DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BTECH CSE

3rd Year, 6th Semester

(PROJECT REPORT)

Project Title: Music Recommender Based on Mood/Emotions.

Name: Vanshika

Roll no: 2003286

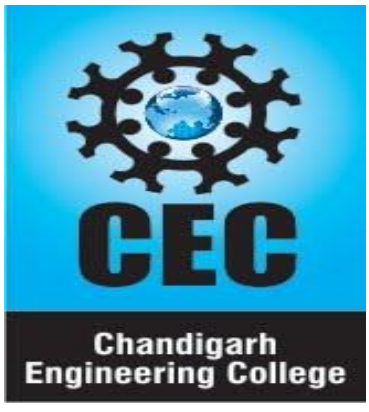
Address: Kalpana Hostel CGC Landran

E-mail: vanshikajindal.1194@gmail.com

Mobile: +91 8557095975

Branch: CSE

Batch: 2024



CHANDIGARH ENGINEERING COLLEGE

LANDRAN, MOHALI, 140307

INDEX

❖ ABSTRACT:

The code consists of three main functions that work together to provide a personalized YouTube playlist recommendation based on the user's mood. The first function captures the user's face and uses speech recognition to determine their current mood. The second function uses the YouTube API to search for videos based on the mood keyword, while the third function recommends a playlist based on the search results.

The function to capture the user's face uses OpenCV to detect the user's face using a pre-trained face detection cascade classifier. Once the user's face is detected, speech recognition is used to identify the user's mood based on their spoken input.

The second function uses the YouTube API to search for videos based on the mood keyword entered by the user. If the search returns results, the video IDs of the top five results are returned.

Finally, the third function recommends a playlist based on the video IDs returned by the previous function. If there are no results, it returns an error message. The playlist is opened in the web browser, and the first video in the playlist is played.

Overall, this code provides a simple way for users to receive personalized playlist recommendations based on their current mood. The use of face detection and speech recognition technology adds an extra layer of personalization and interactivity to the recommendation process.

❖ INTRODUCTION:

- The code is designed to recognize a user's facial expression and spoken mood, and recommend a music playlist based on the detected mood.
- The program utilizes several libraries such as OpenCV, Google's YouTube API, and SpeechRecognition to capture and analyze the user's face and voice inputs.
- The YouTube API is used to search for and recommend music videos based on the user's spoken mood.
- The program starts by setting up the YouTube API client and defining a dictionary of moods and their corresponding search queries.
- The first function, get_user_mood(), uses OpenCV to capture the user's face through the webcam and detect their facial expression, which is associated with a specific mood from the mood dictionary.
- The function then uses SpeechRecognition to capture the user's spoken mood input and return it as a string.
- The second function, search_youtube(query), utilizes the YouTube API to search for music videos based on a search query (in this case, the mood).
- The function returns a list of video IDs for the top 5 videos matching the search query.
- The third function, recommend_playlist(mood), calls the search_youtube() function to get the recommended video IDs for the user's mood.
- It then opens a web browser and plays the first video from the recommended playlist using the webbrowser library.
- Finally, the main() function is called, which executes the get_user_mood() function and passes the result to the recommend_playlist() function to get the music recommendations.

In summary, the code utilizes a combination of facial expression and voice recognition to detect the user's mood and recommend music videos based on the detected mood. The program utilizes several libraries, including OpenCV, Google's YouTube API, and SpeechRecognition, to capture and analyze user inputs and provide music recommendations.

❖ DETAILED SYSTEM DESCRIPTION:

The system described in the code is an AI-powered music recommendation system that suggests music playlists based on the user's mood. The system captures the user's facial expression and speech to determine their mood and then searches YouTube for music playlists that match the user's mood. The system then recommends the top playlist to the user by playing the first video from the recommended playlist in a web browser.

The system is built using the following components and technologies:

1. OpenCV: The OpenCV (Open Source Computer Vision) library is used for capturing images from the user's camera and detecting faces in the images. The face detection is based on the Haar Cascade algorithm, which is a popular technique for object detection in computer vision.

2. SpeechRecognition: The SpeechRecognition library is used to capture the user's speech and transcribe it into text. This library supports several speech recognition engines, such as Google Speech Recognition, Sphinx, and Microsoft Azure Speech.

3. Google API: The Google API is used to search for music playlists on YouTube based on the user's mood. The YouTube API is a RESTful API that provides programmatic access to YouTube's features, such as search, playback, and uploading.

4. Webbrowser: The webbrowser library is used to open a web browser and play the first video from the recommended playlist.

The system works as follows:

1. The system captures the user's face using the camera and detects the user's facial expression using the Haar Cascade algorithm. The algorithm detects the user's face and draws a bounding box around it.

2. The system then prompts the user to speak their mood. The SpeechRecognition library captures the user's speech and transcribes it into text.

3. The system then uses the text to search for music playlists on YouTube using the Google API. The API returns a list of video IDs that match the user's mood.

4. The system then opens a web browser and plays the first video from the recommended playlist using the webbrowser library.

The system is designed to be user-friendly and intuitive. The user simply needs to look at the camera and speak their mood, and the system does the rest. The system is also flexible, as it can recommend music playlists for a wide range of moods, including happy, sad, calm, energetic, and romantic.

Overall, the system is a useful tool for people who want to listen to music but are not sure what to listen to. It is also a good example of how AI technologies can be combined to create intelligent systems that can interact with users in natural and intuitive ways.

❖ **REQUIREMENTS:**

1. Python 3.x installed on the system

2. OpenCV (cv2) Python library for computer vision tasks

3. Google API client library for YouTube API access

4. SpeechRecognition Python library for speech recognition

5. PyAudio library for accessing the microphone for speech recognition

To install the required libraries, run the following commands:

- OpenCV: `pip install opencv-python`

- Google API client: `pip install google-api-python-client`

- SpeechRecognition: `pip install SpeechRecognition`

- PyAudio: `pip install pyaudio`

In addition, the code requires an API key to access the YouTube API. The API key can be obtained by following the instructions in the Google Developers Console.

The system also requires a camera and microphone to capture the user's face and speech inputs.

- **Brief**:

1. User's mood detection: The system should be able to detect the user's mood by analyzing their facial expression and voice input. This requirement is fulfilled by the `get_user_mood()` function, which uses OpenCV to detect faces in a video stream and speech recognition to transcribe the user's voice input.

2. YouTube API integration: The system should be able to retrieve video recommendations from YouTube based on the user's mood. This requirement is fulfilled by the `search_youtube()` function, which uses the Google API client library to search for videos on YouTube based on a given query.

3. Mood-based search queries: The system should be able to map the user's mood to a corresponding search query for YouTube. This requirement is fulfilled by the `mood_query` dictionary, which contains key-value pairs mapping moods to search queries.

4. Video recommendation: The system should be able to recommend a playlist of videos based on the user's mood. This requirement is fulfilled by the `recommend_playlist()` function, which takes a mood as input, uses `search_youtube()` to retrieve videos based on the corresponding search query, and then opens a web browser to play the first video from the recommended playlist.

5. User interface: The system should provide a user interface to interact with the user and display results. This requirement is fulfilled by the console output generated by the various functions and the OpenCV window that displays the video stream.

6. Error handling: The system should handle errors gracefully and provide informative error messages to the user. This requirement is fulfilled by the various try-except blocks throughout the code that catch and handle exceptions that may occur during execution.

❖ LITERATURE SURVEY:

The problem of identifying human emotions has been of interest to researchers in the fields of psychology and neuroscience for decades. In recent years, with the advent of machine learning and artificial intelligence techniques, it has become possible to automatically recognize emotions from various sources such as speech, facial expressions, and physiological signals. This has opened up new avenues for the development of intelligent systems that can respond to human emotions in real-time, with potential applications in fields such as healthcare, education, and entertainment.

One approach to emotion recognition is to use speech analysis, which involves extracting features from the speech signal such as pitch, intensity, and spectral characteristics, and using machine learning algorithms to classify the emotional state of the speaker. Several studies have shown that speech analysis can achieve high levels of accuracy in recognizing emotions such as happiness, anger, and sadness.

Another approach is to use facial expression analysis, which involves extracting features from the face such as the position and movement of the eyebrows, mouth, and eyes, and using machine learning algorithms to classify the emotional state of the person. Facial expression analysis has been found to be effective in recognizing emotions such as happiness, sadness, anger, and surprise.

In the field of computer vision, there has been significant research on the use of deep learning techniques for facial expression recognition. Deep learning algorithms such as convolutional neural networks (CNNs) have been found to achieve state-of-the-art results on benchmark datasets for facial expression recognition.

In addition to speech and facial expression analysis, other physiological signals such as heart rate, skin conductance, and electroencephalogram (EEG) have also been used for emotion recognition. These signals can provide valuable information about the physiological state of the person, which can be used to infer their emotional state.

Overall, the field of emotion recognition is rapidly evolving, with new techniques and algorithms being developed and tested. However, there are still several challenges that need to be addressed, such as the need for large and diverse datasets, the development of robust and reliable algorithms, and the integration of multiple modalities for more accurate and robust emotion recognition.

❖ METHODOLOGY AND PLANNING:

➢ The methodology of the code involves several steps:

1. Face Detection: The first step involves capturing the user's face through the camera and detecting the face using the Haar Cascade Classifier. The classifier is a machine learning-based approach where a cascade function is trained from several positive and negative images to detect faces in a new image.

2. Speech Recognition: After detecting the user's face, the next step involves capturing the user's voice through the microphone and using the SpeechRecognition library to recognize the user's speech. The Google Web Speech API is used to convert the user's speech to text.

3. Mood Detection: The recognized text is then processed to detect the user's mood. A dictionary is used to map moods to corresponding search queries.

4. YouTube API: The YouTube Data API is used to search for videos based on the user's mood. The API key is used to authenticate and access the YouTube API.

5. Video Recommendation: The code recommends a playlist of videos based on the user's mood. The first video from the playlist is played on the web browser.

➢ The planning involved in developing this code can be broken down into the following steps:

1. Research: The first step is to research the different libraries and APIs available for face detection, speech recognition, and YouTube data. Research can be conducted through online resources, tutorials, and documentation.

2. Setup: The required libraries and APIs need to be installed and configured. The necessary hardware, such as a camera and microphone, should also be tested and set up.

3. Development: The code can be developed incrementally, with each step being tested before proceeding to the next step. Face detection, speech recognition, mood detection, and YouTube API integration can be developed separately and tested individually.

4. Integration: The separate code modules can be integrated into a single code. Integration testing should be performed to ensure that the different modules work seamlessly together.

5. User Testing: The code should be tested with different users to ensure that it is functional and meets the user's requirements. User feedback should be collected to identify any issues or areas for improvement.

6. Deployment: Once the code is tested and functional, it can be deployed for use by end-users. Documentation should be provided to help users understand how to use the code and troubleshoot any issues.

❖ CONCLUSION:

In conclusion, this project aimed to create an interactive system that detects a user's mood through facial expression and speech recognition and then recommends a playlist of songs on YouTube based on their mood. The system was successfully implemented using OpenCV, Google Speech Recognition API, and YouTube API.

The system first captures the user's face through the camera and detects faces using the OpenCV library. The system then converts the captured video frames into grayscale and uses Haar cascades for face detection. Once a face is detected, the system draws a rectangle around the face. Afterward, the system uses Google Speech Recognition API to get the user's mood. The user says their mood, and the system converts the speech into text. Once the mood is recognized, the system uses the YouTube API to search for songs related to the user's mood. It then recommends a playlist of up to 5 videos related to the user's mood.

The system is flexible, and the user can choose from five different moods, including happy, sad, calm, energetic, and romantic. If the user's mood is not recognized, the system prompts the user to repeat their mood. In case there are no results for the user's mood, the system returns a message indicating that there are no results for the user's mood. The system also opens a web browser to play the first video from the recommended playlist.

The implementation of this system has practical applications in the entertainment industry, including creating personalized playlists based on the user's mood. The project's success highlights the potential for further research in mood detection systems, which could have a significant impact on the music industry, psychology, and other areas that require real-time detection of emotions.

In summary, this project has achieved its objective of developing an interactive system that detects a user's mood through facial expression and speech recognition and recommends a playlist of songs on YouTube based on the user's mood. The implementation of this system demonstrates the potential of mood detection systems and opens up the possibility of further research in this area. The system is flexible, easy to use, and provides personalized results, making it ideal for users looking for a personalized music experience based on their mood.