks

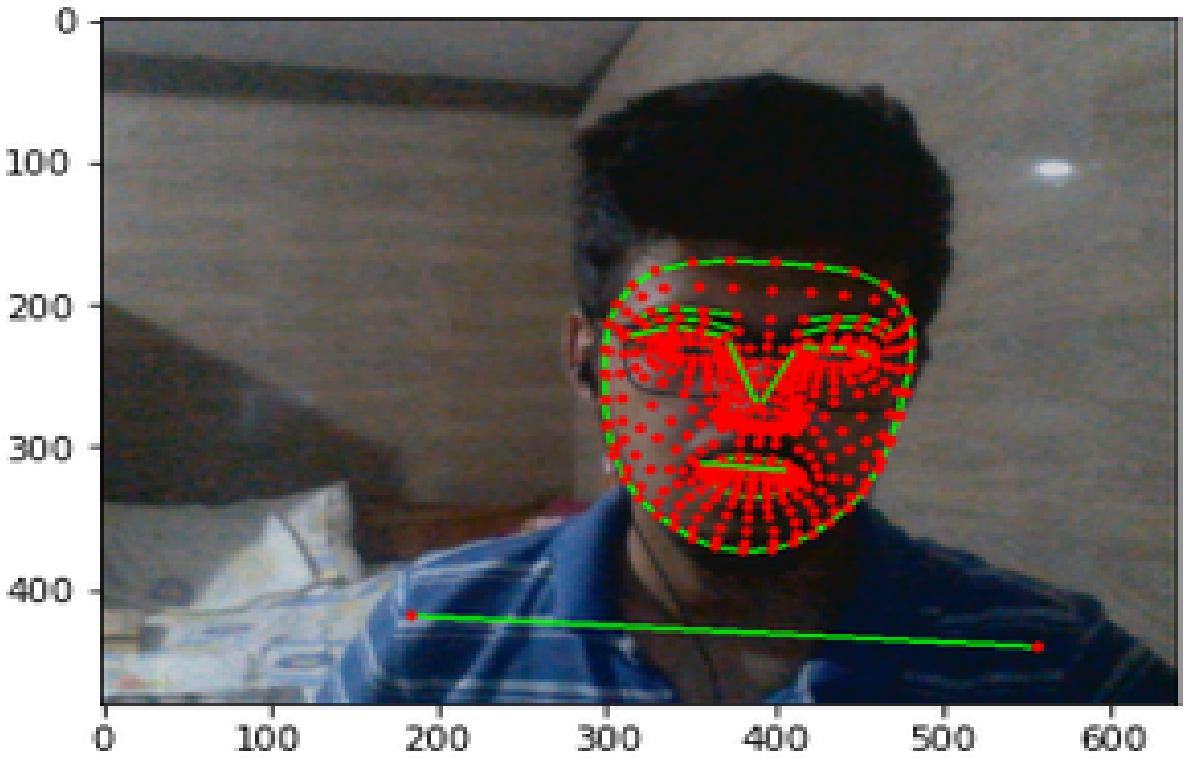



Figure 1.1.3: Face landmarks

## B. OpenCV

In the field of artificial intelligence, computer vision is one of the most interesting and challenging tasks. Computer vision acts as a link between computer software and visualization. Computer vision allows computer software to know and study visualizations of the environment. Tell us a pattern: Based on the shape, color and size that govern the fruit.

This task is very simple for the human brain, but in the Computer Vision system we want to collect data first, then we do data processing operations, then we train and explain the pattern to realize how outcomes are distinguished. fruit size, shape and color.

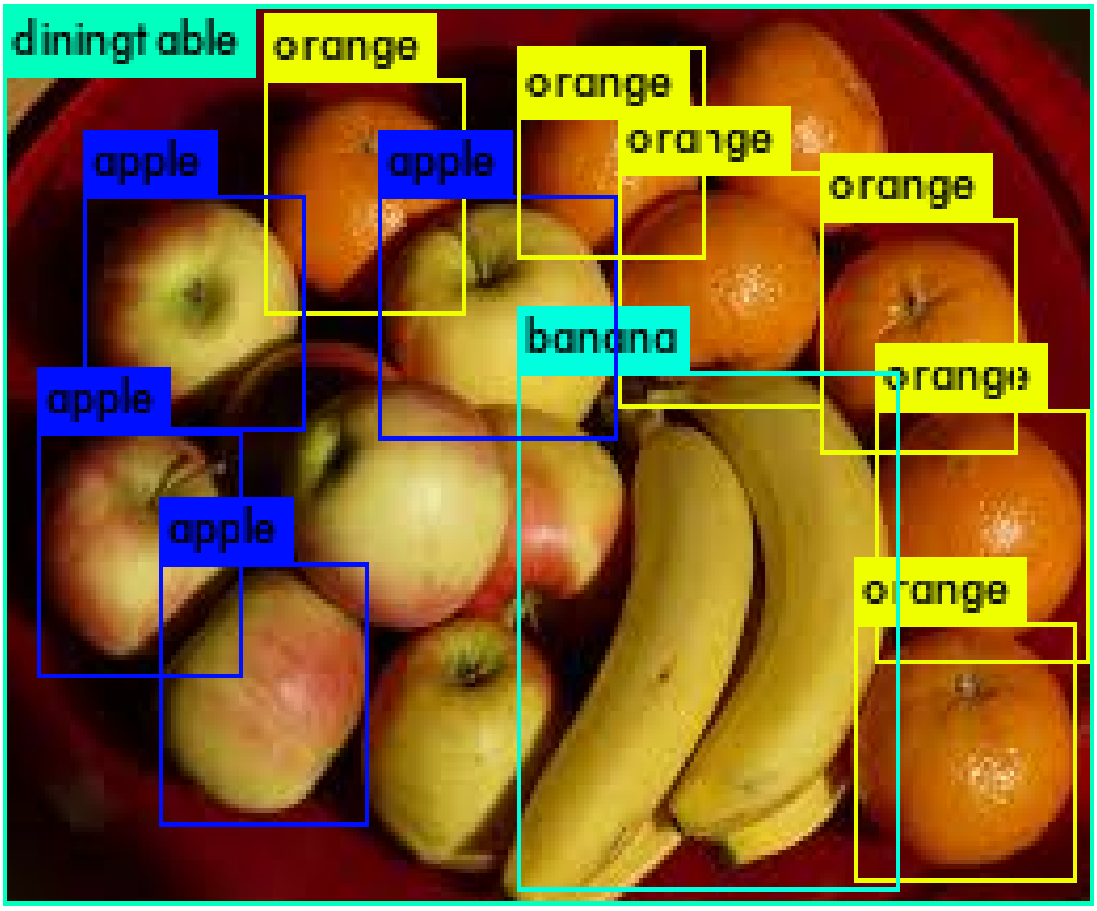



Figure 1.1.4: Detecting objects

## 1.2 Problem Definition

People tend to express their feelings, substantially, by their facial expressions. Music has always been known to alter the mood of an existent. Landing and feting the emotion being raised by a person and displaying applicable songs matching the bone's mood and can decreasingly calm the mind of a stoner and overall end up giving a pleasing effect. The design aims to capture the emotion expressed by a person through facial expressions. A music player is designed to capture mortal emotion through the web camera interface available on calculating systems. The software captures the image of the user and also with

the help of image segmentation and image processing ways excerpts features from the face of a target human being and tries to describe the emotion that the person is trying to express.

## 1.3 Project Overview

In this conception music is recommended to the stoner by detecting the real time capturing of stoner's feelings. Being fashionable we use collaboration fashion which will use former stoner data to recommend music and This fashion requires a lot of homemade work so, we proposed a system to arrange different music in different orders similar as happy, sad or angry etc. Emotion- Grounded-music-player It's a music player with chrome as frontal- End which has the capability to describe feelings i.e, the face of a stoner with the help of machine literacy algorithm using python. Grounded on the detected stoner's mood, the song list will be displayed/ recommended to the stoner. In this operation image of a person is captured using a real time machine that has the access to the original machinery and depending on the captured image it compares the database data sets that formerly saved in the original device through recycling it defines the present mood of the stoner in numerical form grounded on this music will be played other than that we've some common features that are a line playlist so that we can have an individual playlist and the last one is arbitrary. It uses the python Eel library so that it can pick an arbitrary song without any order.

## 1.4 Hardware Requirements

Computer hardware is the set of physical components that a computer system needs to function. It covers everything with a circuit board that works in a PC or laptop. The hardware requirements are:

- 4 Gigabyte (GB) RAM (Minimum)
- A Webcam (for Desktop)
- Minimum 13 Megapixel Resolution camera (for android device)
- 20 MB Memory space (approx. value)

## 1.5 Software Requirements

The functions performed by the computer are the result of the software, the computer is capable of performing many different functions. Some basic initial Softwares must be installed on a computer for an application to function optimally. These libraries are not included in the software installation package and must be installed separately prior to installing the music recommendation software.

The software requirements that are required for this project are:

- Python 3.8.8
- Keras
- Tensorflow
- Streamlit modules
- OpenCV 3.1
- VS Code IDE or WINDOWS Powershell

# 2. LITERATURE REVIEW

## 2.1 Existing System :

The process of multidimensional reduction by taking the primary data that's lowered to numerous other classes for sorting out or organizing. Emotion of a stoner is uprooted by landing the image of the stoner through a webcam. The captured image is enhanced by the process of dimensional reduction by tracking the primary data. This data is converted into double image format and the face is detected using Fisher Face and Haarcascade styles.

The initial or the primary data taken from the mortal face that's lowered to numerous other classes. These classes are sorted and organized using the below styles. Emotion is detected by rooting the point from the mortal face. The main end in point rooting module is to dwindle the number of coffers needed from the large sets of data.

Features in an image consist of 3 corridors.

1. Boundaries/ edges
2. Corners/projection points
3. Field points

### 2.1.1 Fisher Face Algorithm

This image processing system is used for reducing the face space confines using the top element analysis (PCA) system and also it applies fishermen direct discriminant (FDL) or the LDA system to gain the point of the image characteristics, we especially use this because it maximizes the separation between classes in the training process. This algorithm helps to reuse for image recognition is done in fisher face while for matching faces algorithm we use minimal euclidean it helps us to classify the expression that implies the emotion of the user.

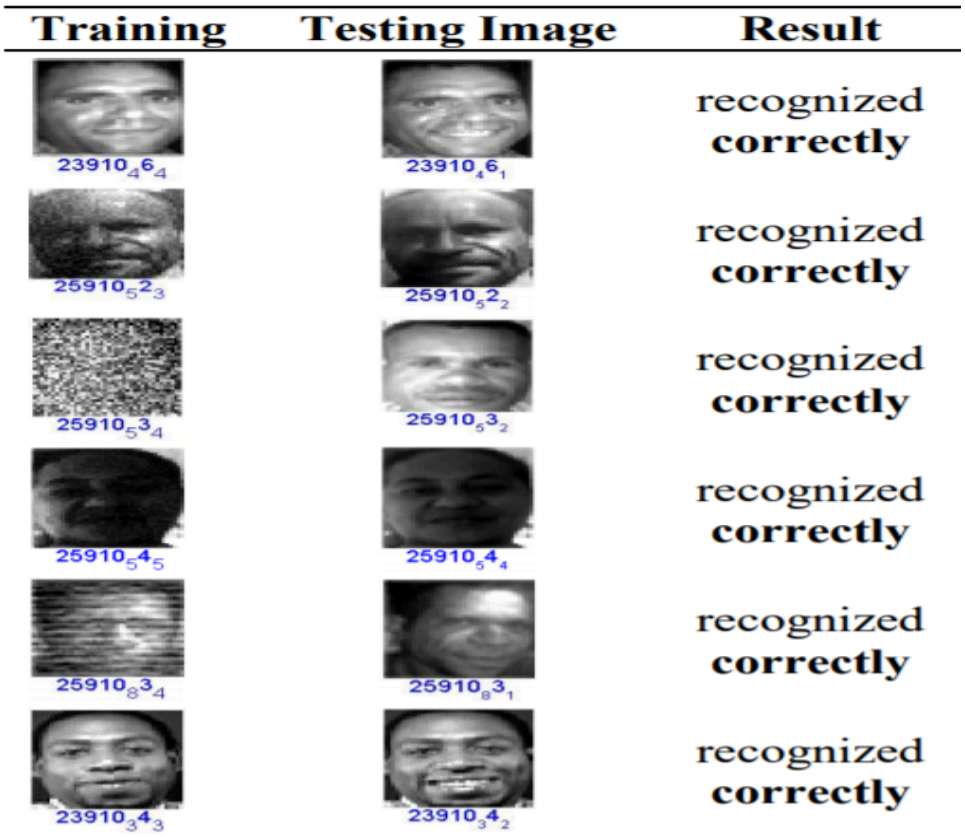



Fig. 2.1.1 : Fisher algorithm

## 2.2 Proposed System

Humans have a tendency to show their feelings intentionally; they reflect the face. The proposed system helps us to create a commerce between the stoner and the music system. This design substantially focuses on the stoner's favored music that's recommended due to the emotional mindfulness. In the original stage of the proposed system The projected system is an A.I.-based music recommendation system that is based on the Streamlite web application which takes input like the singer's name and language from the user after clicking the '**recommend me a song**' button it will automatically sense emotion. An external or in-built webcam is

used to capture the real-time image of the user that will be used as input for the programmed system to verify human emotion,


Figure 2.2 : Different types of emotion

### 2.2.1 Techniques to extract Facial Emotions

Facial emotions are one of the most important factors which enable us to recognize someone's intentions. Generally, people deduce someone's emotional states such as anger, joy, or sadness through two main factors: expression of the face and his vocal tone. Study shows that two-thirds of people's communication is carried in non-verbal communication while facial expression represents the highest factor in this study. While only one third of emotion is carried out in verbal communication. For that reason, nowadays one of the main fields in the

era of artificial intelligence is facial emotion recognition not only due to its academic importance but also its world-wide commercial potential. The studies in this arena shows that the methods used by researchers can be divided into two key classes: conventional method and neural networks-based method.

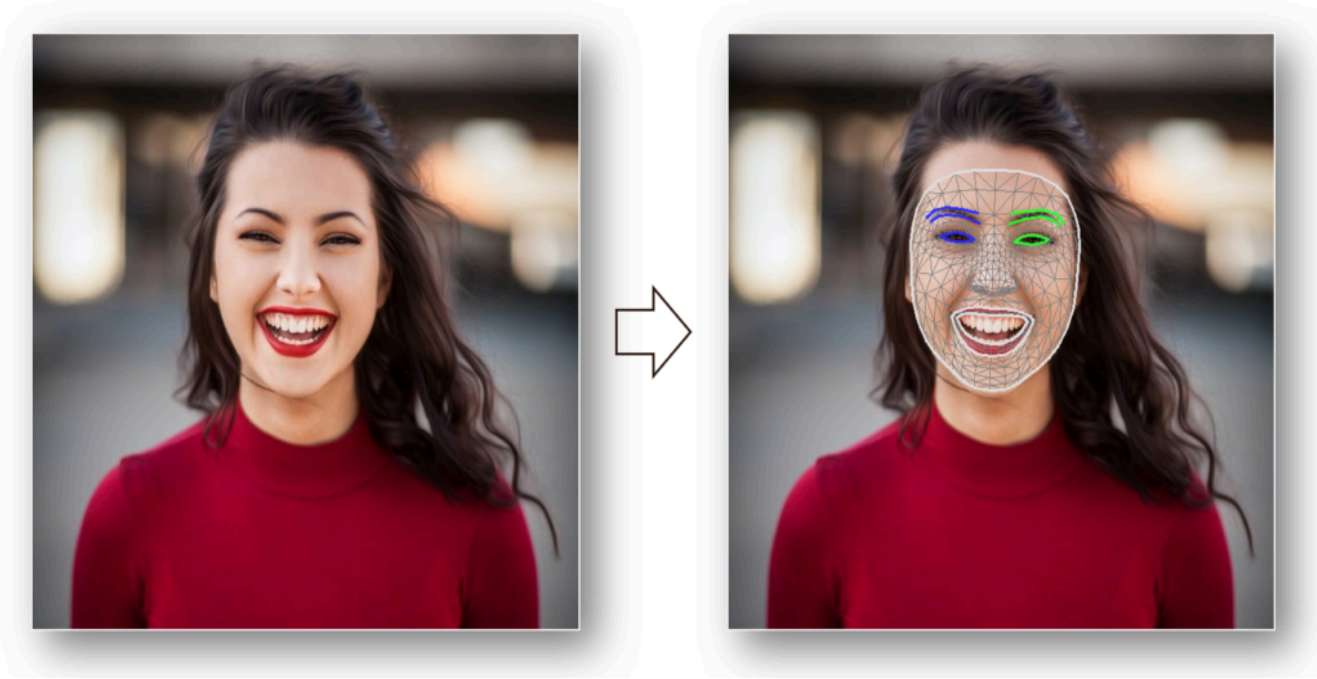


Figure 2.2.1 : Extract Facial Emotion

In the traditional method of inferring human emotions from still images, researchers first work with face and landmark recognition (e.g. eyes, nose, mouth) as a first step, then extract spatial features and end the process with classification. Based on extracted features, draw conclusions and display the results.

### 2.2.2. Music’s effect on a person

Different studies have looked at music and how it will affect people in different ways, some researchers have studied how music affects people's emotions in the short or long term. , while others pay attention to its effect on customer behavior and measure their approval of the service provided. Very few researchers have proven its usefulness in treating patients. Since the main concern of the proposed system is to detect the user's emotions and improve their mood through music, we are going to deal with given types of emotions and we need to choose the right type of music so as to merge with them.

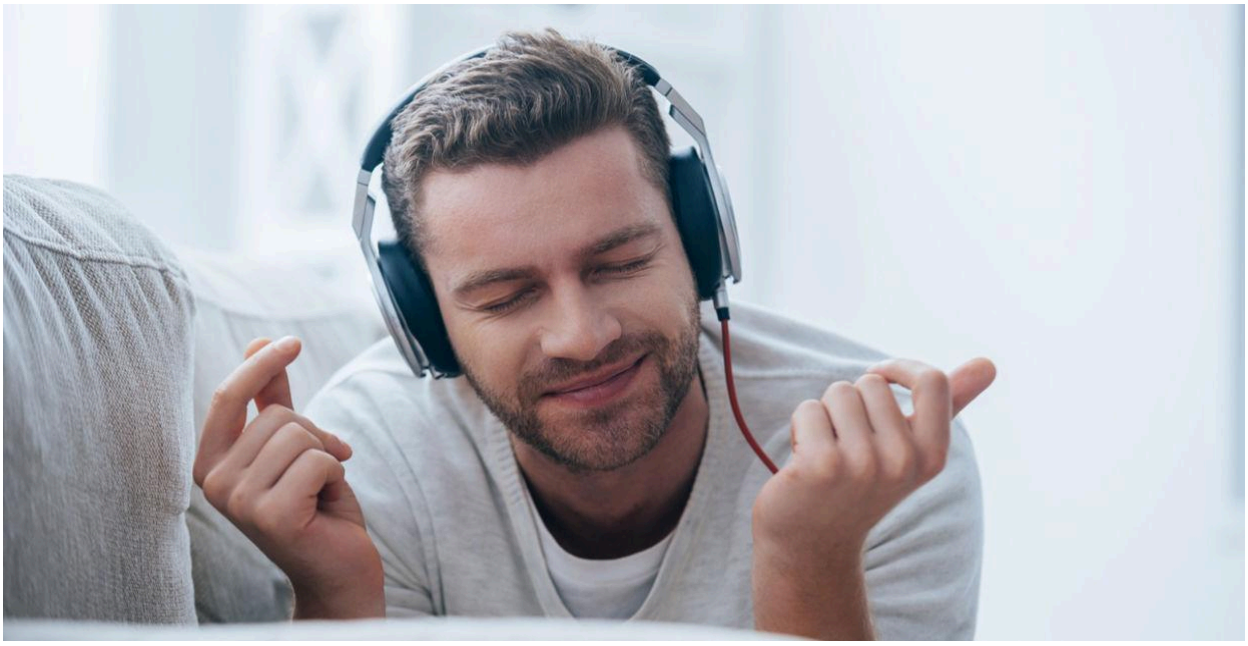


Figure 2.2.2 : Music Effect on Person

Figure 2.2.3 : Listening Music can really reduce pain

A recent analysis has found that surgery patients who listen to music may experience lower levels of pain, anxiety, blood pressure, and heart rate than people who do not listen to music. The study, “Effects of Art on Surgical Patients: A Systematic Review and Meta-analysis,” was published in the Annals of Surgery scientific journal. Researchers with the University of Zurich in Switzerland analyzed data from 47 different studies over the last 15 years, including 26 that looked at the effect of music before surgery, 25 looking at music during surgery, and 25 looking at music during recovery. The researchers found that 31 percent of people reported experiencing less pain while listening to music, 29 percent had lower odds of using

pain killers, and 34 percent reported less anxiety. Listening to music was also tied to 40 percent lower blood pressure and 27 percent lower heart rate. These benefits for pain and anxiety greatly increased when patients were allowed to select their own playlists.

“As many patients have smartphones with personal music playlists, informing our patients before scheduled surgeries of the positive effect of music on their wellbeing could be a low cost intervention that may enhance wellbeing and possibly faster recover,”

## 2.3 SYSTEM ARCHITECTURE

In this paper, In this project, by running the main Streamlight Web page it will take some input like enter the language and singer name from the user. After clicking on a song button it automatically starts detecting the emotion using a webcam. OpenCV helps in capturing images from the webcam by importing the webrtc_streamer library as well as for processing purposes. Emotion is detected with the help of mediapipe we want to predict let take the example of our hand our have some landmarks on it and the mediapipe will easily detect these landmarks at very fast frame like 24/25 rate per second.Now our model able to take these points and predict the particular emotion associated to that particular face we collect different landmarks points of our face. one particular sample and one particular row would be

comprising all the key points all the face key points all the left hand landmarks and all the right hand landmarks one particular sample have all of that and based on that we will train our model  after detection it will trigger the user to YouTube and show to user many playlist according to its emotion.

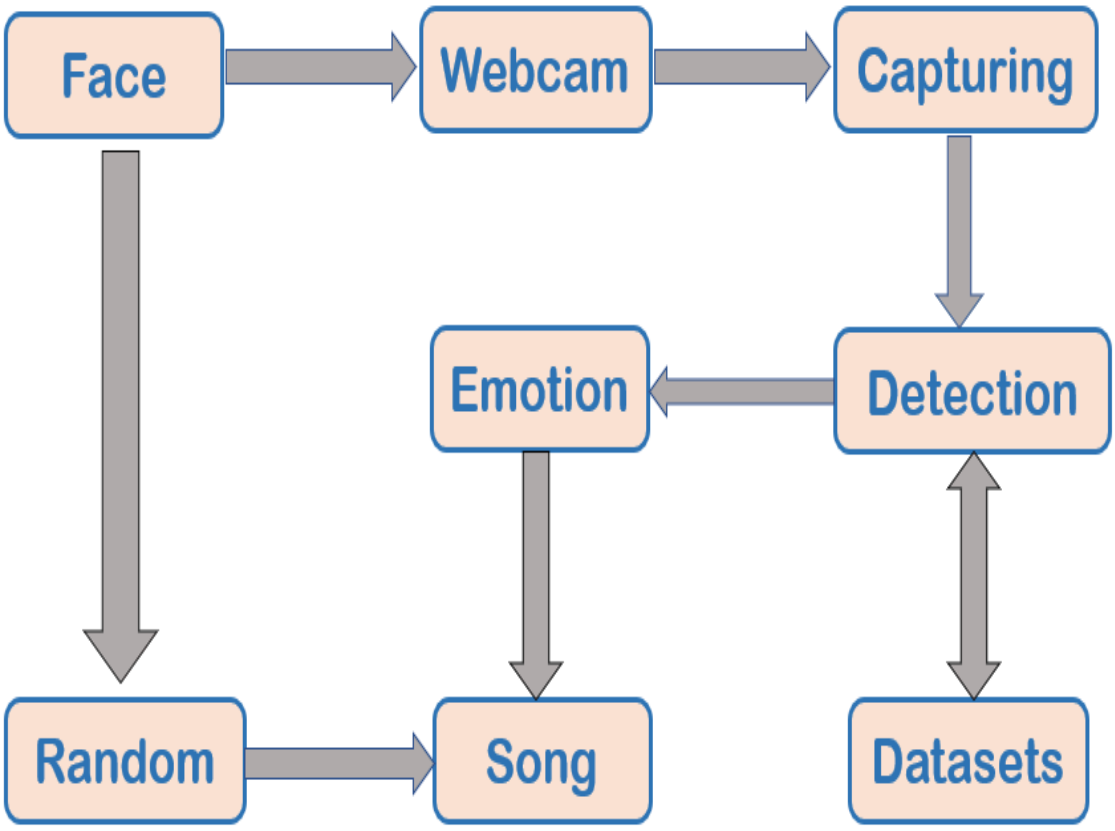



Figure 2.3 : System Architecture

### 2.3.1. Classification

There is a division of classes of the inputs; the system produces a model from training data wherein it assigns new inputs to one of these classes. It falls under the umbrella of supervised learning. A real-life example can be spam filtering, where emails are the input that is classified as “spam” or “not spammed”.

### 2.3.2. Regression

Regression algorithm also is a part of supervised learning, but the difference is that the outputs are continuous variables and not discrete. We have used unsupervised machine learning and KNN algorithms can be used for both classification and regression problems. The KNN algorithm uses ‘feature similarity’ to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.

## 2.4. Python

What exactly is Python? You may be wondering about that. You may be referring to this book because you wish to learn editing but are not familiar with editing languages. Alternatively, you may be familiar with programming languages such as C, C ++, C #, or Java and wish to learn more about Python language and how it compares to these “big word” languages.

### 2.4.1. Python Concepts

You can skip to the next chapter if you are not interested in how and why Python. In this chapter, I will try to explain why I think Python is one of the best programming languages available and why it is such a great place to start. Python was developed into an easy-to-use programming language. It uses English words instead of punctuation, and has fewer syntax than other languages. Python is a highly developed, translated, interactive, and object-oriented language. Python translated - Interpreter processing Python during launch. Before using our software, you do not need to install it. This is similar to PERL and PHP editing languages. Python interactive - To write your own applications, you can sit in Python Prompt and communicate directly with the interpreter. Python Object-Oriented - Python supports the Object-Oriented program style or method, encoding the code within objects. Python is a language for beginners - Python is an excellent language for beginners, as it allows for the creation of a variety of programs, from simple text applications to web browsers and games.

## 2.4.2. Python Features

Python features include - Easy-to-learn - Python includes a small number of keywords, precise structure, and well-defined syntax. This allows the student to learn the language faster. Easy to read – Python code is clearly defined and visible to the naked eye. Easy-to-maintain - Python source code is easy to maintain. Standard General Library - Python's bulk library is very portable and shortcut compatible with UNIX, Windows, and Macintosh. Interaction mode - Python supports interaction mode that allows interaction testing and correction of captions errors. Portable - Python works on a variety of computer systems and has the same user interface for all. Extensible - Low-level modules can be added to Python interpreters. These Modules allow system developers to improve the efficiency of their tools either by installing or customizing them. Details - All major commercial information is provided by Python ways of meeting GUI Programming - Python assists with the creation and installation of a user interface for images of various program phones, libraries, and applications, including Windows MFC, Macintosh, and Unix's X Window.

Scalable – Major projects benefit from Python building and support, while Shell writing is not.

Aside from the characteristics stated above, Python offers a long list of useful features, some of which are described below. –

1.It supports OOP as well as functional and structured programming methodologies.

2.It can be used as a scripting language or compiled into byte-code for large-scale application development.

3.It allows dynamic type verification and provides very high-level dynamic data t ypes.

4.Automatic garbage pickup is supported by IT.

## 2.4.3. Datatypes Python has five standard data types –

1.Numbers

2.String

3.List

4.Tuple

5.Dictionary

## 2.4.4. Python Strings

In this python a string is defined as a collection set of characters enclosed in quotation marks. Python allows you to use any number of quotes in pairs. The slice operator ([ ] and [:] ) can be used to extract subsets of strings, with indexes starting at 0 at the start of the string and working their way to -1 at the end.

## 2.4.5. Python Lists

The most diverse types of Python data are lists. Items are separated by commas and placed in square brackets in the list ([]). Lists are similar to C-order in some ways. Listings can be for many types of data, which is one difference between them. The slide operator ([] and [:]) can be used to retrieve values stored in the list, the indicators start with 0 at the beginning of the list and work their way to the end of the list. The concatenation operator of the list is a plus sign (+), while the repeater is an asterisk (*).

## 2.4.7. Python Tuples

A cone is a type of data type similar to a sequence of items. Cone is a set of values separated by commas. The pods, unlike the list, are surrounded by parentheses. Lists are placed in parentheses ([]), and the elements and sizes can be changed, but the lumps are wrapped in brackets (()) and cannot be sorted. Powders are the same as reading lists only.

## 2.4.8. Python Dictionary

Python dictionaries in Python are a way of a hash table. They are similar toPerl's combination schemes or hashes, and are made up of two key numbers. The dictionary key can be any type of Python, but numbers and strings are very common. Prices, on the other hand, can be anything you choose Python. Curly braces () surround dictionaries, and square braces ([) are used to assign and access values.

### 2.4.9. Different Modes In Python

Python Normal and interactive are the two basic Python modes. The scripted and completed.py files are executed in the Python interpreter in
the regular manner. Interactive mode is a command line shell that provides instant response for each statement while simultaneously running previously provided statements in inactive memory. The feed programme is assessed in part and whole as fresh lines are fed in the interpreter. Python is a full featured for general purpose programming language. It is a mature and fast expanding platform for scientific research and numerical computing. Python hosts numerous open source libraries and almost all general-purpose libraries for machine learning which can be further used for deep learning models. All this benefits from the python ecosystem lead to the top two libraries for numerical analysis of deep learning developed for python language, which is Pandas and numpy.

### 2.4.10. Pandas

Pandas is an open source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics. It allows us to store and manipulate tabular data as a 2-D data structure. Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code that is purely written in C or Python. We can analyze data in pandas with: Series.

Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science. Pandas gives you answers about the data.

Like:

· Is there a correlation between two or more columns?

· What is the average value?

· Max value?

· Min value?

Pandas are also able to delete rows that are not relevant, or contain wrong values, like empty or NULL values. This is called cleaning the data.

### 2.4.11. Numpy

NumPy is the fundamental package for scientific computing in Python.

NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called nd array , it provides a lot of supporting functions that make working with nd array very easy. Arrays are very frequently used in data science, where speed and resources are very important.

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called nd array, it provides a lot of supporting functions that make working with nd array very easy. Arrays are very frequently used in data science, where speed and resources are very important. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently.

This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with the latest CPU architectures. NumPy is a Python library and is written  partially in Python, but most of the parts that require fast computation are written in Cor C++. The Pandas module mainly works with the tabular data, whereas the NumPy module works with the numerical data. ... NumPy library provides objects for multi-dimensional arrays, whereas Pandas is capable of offering an in-memory 2d table object called DataFrame. NumPy consumes less memory as compared to Pandas.

# 3. PROBLEM FORMULATION

During software development, clones can occur in software intentionally or unintentionally. Developers tend to clone fragments of software during development to save efforts and expedite the development process

From the literature review, it is observed that studies highlight the need for an efficient and scalable approach for detecting code clones having software vulnerability. The existing techniques are not able to detect all types of vulnerable code clones. Different approaches suffer from high false negative rates and are not scalable to large software systems due to high time complexity.

**Existing techniques** use previous data to suggest music which is stored in a database and the other algorithms used are normally slow, usually they are less accurate and it even require additional hardware like EEG or physiological sensors.

Existing techniques were using collaboration techniques which will use previous user data to recommend music and This technique requires a lot of manual work.

**In this project,** use of physiological sensors has been annihilated enabling us to capture facial emotions at zero cost.

A fast and accurate as much as 75% of a model using OpenCV and Streamlit libraries has been used in the software so as to instantly capture human emotion and to recommend music based on their preference from youtube providing long range of choice instead of previously stored and limited music files in the user storage system.

# SOURCE CODE:

```
import requests

import streamlit as st

from PIL import Image

from streamlit_lottie import st_lottie

from streamlit_webrtc import webrtc_streamer

import av

import cv2

import numpy as np

import mediapipe as mp

from keras.models import load_model

import webbrowser


withopen('style.css') as f:

    st.markdown(f'<style>{f.read()}</style>', unsafe_allow_html=True)


model  = load_model("model.h5")

label = np.load("labels.npy")

holistic = mp.solutions.holistic

hands = mp.solutions.hands

holis = holistic.Holistic()

drawing = mp.solutions.drawing_utils



defload_lottieurl(url):

    r= requests.get(url)

    if r.status_code !=200:
```

```
       return none
    return r.json()


lottie_coding =
load_lottieurl("https://assets6.lottiefiles.com/packages/lf20_ikk4jhps.json")
lottie_maxican =
load_lottieurl("https://assets2.lottiefiles.com/packages/lf20_ydhm6y.json")
lottie_camera =
load_lottieurl("https://assets7.lottiefiles.com/packages/lf20_m5evszda.json")


col3,col1, col2,col4 = st.columns([0.5,1,3,1])


with col1:
     st_lottie(lottie_coding,height=110,width=350,key="coding")


with col2:
     st.markdown("<h1 style=' color: blue; text-align:left; '>Music Recommendation
System</h1>", unsafe_allow_html=True)


     img = Image.open("image/music.jpg")
     st.image(img)


new_title ='<p style="font-family:sans-serif; color:pink; text-align:
center;font-family:Courier; font-size: 42px;">  Hii, I am Moody</p>'
st.markdown(new_title, unsafe_allow_html=True)


if"run"notin st.session_state:
    st.session_state["run"] ="true"
```

```
try:
    emotion = np.load("emotion.npy")[0]
except:
    emotion=""

ifnot(emotion):
    st.session_state["run"] ="true"
else:
    st.session_state["run"] ="false"

classEmotionProcessor:
    defrecv(self, frame):
        frm = frame.to_ndarray(format="bgr24")

        ##############################
        frm = cv2.flip(frm, 1)

        res = holis.process(cv2.cvtColor(frm, cv2.COLOR_BGR2RGB))

        lst = []

        if res.face_landmarks:
            for i in res.face_landmarks.landmark:
                lst.append(i.x - res.face_landmarks.landmark[1].x)
                lst.append(i.y - res.face_landmarks.landmark[1].y)

            if res.left_hand_landmarks:
                for i in res.left_hand_landmarks.landmark:
                    lst.append(i.x - res.left_hand_landmarks.landmark[8].x)
                    lst.append(i.y - res.left_hand_landmarks.landmark[8].y)
```

```
            else:
                for i inrange(42):
                    lst.append(0.0)

            if res.right_hand_landmarks:
                for i in res.right_hand_landmarks.landmark:
                    lst.append(i.x - res.right_hand_landmarks.landmark[8].x)
                    lst.append(i.y - res.right_hand_landmarks.landmark[8].y)
            else:
                for i inrange(42):
                    lst.append(0.0)

            lst = np.array(lst).reshape(1,-1)

            pred = label[np.argmax(model.predict(lst))]

            print(pred)
            cv2.putText(frm, pred, (50,50),cv2.FONT_ITALIC, 1, (255,0,0),2)

            np.save("emotion.npy", np.array([pred]))



        drawing.draw_landmarks(frm, res.face_landmarks, holistic.FACEMESH_TESSELATION,
landmark_drawing_spec=drawing.DrawingSpec(color=(0,0,255), thickness=-1,
circle_radius=1),
connection_drawing_spec=drawing.DrawingSpec(thickness=1))
        drawing.draw_landmarks(frm, res.left_hand_landmarks, hands.HAND_CONNECTIONS)
        drawing.draw_landmarks(frm, res.right_hand_landmarks, hands.HAND_CONNECTIONS)
```

```
        ##############################

        return av.VideoFrame.from_ndarray(frm, format="bgr24")



col3,col1, col2 = st.columns([2,2,2])

with col1:

    lang = st.text_input("Enter the language",key=1)

    singer = st.text_input("Enter the name of singer",key=2)

with col2:
    st_lottie(lottie_maxican,height=200,width=200,key="ani")


if lang and singer and st.session_state["run"] =="true":
    webrtc_streamer(key="key", desired_playing_state=True,
                video_processor_factory=EmotionProcessor)



col1,col2, col3 , col4, col5 = st.columns([2,1.2,2,1,1])

with col1:
    pass
with col2:
    st_lottie(lottie_camera,height=75,width=430,key="cam")
```

```
with col4:
    pass
with col5:
    pass
with col3 :
    btn = st.button("Recommend me songs")









if btn:
    ifnot(emotion):
        st.warning("Let us capture your emotion ")
        st.session_state["run"] ="true"
    else:

webbrowser.open(f"https://www.youtube.com/results?search_query={emotion}+{lang}+songs+
by+{singer}")
        np.save("emotion.npy", np.array([""]))
        st.session_state["run"] ="false"
```

# Code for UserInterface:

```
.css-1hm2aoh {
    font-size: 20px;
    color: rgb(231150150);
    margin-bottom: 7px;
    height: auto;
    min-height: 1.5rem;
    vertical-align: middle;
    display: flex;
    flex-direction: row;
    -webkit-box-align: center;
    align-items: center;
    font-family: 'Baloo Bhai', cursive;
}


/* .css-16huue1 {
    font-size: 20px;
    color: rgb(236, 149, 149);
    margin-bottom: 7px;
    height: auto;
    min-height: 1.5rem;
    vertical-align: middle;
    display: flex;
    flex-direction: row;
    -webkit-box-align: center;
    align-items: center;
    font-family: 'Baloo Bhai', cursive;
```

```
} */


.css-12nj2tl {


    font-family: 'Baloo Bhai', cursive;

    margin-bottom: -1rem;

}

.css-1v0mbdjimg{

    height: 205px;

    width: 610px;

}



.css-10trblm {

    position: relative;

    flex: 110%;

    margin-left: calc(3rem);

    font-family: 'Rubik Moonrocks', cursive;

}


/* For button */


.css-1sz81cb {

    padding: 4px10px;

    border: 2pxsolidwhite;

    background-color: brown;

    border-radius: 8px;

    color: white;

    cursor: pointer;

    font-family: 'Baloo Bhai', cursive;
```

```
    font-size: 20px;
}
.css-1sz81cb.edgvbvh1:hover{
    background-color: red;
    color: white;
    border-color: white;
}

/* for image  */
.css-1v0mbdjimg {
    height: 207px;
    width: 760px;
    margin: 0px-50px;
    padding: 0px40px;
}

/* for music recommendation heading */

.css-10trblm {
    position: relative;
    flex: 110%;
    margin-left: calc(0rem);
    font-family: 'Rubik Moonrocks', cursive;
    font-size: 55px;
}

*, :after, :before {
    box-sizing: inherit;
    padding: 0px0px;
}
```

# 4. ALGORITHM

## 1. Collaborative filtering algorithms

### 1.1. KNN

The aim of this algorithm is to learn a function that can predict if a user will benefit from an item — meaning the user will likely listen to a song. This can be done by using ratings. There are two ways to collect user ratings: Explicit Rating and implicit Rating. We used the K-Nearest Neighbors Algorithm.

### Explicit Rating

This means we explicitly ask the user to give a rating. This represents the most direct feedback from users to show how much they like a song. The dictionary meaning of explicit is to state clearly and in detail. Explicit feedback data as the name suggests is an exact number given by a user to a product. Some of the examples of explicit feedback are ratings of movies by users on Netflix, ratings of products by users on Amazon

.

## Implicit Rating

We examine whether or not a user listened to a song, for how long or how many times, which may suggest that he/she liked that particular song. Implicit ratings include measures of interest such as whether the user listens. A song and, if so, how much time the user spent reading it.

The main motivation for using implicit ratings is that it removes the cost to the evaluator of examining and rating the item.

## Interaction matrices

These are based on the many entries that include a user-song pair as well as a value that represents the user's rating for that song. We are going to use listen_count, the number of times a user listened to a song as an implicit rating. Doing some exploratory analysis, we can discover that a user listens to a mean number of 26 songs and a median of songs of 16. We can quickly see that not all users listen to all songs. So a lot of value in the song x users matrix is going to be zero. Thus, we'll be dealing with extremely sparse data. We now did work with a scipy-sparse matrix to avoid overflow and wasted memory. For that purpose, we'll use the csr_matrix function from scipt. sparse. First, we reshaped the data based on unique values from song_id as index and user_id as columns to form axes of the resulting DataFrame.. Then, we'll use the function pivot to produce a pivot table. Then, we'll convert this table to a sparse matrix. As we can observe many of the values are equal to zero. This indicates that the user has

not listened to that song. There are 2 main approaches for collaborative filtering: It finds users with similar interests and behavior, and considering what those similar users listened to, it makes a recommendation. This technique is known as user-based approach (or user-item). Or it can take into account what songs the user has considered in the past and recommend new similar songs that the user can enjoy; a technique that is called item-based approach (or item-item).

## KNN:

K - Nearest Neighbors (KNN) is considered the standard method when it comes to both user-based and item-based collaborative filtering approaches. The KNN algorithm is a supervised non-parametric Lazy Learning method used for both classification and regression. It considers a graph based on the rating and plots the rating of the input song in the graph and calculates the distance with all the other songs using cosine similarity and recommends the song based on the distance . Least distance is compared. KNN is a machine learning algorithm to find clusters of similar users based on common book ratings, and make predictions using the average rating of top-k nearest neighbors. We use unsupervised algorithms with sklearn.neighbors. The algorithm we use to compute the nearest neighbors is “brute”, and we specify “metric=cosine”so that the algorithm will calculate the cosine similarity between rating vectors. Finally, we fit the model.

**1.metric**: the distance metric to use. We are going to use cosine.

**2.algorithm:** Algorithm used to compute the nearest neighbors. We are going to see brute force algorithm

**3.n_neighbors:** Number of neighbors to use for queries. We are going to use20. Remember that we mentioned before that this parameter is very important. The KNN algorithm measures distance to determine the “closeness”of instances. It then classifies an instance by finding its nearest neighbors, and picks the most popular class among the neighbors. This method will make a prediction based on the entire data set. When we want to predict a new value, the algorithm will look for the K instances of these closest to it. Then, it will use the output values of the closest K neighbors to compute the value of the variable that needs to be predicted. Parameter K is to be determined, a sufficiently high value is needed to avoid underfitting, however if the value is too high there is a risk of overfitting and so poor generalization on unseen data. A compromise has to be found. For each new item, the first step is to calculate its distance to all the other values of the dataset and retain the K items for which the distance is minimal. Then the optimal K is used to make the prediction: in the case of a regression, the next step is to calculate the mean (or median) of the output values of the selected K-neighbors.

K Nearest Neighbor considers K Nearest Neighbors (Data points) to

predict the class or continuous value for a new Data point as the name suggests

**1)** Instance-based learning uses full training instances to predict output for unknown data, rather than learning weights from training data to predict output (as in model-based algorithms).

**2) Lazy Learning:** The model is not learned using training data before the prediction is required on the new instance, and the learning process is postponed until the prediction is asked.

3) Non-Parametric: In KNN, the mapping function has no specified form.

# 5 . RESEARCH OBJECTIVES

The project aims to capture the emotion expressed by an individual through facial expressions. A music player is meant to capture human emotion through the online camera interface available on computing systems. The software captures the image of the user then with the assistance of image segmentation and image processing techniques extracts features from the face of a target person and tries to detect the emotion that the person is trying to precise . 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 the past the simplest sort of expression analysis known to humankind is countenance recognition. The simplest possible way in which people tend to research or conclude the emotion or the sensation or the thoughts that another person is trying to precise is by countenance . In some cases, mood alteration can also help in overcoming situations like depression and sadness. With the help of expression

analysis, many health risks are often avoided, and also there are often steps taken that help bring the mood of a user to a far better stage.

In our proposed system, a mood-based music player is created which performs real time mood detection and suggests songs as per detected mood. This becomes an additional feature to the traditional music player apps that come pre-installed in our mobile phones. An important benefit of incorporating mood detection is customer satisfaction. The objective

of this system is to analyze the user's image, predict the expression of

the user and suggest songs suitable to the detected mood.

The objective was to develop a system that can analyze the image and predict the expression of the person. The study proved that this procedure is workable and produces valid results.

# 5. METHODOLOGY

The mood-based music recommendation system is an application that focuses on implementing real time mood detection. It is a prototype of a new product that comprises two main modules: Facial expression recognition/mood detection and Music recommendation.

The following methodology will be followed to achieve the objectives defined for proposed research work:

1. Detailed study of software vulnerabilities, their types and impact of code cloning practice on software vulnerabilities will be done.
2. Installation of clone detection tools and hand on experience on existing approaches of clone detection applicable for software code clone vulnerability will be done. Relative pros and cons will be identified.
3. A Vulnerability database will be created, which corresponds to the C/C++ open source projects that have some vulnerabilities according to National Vulnerability Database (NVD) [36] and Open Source Vulnerability Database (OSVDB)[51].
4. Different clone detections techniques will be analyzed to figure out which clone detection technique is appropriate for a particular type of vulnerability.
5. An approach will be developed for vulnerable code clone detection.

6. Various parameters will be identified to evaluate the proposed system.
7. Comparison of new implemented approaches with existing approaches will be done.

The projected system is an A.I.-based music recommendation system that is based on **streamlite web application** which takes input like the singer's name and language from the user after clicking the 'recommend me a song' button it will automatically sense emotion. An external or in-built webcam is used to capture the real-time image of the user that will be used as input for the programmed system to verify human emotion, then the appearance in the given image is extracted and detected to classify it to one of emotions such as 'Happy', 'Rocking', 'Neutral', 'Sad', or 'Surprised', as shown in following figure:


Figure 5.1 Emotions Fed in the program

Reading the results which come from the appearance that are perceived, the matching music from YouTube will be presented to the user. If the user's real time facial expression is recognised as extremely kicking, then the rock music playlist from youtube will be permitted to the users to select from the music clips..

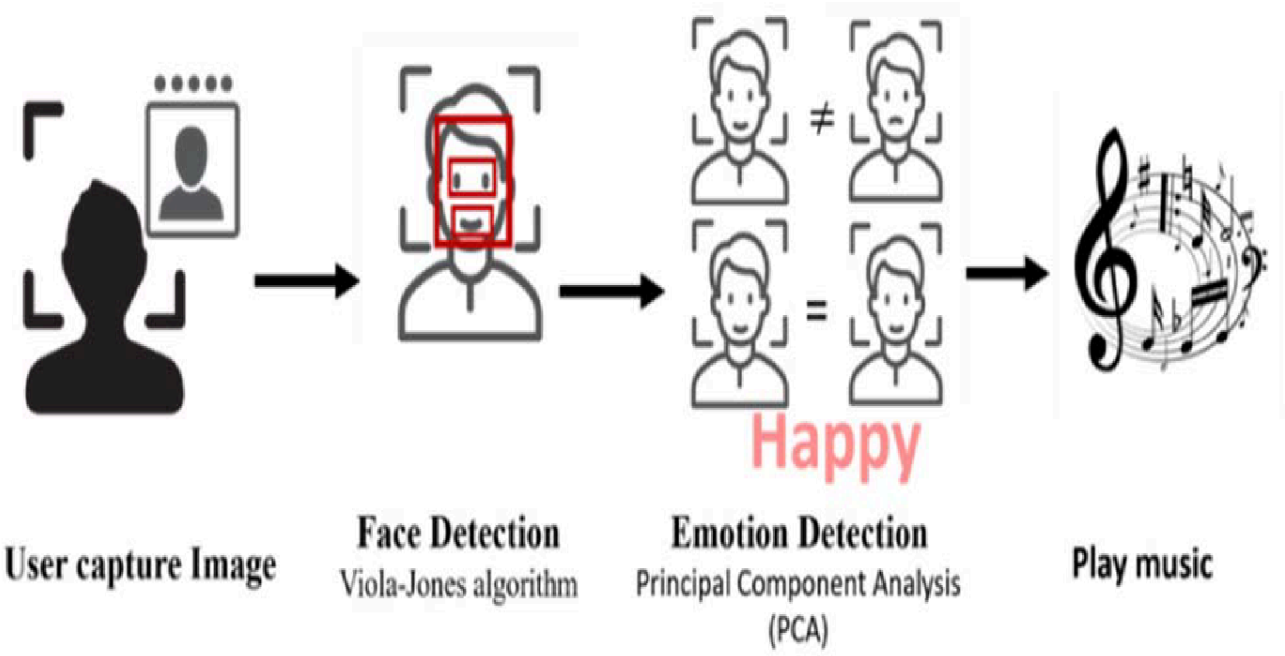



Figure 5.2 System Architecture

## *A. Acquisition of real-time user's image*

In our proposed system, we start by taking user input like "Enter language" and "Enter singer name", and after clicking the "Recommend me song" button, it starts capturing the user's face image using either an external webcam or the in-built laptop webcam. An image of a face correctly processed in the recommendation system should have the face in a frontal position .

## *B. Face Detection*

After receiving the image, the system will start to detect the face by applying theMediaPipe face mesh algorithm, based on real-time deep learning. Additionally, the generated keypoints are coded using a carefully designed series of angular coding modules and grid generators. In addition, feature analysis was performed using principal component analysis (PCA). This phase is implemented to improve the accuracy of emotion detection. Finally, the decomposed features are fed into machine learning (ML) techniques that depend on support vector machines (SVM), k-Nearest Neighbors (KNN) and Logistic Regression (LR). The techniques presented were evaluated on different data sets with different evaluation measures. The results show that they achieve outstanding performance with a human emotion detection accuracy of 92%, ensuring superiority among efforts in this field.

## *C. Real-time detection of human emotion*

Then you need to understand the user's feelings. We use the Deep learning-based face recognition using keras. Deep learning is one of the latest ways to improve the accuracy of facial recognition software. Deep learning extracts unique combinations of faces from images of faces and uses the trained model to recognize photos from a database in other photos and videos.

### *D. Corresponding playlist*

The system will display the corresponding music playlist according to the detected sentiment. At the same time, we have different emotions. For happy emotions, users send YouTube music playlists. For negative emotions, surprises, and sadness, YouTube provides the appropriate music playlists to improve your mood.

## 6. RESULTS AND DISCUSSION

We use the Deep learning-based face recognition using keras. It extracts unique combinations of faces from images of faces and uses the trained model to recognize photos from a database in other photos and Videos  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, to be optimal according to the music recommended to improve the user's mode. was able to propose music.

**Following figures show screenshots of the projected system.**

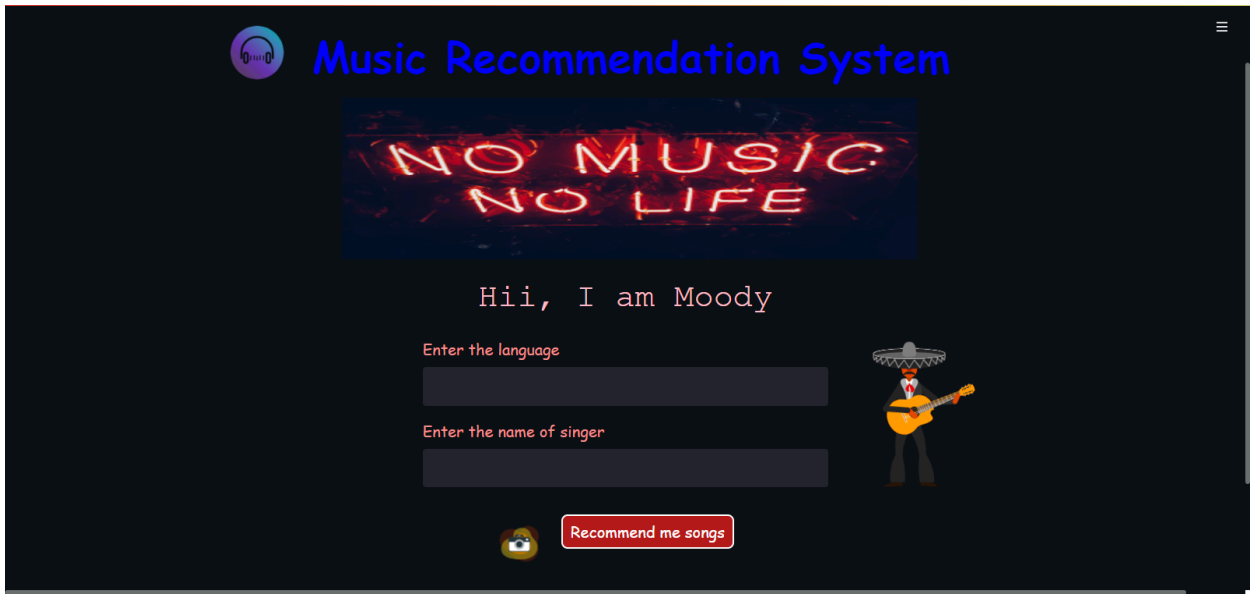



**Figure-6.1-**explain the UI of Emotion based music recommendation system

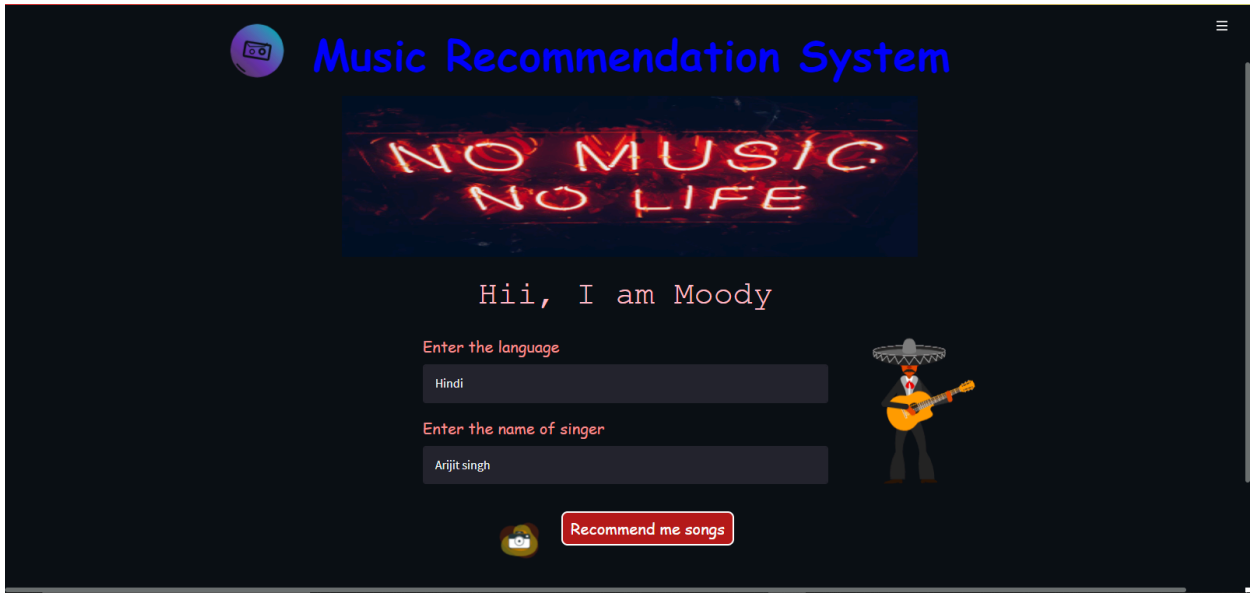



**Figure 6.2** -Explain the Input required values

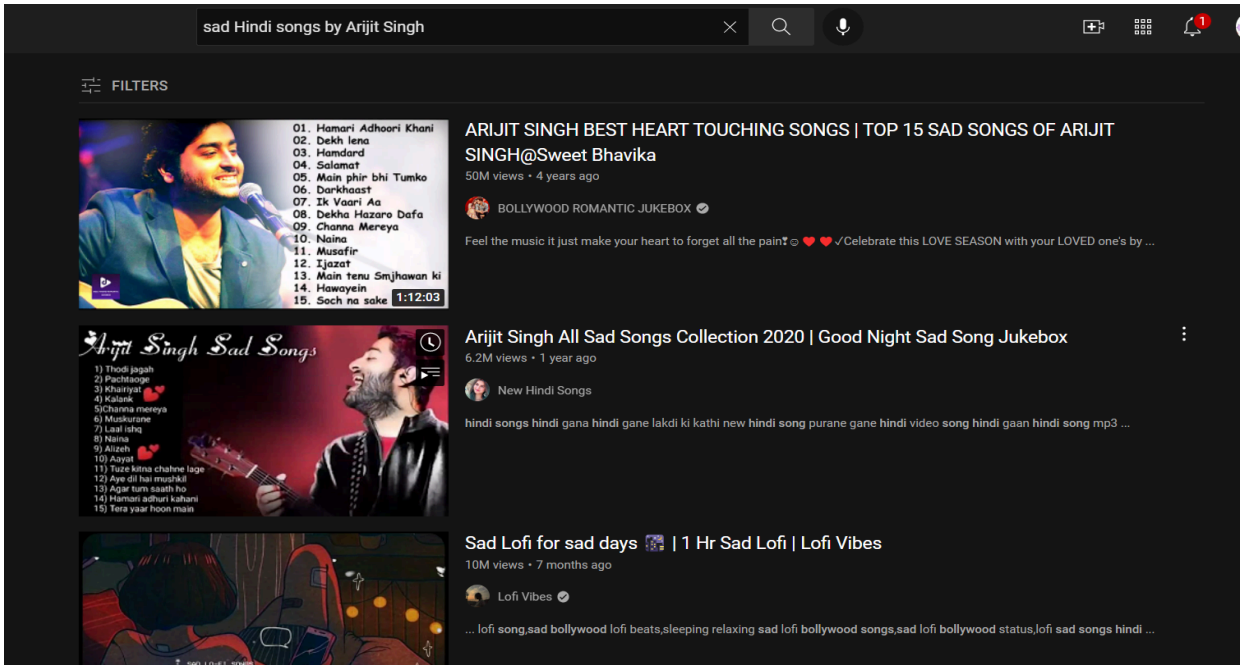



**Figure 6.3** Output as a song

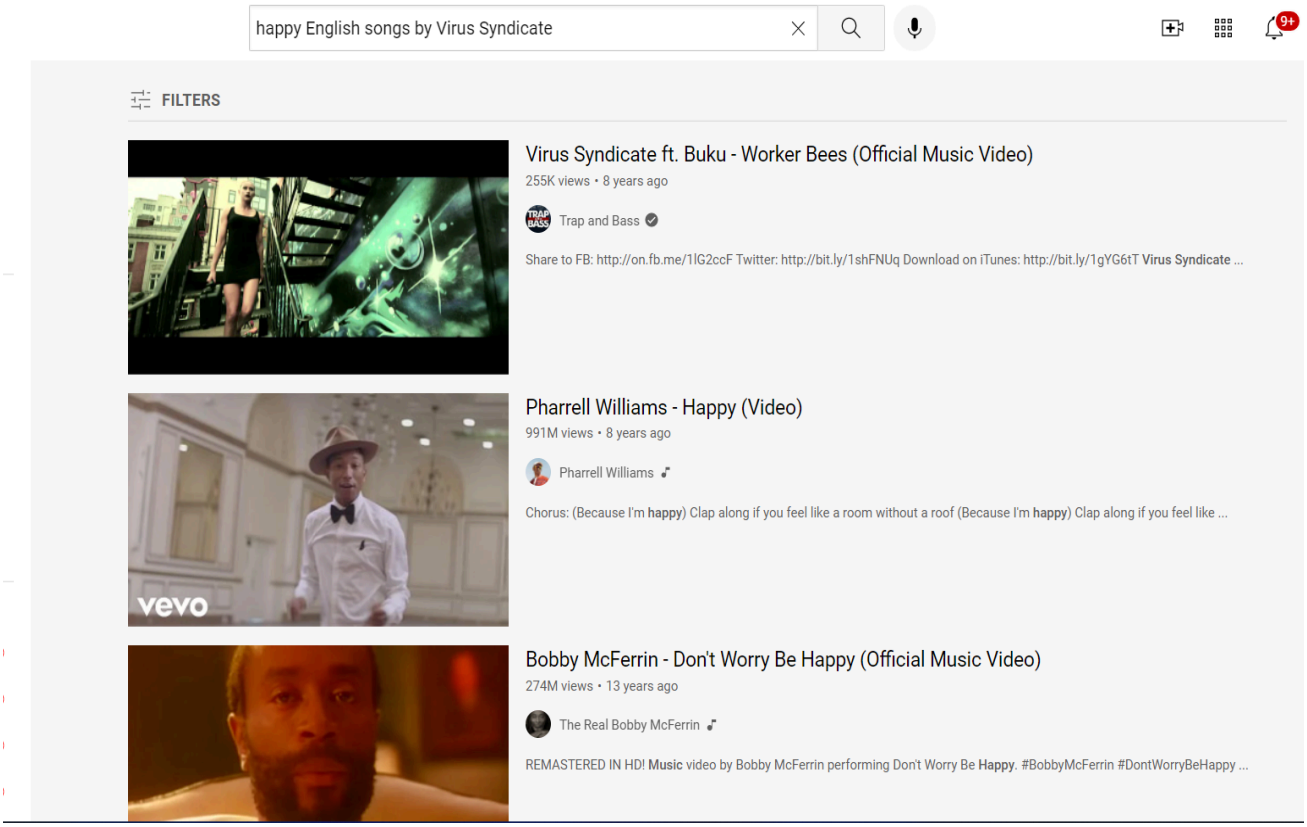



**Figure 6.4 -** Detect facial emotion

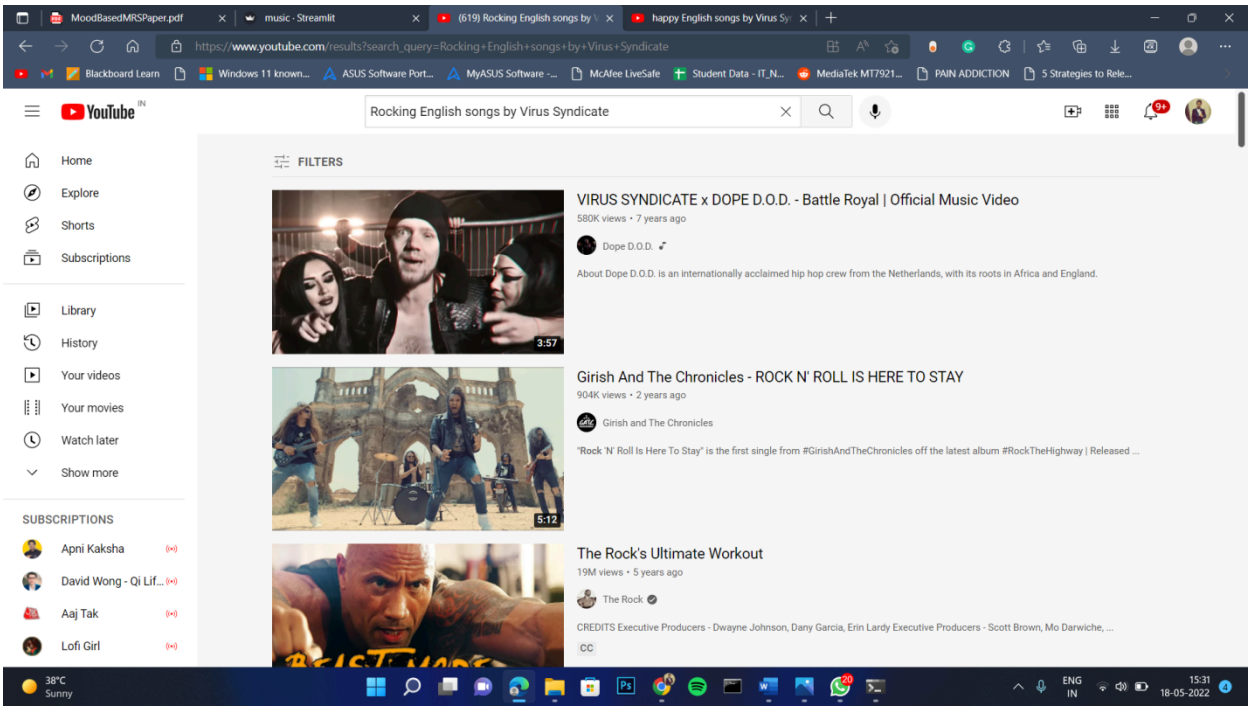



**Figure 6.5 -** Output as a song

# 7. CONCLUSION

In this project, we presented a model to recommend music based on the emotion which is detected from the facial expression. This project proposed, designed and developed an emotion-based music recommendation system using a face recognition system. Music is the one that has the power to heal any stress or any kind of emotions. Recent development promises a wide scope in developing emotion-based music recommendation systems. Thus, the proposed system presents a face based emotion recognition system to detect the emotions and play music from the emotion detected.

Emotion recognition using facial expressions is one of the important topics of research and has gathered much attention in the past. It can be seen that the problem of emotion recognition with the help of image processing algorithms has been increasing day by day. Researchers are continuously working on ways to resolve this by the use of different kinds of features and image processing methods. The applications of Image processing algorithms in the field of both medical science and human science are of vast importance. There are continuously new ways and methods being developed that make use of image processing algorithms to extract the emotion of the user and make use of the extracted emotion to treat the user. Emotion recognition has gained a lot of importance in all aspects of life and if a robust algorithm implemented which can

accurately classify the emotions of the person, then a great deal of advancement in the industry can be achieved with the help of this. 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. However, it has to be 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 .The system was designed using the facial landmarks scheme and was tested under various scenarios for the result that would be obtained.

Due to the subjective nature in music and the issues existing in the previous methods, two human-centered approaches are proposed. By considering affective and social information, emotion-based model and context-based model largely improved the quality of recommendation.

However, this research is still at an early stage.

As we can see from the development of music recommenders over the past years, the given results tend to be more personalized and subjective. Only considering the music itself and human ratings are no longer sufficient.

A great amount of work in recent years has been done in music perception, psychology,neuro- science and sport which study the relationship between music and the impact of human behavior.

Undoubtedly, music has always been an important component of our life, and now we have greater access to it. Human emotions are very complex, but it is possible to train machine-learning models to accurately recognize different emotions that can

be distinguished from each other based on specific facial expressions. You can use a person's facial expressions to identify their emotion, and once a particular emotion is identified, you can suggest music that suits that person's specific mood. Our model, having the accuracy of approximately 92%, is able to detect five emotions accurately: happy, sad, surprise, rocking and neutral; and the redirected youtube link will allow the user to play the music that would be suitable for the detected emotion.

# 8. REFERENCES


[1] D. Rattan, R. Bhatia, and M. Singh, "Software clone detection: A systematic review," *Information and Software Technology*, vol. 55, no. 7, pp. 1165–1199, Jul. 2013.

[2] J. F. Islam, M. Mondal, and C. K. Roy, "Bug Replication in Code Clones: An Empirical Study," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp.68–78.

[3] M. R. Islam and M. F. Zibran, "A Comparative Study on Vulnerabilities in Categories of Clones and Non-cloned Code," in *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, 2016, pp.8–14.

[4] M. R. Islam, M. F. Zibran, and A. Nagpal, "Security Vulnerabilities in Categories of Clones and Non-Cloned Code: An Empirical Study," in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2017, pp. 20–29.

[5] C. K. Roy, M. F. Zibran, and R. Koschke, "The vision of software clone management: Past, present, and future (Keynote paper)," in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)*, 2014, pp.18–33.

[6] J. Krinke, "A Study of Consistent and Inconsistent Changes to Code Clones," in *14th Working Conference on Reverse Engineering (WCRE 2007)*, 2007, pp.170–178.

[7] D. Chatterji, J. C. Carver, N. A. Kraft, and J. Harder, "Effects of cloned code on software maintainability: A replicated developer study," in *2013 20th Working Conference on Reverse Engineering (WCRE)*, 2013, pp. 112–121.

[8] D. Rattan, R. Bhatia, and M. Singh, "An Empirical Study of Clone Detection in MATLAB/ Simulink Models," *International Journal of Information and CommunicationTechnology*.

[9] D. Rattan, R. Bhatia, and M. Singh, "Detecting High Level Similarities in SourceCode and Beyond," *International Journal of Energy, Information and Communications*, vol. 6, no. 2, pp. 1–16, 2015.

[10] D. Rattan, R. Bhatia, and M. Singh, "Detection and Analysis of Clones in UML Class Models," *International Journal of Software Engineering, IJSE*, vol. 8, no. 2, pp. 66– 99, 2015.

[11] D. Rattan, R. Bhatia, and M. Singh, "Model clone detection based on tree comparison," in *2012 Annual IEEE India Conference (INDICON)*, 2012, pp. 1041– 1046.