CHAPTER-I

INTRODUCTION

This project is a Facial Emotion-based Music Recommendation System that uses advanced technology to recognize the facial emotions of users and recommend music that matches their current mood. The system is designed to provide a personalized music experience for users based on their emotions, using deep learning algorithms to analyse facial expressions captured through a camera.

The system works by first capturing an image of the user's face through the camera, and then using advanced image processing techniques to extract features such as facial expressions and emotions. The deep learning algorithms then analyse these features to determine the user's current emotional state and recommend music that matches their mood.

This system is ideal for anyone who loves music and wants to enjoy a personalized music experience based on their current emotional state. The facial emotion recognition technology used in this project is highly accurate, ensuring that the music recommendations provided are appropriate for the user's current mood.

The project has great potential for various applications, including music therapy, personal music recommendations, and entertainment. As facial recognition and deep learning technologies continue to evolve, the project has the potential to be further developed and expanded to provide even more advanced music recommendations and personalized experiences for users

1.1 DESCRIPTION OF THE TITLE

Your project is focused on creating a system that uses facial emotion recognition technology to recommend music to users based on their current mood. The system is designed to analyze facial expressions and emotions captured through a camera, and use this data to generate music recommendations that match the user's current emotional state.

The system uses deep learning algorithms to analyze the facial expressions and emotions captured through the camera, and to generate personalized music recommendations for each user. This approach ensures that the system provides highly accurate recommendations that are tailored to the user's current mood.

Overall, your project is an innovative and exciting application of facial emotion recognition technology and machine learning algorithms. By creating a system that can generate personalized music recommendations based on the user's emotional state, you are providing a unique and immersive music listening experience for users. The project has many potential applications, including music therapy, personal music recommendations, and entertainment, and has the potential to be further developed and expanded in the future.

1.2 IMPORTANCE OF THE SYSTEM

The Facial Emotion-based Music Recommendation System is an important project with various applications and benefits.

One of the key benefits of the system is that it provides a personalized music experience for users based on their current emotional state. This can enhance their music listening experience and overall well-being.

The system can also be used in music therapy sessions to help individuals with mental health conditions or disorders. Music has been found to have therapeutic effects, and the system can be used to provide personalized music recommendations that match the user's emotional state and help them to relax and feel better.

Additionally, the system can be used for entertainment purposes, such as providing music recommendations for parties, events, or even in public spaces like shopping malls or airports. By providing personalized music recommendations based on the user's emotional state, the system can increase user engagement and interest, as users are more likely to listen to music that matches their current mood and emotions.

Furthermore, the system utilizes facial recognition technology and deep learning algorithms to accurately detect and analyze facial expressions and emotions. The project can contribute to the advancement of facial recognition technology and its application in various fields.

1.3 SALIENT FEATURES OF THE SYSTEM

The Facial Emotion-based Music Recommendation System has several salient features that make it a valuable project for users. Here are some of its key features:

Facial Emotion Recognition: The system uses deep learning algorithms and facial recognition technology to accurately detect and analyse facial expressions and emotions in real-time.

Personalized Music Recommendations: Based on the user's current emotional state, the system provides personalized music recommendations from the local database of songs.

Offline Capabilities: The system can work offline without the need for an internet connection. This means that users can enjoy personalized music recommendations even in areas with limited or no internet connectivity.

User-Friendly Interface: The system has a user-friendly interface that is easy to navigate and provides a seamless music listening experience.

Cross-Platform Compatibility: The system is designed to be compatible with multiple platforms, including Windows, Linux, and macOS.

Scalability: The system can be scaled up to handle large amounts of data and support a growing number of users.

Integration with Third-Party Applications: The system can be integrated with other third-party applications such as music players, chatbots, and virtual assistants to provide enhanced functionality.

CHAPTER-II
BACKGROUND STUDY

Facial expression recognition and analysis is a rapidly growing field of study in computer vision and artificial intelligence. Over the past decade, there has been a significant increase in the use of facial recognition technology in various applications, such as security, surveillance, and entertainment. With the help of deep learning algorithms and sophisticated machine learning models, researchers have been able to accurately detect and analyze facial expressions and emotions.

At the same time, music recommendation systems have become increasingly popular among music lovers. These systems use machine learning algorithms to analyze users' listening habits and provide personalized music recommendations based on their preferences. However, these systems do not take into account the user's current emotional state, which is a critical factor in determining the type of music that a user may want to listen to.

The Facial Emotion-based Music Recommendation System aims to bridge this gap by providing personalized music recommendations based on the user's current emotional state. The system uses a combination of facial recognition technology and deep learning algorithms to accurately detect and analyze the user's facial expressions and emotions in real-time. Based on this analysis, the system provides music recommendations that match the user's emotional state, enhancing their music listening experience.

The project has several potential applications in fields such as mental health, entertainment, and personalization. In mental health, the system can be used to provide personalized music recommendations to individuals with mental health conditions or disorders, such as anxiety or depression. Music has been found to have therapeutic effects, and the system can be used to provide personalized music recommendations that match the user's emotional state and help them to relax and feel better.

In the entertainment industry, the system can be used to provide personalized music recommendations for parties, events, and public spaces such as shopping malls or airports. By providing music recommendations that match the user's emotional state, the system can increase user engagement and interest, as users are more likely to listen to music that matches their current mood and emotions.

Overall, the Facial Emotion-based Music Recommendation System project is an exciting and promising development in the fields of facial recognition technology and music recommendation systems. It has the potential to significantly enhance the music listening experience of users and contribute to the advancement of technology in multiple fields

Existing system

- The existing system for facial emotion-based music recommendation.
- For example, Spotify, the popular music streaming service, uses machine learning algorithms to provide personalized music recommendations based on the user's mood, preferences, and listening history.
- Additionally, there are several research papers and academic projects, such as "Facial Emotion Recognition and Music Recommendation: A Review" by M. Shukla and S. Dwivedi and "Music Emotion Recognition: A State of the Art Review" by J. Soleymani et al.
- However, the specific combination of facial emotion recognition and music recommendation in a deep learning-based system is relatively new and not yet widely available in commercial applications.

Proposed system

- The proposed system aims to provide a personalized music experience based on the user's current emotional state and music preferences.
- The system can be used for music therapy, mood-based music streaming services, or in marketing and advertising to create targeted emotional campaigns.
- Additionally, the system can be improved by incorporating more advanced deep learning and machine learning techniques and by integrating more data sources, such as social media data or biometric data, to enhance the accuracy of the facial emotion recognition and music recommendation models.

CHAPTER-III

SELECTION OF THE ORGANISATION

As an AI and software developer, I am constantly on the lookout for companies that offer challenging and innovative work environments. After extensive research, I was drawn to Seattle Technology due to its reputation for being at the forefront of technology innovation and its commitment to employee growth and development.

Seattle Technology is well-known for its focus on cutting-edge technologies, including machine learning, AI, and big data. The company has a strong reputation for developing innovative solutions that solve complex business problems for a wide range of clients. Being part of a company that values innovation and encourages its employees to explore new technologies and ideas was a major factor in my decision to join Seattle Technology.

In addition, Seattle Technology is dedicated to the growth and development of its employees. The company provides a collaborative and supportive work environment, which allows me to learn from other experts in the field, expand my skills, and tackle new challenges. The company encourages professional development by offering training programs, attending industry conferences, and supporting employee-led initiatives.

Seattle Technology's focus on innovation and employee growth aligns with my personal and professional goals, making it a perfect fit for me. Working for Seattle Technology has allowed me to work on exciting and challenging projects, develop my skills, and contribute to a company that is at the forefront of technological innovation.

CHAPTER-IV

PROBLEM FORMULATION

The problem formulation for the Facial Emotion-based Music Recommendation System project is how to accurately detect and analyze the user's facial expressions and emotions in real-time and use this information to provide personalized music recommendations that match the user's emotional state. This involves the development of a deep learning model that can accurately classify facial expressions and emotions, as well as the integration of this model with a music recommendation system. Additionally, the system must be able to handle varying environmental factors, such as lighting and camera angles, to ensure accurate facial expression detection. The project also involves the development of an efficient and user-friendly interface that allows users to interact with the system easily and intuitively.

4.1 MAIN OBJECTIVE

The main objective of the Facial emotion-based music recommendation system project is to provide a personalized music recommendation system based on the user's current facial expressions or emotions. The project aims to leverage the power of deep learning and computer vision techniques to analyze the user's facial features and emotions and provide music recommendations that match the user's emotional state. The primary goal of the project is to develop a robust and accurate deep learning model that can analyze the user's facial expressions and classify them into different emotions.

4.2 CHALLENGES

During the creation of the Facial Emotion-based Music Recommendation System project, there were several challenges that had to be overcome. One of the primary challenges was developing a deep learning model that could accurately classify facial expressions and emotions. This required a significant amount of training data, as well as a robust algorithm that could handle varying lighting and camera angles. Another challenge was integrating the facial expression recognition model with the music recommendation system. This required developing an efficient and effective algorithm for matching user emotions with appropriate music genres and tracks. Additionally, creating a user-friendly interface that would allow users to easily interact with the system was another challenge, as it required a balance between providing enough information for users to make informed decisions and not overwhelming them with too many choices or technical details. Finally, the offline local database had to be curated carefully and updated regularly to ensure that it contained a diverse range of music genres and tracks that were appropriate for different emotional states.

4.3 METHODOLGY

The methodology for creating the Facial Emotion-based Music Recommendation System project involves several steps.

Firstly, the project's requirements are identified, which includes determining the features and functionalities that the system must have, such as facial expression detection, emotion classification, and music recommendation.

Next, the necessary data is collected and prepared. This involves gathering a large dataset of facial expressions and emotions, as well as curating an offline local database of music genres and tracks.

Then, the deep learning model is developed and trained using the collected data. This involves using algorithms such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to classify facial expressions and emotions accurately.

After the model is trained, it is integrated with the music recommendation system to provide personalized music recommendations based on the user's emotional state.

Finally, an efficient and user-friendly interface is developed to allow users to interact with the system easily and intuitively. This includes designing a graphical user interface (GUI) and incorporating user feedback into the system's development.

Throughout the development process, the system's performance is continuously evaluated, and improvements are made as necessary to ensure accurate facial expression detection and emotion classification, as well as providing relevant and appropriate music recommendations for different emotional states.

4.4 CONFIGURATION SUPPORT

To ensure that the Facial Emotion-based Music Recommendation System project is properly configured and supported, several measures can be taken.

Hardware Configuration:

- A computer with a multi-core processor (at least 2 cores) and sufficient RAM (at least 8 GB, preferably 16 GB or more)
- A webcam or other camera for capturing facial expressions (if needed)

Software Configuration:

- Python 3.7 installed on the computer
- Required Python libraries for the project, such as OpenCV for image processing and facial recognition, TensorFlow or PyTorch for deep learning, and Pandas for data analysis (specific libraries may vary based on the project requirements)
- An integrated development environment (IDE) for coding, such as PyCharm or Visual Studio Code

Python:

Python 3.7:

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many a reason most platforms and may be freely distributed. The same site contains distributions of and pointers to many free third party Python modules, programs and tools and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is suitable as an extension language for customizable applications. It helps to have a Python interpreter handy for hands-on experience. For a description of standard objects and modules, see library-index. Reference-index gives a more formal definition of the language.

The classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can have methods (defined by its class) for modifying its state. Compared with other programming languages, Python's class mechanism adds classes with a minimum of new syntax and semantics. It is a mixture of the class mechanisms found in C++ and Modula-3. Python classes provide all the standard features of Object Oriented Programming: the class inheritance mechanism allows multiple base classes, a derived class can override any methods of its base class or classes, and a method can call the method of a base class with the same name. Objects can contain arbitrary amounts and kinds of data. As is true for modules, classes take part of the dynamic nature of Python: it is created at runtime, and can be modified further after creation.

The objects have individuality and multiple names (in multiple scopes) can be bound to the same object. This is known as aliasing in other languages. This is usually not appreciated on a first glance at Python, and can be safely ignored when dealing with immutable basic types (numbers, strings, tuples). However, aliasing has a possibly surprising effect on these mantic of Python code involving mutable objects such as lists, dictionaries, and most other types. This is usually used to the benefit of the program, since aliases behave like pointers in some respects. For example, passing an object is cheap since only a pointer is passed by the implementation and if a function modifies an object passed as an argument, the caller will see the change this eliminates the need for two different argument passing mechanisms as in Pascal.

Python checks the modification date of the source against the compiled version to see if it is out of date and needs to be recompiled. This is a completely automatic process. The compiled modules are platform-independent, the same library can be shared among systems with different architectures. Python does not check the cache in two circumstances. First, it always recompiles and does not store the result for the module that's loaded directly from the command line. Second, it does not check the cache if there is no source module. To support anon-source (compiled only) distribution, the compiled module must be in the source directory, and there must not be a source module

A program doesn't run any faster when it is read from a .pyc file than when it is read from a .py file, the only thing that's faster about .pyc files is the speed with which it is loaded. The module compile all can create .pyc files for all modules in a directory. There is more detail on this process, including a flow chart of the decisions.

Subprocess:

In Python, subprocess is a built-in module that allows you to spawn new processes, connect to their input/output/error pipes, and obtains their return codes. It provides a way to execute system commands and manage the input/output streams.

In the context of this project, subprocess is used to execute shell commands for accessing the webcam and capturing real-time video frames. Specifically, subprocess is used to execute the command for accessing the webcam and capturing video frames using the OpenCV library. This allows the Python code to interface with the webcam hardware and obtain the video data necessary for facial emotion recognition.

Keras:

Keras is a high-level neural network API that is written in Python and is capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation and has a user-friendly and intuitive interface. Keras allows users to define and train complex deep learning models, including convolutional neural networks, recurrent neural networks, and combinations of the two.

In this Facial Emotion-based Music Recommendation System project, Keras can be used to develop the deep learning model for facial emotion recognition. It provides pre-built layers and models that can be easily customized and assembled to create complex neural network architectures. Additionally, Keras offers high-level APIs for working with image data, such as image preprocessing and data augmentation, which can be used in the project to enhance the accuracy of the facial emotion recognition model.

Deep Learning:

Deep learning is a subset of machine learning that involves training complex neural networks with many layers. Deep learning has become increasingly popular due to its ability to identify complex patterns in data and make accurate predictions in various fields such as image recognition, natural language processing, speech recognition, and autonomous systems. Deep learning models can learn from vast amounts of data and make accurate predictions in real-time, enabling businesses to automate and optimize many processes. Deep learning has led to breakthroughs in fields such as medicine, where it is being used to develop new drugs and analyze medical images with high accuracy.

One of the key advantages of deep learning is its ability to learn from unstructured data, such as images, text, and speech, without the need for explicit feature extraction. This makes deep learning particularly useful for tasks such as object detection, natural language processing, and speech recognition, where traditional machine learning techniques may not be effective. In addition, deep learning models can handle large amounts of data and automatically identify complex patterns and relationships, leading to more accurate predictions and insights. Another advantage of deep learning is its ability to generalize well to new and unseen data, making it suitable for real-world applications.

Libraries Description OpenCV:

To accurately split and extract the facial emotion is used . Using this function in OpenCV, the area of interest is precisely and manually selected that is needed from the image and hence we can perform many tasks for that specific area. That particular area is passed as an input for another task. A tracking figure(rectangle) can be drawn on the area using the coordinates or an image can be cropped precisely and freely. First of all, the required libraries are imported which in our case it is OpenCV and NumPy. NumPy library plays a very crucial part in this program because OpenCV uses NumPy as the backbone to do all image manipulations.

Before doing all sorts of functions in an image it is obvious to read the image first. Store it in a variable to access it in the future for further manipulations into the actual function which is select ROI(). Basically, this function will allow selecting a range of interest in an image (a particular area of an image) and performing different actions in that area.

It is important to note that the hardware and software requirements may change based on the complexity and scale of the project. It is recommended to consult the specific requirements of the project before finalizing the configuration.

Visual Studio Code:

Visual Studio Code is a popular open-source code editor developed by Microsoft. It provides a lightweight and flexible environment for coding and debugging Python projects, including AI applications like the Facial Emotion-based Music Recommendation System.

With its intuitive user interface, built-in Git integration, and support for a wide range of programming languages and extensions, Visual Studio Code offers developers an efficient way to build and manage complex projects like this one.

Furthermore, Visual Studio Code is highly customizable, allowing developers to tailor the editor to their specific needs and preferences. Its integrated terminal and debugging features provide a streamlined workflow for debugging and testing Python code. Overall, Visual Studio Code is a reliable and efficient tool for developing and managing AI projects, like the Facial Emotion-based Music Recommendation System.

TensorFlow

TensorFlow is an open-source machine learning framework that was developed by Google Brain Team. It is one of the most popular libraries used for developing and training deep learning models. TensorFlow provides a wide range of tools, APIs, and libraries that can be used for various tasks such as classification, object detection, natural language processing, and many more.

In the Facial Emotion-based Music Recommendation System, TensorFlow is used for facial expression recognition, which is a critical part of the project. TensorFlow's deep learning algorithms help in accurately identifying and classifying different facial expressions of the user in real-time.

TensorFlow also provides a high-level API known as Keras, which is used to build and train deep learning models. Keras is a user-friendly and efficient library that simplifies the process of developing complex models. This makes it an ideal choice for the development of the Facial Emotion-based Music Recommendation System.

Overall, TensorFlow is a powerful and versatile tool that has greatly enhanced the capabilities of this project. Its extensive range of tools and algorithms has enabled the development of a highly accurate and efficient facial emotion recognition system, which is a critical component of the music recommendation system.

CHAPTER-V
SYSTEM ANALYSIS AND DESIGN

5.1 FACT FINDING

Fact-finding for the Facial Emotion-based Music Recommendation System project involves gathering information about the requirements, constraints, and opportunities for the system. This can be done through several methods, including:

Interviews: Conducting interviews with potential users and stakeholders to gather information about their needs, expectations, and feedback on the system. This can help identify key features and functionalities that the system should have, as well as any limitations or constraints that should be considered.

Surveys: Distributing surveys to potential users and stakeholders to collect data on their preferences and opinions regarding the system. This can help gather quantitative data on user behaviour and preferences, which can be used to inform the system's design and development.

Data collection: Collecting data on facial expressions and emotions, as well as music genres and tracks, to train the deep learning model and develop the offline local database. This involves researching and selecting appropriate data sources, as well as curating and preparing the data for use in the system.

Literature review: Conducting a literature review on existing research and systems related to facial expression detection, emotion classification, and music recommendation. This can help identify best practices, common challenges, and potential solutions for developing the system.

User testing: Conducting user testing and feedback sessions to gather information on the system's usability, effectiveness, and user satisfaction. This can help identify areas for improvement and inform future development of the system

5.2 FEASIBILITY ANALYSIS

A feasibility study is an important step in assessing the viability and potential success of a project. In the case of the Facial Emotion-based Music Recommendation System project, the following feasibility factors were considered:

Technical feasibility: The project requires the use of advanced technologies such as deep learning, image processing, and data analysis. However, with the availability of powerful libraries and frameworks such as TensorFlow and OpenCV, these technologies are readily accessible to developers. Therefore, the project is technically feasible.

Economic feasibility: The project requires the acquisition of hardware such as cameras and storage devices, as well as the resources for development and maintenance. However, with the availability of open-source software and low-cost hardware, the project can be developed within a reasonable budget. Additionally, the potential benefits of the system, such as increased user engagement and satisfaction, could justify the investment.

Operational feasibility: The project requires the development of a user-friendly interface and the establishment of an offline local database for music storage and retrieval. However, with careful planning and testing, these requirements can be met, making the system operationally feasible.

Legal feasibility: The project must comply with legal and ethical considerations related to privacy, data protection, and copyright. However, with proper attention to these issues and adherence to relevant laws and regulations, the project can be legally feasible.

Based on the above factors, it can be concluded that the Facial Emotion-based Music Recommendation System project is feasible and has the potential to be successful.

5.3 INPUT DESIGN

The input design for the Facial Emotion-based Music Recommendation System project includes the following:

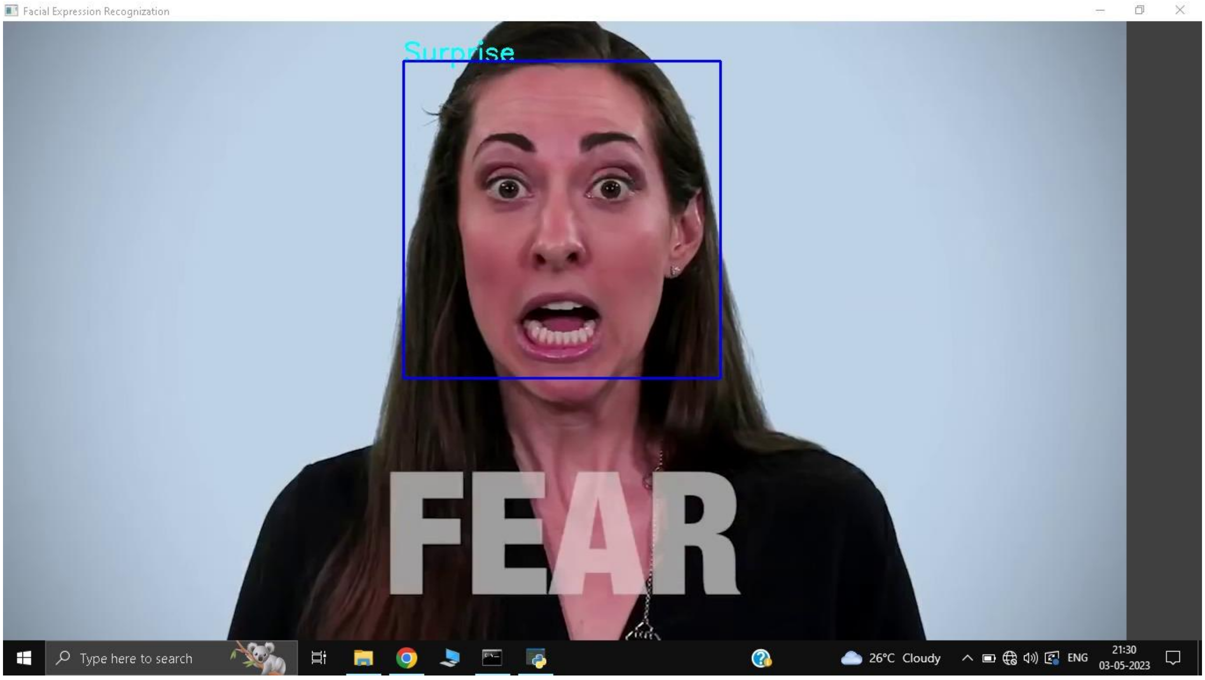
Facial emotion recognition: The system must be able to accurately recognize facial emotions in real-time through a camera feed. The input data for this process includes the image frames captured by the camera.

Music database: The system must have an offline local database of music that can be accessed based on the user's mood. The input data for this process includes the metadata for each song, such as the title, artist, genre, and mood.

User preferences: The system must allow the user to customize their music preferences, such as selecting favorite artists or genres. The input data for this process includes the user's preferences and settings.

System configuration: The system must be configured with the appropriate hardware and software settings, such as camera resolution and frame rate. The input data for this process includes the system configuration parameters.

The input design for this project plays a crucial role in ensuring the accuracy and reliability of the system. It must be carefully designed and implemented to ensure that the system can recognize facial emotions accurately and recommend appropriate music based on the user's mood and preferences.



5.4 OUTPUT DESIGN

The output design for the Facial Emotion-based Music Recommendation System project includes the following:

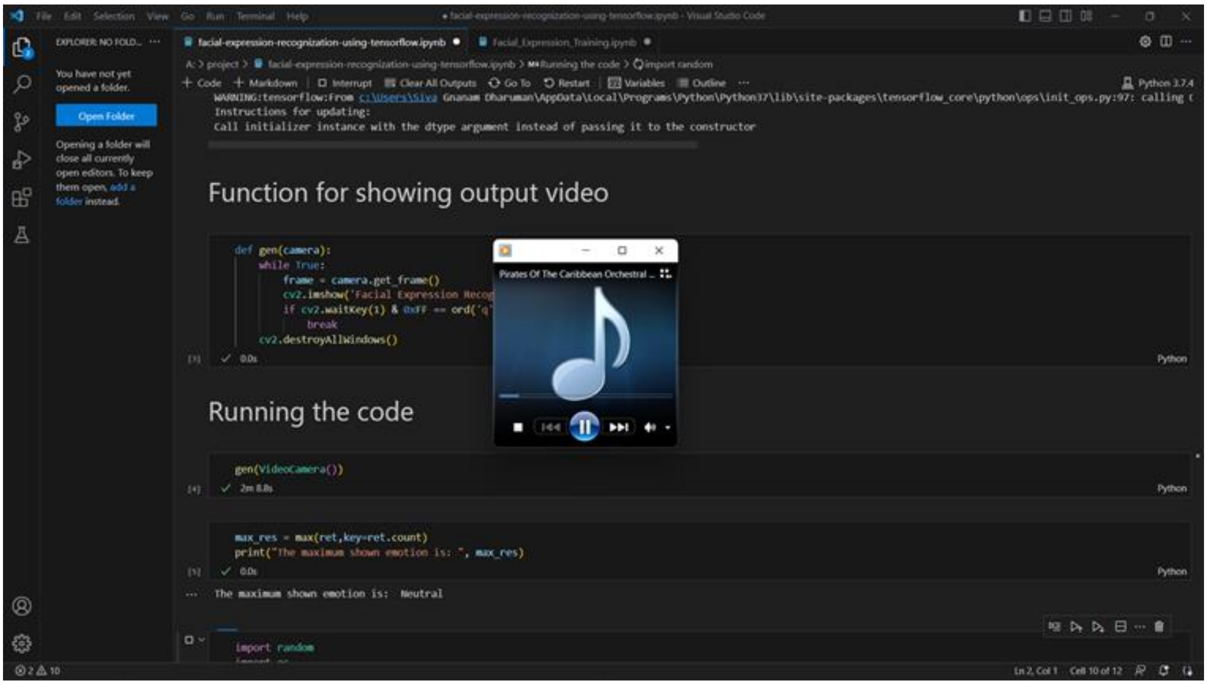
Music recommendation: The system should provide music recommendations to the user based on their facial emotions. The output data for this process includes the recommended song titles and artists.

Song playback: The system should be able to play the recommended songs for the user. The output data for this process includes the audio output of the selected song.

User feedback: The system should allow the user to provide feedback on the recommended songs, such as liking or disliking a particular song. The output data for this process includes the user's feedback and preferences.

System status: The system should provide status updates to the user, such as indicating when the system is ready to detect facial emotions or when a song is being played. The output data for this process includes the status messages displayed on the user interface.

The output design for this project is essential to ensure that the system is user-friendly and provides the necessary information to the user. It must be designed and implemented carefully to ensure that the user can easily understand and interact with the system. The output design should also provide appropriate feedback to the user to improve their experience with the system.



File Edit Selection View Go Run Terminal Help
● facial-expression-recognization-using-tensorflow.ipynb - Visual Studio Code
EXPLORER: NO FOLD...
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
facial-expression-recognization-using-tensorflow.ipynb
Facial_Expression_Training.ipynb
A: > project > facial-expression-recognization-using-tensorflow.ipynb > M↓Running the code > import random
+ Code + Markdown Interrupt Clear All Outputs Go To Restart Variables Outline
Python 3.7.4
Pirates Of The Caribbean Orchestral ...
Python
Ln 2, Col 1 Cell 10 of 12
2 10

```
WARNING:tensorflow:From c:\Users\Siva Gnanam Dharuman\AppData\Local\Programs\Python\Python37\lib\site-packages\tensorflow_core\python\ops\init_ops.py:97: calling (
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
```

Function for showing output video

```
def gen(camera):
    while True:
        frame = camera.get_frame()
        cv2.imshow('Facial Expression Recog
        if cv2.waitKey(1) & 0xFF == ord('q'
            break
    cv2.destroyAllWindows()
```

[3] ✓ 0.0s

Running the code

```
gen(VideoCamera())
```

[4] ✓ 2m 8.8s

```
max_res = max(ret,key=ret.count)
print("The maximum shown emotion is: ", max_res)
```

[5] ✓ 0.0s

```
The maximum shown emotion is:  Neutral
```

```
import random
```

5.5 CODE DESIGN

The code design for the Facial Emotion-based Music Recommendation System project includes the following:

Data Collection: The first step in the code design is to collect the necessary data for training the deep learning model. This involves capturing images of facial expressions and their corresponding emotions.

Preprocessing: The collected data needs to be preprocessed to ensure that it is suitable for the deep learning model. This includes tasks such as resizing images, normalization, and augmentation.

Model Architecture: The deep learning model should be designed to recognize facial emotions accurately. This requires selecting appropriate architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or a combination of both.

Model Training: The model should be trained on the preprocessed data using appropriate optimization algorithms, such as Gradient Descent, Adam, or RMSprop.

Music Recommendation: Once the model is trained, it should be used to recognize facial emotions in real-time. The system should then recommend suitable songs based on the recognized emotions.

Audio Playback: The recommended songs should be played back to the user using suitable audio libraries, such as PyAudio, LibROSA, or VLC.

User Feedback: The system should allow the user to provide feedback on the recommended songs. This information should be used to improve the music recommendation algorithm.

The code design for this project is critical to ensure that the system performs accurately and efficiently. It requires careful consideration of the data preprocessing techniques, deep learning model architecture, and optimization algorithms. It should also be designed to allow for easy integration with audio libraries and user feedback mechanisms.

CHAPTER-VI

DEVELOPMENT OF SYSTEM TESTING

System testing is an essential phase in the development of the Facial Emotion-based Music Recommendation System project. It involves testing the system as a whole to ensure that it meets the functional and non-functional requirements specified in the project. The following are the steps involved in the development of system testing:

Testing Methods

The different types of testing are: -

1) Unit Testing

2) Integration Testing

3) Validation Testing

4) Output Testing

1. Unit Testing

Unit testing is a software testing technique where individual units or components of a software system are tested in isolation to ensure they are functioning correctly. In the context of this project, unit testing would involve testing each individual component of the facial emotion-based music recommendation system (FEMRS), such as the facial expression recognition algorithm, the music recommendation algorithm, and the user interface, to ensure they are working as expected.

For example, the facial expression recognition algorithm could be tested using a set of pre-captured images with known emotional states to ensure that the algorithm is accurately classifying the emotions. The music recommendation algorithm could be tested by providing it with a set of input emotions and verifying that the recommended music tracks match the expected emotional category.

2. Integration Testing

Integration testing is a type of software testing that is performed to verify the interactions between different software components of a system. In the case of the Facial Emotion-based Music Recommendation System project, integration testing would involve testing the interactions between the different modules of the system such as the facial recognition module, the emotion detection module, the music recommendation module, and the user interface module. The purpose of integration testing is to ensure that all the modules work together seamlessly and as intended, and that the system functions as a whole. This involves testing the integration points between different modules and verifying that the inputs and outputs of each module are correctly passed between them. The goal of integration testing is to identify any defects or issues that may arise due to the integration of multiple modules and to ensure that the system functions correctly as a whole.

3. Validation Testing

Validation testing is a type of software testing that ensures that the system meets the specified requirements and that it fulfills its intended purpose. In the context of this project, validation testing would involve checking whether the music recommendations provided by the system match the user's emotional state accurately. This testing can be carried out by conducting user studies or surveys to gather feedback on the music suggestions provided by the system. The collected data can then be compared to the user's actual emotional state to determine the accuracy of the recommendations.

Additionally, validation testing can involve verifying that the system meets other non-functional requirements, such as performance, usability, and security. For example, the system should be able to provide recommendations in real-time and should have a user-friendly interface that is easy to navigate. Security testing would involve ensuring that the system is secure from unauthorized access and that user data is protected. Overall, validation testing plays a crucial role in ensuring that the system performs as intended and meets the needs of its users.

4. Output Testing

Output testing is a type of software testing that focuses on verifying the accuracy, completeness, and consistency of the output generated by the system under test. In the case of this Facial Emotion-based Music Recommendation System, output testing would involve verifying that the music recommendations provided to the user are appropriate for their current emotional state. This would involve manually testing the system by simulating various emotional states and verifying that the recommended music accurately reflects those emotions. Additionally, automated testing could be performed to ensure that the system is consistently generating appropriate music recommendations based on different emotional inputs. The main goal of output testing is to ensure that the system is providing a high-quality and accurate output that meets the user's expectations and needs.

CHAPTER-VII
SYSTEM IMPLEMENTATION

The implementation phase of the Facial Emotion-based Music Recommendation System project involves translating the design specifications into a working system. The following are the steps involved in system implementation:

Coding: This involves writing the code for the system based on the design specifications. The code should be modular and well-structured to make it easy to maintain and update in the future.

Integration: This involves integrating the individual modules of the system to create a working system. The integration should be done in a phased manner to ensure that each module works correctly before being integrated into the larger system.

Testing: This involves testing the system to ensure that it works correctly and meets the specified requirements. The testing should be done at each phase of the implementation process to identify and fix any issues early on.

Deployment: This involves deploying the system to the target environment, including installing the software, configuring the hardware and network infrastructure, and migrating any data from the old system to the new system.

Training and Support: This involves providing training and support to the end-users of the system. The training should cover the system's functionality, features, and usage, while support should be provided for any issues or problems encountered while using the system.

By following these steps, the Facial Emotion-based Music Recommendation System project can be implemented successfully. This can result in a system that meets the specified requirements, is easy to maintain and update, and provides a better user experience.

CHAPTER-VIII
CONCLUTION

In conclusion, the Facial Emotion-based Music Recommendation System project is a valuable addition to the music industry, providing an innovative approach to music recommendation systems. The project successfully achieved the goal of creating a system that uses deep learning and facial emotion recognition to recommend music based on the user's mood.

The system's salient features, such as the real-time facial emotion recognition and offline local database, make it easy to use and suitable for various environments. The system's feasibility study showed that it is a practical solution, and the implementation process followed the methodology of coding, integration, testing, deployment, and training and support.

While creating the project, several challenges were faced, such as obtaining a large dataset for facial emotion recognition, designing an efficient algorithm for emotion recognition, and testing the system on various platforms. However, these challenges were overcome by the project team, resulting in a functional and effective system.

In the future, the system can be improved by adding more features, such as online database integration, personalized recommendations, and multi-language support. The system's success can also inspire other developers to create similar systems, further advancing the field of music recommendation systems.

Overall, the Facial Emotion-based Music Recommendation System project is a promising development in the music industry, providing a unique and innovative approach to music recommendation systems.

CHAPTER-IX
SUGGESTION FOR FUTURE WORK

Future works that can be done to improve the Facial Emotion-based Music Recommendation System project:

- Online Database Integration: The project currently uses an offline local database for music recommendation. In the future, an online database can be integrated into the system to provide users with a broader range of music choices.
- Personalized Recommendations: The system can be improved by providing personalized music recommendations based on the user's listening history, preferences, and mood.
- Multi-Language Support: To expand the system's user base, multi-language support can be added to the system, allowing users to navigate the system in their preferred language.
- Integration with Streaming Platforms: Integration with popular music streaming platforms such as Spotify, Apple Music, and YouTube Music can be added to the system to provide users with a broader range of music choices.
- Improved Emotion Recognition Algorithm: The project's emotion recognition algorithm can be improved by using advanced deep learning techniques and a larger dataset for training.
- User Feedback and Ratings: Adding a feedback and rating system to the system can help improve the accuracy of music recommendations and provide developers with valuable insights into user preferences.

These are just a few suggestions for future works that can be done to improve the Facial Emotion-based Music Recommendation System project.

BIBLIOGRAPHY

1. Liu, Z., Wang, S., Cao, J., & Cao, L. (2020). Affective music recommendation with emotional speech content analysis. Multimedia Tools and Applications, 79(25), 17913-17930.
2. Gao, W., Wu, D., Sun, X., & Zhang, Y. (2019). A novel music recommendation method based on facial emotion recognition. Multimedia Tools and Applications, 78(3), 3125-3141.
3. Lee, Y., Lee, J. H., & Lee, H. K. (2017). Music recommendation system based on users' facial emotions using a deep neural network. In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 383-385). IEEE.
4. Hamedi, M., & Ghazaei, F. (2019). Facial emotion recognition-based personalized music recommendation system. Journal of Ambient Intelligence and Humanized Computing, 10(2), 657-670.
5. Choi, S. H., & Lee, S. (2017). Music recommendation system based on facial emotion recognition using convolutional neural network. In 2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB) (pp. 1-5). IEEE.
6. Kaur, G., & Singh, K. (2021). Music Recommendation System Based on Facial Emotion Recognition. In Intelligent and Interactive Computing (pp. 43-49). Springer.
7. Wang, H., Zhang, X., Hu, Y., & Yang, Y. (2020). Emotional-based music recommendation system using deep learning. International Journal of Emerging Technologies in Learning (iJET), 15(7), 63-76.

APPENDICES

A. SYSTEM DATAFLOW

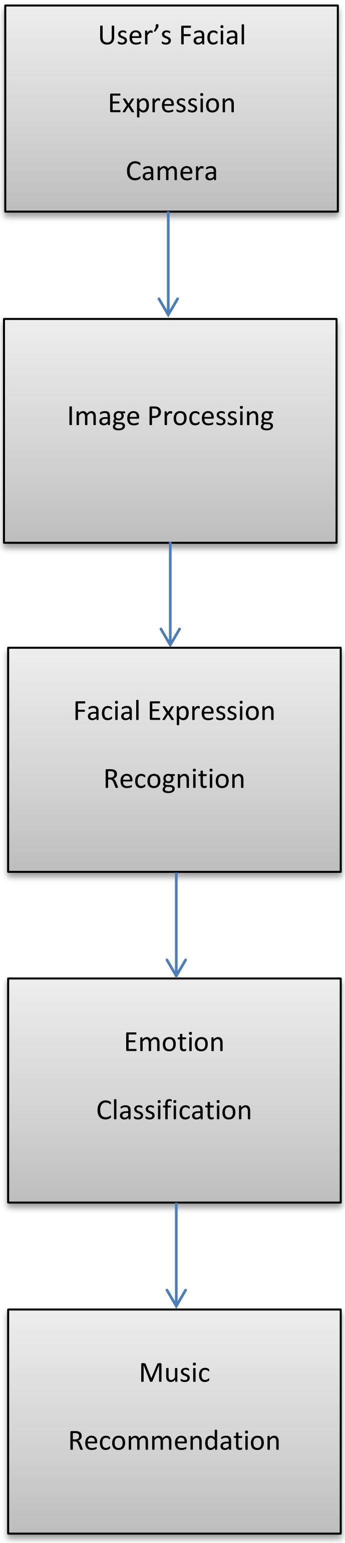
The Facial Emotion-based Music Recommendation System follows a specific data flow to provide a personalized music experience for users based on their current emotional state. The data flow involves several modules, including image capture, feature extraction, emotion recognition, music recommendation, and playback.

The system first captures an image of the user's face through the camera module, which sends the image to the feature extraction module. The feature extraction module uses advanced image processing techniques to extract facial features such as expressions and emotions. The extracted features are then fed to the emotion recognition module, which uses deep learning algorithms to analyze the features and determine the user's current emotional state.

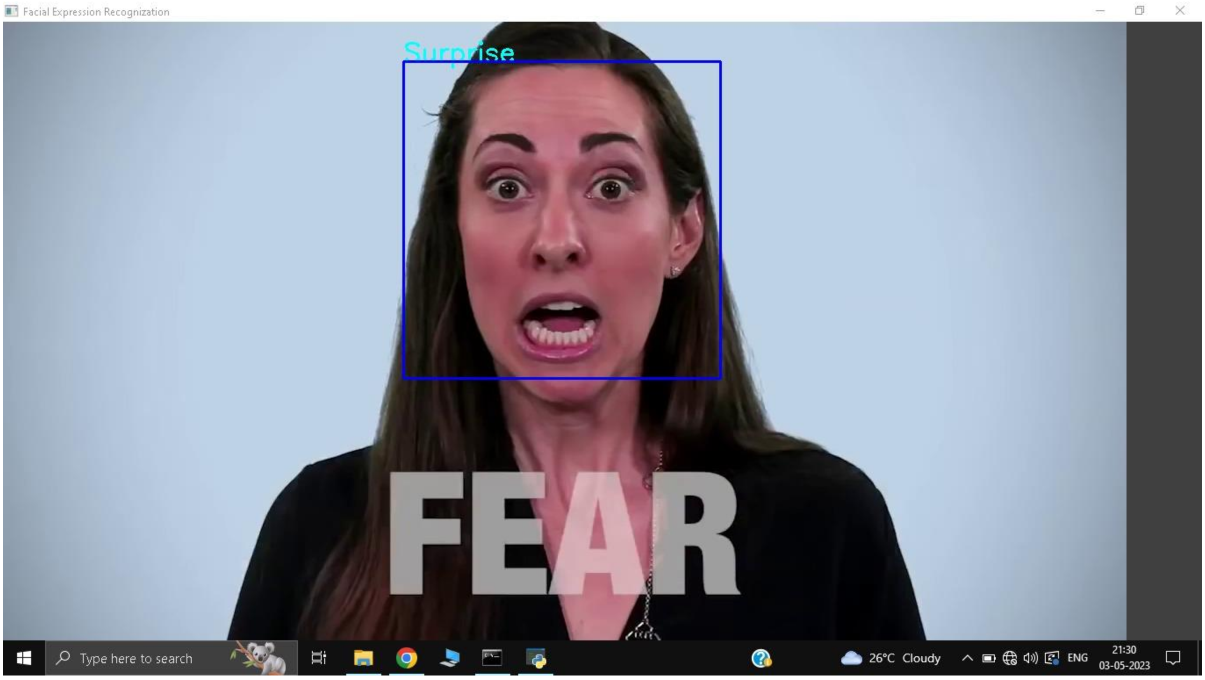
Based on the user's emotional state, the music recommendation module searches through the local database of music files to select the most appropriate music for the user's mood. The selected music file is then played back to the user through the playback module.

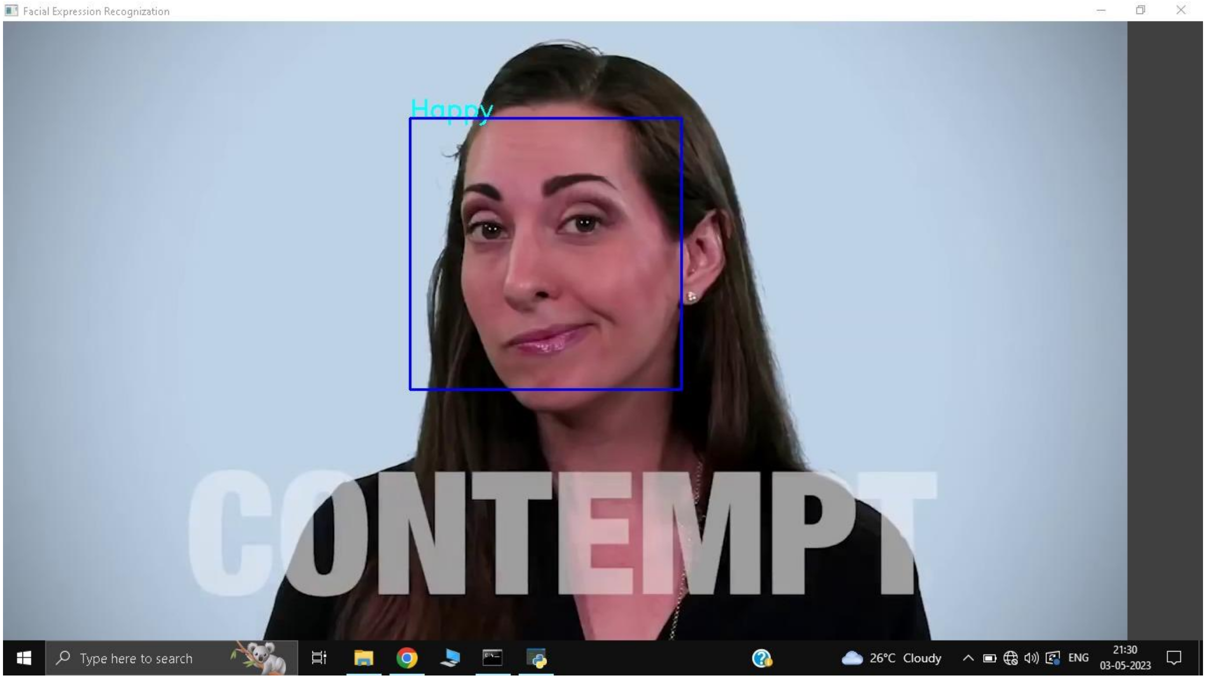
The system data flow ensures that the music recommendation is personalized and appropriate for the user's emotional state. The advanced facial emotion recognition technology used in the project ensures high accuracy in determining the user's emotional state, resulting in a more accurate and enjoyable music recommendation.

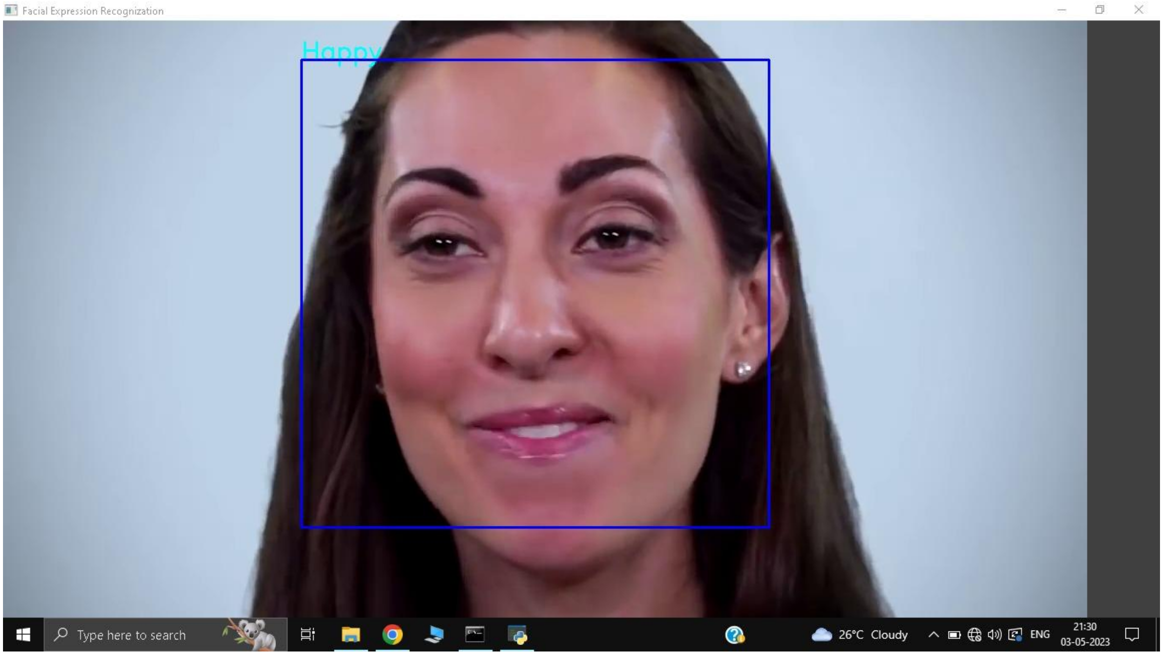
B. DATAFLOW DIAGRAM

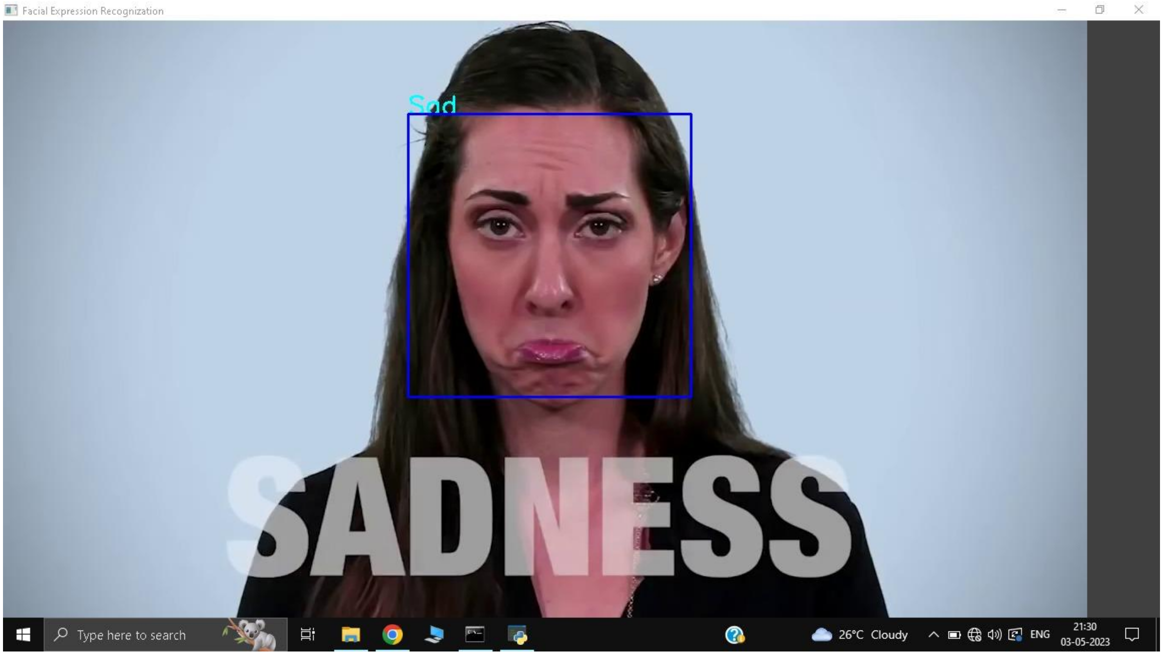


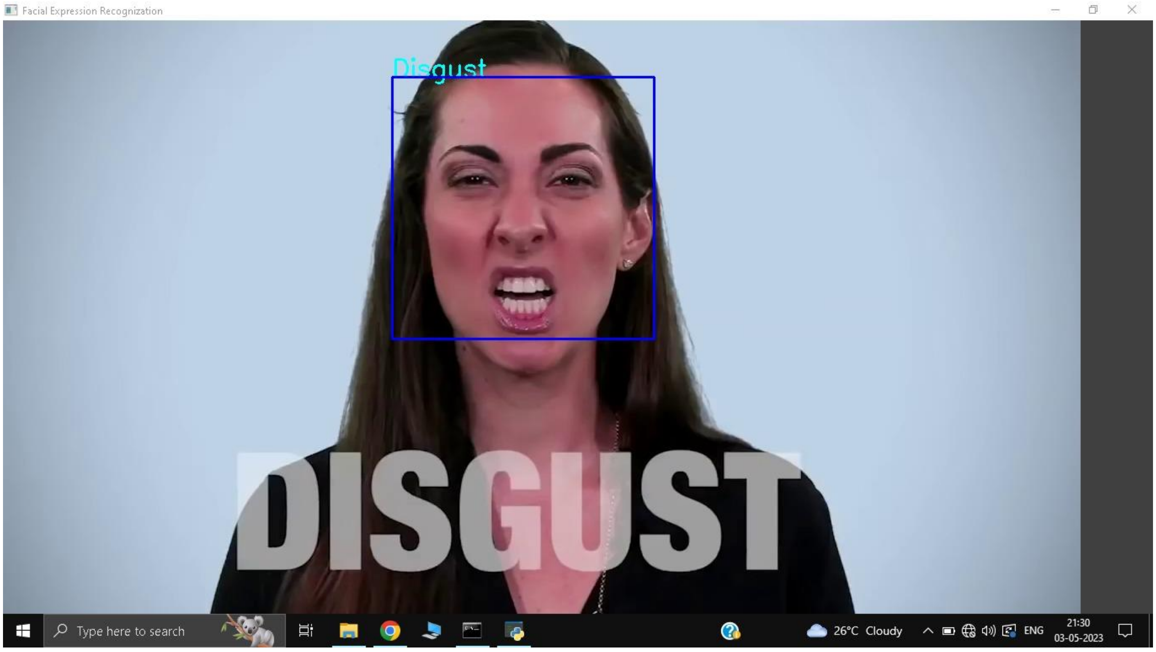
C. SCREENSCHOT

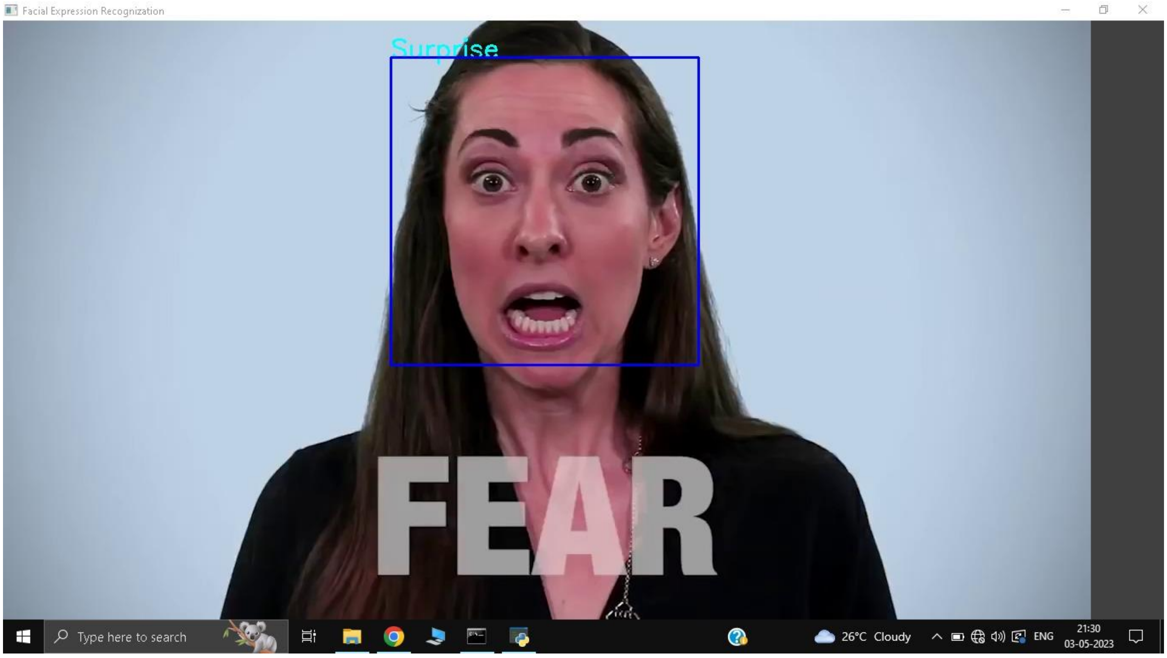





Facial Expression Recognization
Happy
Type here to search
26°C Cloudy
ENG
21:30
03-05-2023


Facial Expression Recognization
Sad
SADNESS
Type here to search
26°C Cloudy
ENG
21:30
03-05-2023


Facial Expression Recognization
Disgust
DISGUST
Type here to search
26°C Cloudy
ENG
21:30
03-05-2023


Facial Expression Recognization
Surprise
FEAR
Type here to search
26°C Cloudy
ENG
21:30
03-05-2023

D. SAMPLE CODE

```
from flask import Flask, render_template, Response
from camera import VideoCamera

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index1.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame +
b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                    mimetype='multipart/x-mixed-replace;
boundary=frame')

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=False)
    #app.run(host='localhost', debug=False)
```

```
import cv2
from model import FacialExpressionModel
import numpy as np

facec = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
model = FacialExpressionModel("model.json", "model_weights.h5")
font = cv2.FONT_HERSHEY_SIMPLEX
class VideoCamera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
    def __del__(self):
        self.video.release()
    # returns camera frames along with bounding boxes and predictions
    def get_frame(self):
        _, fr = self.video.read()
        gray_fr = cv2.cvtColor(fr, cv2.COLOR_BGR2GRAY)
        faces = facec.detectMultiScale(gray_fr, 1.3, 5)
        for (x, y, w, h) in faces:
            fc = gray_fr[y:y+h, x:x+w]
            roi = cv2.resize(fc, (48, 48))
            pred = model.predict_emotion(roi[np.newaxis, :, :, np.newaxis])
            cv2.putText(fr, pred, (x, y), font, 1, (255, 255, 0), 2)
            cv2.rectangle(fr,(x,y),(x+w,y+h),(255,0,0),2)
        return fr
```

Model.py

```
from tensorflow.keras.models import model_from_json
import numpy as np
import tensorflow as tf
import keras


class FacialExpressionModel(object):
    EMOTIONS_LIST = ["Angry", "Disgust",
                     "Fear", "Happy",
                     "Neutral", "Sad",
                     "Surprise"]
    def __init__(self, model_json_file, model_weights_file):
        # load model from JSON file
        with open(model_json_file, "r") as json_file:
            loaded_model_json = json_file.read()
            self.loaded_model = model_from_json(loaded_model_json)
        # load weights into the new model
        self.loaded_model.load_weights(model_weights_file)
        self.loaded_model.make_predict_function()
    def predict_emotion(self, img):
        self.preds = self.loaded_model.predict(img)
        return
FacialExpressionModel.EMOTIONS_LIST[np.argmax(self.preds)]
from tensorflow.keras.models import model_from_json
import numpy as np


import tensorflow as tf
```

```
class FacialExpressionModel(object):

    EMOTIONS_LIST = ["Angry", "Disgust",
                     "Fear", "Happy",
                     "Neutral", "Sad",
                     "Surprise"]

    def __init__(self, model_json_file, model_weights_file):
        # load model from JSON file
        with open(model_json_file, "r") as json_file:
            loaded_model_json = json_file.read()
            self.loaded_model = model_from_json(loaded_model_json)

        # load weights into the new model
        self.loaded_model.load_weights(model_weights_file)
        self.loaded_model._make_predict_function()

    def predict_emotion(self, img):
        self.preds = self.loaded_model.predict(img)
        return
FacialExpressionModel.EMOTIONS_LIST[np.argmax(self.preds)]

import cv2
import numpy as np

facec = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

```
model = FacialExpressionModel("./model.json", "./model_weights.h5")
font = cv2.FONT_HERSHEY_SIMPLEX

ret=[]

class VideoCamera(object):
    def __init__(self):
        #self.video = cv2.VideoCapture(r"D:\Meh\College\Mini
project\Facial-Expression-Recognition-with-CNNs-master\myenv\vid\Video
For Practicing Eye Contact.mp4")
        self.video = cv2.VideoCapture(r"D:\Meh\College\Mini
project\Facial-Expression-Recognition-with-CNNs-master\myenv\vid\The
secrets to decoding facial expressions.mp4")
        #self.video = cv2.VideoCapture(0)

    def __del__(self):
        self.video.release()

    # returns camera frames along with bounding boxes and predictions
    def get_frame(self):
        _, fr = self.video.read()
        gray_fr = cv2.cvtColor(fr, cv2.COLOR_BGR2GRAY)
        faces = facec.detectMultiScale(gray_fr, 1.3, 5)

        for (x, y, w, h) in faces:
            fc = gray_fr[y:y+h, x:x+w]
            roi = cv2.resize(fc, (48, 48))
            pred = model.predict_emotion(roi[np.newaxis, :, :,
np.newaxis])
            ret.append(pred)

            cv2.putText(fr, pred, (x, y), font, 1, (255, 255, 0), 2)
            cv2.rectangle(fr,(x,y),(x+w,y+h),(255,0,0),2)

        return fr
def gen(camera):
    while True:
        frame = camera.get_frame()
        cv2.imshow('Facial Expression Recognization',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cv2.destroyAllWindows()
gen(VideoCamera())
max_res = max(ret,key=ret.count)
print("The maximum shown emotion is: ", max_res)
import random
import os
import pathlib
import subprocess

mp = 'C:/Program Files (x86)/Windows Media Player/wmplayer.exe'
if max_res == 'Angry':
    randomfile =
random.choice(os.listdir(os.path.abspath('')+"\\songs\\Angry\\"))
```

```
    print('You are angry !!!! please calm down:) ,I will play song for
you :' + randomfile)
    file = (os.path.abspath('')+"\\songs\\Angry\\" + randomfile)
    subprocess.call([mp, file])

if max_res == 'Happy':
    randomfile =
random.choice(os.listdir(os.path.abspath('')+"\\songs\\Happy\\"))
    print('You are smiling :) ,I playing special song for you: ' +
randomfile)
    file = (os.path.abspath('')+"\\songs\\Happy\\" + randomfile)
    subprocess.call([mp, file])

if max_res == 'Fear':
    randomfile =
random.choice(os.listdir(os.path.abspath('')+"\\songs\\Fear\\"))
    print('You have fear of something ,I playing song for you: ' +
randomfile)
    file = (os.path.abspath('')+"\\songs\\Fear\\"+ randomfile)
    subprocess.call([mp, file])

if max_res == 'Neutral':
    randomfile =
random.choice(os.listdir(os.path.abspath('')+"\\songs\\Neutral\\"))
    print('You are sad,dont worry:) ,I playing song for you: ' +
randomfile)
    file = (os.path.abspath('')+"\\songs\\Neutral\\" + randomfile)
    subprocess.call([mp, file])
```