*A Mini Project Report on*

**EMOTION BASED MUSIC RECOMMENDER USING MACHINE LEARNING**

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

### BACHELOR OF TECHNOLOGY

In

### CSE(DATA SCIENCE)

By

|  |  |
| --- | --- |
| **A Rushmitha** | **(22AG1A6770)** |
| **Huda Nazneen** | **(22AG1A6792)** |
| **S Pradeep** | **(22AG1A67B6)** |
| **V Sai Prabhas** | **(22AG1A67C5)** |

Under the guidance of

#### Mrs.T.Swathi

Assistant Professor



## DEPARTMENT OF CSE(DATA SCIENCE)

**ACE Engineering College**

**Ankushapur(V),Ghatkesar(M),Medchal Dist-501301**

***(An Autonomous Institution,Affiliated to JNTUH,Hyderabad)***

[www.aceec.ac.in](http://www.aceec.ac.in/)

**2024-2025**

## 

## DEPARTMENT OF CSE(DATA SCIENCE)



**CERTIFICATE**

This is to certify that the major project report entitled ***“ Emotion Based music recommender using Machine Learning”*** is a Bonafide work done by **A.Rushmitha (22AG1A6770) , Huda Nazneen (22AG1A6792), S.Pradeep (22G1A67B6),** and **V.Sai Prabhas** (**22AG1A67C5)**in partial fulfillment for the award of Degree of BACHELOR OF TECHNOLOGY in CSE(Data Science) from JNTUH University, Hyderabad during the academic year 2024 - 2025. This record of bonafide work carried out by them under our guidance and supervision.

The results embodied in this report have not been submitted by the student to any other University or Institution for the award of any degree or diploma.

|  |  |  |
| --- | --- | --- |
| **Mrs.T.Swati**  Assistant Professor | **Dr. P Chiranjeevi**  Associate Professor | **External** |
| Internal Guide | HOD-CSE(DS) |  |

### ACKNOWLEDGEMENT

We would like to express my gratitude to all the people behind the screen who have helped me transform an idea into a real-time application.We would like to express my heart-felt gratitude to my parents without whom we would not have been privileged to achieve and fulfill my dreams.

A special thanks to our General Secretary,**Prof.Y.V.Gopala Krishna Murthy,**for having founded such an esteemed institution.Sincere thanks to our Joint Secretary **Mrs.M.Padmavathi**,for support in doing project work.I am also grateful to our beloved principal, **Principal** for permitting us to carry out this project.

We profoundly thank **Dr.P.Chiranjeevi**,Associate Professor and Head of the Department of Computer Science and Engineering(Data Science),who has been an excellent guide and also a great source of inspiration to my work.

We extremely thank **Mr.Parwateeswar ,** Assistant Professor,Project coordinator,who helped us in all the way in fulfilling all aspects in completion of our Mini-Project.

We are very thankful to my internal guide **Mrs.T.Swathi** who has been excellent and also given continuous support for the Completion of my project work.The satisfaction and euphoria that accompany the successful completion of the task would be great,but incomplete without the mention of the people who made it possible,whose constant guidance and encouragement crown all the efforts with success.In this context,we would like to thank all the other staff members,both teaching and non-teaching,who have extended their timely help and eased my task.

A Rushmitha (22AG1A6770)

Huda Nazneen (22AG1A6792)

S Pradeep (22AG1A67B6)

V Sai Prabhas (22AG1A67C5)

# DECLARATION

We here by declare that the results embodied in this project report entitled

**“EMOTION BASED MUSIC RECOMMENDER USING MACHINE LEARNING”** is carried out by us during the year 2024-2025 for the partial fulfilment of the award of **Bachelor of Technology in Computer Science and Engineering (Data Science)** , from **ACE ENGINEERING COLLEGE.** We have not submitted this project report to any other Universities/Institute for the award of any degree.

A Rushmitha (22AG1A6770)

Huda Nazneent (22AG1A6792)

S Pradeep (22AG1A67B6)

V Sai Prabhas (22AG1A67C5)

**EMOTION BASED MUSIC RECOMMENDER USING MACHINE LEARNING**

#### ABSTRACT

People often express their emotions through facial expressions. Music is a powerful tool that can influence and improve a person’s mood. This project aims to detect a person’s emotion—either through facial expressions or manual text input and recommend songs that match their mood, creating a calming and enjoyable experience.The system uses a webcam to capture the user's facial expressions and applies image processing techniques to analyze facial features. With the help of a trained deep learning model, it identifies the user’s emotion—such as happy, sad, angry, or surprised. Alternatively, users can also manually enter their current emotion through a text input option. Based on the detected or entered emotion, language preference, and selected music platform, the system suggests suitable songs. For example, it may suggest pop and dance songs for happy moods or soft rock and acoustic songs for sad moods etc.

This emotion-based music recommendation system not only enhances the user experience but also contributes to mental well-being. By understanding and responding to users' emotional states, the system creates a more personalized and empathetic music experience, potentially helping users manage stress, anxiety, and emotional imbalance through music therapy.The application provides a simple and interactive interface using PyQt5, making it easy for users to receive personalized music recommendations. It also includes a feedback system to improve the accuracy of emotion detection over time.This emotion-based music recommendation system not only enhances the user experience but may also contribute to better emotional well-being by suggesting music that matches and improves the user’s mood.

**CONTENTS**

1. **NO CHAPTER NAME PAGE NO**

**1 INTRODUCTION**

1.1 Background and context of project 1

1.1.1 Applications of Facial Emotion Recognition 1

1.1.2 Importance of Emotion Detection 2

in Various Field

1.1.3 Benefits of Using Machine Learning 2

for Emotion Detection

1**.**1.4 Comparison with Traditional 2

Music Recommendation Methods

1.2 Problem Statement and Objectives 3

1.3 Significance 4

1.4 Scope of the Project 4

**2 LITERATURE SURVEY**

2.1 Existing System 5-7

2.2 Proposed System 7-9

**3 REQUIREMENT ANALYSIS**

3.1 Software Requirements 10

3.2 Hardware Requirements 11

3.3 Functional Requirements 11

3.4 Non-Functional Requirements 12

## 3.5 Environment Setup 13

**4 SYSTEM ANALYSIS**

4.1 Methodology 14

4.2 Modules

4.2.1 Data Collection 15

4.2.2 Face Detection 16

4.2.3 Emotion Recognition 16

**5 SYSTEM DESIGN**

5.1 System Architecture

5.1.1 System Components 17-18

5.1.2 System Workflow 18-19

5.2 UML Diagrams 20

5.2.1 Class Diagram 21

5.2.2 Use Case Diagram 22

5.2.3 Sequence Diagram 23

5.2.4 Activity Diagram 24

**6 IMPLEMENTATION**

6.1 Code Structure Overview 25-29

6.2 Algorithm 29-31

**7 SYSTEM TESTING**

### 7**.1.1 Objective of Testing 32**

### **7.1.2 Test Environment 32**

### 7.1.3 Test Cases 33

### **7.1.4 Feedback Integration Testing 34**

### **7.1.5 Exception Handling 34**

### **7.1.6 Test Summary 34**

**8 OUTPUT SCREENS**  35-38

**9 CONCLUSION AND FUTURE SCOPE** 39-40

**10 REFERENCES** 41

**List Of Figures**

**FIGURE NO FIGURE NAME PAGE NO**

1. Methodology 14
2. Emotion Recognition 16
3. System Architecture 17
4. Class Diagram 21
5. Use Case Diagram 22
6. Sequence Diagram 23
7. Activity Diagram 24
8. Output 1 & 2 35
9. Output 3 & 4 36
10. Output 5 & 6 37
11. Output 7 38

# 1.INTRODUCTION

##### 1.1 Background and Context of the Project:

In the modern digital landscape,recommendation systems play a vital role in enhancing user experiences across platforms such as e-commerce,entertainment,and education.Traditionally,these systems rely on user behavior—such as listening history,search logs,and ratings—to suggest personalized content.However,such methods often fail to capture the real-time emotional state of users,which is a critical factor in content preference,especially for media like music.

To overcome these limitations,this project introduces an **Emotion-Based Music Recommender System** that leverages **facial emotion recognition** to understand the user's current mood and suggest relevant music accordingly.The system uses **MediaPipe** for facial landmark detection and a **pre-trained** model for classifying emotions.A user-friendly interface built with **PyQt5** allows users to either detect emotion through webcam input or enter it manually.Based on the detected emotion and preferred language,the system recommends music via platforms like **YouTube**,**Spotify**,or **Apple Music**.

* + 1. **Applications of Facial Emotion Recognition:**

Facial emotion recognition has wide-ranging applications across various domains,including:

1. **Healthcare**:Monitoring patient emotions for mental health assessments.
2. **Education**:Enhancing e-learning platforms by understanding student engagement.
3. **Entertainment**:Personalizing media content like music or videos.
4. **Human-Computer Interaction**:Making interfaces more responsive and adaptive.

In this project,facial emotion recognition is used to personalize music recommendations—offering songs that align with the user’s detected emotion to improve satisfaction and engagement.

**1.1.2 Importance of Emotion Detection in Various Fields:**

Emotion detection bridges the gap between human perception and machine understanding.In music applications,understanding the listener's current emotional state can dramatically improve recommendation quality.Unlike static behavior-based systems,emotion-aware systems can:

1. Respond in real-time.
2. Adapt to changing moods.
3. Offer context-sensitive content.
4. Increase emotional satisfaction and user retention.

**1.1.3 Benefits of Using Machine Learning for Emotion Detection**

The system uses aCuston trained CNN trained on facial features to classify emotions like **Happy**,**Sad**,**Angry**,**Neutral**,etc.CNNs are ideal for emotion recognition due to their ability to:

1. Extract spatial hierarchies in facial images.
2. Learn patterns without manual feature engineering.
3. Generalize across different faces and lighting conditions.

This enhances both the **accuracy** and **reliability** of real-time emotion classification.

**1.1.4 Comparison with Traditional Music Recommendation**

Traditional methods of emotion detection often rely on psychological questionnaires,manual observation,or textual sentiment analysis.These approaches:

1. Are time-consuming and subjective.
2. Lack real-time adaptability.
3. Do not work effectively in dynamic,interactive environments.

In contrast,this project uses **automated,vision-based emotion detection** that operates in real-time and integrates seamlessly into the user interface,thereby offering a **more scalable and immediate** solution.

* 1. **Problem Statement and Objectives:**

#### ****Problem Statement:****

Traditional music recommendation systems rely heavily on past user behavior,such as listening history,likes/dislikes,and search patterns.While this data-driven approach has proven effective to an extent,it lacks sensitivity to the user’s **current emotional state**,which plays a crucial role in music preference.A user feeling sad may not enjoy songs recommended based on previously liked upbeat music.Moreover,most systems are not capable of adapting their recommendations in **real-time** or based on contextual cues like facial expressions or mood.

#### ****Objectives:****

This project addresses the gap by developing an **Emotion-Based Music Recommender System** that detects a user’s current emotional state through facial expressions or manual text input using computer vision and machine learning,and recommends music accordingly.

1. To build a real-time facial emotion recognition system using computer vision and deep learning(CNN).
2. To integrate MediaPipe for efficient face landmark detection.
3. To create a GUI interface using PyQt5 that allows camera-based or text-based emotion input.
4. To map detected emotions to suitable songs and redirect users to music platforms(YouTube,Spotify,Apple Music).
5. To implement a feedback mechanism to refine emotion prediction over time based on user responses.

**1.3 Significance**

Music has a profound impact on human emotions.People often turn to music not just for entertainment,but to reflect or shift their mood.Integrating **emotion recognition** into recommendation systems brings a new level of personalization that goes beyond passive data collection.

The motivation behind this project is rooted in enhancing user experience by enabling systems to understand and respond to **emotional context**.This also aligns with trends in human-computer interaction,where systems are becoming more **empathetic** and **adaptive**.

This system can also serve as a foundational framework for:

1. Mental wellness apps.
2. Emotion-aware virtual assistants.
3. Smart entertainment systems.

**1.4 Scope of the project:**

The scope of this project involves the development of an Emotion-Based Music Recommender System that intelligently suggests music to users based on their emotional state. The system leverages facial expression recognition using MediaPipe for facial and hand landmark detection, combined with a pre-trained Convolutional Neural Network (CNN) model to classify emotions such as happy, sad, angry, or surprised. It also offers flexibility by allowing users to manually input their current mood through a text field. Based on the detected or entered emotion, along with the user’s preferred language and selected music platform (such as YouTube etc), the system recommends suitable music genres—like pop and dance.

The application provides an intuitive graphical user interface built with PyQt5, enabling smooth interaction and easy switching between camera-based and text-based input modes. To improve system accuracy, a feedback mechanism is included, which prompts users to confirm whether the emotion detection was correct.The integration of technologies such as computer vision, deep learning, and web-based music platforms makes the system adaptable and extendable. This project has a wide scope for future development, including adding voice-based emotion recognition, improving emotion classification models with larger datasets, integrating in-app music players, and expanding to mobile or web platforms. Overall, the project not only enhances user experience but also contributes to emotional well-being through personalized music recommendations.

**2.LITERATURE SURVEY**

Emotion-based music recommendation systems are designed to suggest songs based on the user's current emotional state. Traditional music recommenders use methods like collaborative filtering or past user behavior, but these systems do not consider the user’s emotions at the moment. Emotion-aware systems solve this by detecting emotions in real-time using facial expressions, speech, or physical signals. In this project, facial expressions are used to detect emotions with the help of computer vision and machine learning, and songs are recommended based on the detected emotions.

Machine learning plays a major role in emotion classification. In this project, a deep learning model built with Keras takes input features from the user’s face and hands and predicts the emotion. These features are 1020 values collected from Mediapipe’s landmark positions. This approach is similar to the work done by researchers like Tang (2013), who used CNNs on facial datasets like FER2013 for emotion recognition.

For recommending music, many existing systems like Spotify and Apple Music use user history or popularity. But more advanced systems now include context such as mood or time. For example, Baltrunas et al. (2011) worked on using user context for better recommendations. Kim et al. (2010) used physical signals such as heart rate to suggest music that matches the user’s mood. This project uses detected emotions to suggest calm and relaxing songs for sad or angry emotions, and party songs for neutral or surprised moods.

Some systems also combine multiple sources like facial expressions, speech, and text for better emotion detection. Zhou et al. (2017) showed that using more than one source improves accuracy. Although this project only uses visual input, it can be improved later by adding other inputs like voice.

The tools used in this project include Mediapipe for getting facial and hand landmarks, OpenCV for capturing video from the webcam, PyQt5 for the graphical user interface, and Keras with TensorFlow for the emotion detection model. The GUI allows users to choose between camera and text input, and also to pick their preferred music platform and language. A feedback system is included to help improve future emotion predictions.

### ****2.1 Existing System****

Music recommendation systems have evolved considerably over the past decades,primarily focusing on understanding user preferences through **historical usage data**.The most common techniques employed include:

* **Collaborative Filtering(CF):**This method recommends songs based on the preferences of users with similar listening behaviors.For instance,if User A and User B share many liked songs,the system will suggest to User A the songs favored by User B that User A has not yet explored.
* **Content-Based Filtering:**This approach analyzes the audio features of songs such as tempo,rhythm,melody,and genre.It recommends tracks similar to those a user has previously enjoyed.
* **Hybrid Models:**Many platforms combine CF and content-based methods to enhance accuracy.

Popular music streaming services such as **Spotify**,**Apple Music**,and **YouTube Music** use complex hybrid recommendation models that also include natural language processing(NLP)techniques to analyze song metadata,user reviews,and lyrics.Despite their sophistication,these models are largely static in the sense that they reflect **long-term user preferences** rather than transient,momentary states.

Some research efforts have explored **sentiment analysis** using textual data mined from social media posts,chat messages,or user-generated comments.These methods attempt to infer the user’s mood by analyzing the polarity and emotional content of+text.While this adds some contextual awareness,it is limited by:

* **Latency issues:**Textual inputs may be delayed or sparse,limiting real-time responsiveness.
* **Single modality limitation:**Emotions are complex and often better captured through multiple channels,including facial expressions and voice tone.
* **Dependence on explicit user input:**Many systems require the user to consciously express their mood or emotion,reducing ease of use.

On the facial emotion recognition front,earlier approaches typically relied on **rule-based facial landmark detection** using geometric measurements of key facial points(e.g.,eye corners,lip edges).While these techniques laid foundational work,they suffer from:

* Sensitivity to lighting,occlusion,and pose variations,
* Limited adaptability to diverse facial expressions,
* Dependence on manually designed rules,leading to poor scalability.

More recent deep learning methods,including CNNs trained on large facial emotion datasets,have significantly improved accuracy and robustness.However,these models are not yet widely integrated into real-time music recommendation systems.

### ****2.2 Proposed System****

The proposed system aims to fill the identified gaps by integrating **real-time facial emotion recognition** into the music recommendation process and through text input,thus creating a more **emotion-aware** and personalized user experience.

#### Key Features:

1. **Facial Landmark Detection with MediaPipe:**MediaPipe,developed by Google,offers an efficient and lightweight framework for detecting facial landmarks in real time.It provides precise tracking of 468 3D facial landmarks,enabling robust capture of subtle facial movements and expressions even under variable lighting and orientations.This landmark detection serves as the foundational input for emotion classification.
2. **Emotion Classification using CNN:**A Convolutional Neural Network,pre-trained on extensive facial emotion datasets(such as FER-2013 or CK+),processes the extracted landmarks or face images to categorize user emotions into classes like Happy,Sad,Angry,Fearful,Neutral,etc.CNNs excel in learning hierarchical patterns from visual data,offering superior accuracy compared to rule-based systems.
3. **User Interface with PyQt5:**To maximize usability,the system employs a graphical interface using PyQt5.This interface allows users to interact via:

* **Live webcam input:**Captures the user’s current facial expression continuously,
* **Manual text input:**Provides a fallback for users preferring to specify their mood explicitly or when the webcam is unavailable.

1. **Emotion-based Query Construction:**The system maps detected emotions to mood-specific song queries,factoring in language preference.It then launches the relevant music streaming platform(YouTube,Spotify,or Apple Music)with tailored recommendations,ensuring an immediate and intuitive response to the user’s emotional state.
2. **Feedback Mechanism:**Recognizing that emotion recognition is inherently challenging and not always perfect,the system incorporates a feedback module.Users can confirm or correct the detected emotion,enabling the collection of labeled data for continuous model refinement.Over time,this iterative learning improves system accuracy and user satisfaction.

#### Advantages and Impact:

1. **Real-time Responsiveness:**Unlike traditional models dependent on historical data,this system dynamically adapts to the user’s fleeting emotional state,capturing transient moods that strongly influence music preference.
2. **Multimodal Emotion Detection:**By combining visual emotion recognition with optional textual input,the system supports richer and more flexible emotion sensing.
3. **Personalized,Empathetic Interaction:**Emotion-aware recommendations foster a more engaging and supportive user experience,making music selection feel more intuitive and responsive to the user's inner feelings.
4. **Integration of Affective Computing and Multimedia:**This system exemplifies a cutting-edge application of affective computing principles within intelligent multimedia recommendation,pushing forward the frontier of emotionally intelligent user interfaces.

In summary,the proposed system not only enhances recommendation accuracy by incorporating emotional context but also pioneers a seamless,empathetic approach to human-computer interaction in digital music platforms.

**Algorithms and Tools:**

This code implements an Emotion-Based Music Recommender System using Machine Learning and Computer Vision. The project utilizes a Convolutional Neural Network (CNN) for emotion classification, which is trained to recognize emotions from facial landmarks. These landmarks are extracted in real-time using MediaPipe Holistic, which detects facial and hand features from webcam input. The application is built using PyQt5 for the graphical interface, allowing users to choose between facial detection or manual text input. OpenCV handles webcam integration and image processing, while NumPy supports numerical computations. The Keras library is used to load the pre-trained emotion recognition model. Based on the detected or entered emotion and and language input, webbrowser is used to open music recommendations on platforms like YouTube, Spotify, or Apple Music. User feedback is collected and stored using JSON, enabling future improvements in detection accuracy over time.

**3.REQUIREMENT ANALYSIS**

**3.1 Software Requirements**

**1.Programming Language:**

Python:It simplifies complex operations like data preprocessing,model training,and real-time detection.Libraries like NumPy,Pandas,and Scikit-learn streamline data handling,while TensorFlow supports model building and training.

**2.Operating System(OS):**Windows 10/11,macOS,or Linux(with GUI support)

**3.OpenCV:** For image and video processing:OpenCV is used for capturing and processing real-time images or video feeds.It handles tasks like face detection,image with camera hardware.

**4.Machine Learning Frameworks:**

* NumPy(array operations)
* Mediapipe(for facial landmark detection)
* TensorFlow/Keras(to load and run emotion detection model)

**5.User Interface:**

PyQt5(for GUI):It is used to create the app’s user interface in a simple and attractive way.It helps show live video,take user input,and manage buttons and options easily.

**6.Other:**

* **Web Browser:**Default browser to open music recommendation links(YouTube,Spotify,Apple Music)
* **Pre-trained Model Files:**
* model.h5(emotion detection model)
* lables.npy(emotion labels)
* **JSON Module:**for storing and retrieving user feedback data.

### 3.2 Hardware Requirements

1. **Webcam:**A functional webcam with at least 640x480 resolution to capture user facial expressions.
2. **Processor:**Minimum Intel Core i3 or equivalent to handle real-time video processing and model inference.
3. **Memory:**Minimum 4 GB RAM for smooth operation.
4. **Storage:**At least 100 MB free space for installing dependencies and storing feedback files.

## 3.3 Functional Requirements

The following are the core functional requirements for the Emotion-Based Music Recommender System:

1. **Real-Time Video Capture:**  
   The system shall capture real-time video from the user's webcam to analyze facial expressions.
2. **Facial Landmark Detection:**  
   The system shall use Mediapipe to detect facial landmarks from the video frames.
3. **Emotion Prediction:**  
   The system shall use a pre-trained Keras model to predict the user's emotion based on the detected facial landmarks.
4. **Manual Emotion Input:**  
   The system shall allow users to manually enter their emotion via text input as an alternative to automatic detection.
5. **Emotion-Based Music Recommendation:**  
   The system shall recommend songs that match the detected or manually entered emotion.
6. **Music Platform Integration:**  
   The system shall open the selected music platform(YouTube,Spotify,or Apple Music)in the default web browser using an emotion-based search query.
7. **User Feedback Collection:**  
   The system shall allow users to provide feedback on whether the detected emotion was accurate.

## 3.4 Non-Functional Requirements

## Here are the non-functional requirements for the facial emotion recognition system:

1. **Video Performance:**  
   The system shall maintain a video frame update rate of at least 30 frames per second to ensure smooth real-time detection.
2. **User Data Privacy:**  
   The system shall ensure user data privacy by storing feedback only on the local machine,with no external data transmission.
3. **User Interface Usability:**  
   The system shall provide a clean,responsive,and intuitive graphical user interface(GUI)using PyQt5.
4. **Error Handling:**  
   The system shall handle common errors gracefully,such as unavailable webcam or failure to load the model,and notify the user appropriately.
5. **Cross-Platform Compatibility:**  
   The system shall be compatible with major operating systems(Windows,macOS,and Linux)that support Python and PyQt5.
6. **Low-Latency Feedback:**  
   The system shall process user input and provide emotion detection results within one second to ensure responsiveness.

## 3.5 Environment Setup

This section outlines the steps required to set up the development and runtime environment for the Emotion-Based Music Recommender system.

### 3.5.1 Prerequisites

1. **Python Version:**Python 3.7 or above
2. **Operating Systems Supported:**Windows 10/11,macOS,Linux
3. **Webcam:**Required for live emotion detection
4. **Internet Connection:**Required for accessing YouTube,Spotify,or Apple Music in the browser

### 3.5.2 Installing Dependencies

### It is recommended to use a virtual environment for managing dependencies.

#### Step 1:Create and Activate a Virtual Environment

#Windows

python-m venv env

env\Scripts\activate

#macOS/Linux

python3-m venv env

source env/bin/activate

#### Step 2:Install Required Packages:

pip install pyqt5 opencv-python mediapipe tensorflow numpy

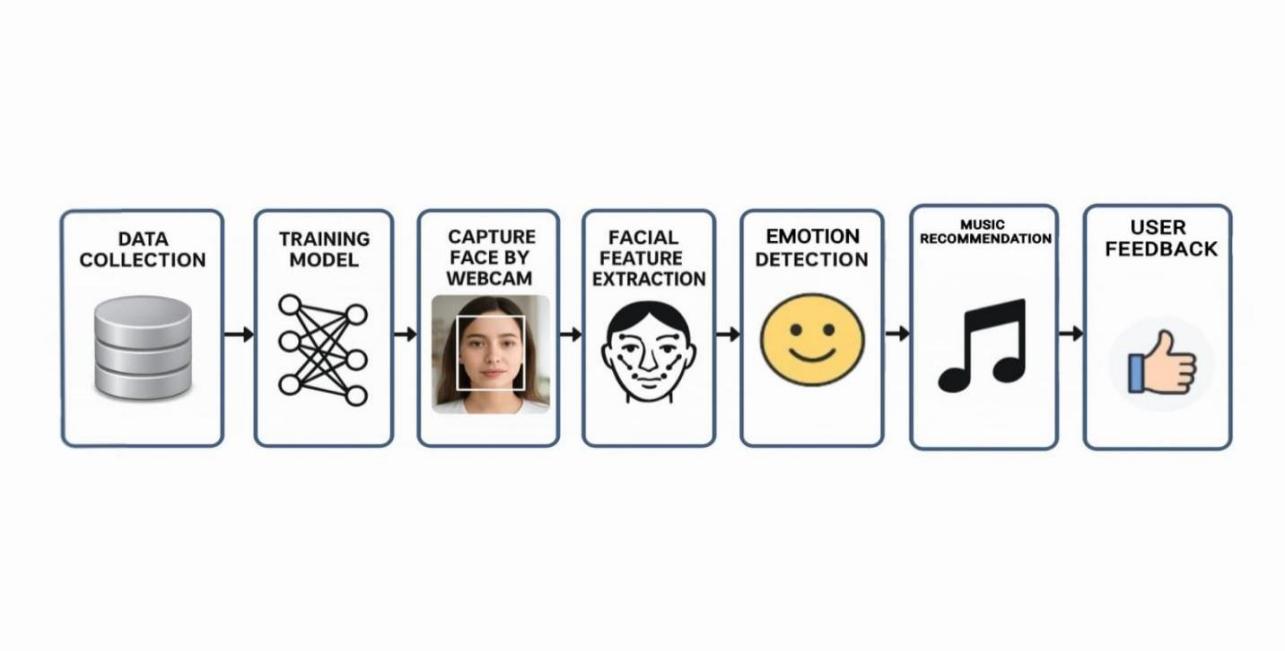
### 3.5.3 Project File Setup

Ensure the following files are present in the project directory:

1. emotion\_music\_app.py–Main application script
2. model.h5–Pre-trained Keras model for emotion detection
3. labels.npy–Numpy file containing emotion class labels
4. feedback.json–(Optional)Feedback data file;auto-created if not present

**4.SYSTEM ANALYSIS**

* 1. **Methodology**



# Fig 1 Methodology

The system utilizes a **hybrid emotion recognition approach** by offering two modes:

1. **Automatic mode(via webcam):**Uses a live video feed to detect facial expressions,extract features using Mediapipe,and predict emotions with a deep learning model.
2. **Manual mode(text input):**Allows the user to input their emotional state directly.

Once the emotion is recognized or entered,the system generates a song recommendation tailored to the emotion and selected platform(YouTube,Spotify,or Apple Music).The system also incorporates **feedback loops** where users confirm or correct detected emotions,enabling future accuracy improvements.

The overall flow can be summarized as:

1. Input(Webcam or Text)
2. Facial landmark detection(if webcam used)
3. Emotion classification(via pre-trained Keras model)
4. Music recommendation based on emotion and language
5. Platform redirection(opens in browser)
6. Feedback collection and learning

### 4.2 Modules

#### 4.2.1 Data Collection

**Purpose:** To gather visual input from the user for emotion recognition.

**Objective:**

The data collection module in the Emotion-Based Music Recommender application primarily involves capturing real-time facial and hand landmarks from the user through a webcam. This is accomplished using MediaPipe Holistic, which detects and tracks multiple body landmarks, including the face and hands, to extract relevant features that represent the user's emotional state.

**Workflow:**

### Data Collection:

* User chooses between Camera mode or Text mode
* Capturing live video frames from the webcam using OpenCV.
* Detecting facial and hand landmarks in each frame with MediaPipe Holistic.
* Collecting user feedback on detected emotions (stored in JSON) to improve the system over time.

### Preprocessing :

* Flipping the video frames horizontally to provide a mirror-like view.
* Converting frames from BGR (OpenCV format) to RGB (MediaPipe format) color space.
* Normalizing facial landmark coordinates relative to a reference point to make the data position-invariant.
* Combining face and hand landmark coordinates into a single fixed-length feature vector (padding with zeros if needed) for consistent input size.
* Formatting this feature vector as a NumPy array to feed into the deep learning emotion classifier.

**Quality Assurance:**

Perform quality control checks to ensure dataset integrity, correctness of labeling, and consistency in image quality.

#### 4.2.2 Face Detection

**Tool Used:**Mediapipe Holistic model

**Function:**

1.Extracts 3D coordinates of facial landmarks(e.g.,eyes,nose,mouth).

2.Normalizes the coordinates relative to a base point for consistent feature extraction.

**Output:**A fixed-size feature vector representing the user's facial expression.

#### 4.2.3 Emotion Recognition

**Model Used:**Pre-trained Keras model(model.h5)

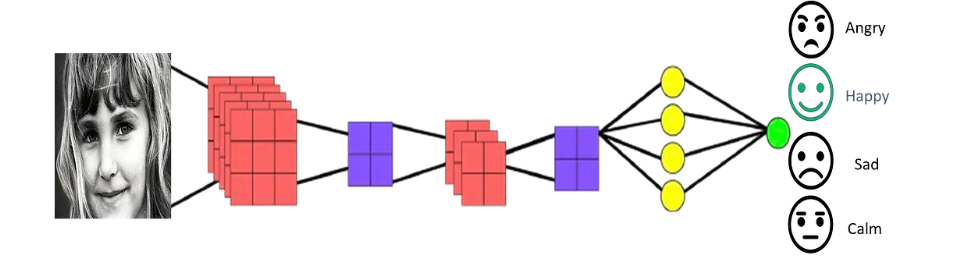
**Functionality:**

1. Takes the normalized facial landmark features as input.

2.Outputs a probability distribution across multiple emotion classes(e.g.,Happy,Sad,Angry,etc.).

3.Predicts the most probable emotion,which is then displayed and used for music recommendation.

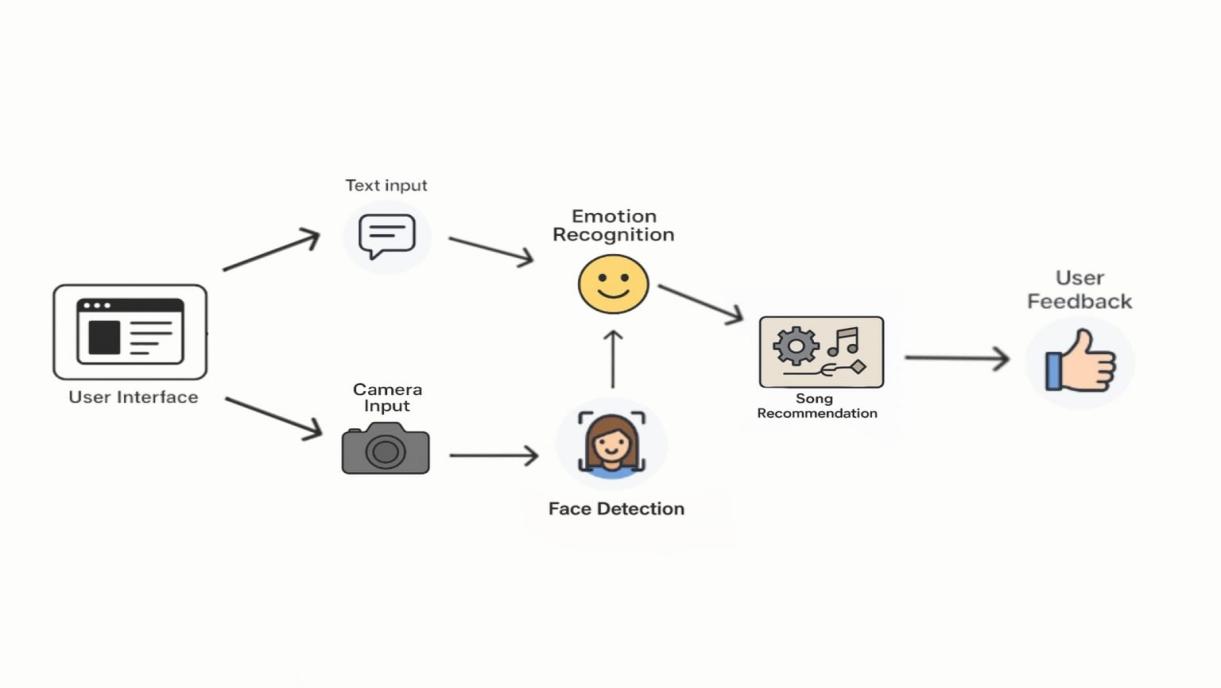
**Adaptation:**Incorporates user feedback from feedback.json to adjust the final prediction dynamically(e.g.,correcting over-detection of“Angry”).

****

**Fig 2. Emotion Recognition Diagram**

**5.SYSTEM DESIGN**

##### 5.1 System Architecture

****

**Fig 3. System Architecture**

**5.1.1 System Components:**

**1.User Interface(GUI–PyQt5):**

* Displays webcam feed or accepts manual text input.
* Shows detected emotion and provides platform options(YouTube,Spotify,Apple Music).
* Enables interaction with detection and music recommendation buttons.

1. **Input Module:**

**Camera Input:**Captures real-time video using OpenCV.

**Manual Input:**Accepts text-based emotion input if the user prefers not to use the camera.

**3.Facial Landmark Detection Module:**

* Utilizes **MediaPipe Holistic** for detecting facial landmarks(eyes,nose,mouth,etc.).
* Outputs landmark coordinates used as feature vectors.

**4.Emotion Classification Module:**

* Loads a pre-trained **Keras model**(model.h5)to classify emotions based on landmark data.
* Uses labels.npy for mapping predictions to readable emotion labels.

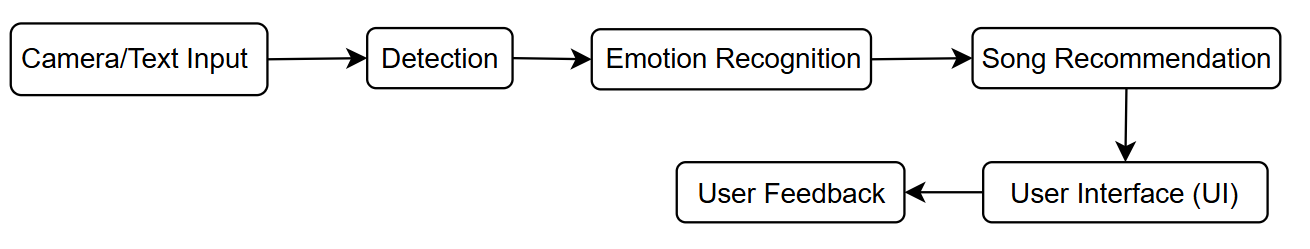
**5.Feedback System:**

* Prompts users for feedback on predicted emotion accuracy.
* Stores responses in feedback.json.
* Uses historical feedback to adjust prediction outcomes for recurring misclassifications.

**6.Music Recommendation Module:**

* Maps detected emotion to relevant songs.
* Constructs platform-specific search URLs using emotion+language.
* Opens the result in the system’s default web browser.

##### Simple Diagram:



This simplified design allows the system to detect faces, recognize their emotions, and give songs in real-time.

#### ****5.1.2 System Workflow****

**Workflow for the facial emotion recognition system:**

**1.Input Selection:**

User selects between"Camera"and"Text"modes.

In camera mode,webcam is activated;in text mode,user inputs an emotion manually.

**2.Face Detection:**

If using the camera,MediaPipe detects facial landmarks from the video stream.

**3.Emotion Prediction:**

Landmark data is processed and passed to the Keras model.

Predicted emotion is displayed.

**4.Feedback Mechanism:**

After prediction,the user is prompted for feedback.

If marked inaccurate,the feedback is recorded and used in future predictions.

**5.Music Recommendation:**

User enters language preference and selects a platform.

Application launches a browser tab with a search query tailored to the user's mood and preferences.

**6.**  **Output:**

The final output is a live display of detected faces, their corresponding emotions, and Song Recommendation. This workflow ensures the system continuously detects faces, classifies emotions, and provides real-time feedback and song suggestions.

**5.2 UML Diagrams:**

UML stands for Unified Modelling Language.UML is a standardized general purpose modelling language in the field of object-oriented software engineering.The standard is managed,and was created by,the Object Management Group.The goal is for UML to become a common language for creating models of object-oriented computer software.In its current form UML comprises two major components,a Meta-model and a notation.In the future,some form of method or process may also be added to or associated with,UML.

The Unified Modelling Language is a standard language for specifying,Visualization,Constructing and documenting the artifacts of software systems as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.The UML is a very important part of developing object oriented software and the software development process.The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1.Provide users a ready-to-use,expressive visual modeling Language so that they can develop and exchange meaningful models.

2.Provide extendibility and specialization mechanisms to extend the core concepts.

3.Be independent of particular programming languages and development process.

4.Provide a formal basis for understanding the modeling language.

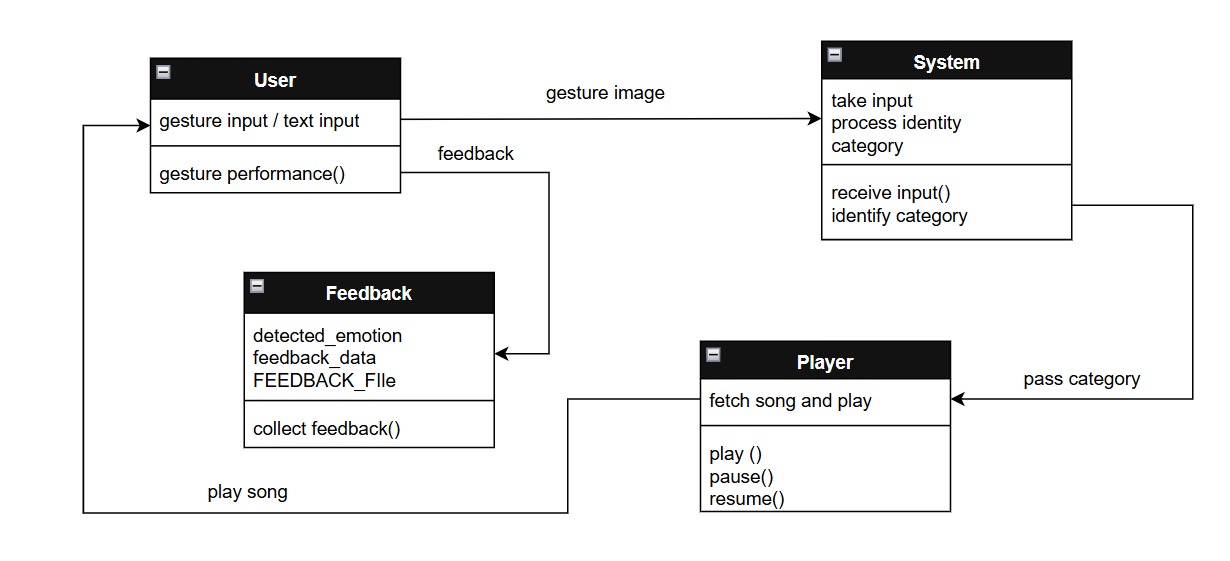
5.Encourage the growth of OO tools market.

6.Support higher level development concepts such collaborations,frameworks,patterns and components.

7.Integrate best practices.

**5.2.1 CLASS DIAGRAM:**

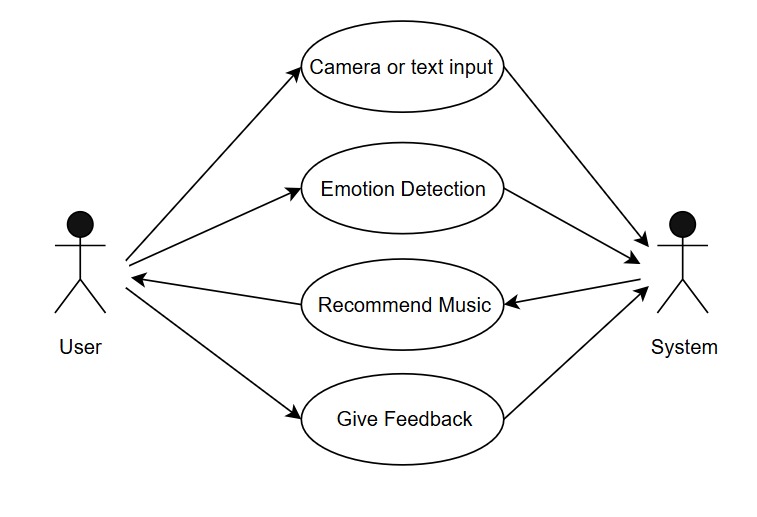
The class diagram illustrates the interaction between three core components of the system:**User**,**System**,and **Player**.Each class represents a major entity in the architecture of a gesture-controlled music playback application.



**Fig 4. Class Diagram**

**5.2.2 USE CASE DIAGRAM:**

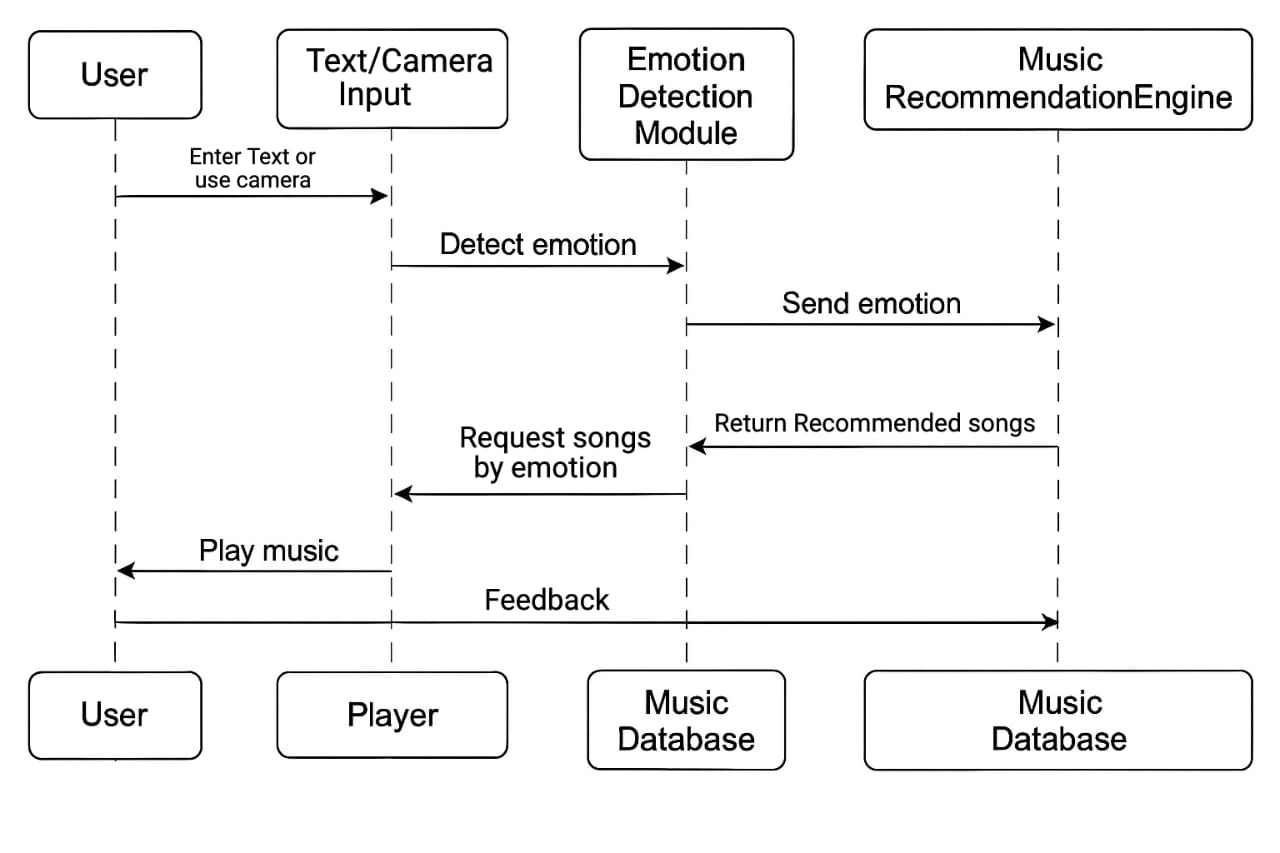
A use case diagram in the Unified Modeling Language(UML)is a type of behavioral diagram defined by and created from a Use-case analysis.Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors,their goals(represented as use cases),and any dependencies between those use cases.The main purpose of a use case diagram is to show what system functions are performed for which actor.Roles of the actors in the system can be depicted.



**Fig 5. Use Case Diagram**

**5.2.3 SEQUENCE DIAGRAM:**

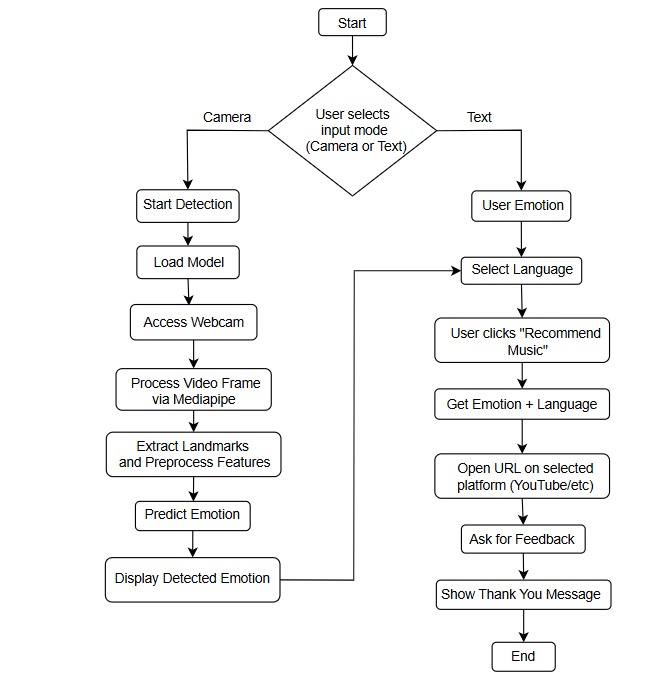
This sequence diagram demonstrates the **dynamic interaction** between different components of an **Emotion-Based Music Recommendation System**.The process begins with the user and continues through emotion detection to song recommendation and playback.



**Fig 6. Sequence Diagram**

**5.2.4 ACTIVITY DIAGRAM:**

The **activity diagram** represents the dynamic workflow of the emotion-based music recommendation system.It outlines how a user interacts with the system to receive music suggestions based on either automatically detected or manually input emotions.The diagram ensures a clear view of the system’s operations from input to intelligent response generation.

****

**Fig 7. Activity Diagram**

# 6.MPLEMENTATION

* 1. **Code Structure Overview**

1. **Imports and Setup**

**import sys**

**import os**

**import json**

**import cv2**

**import numpy as np**

**import mediapipe as mp**

**import webbrowser**

**from PyQt5.QtWidgets import (**

**QApplication, QWidget, QLabel, QPushButton, QVBoxLayout,**

**QComboBox, QLineEdit, QMessageBox, QHBoxLayout**

**)**

**from PyQt5.QtGui import QImage, QPixmap, QFont**

**from PyQt5.QtCore import QTimer, Qt**

**DATA COLLECTION**

def update\_frame(self):

if not self.cap or not self.cap.isOpened():

return

ret, frame = self.cap.read()

if not ret:

return

frame = cv2.flip(frame, 1)

rgb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

results = holistic.process(rgb)

features = []

if results.face\_landmarks:

origin = results.face\_landmarks.landmark[1]

for lm in results.face\_landmarks.landmark:

features.extend([lm.x - origin.x, lm.y - origin.y])

if results.left\_hand\_landmarks:

for lm in results.left\_hand\_landmarks.landmark:

features.extend([lm.x, lm.y])

if results.right\_hand\_landmarks:

for lm in results.right\_hand\_landmarks.landmark:

features.extend([lm.x, lm.y])

features = features[:1020] + [0.0] \* max(0, 1020 - len(features))

features\_np = np.array(features).reshape(1, -1)

if model\_loaded and results.face\_landmarks:

pred = model.predict(features\_np, verbose=0)

label = labels[np.argmax(pred)]

self.detected\_emotion = label

self.emotion\_label.setText(f"🎭 Detected Emotion: {label}")

else:

self.emotion\_label.setText("😶 No face detected")

h, w, ch = frame.shape

img = QImage(frame.data, w, h, ch \* w, QImage.Format\_BGR888)

self.video\_label.setPixmap(QPixmap.fromImage(img))

**DATA TRAINING**

#Pre-trained

from keras.models import load\_model

model = load\_model("model.h5", compile=False)

**VALIDATION & TESTING**

import cv2

import numpy as np

import mediapipe as mp

from keras.models import load\_model

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

cap = cv2.VideoCapture(0)

while True:

frm = cap.read()

    frm = cv2.flip(frm, 1)

    res = holis.process(cv2.cvtColor(frm, cv2.COLOR\_BGR2RGB))

    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)

        else:

            for i in range(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)

drawing.draw\_landmarks(frm,res.face\_landmarks, holistic.FACEMESH\_CONTOURS)

drawing.draw\_landmarks(frm,res.left\_hand\_landmarks,

    if cv2.waitKey(1) == 27:

        cv2.destroyAllWindows()

        cap.release()

        break

**USER INTERFACE**

from PyQt5.QtWidgets import (

QApplication, QWidget, QLabel, QPushButton, QVBoxLayout,

QComboBox, QLineEdit, QMessageBox, QHBoxLayout

)

from PyQt5.QtGui import QFont

from PyQt5.QtCore import Qt, QTimer

class EmotionMusicApp(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setWindowTitle("🎵 Emotion-Based Music Recommender")

self.setGeometry(100, 100, 900, 650)

self.setStyleSheet("""

QWidget { background-color: #1e1e1e; color: #f0f0f0; font-family: 'Segoe UI'; font-size: 15px; }

QPushButton { background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 #4e54c8, stop:1 #8f94fb); color: white; border-radius: 10px; padding: 8px; }

QPushButton:hover { background-color: #6C63FF; }

QLineEdit, QComboBox { background-color: #2b2b2b; color: white; border: 1px solid #555; border-radius: 8px; padding: 6px; }

QLabel { font-weight: bold; }

""")

self.layout = QVBoxLayout()

# Title Label

self.title\_label = QLabel("🎧 Emotion-Based Music Recommender")

self.title\_label.setAlignment(Qt.AlignCenter)

self.title\_label.setFont(QFont("Segoe UI", 20, QFont.Bold))

self.layout.addWidget(self.title\_label)

# Video Display

self.video\_label = QLabel()

self.video\_label.setFixedSize(640, 480)

self.video\_label.setStyleSheet("border: 2px solid #555; border-radius: 10px;")

self.layout.addWidget(self.video\_label, alignment=Qt.AlignCenter)

# Input Mode Selection

controls\_layout = QHBoxLayout()

self.input\_mode\_combo = QComboBox()

self.input\_mode\_combo.addItems(["Camera", "Text"])

self.text\_input = QLineEdit()

self.text\_input.setPlaceholderText("Enter Emotion (e.g. Happy)")

self.text\_input.setVisible(False)

controls\_layout.addWidget(self.input\_mode\_combo)

controls\_layout.addWidget(self.text\_input)

self.layout.addLayout(controls\_layout)

# Emotion Label

self.emotion\_label = QLabel("🎭 Detected Emotion: Not Detected")

self.emotion\_label.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.emotion\_label)

# Language Input

self.lang\_input = QLineEdit()

self.lang\_input.setPlaceholderText("Enter Language (e.g. Hindi, English, Tamil)")

self.layout.addWidget(self.lang\_input)

# Music Platform Dropdown

self.platform\_dropdown = QComboBox()

self.platform\_dropdown.addItems(["YouTube", "Spotify", "Apple Music"])

self.layout.addWidget(self.platform\_dropdown)

# Buttons Layout

btn\_layout = QHBoxLayout()

self.detect\_button = QPushButton("🎥 Start Detection")

self.music\_button = QPushButton("🎶 Recommend Music")

btn\_layout.addWidget(self.detect\_button)

btn\_layout.addWidget(self.music\_button)

self.layout.addLayout(btn\_layout)

self.setLayout(self.layout)

* 1. **Algorithm**

**MODEL BUILDING**

ip = Input(shape=(X.shape[1])) # Input layer: shape = (1020,)

m = Dense(512, activation="relu")(ip) # Hidden layer 1: 512 units

m = Dense(256, activation="relu")(m) # Hidden layer 2: 256 units

op = Dense(y.shape[1], activation="softmax")(m) # Output layer: units = number of classes

**Start Detection(Camera or Text):**

def start\_detection(self):

self.feedback\_given=False

if self.input\_mode\_combo.currentText()=="Camera":

if not model\_loaded:

QMessageBox.warning(self,"Model Error","Model not loaded!")

return

self.cap=cv2.VideoCapture(0,cv2.CAP\_DSHOW)

self.cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH,640)

self.cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT,480)

if not self.cap.isOpened():

QMessageBox.critical(self,"Camera Error","Webcam not found!")

return

self.timer.start(30)

else:

self.detected\_emotion=self.text\_input.text().strip().capitalize()

self.emotion\_label.setText(f"🎭Detected Emotion:{self.detected\_emotion}")

**Feedback Collection**

def collect\_feedback(self):

if emo not in feedback\_data:

feedback\_data[emo]={"Yes":0,"No":0}

if res==QMessageBox.Yes:

feedback\_data[emo]["Yes"]+=1

else:

feedback\_data[emo]["No"]+=1

with open(FEEDBACK\_FILE,"w")as f:

json.dump(feedback\_data,f,indent=2)

QMessageBox.information(self,"Thank You!","Your feedback helps us improve!")

**Adjust Prediction Based on Feedback**

if model\_loaded and results.face\_landmarks:

pred = model.predict(features\_np, verbose=0)

label = labels[np.argmax(pred)]

self.detected\_emotion = label

self.emotion\_label.setText(f"🎭 Detected Emotion: {label}")

**Songs Database**

def recommend\_music(self):

emotion = self.detected\_emotion.lower()

lang = self.lang\_input.text().strip().lower()

if not lang:

lang = "hindi" if emotion in ["sad", "angry", "fearful"] else "punjabi"

if emotion in ["sad", "angry", "fearful", "disgusted"]:

query = f"{lang} calm relaxing positive songs"

elif emotion in ["neutral", "surprised"]:

query = f"{lang} party dance songs"

else:

query = f"{lang} {emotion} songs"

query = query.replace(" ", "+")

urls = {

"YouTube": f"https://www.youtube.com/results?search\_query={query}",

"Spotify": f"https://open.spotify.com/search/{query}",

"Apple Music": f"https://music.apple.com/us/search?term={query}"

}

webbrowser.open(urls.get(platform, urls["YouTube"]))

self.emotion\_label.setText(f"🎵 Searching for '{query}' on {platform}")

**Cleanup on Close**

def closeEvent(self,event):

self.stop\_camera()

event.accept()

**Main Execution**

if\_\_name\_\_=="\_\_main\_\_":

app=QApplication(sys.argv)

window=EmotionMusicApp()

window.show()

sys.exit(app.exec\_())

**7.SYSTEM TESTING**

System testing is a critical phase in software development where the complete and integrated software is tested to verify that it meets the specified requirements.The following outlines the system testing conducted on the Emotion-Based Music Recommender application.

### ****7.1.1 Objective of Testing****

1.Validate the correctness and robustness of the emotion detection functionality using webcam input or manual entry.

2.Ensure accurate recommendation of music based on detected or provided emotions.

3.Verify integration of key components:GUI,emotion detection,feedback collection,and music recommendation engines.

4.Confirm UI/UX reliability and performance under expected user interaction scenarios.

### ****7.1.2 Test Environment****

**1.Operating System:**Windows 10/11

**2.Python Version:**Python 3.8+

**3.Libraries:**PyQt5–GUI framework

OpenCV–Webcam access and image processing

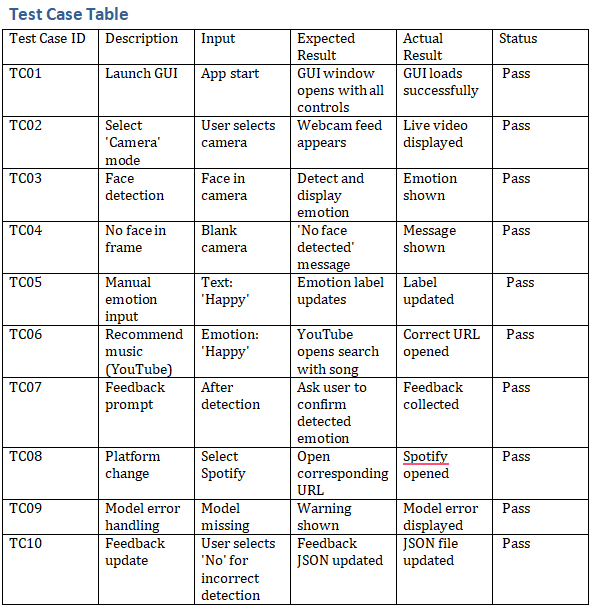
Mediapipe–Facial landmark detection

Keras/TensorFlow–Emotion classification

Webbrowser–Launching platform URLs

JSON–Feedback persistence

**7.1.3 Test Cases:**

****

### ****7.1.4 Feedback Integration Testing****

Tested whether feedback mechanism updates the feedback.json file and influences future label predictions.A high"No"ratio for an emotion like"Angry"led the system to adjust predictions accordingly.

### ****7.1.5 Exception Handling****

The system gracefully handles:

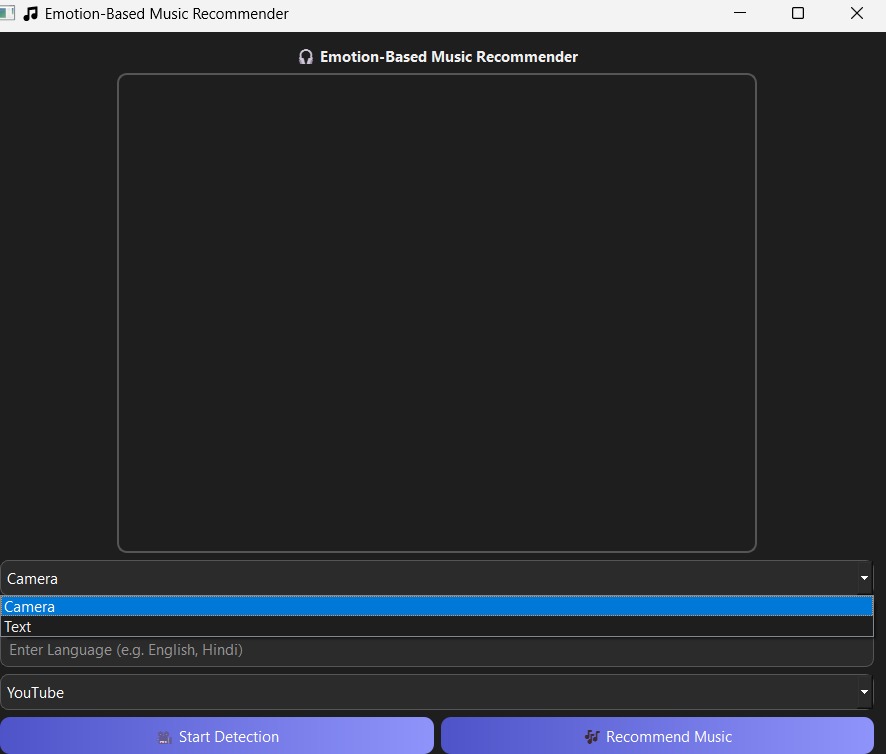
1. Missing model files(model.h5,labels.npy)
2. Inaccessible webcam
3. Empty emotion or language fields
4. Incomplete user inputs

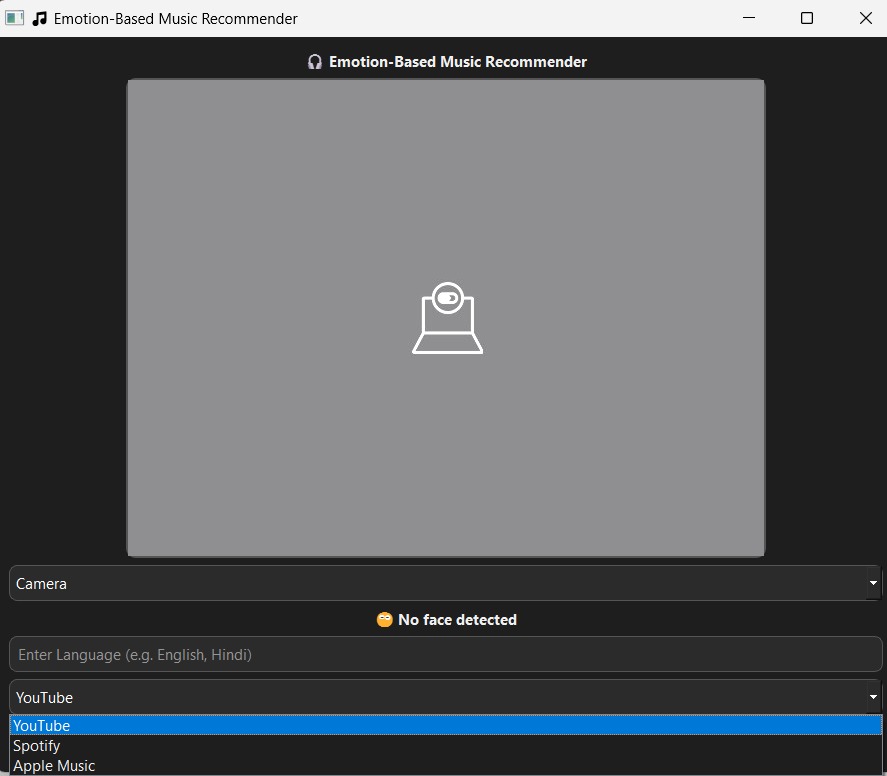
All exceptions are captured using message boxes to ensure a user-friendly interface.

### ****7.1.6 Test Summary****

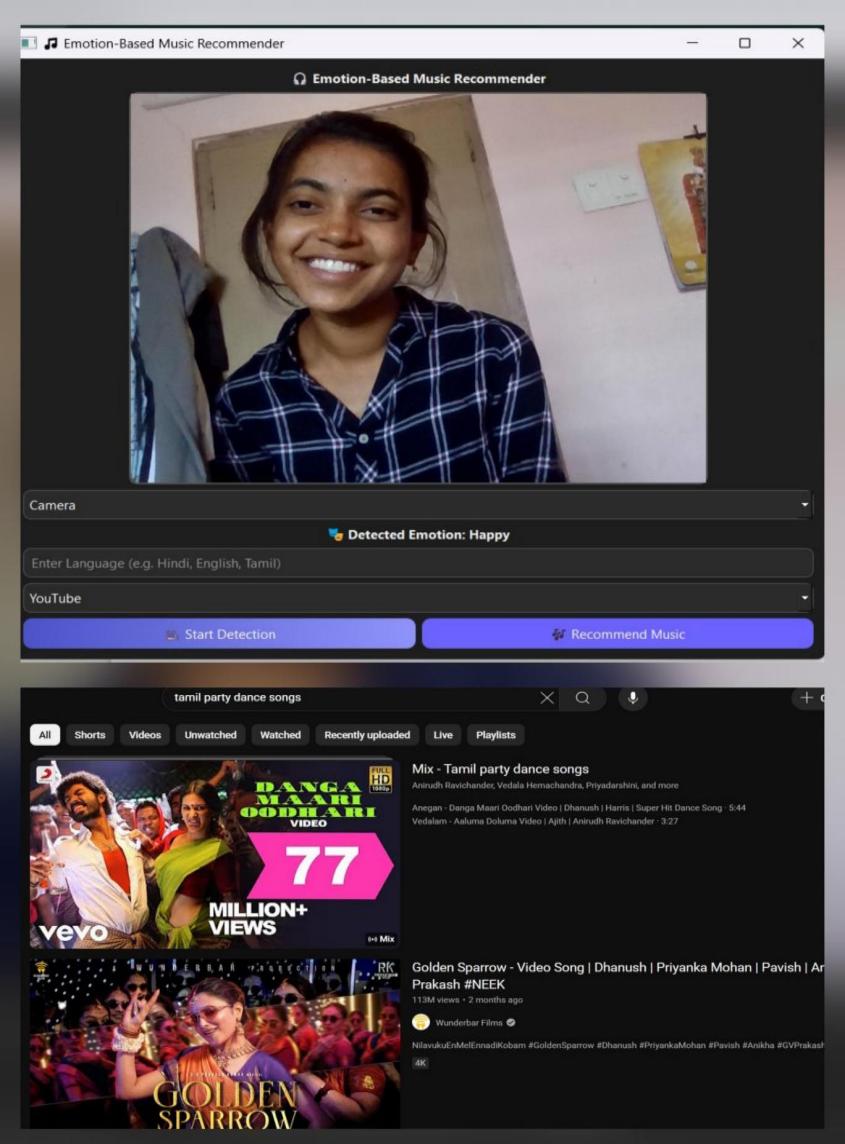
|  |  |
| --- | --- |
| Metric | Result |
| Total Test Cases | 10 |
| Passed | 10 |
| Failed | 0 |
| Overall Result | All functionalities verified successfully |

**8.Output Screens**

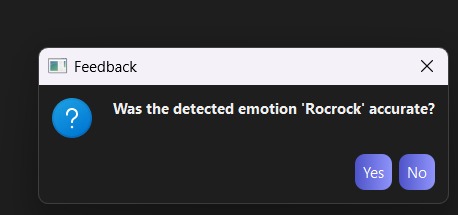
****

****

**Fig 8. Output 1 & 2**

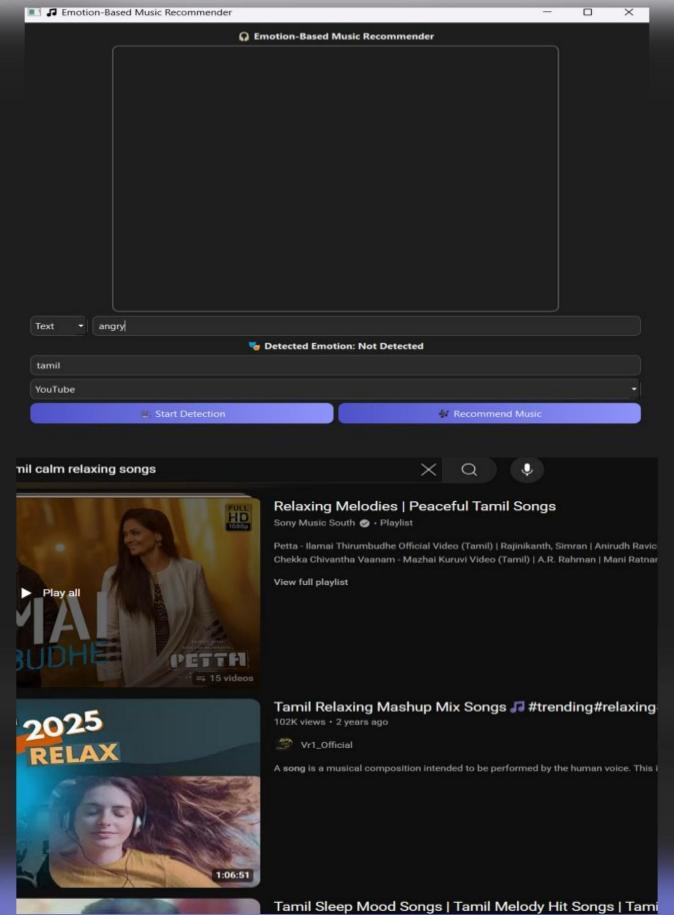
****

### ****WhatsApp Image 2025-06-05 at 15.30.52****

****

**Fig 9. Output 3 & 4**

### ****WhatsApp Image 2025-06-05 at 15.30.38****

****

**Fig 10. Output 5 & 6**

### ****WhatsApp Image 2025-06-05 at 15.30.50****

**Fig 11. Output 7**

**9. CONCLUSION AND FUTURE SCOPE**

### ****Conclusion:****

This project presents an **innovative and user-centric approach** to music recommendation by integrating **real-time facial emotion recognition** with multimedia recommendation systems.The hybrid system bridges the gap between affective computing and traditional recommender models,delivering **emotion-aware and context-sensitive song suggestions** to enhance user experience.

By using **MediaPipe** for accurate facial landmark extraction and a **Convolutional Neural Network(CNN)**for emotion classification,the system effectively detects users'emotional states through a webcam feed.This real-time emotional insight is then used to personalize song recommendations across popular platforms like **YouTube,Spotify,and Apple Music**.

Additionally,the **PyQt5-based GUI** ensures accessibility,offering both live camera and manual input modes for user flexibility.The inclusion of a **feedback mechanism** further strengthens the system by collecting user validation,paving the way for self-improvement through user interaction.

This system not only enhances recommendation accuracy but also brings empathy into human-computer interaction—something that traditional systems often lack.

### ****Future Scope:****

Several enhancements and research directions can improve and expand the system:

1. **Multi-modal Emotion Detection**  
   Integrating **voice tone**,**body posture**,and **text sentiment** with facial analysis can increase emotion detection accuracy and context awareness.
2. **Deep Feedback Learning Loop**  
   Instead of storing simple binary feedback,future versions can include **reinforcement learning models** that adapt the emotion classification and music mapping dynamically over time.
3. **Emotion-to-Genre Mapping Engine**  
   Creating a large-scale emotion-to-music genre mapping dataset,possibly with user clustering or collaborative filtering,would improve personalization
4. **Mobile&Cross-Platform Deployment**  
   Packaging the system into **Android/iOS apps** using frameworks like Flutter,React Native,or PyQt for mobile could increase user reach and real-world impact.
5. **Emotion Variation Over Time**  
   Implementing **emotion tracking over a timeline** would allow the system to understand emotional transitions and recommend a sequence of songs to uplift or stabilize mood
6. **Integration with Wearable Devices**  
   Combining facial emotion with **biometric data(heart rate,skin temperature)**from smartwatches or AR/VR gear could create a holistic emotion recognition ecosystem.
7. **Expanded Music Sources&Personal Libraries**  
   Integrating with personal local libraries or more streaming platforms like **Amazon Music,Gaana,JioSaavn**,etc.,would improve accessibility and variety.

8.**voice input:** for more natural and accessible interaction. Instead of relying solely on text or camera-based emotion detection, users would be able to speak their emotional state, and the system would process the input using speech recognition and natural language understanding.

**REFERENCES**

[1]Emanuel I.Andelin and Alina S.Rusu,”Investigation of facial microexpressions of emotions in psychopathy-a case study of an individual in detention”,2015,Published by Elsevier Ltd.

[2]Paul Ekman,Wallace V Friesen,and Phoebe Ellsworth.Emotion in the human face:Guidelines for research and an integration of findings.Elsevier 2013.[3]F.De la Torre and J.F.Cohn,“Facial expression analysis,”Vis.Anal.Hum.,pp.377–410,2011.

[3]Zhang,Z.Feature-based facial expression recognition:Sensitivity analysis and experiments with a multilayer perceptron.International Journal of Patten Recognition and Artificial Intelligence.

[4]RemiDelbouys,Romain´Hennequin,Francesco Piccoli,Jimena RoyoLetelier,Manuel Moussallam.“Music mood detection based on audio.

[5]nd lyrics with Deep Neural Net”,19th International Society for Music Information Retrieval Conference,Paris,France,2018.

[6]KrittrinChankuptarat,etal,“Emotion Based Music Player”,IEEE 2019 conference.

[7]Kim,Y.:Convolutional Neural Networks for Sentence Classification.In:Proceedings of the 2014 Conference on EMNLP,pp.1746–1751(2014).

[8]Tripathi,S.,Beigi,H.:Multi-Modal Emotion recognition on IEMOCAP Dataset using Deep Learning.In:arXiv:1804.05788(2018).

[9]Tengetal.,”Recognition of Emotion with SVMs”,Lecture Notes in Computer Science,August 2006.