EMOTION-BASED MUSIC RECOMMENDATION SYSTEM

A mini project report submitted by

MANU SASIKANTH O (EKC19CS027)

MOHAMMED NUFAIS (EKC19CS033)

MUHAMMED SHAMEEM(EKC19CS039)

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of the APJ Abdul Kalam Technological University

under the supervision of

Ms. HASEENA M

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ERANAD KNOWLEDGE CITY TECHNICAL CAMPUS
MANJARI, MALAPPURAM–676122
JULY-2022

BONAFIDE CERTIFICATE

Certified that this project report **EMOTION-BASED MUSIC RECOMMENDATION SYSTEM** is the bonafide work of **MUHAMMAD SHAMEEM** who carried out the project work under the supervision of **Ms. HASEENA M**.

SIGNATURE	**SIGNATURE**
Dr. ATHIRA B KAIMAL	**Ms. HASEENA M**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
COMPUTER SCIENCE AND	**COMPUTER SCIENCE AND**
ENGINEERING DEPARTMENT	**ENGINEERING DEPARTMENT**

ACKNOWLEDGEMENT

We would remember with grateful appreciation, the encouragement, and support rendered by ***Dr. VINCE PAUL****, Principal of Eranad Knowledge City Technical Campus, Manjeri.*

We express our deepest sense of gratitude to the Head of the department ***Dr. ATHIRA B KAIMAL*** *and our internal guide* ***Mrs. HASEENA M****, Assistant Professor of Computer Science and Engineering Department for their valuable advice and guidance.*

We would always oblige for the helping hands of all other staff members of the department and all our friends and well-wishers, who directly or indirectly contributed to this venture.

Last but not least, we are indebted to God Almighty for being the guiding light throughout this Project and helping us to complete the same within the stipulated time.

MUHAMMED SHAMEEM(EKC19CS039)

ABSTRACT

This project proposes a recommendation system based on emotional computing, automatic classification, and feature extraction, which recommends music based on the emotion expressed by the human being. The algorithm used to implement in our project is haar-Cascade (Object detection an Object Detection Algorithm used to identify faces in an image or a real-time video.) and CNN model architecture (designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers).

In this system, a data set was used for the training and testing of the system done with a directory containing 35,887 files and 28,709 files for training, and 7,178 files for testing the system.

In this application image of a person is captured using a real-time machine that has the access to the local machinery and depending on the captured image it compares the database data sets that are already saved in the local device through processing it defines the present mood of the user in numerical form based on this music will be recommended and this system is mainly proposed because music play a vital role in recent times that is to reduce stress so, to detect the emotion using the face as a main source of data because normally face expression defines the Emotion so according to the mood and play the music that it can change the user's mood.

CONTENT

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	-	Convolutional Neural Network
OpenCV	-	Open-Source Computer Vision Library
EEL	-	Extensible Embeddable Language
NumPy	-	Numerical Python
SDK	-	Software Development Kit
UI	-	User Interface
AU	-	Action Units
FACS	-	Facial Action Coding System
SaaS	-	Software as a Service
IaaS	-	Infrastructure as a Service
PaaS	-	Platform as a Service
HOG	-	Histogram of Oriented Gradients
SVM	-	Support Vector Machine
LBP	-	Local Binary Pattern
PCA	-	Principal Component Analysis
Ada Boost	-	Adaptive Boosting
EEG	-	Electroencephalogram
HDF5	-	Hierarchical Data Format 5
HTML	-	Hypertext Mark-up Language
CSS	-	Cascading Style Sheets
XML	-	Extensible Markup Language
GUI	-	Graphical User Interface
WSGI	-	Web Server Gateway Interface
GSR	-	Galvanic Skin Response
PPG	-	Plethysmography Physiological Sensors

CHAPTER - 1
INTRODUCTION

A lot of research has been done concerning music-driven influence on the physiological and emotional state of a human. Humans perceive a variety of feelings from different types of music and from ancient times considered music influenced the formation of a personal character and ability to treat diseases. Music listening has a significant impact on human feelings, and thoughts and as a result, it influences mental and physical health, and the topic of music wellbeing support is gaining popularity. A lot of measurements and research has been conducted to understand the impact of music on brain activity.

In the book written by Steven Pinker in 1997 “How the Mind Works”, Steven Pinker dubbed the music "auditory cheesecake", a phrase that in the years since has served as a challenge to the musicologists and psychologists who believe otherwise. Among those to note this stir was Philip Ball in his book The Music Instinct where he noted that music seems to reach to the very core of what it means to be human: "There are cultures in the world were to say 'I'm not musical' would be meaningless," Ball writes, "akin to saying 'I'm not alive.". In a filmed debate, Ball suggests that music might get its emotive power through its ability to mimic people, and perhaps its ability to entice us lies in music's ability to set up an expectation and then violate it [7].

The above paragraphs state the importance of music to enhance the mental challenges faced in everyday life. To overcome this crisis, a music-based song recommendation system to enhance the mood of the human being.

In this concept of an emotion-based music recommendation system, the emotion is detected by using the user to detect the real-time capturing of the user’s emotions. The proposed system detects different types of emotions such as happy, sad, fear, neutral or anger, etc. A music recommendation system with any web browser as the front-End can detect emotions that is, the face of the user with the help of a machine learning algorithm using python. Based on the detected user’s mood.

In this application image of a person is captured using a real-time machine that has the access to the local machinery and depending on the captured image it compares the database data sets that are already saved in the local device through processing it defines the present mood of the user in numerical form based on this music will be recommended and this system is mainly proposed because music play a vital role in recent times that is to reduce stress so, to detect the emotion we are using the face as a main source of data because normally face expression defines the Emotion so according to the mood the music played that it can change the user's mood.

This system uses CNN model architecture. Convolutional neural network (CNN), a class of artificial neural networks that have become dominant in various computer vision tasks, is attracting interest across a variety of domains. CNN is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers.

The data set used for the training and testing of the system is done with a directory containing 35,887 files and 28,709 files for training and 7,178 files for testing the system.

In this system, we are using the harr-cascade algorithm to detect emotion by real-time capture of image and extraction of image. It is an Object Detection Algorithm used to identify faces in an image or a real-time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001 [6].

1.1 OBJECTIVES

To provide an interface between the music system and the user. The music recommendation system provides a new platform for music lovers where they can hear the music based on their emotions.

1.2 PROBLEM STATEMENT

In the present-day environment, there is a lot of stress and fatigue, which leads to more despair and ill health. Medical studies demonstrate that interactive and high-quality games and shows, as well as music tailored to one's mood, can lift one's spirits. Reducing stress hormones can aid in the treatment of depression along with helping in physical aches and pains induced by the high level of stress. However, additional research shows that listening to random music that is unrelated to one's mood can have a stressful effect on people. As a result, listening to music (premium ad-free or with advertisements) to unwind after work can improve one's health.

1.3 BENEFITS

In this project, the emotion is detected and output the music according to the present emotion. By using this system, the user can easily determine the state of a human being and give out the required result. In this case, emotion is detected and outputs the music recommendation.

CHAPTER - 2
LITERATURE SURVEY

2.1 EMOTION-BASED MUSIC RECOMMENDATION SYSTEM [1]

In this concept, music is recommended to the user by detecting the real-time capturing the user's emotions. Emotion-Based-music-player is a music player with chrome as the front-End which can detect emotions I.e., the face of the user with the help of a machine learning algorithm using python. Based on the detected user's mood song the list will be displayed/recommend to the user. In this application image of a person is captured using a real-time machine that has the access to the local machinery and depending on the captured image it compares the database data sets that are already saved in the local device through processing, it defines the present mood of the user in numerical form based on this music will be played other than that we have some common features that are queue playlist so that we can have an individual playlist and the last one is random it uses python Eel library so that it can pick a random song without any order. for this, we have used libraries like OpenCV, EEL, NumPy, etc. this system is mainly proposed because music plays a vital role in recent times which is to reduce stress. so, to detect the emotion, they are using the face as a main source of data because normally facial expression defines Emotion so according to the mood, we play the music that can change the user's mood.

The process of multidimensional reduction is by taking the primary data that is lowered to many other classes for sorting out or organizing. A user's emotion is extracted by capturing the user's image through a webcam. The captured image is enhanced by the process of dimensional reduction by tracking the primary data. These data are converted into binary image format and the face is detected using Fisher Face and Haar cascade methods. In this project, the music recommendation model is based on the emotions that are captured in real-time images of the user. This project is designed to make better interaction between the music system and the user. because Music helps change the mood of the user and for some people, it is a stress reliever. Recent development shows a wide perspective on the development of the emotion-based music recommendation system. Thus, the present system presents a Face(expressions) based recognition system so that it could detect emotions and music will be played accordingly. The Architecture Diagram is represented in figure 2.1.

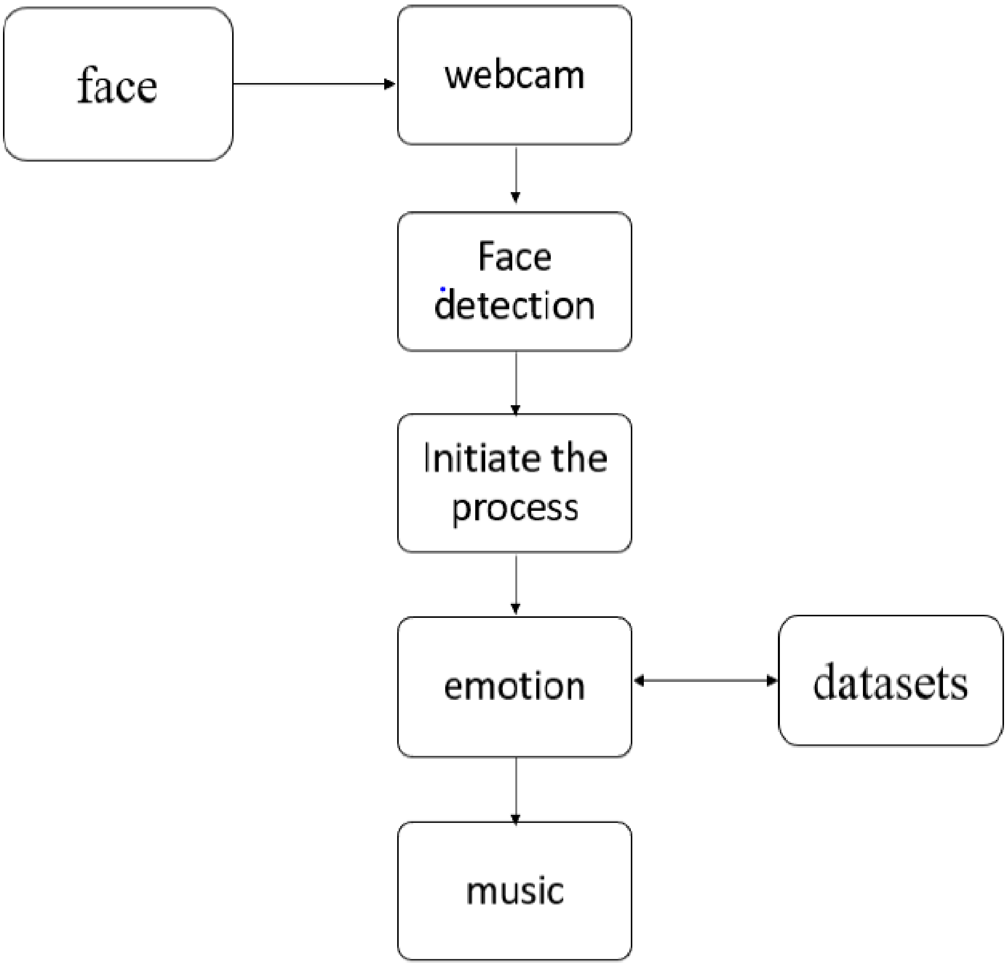


Figure 2.1. Architecture Diagram

2.2 EXPRESSION-X: EMOTION-BASED MUSIC RECOMMENDATION SYSTEM [2]

This Paper proposes a method to recommend a set of songs based on the facial and emotional state of the user. Emotion state of the user is detected with the help of google mobile vision SDK. The detected emotion state is fed to the Expression-X algorithm that would sort the music (based on emotion value keyed in) and generates a playlist that suits the emotional state of the user. In this paper, they are discussing an effective way to capture the facial expression of a user and then process it to figure out the mood and then accordingly produce a playlist for the user's liking. The content discussed in this paper focuses on the following aspect. Capture the facial image of the user using the vision SDK and then process the image for emotion detection. Build an algorithm that would transform the emotion detected and produce a playlist accordingly. Create a user-friendly UI for smooth processing and playing all types of music. Produce a recommendation list based on the user's emotions.

This paper proposed an approach named expression-x for recommending a set of songs based on the emotional states depicted on the user's face. Expression X uses a mobile camera to capture the face of the user, once the user's photo is available, it is processed using Google mobile vision SDK to get the emotional state of the user (on a scale of 0 to 1). The emotional information is fed to the Expression X algorithm, which recommends a set of songs, which will likely please the user. The Proposed System is represented in Figure 2.2.

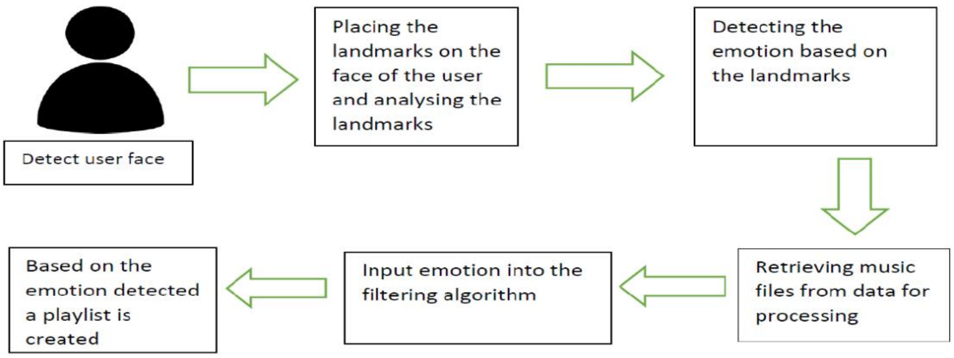

Detect user face
Placing the landmarks on the face of the user and analysing the landmarks
Detecting the emotion based on the landmarks
Retrieving music files from data for processing
Input emotion into the filtering algorithm
Based on the emotion detected a playlist is created

Figure 2.2. Proposed System

2.3 FACIAL EMOTION- BASED SONG RECOMMENDATION SYSTEM [3]

This system is working based on convolutional neural networking (CNN) and machine learning algorithms. The music is recommended by analyzing a structured dataset with the embedded email system with music distribution for getting music, along with the dataset for music name and associate emotion. Dataset of emotion is labeled images for machine learning algorithm training with the help of CNN emotion is detested and the socket program is lined between client and server of the program. Then the data is analyzed and the music is recommended. This system is a mobile application where the user can log in and sign to the system and the connecter to the client-server to communicate. The limitation of this system is the age limitation present system where it can only be used by 12 years and above. The number of datasets and songs is limited in this system. The Hierarchical Task Analysis and Data Flow Diagram are represented in Figures 2.3 and 2.4.

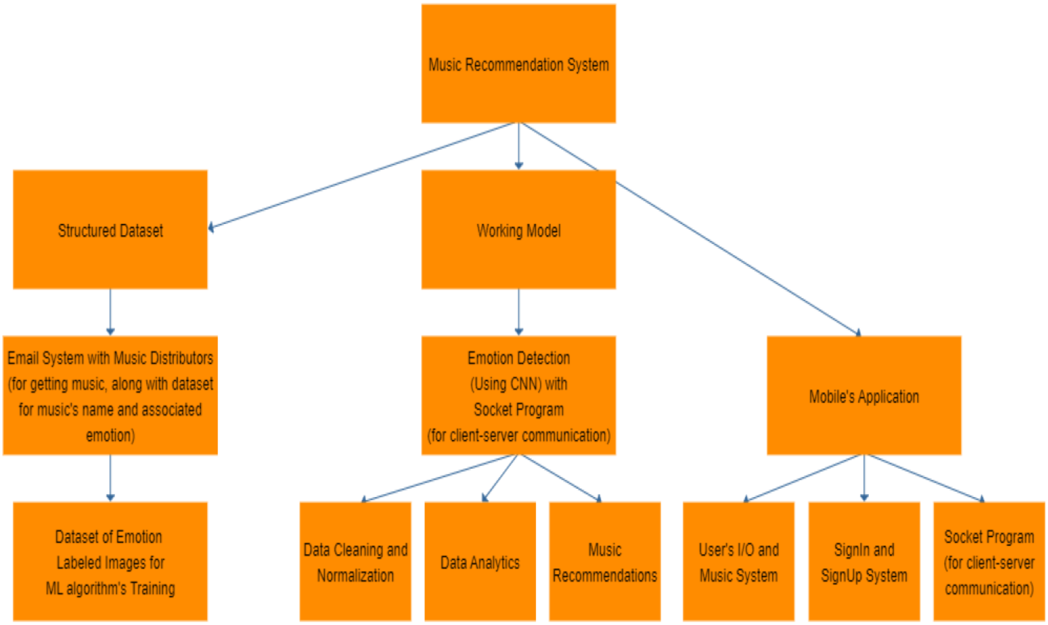

Music Recommendation System
Structured Dataset
Working Model
Email System with Music Distributors (for getting music, along with dataset for music's name and associated emotion)
Emotion Detection (Using CNN) with Socket Program (for client-server communication)
Mobile's Application
Dataset of Emotion Labeled Images for ML algorithm's Training
Data Cleaning and Normalization
Data Analytics
Music Recommendations
User's I/O and Music System
SignIn and SignUp System
Socket Program (for client-server communication)

Figure 2.3. Hierarchical Task Analysis

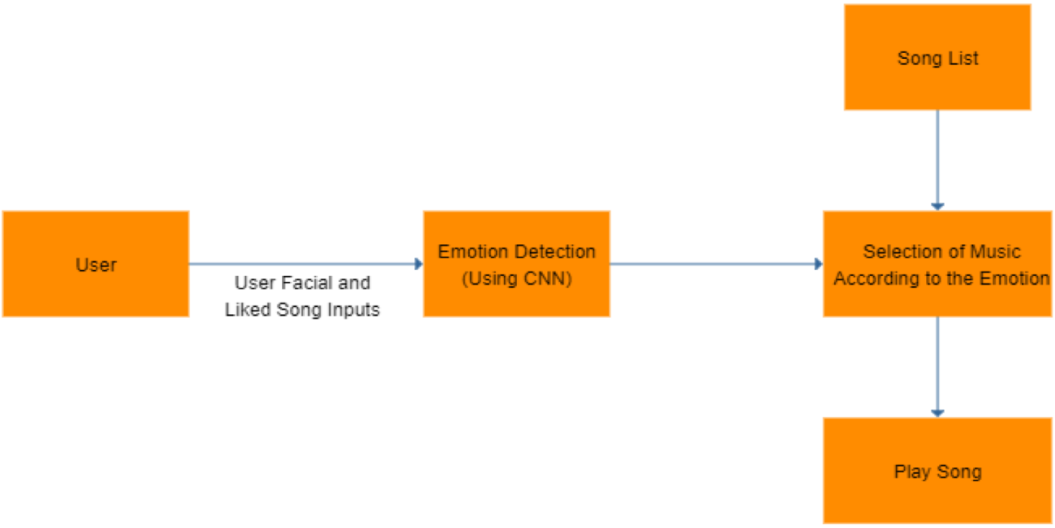

Song List
User
User Facial and Liked Song Inputs
Emotion Detection (Using CNN)
Selection of Music According to the Emotion
Play Song

Figure 2.4. Data Flow Diagram

2.4 EMOTION-BASED MUSIC RECOMMENDATION SYSTEM [4]

In this system, the face of the person is recorded on the webcam. The recorded video is converted into frames. Using Pre-processing the facial expression are converted into a sequence of Action Units (AUs) from the image obtained from the webcam. The Facial Action Coding System (FACS) is a system that describes all the facial expressions using combinations of the 64 AUs. After Feature Extraction, the Emotions are classified whether it is Happy, Angry, Sad, and Surprised faces. The web services are integrated with them. They are SAAS, IAAS, PAAS. The emotions are transferred and the music is played from the emotions detected. The data that are obtained are decomposed into the sampling image using an image pyramid into multiple scales. The use of this technique is simply to extract features while reducing the noise and the other factors. The low pass image pyramid technique (also known as Gaussian pyramid) consists of smoothing the frame and subsampling it by decreasing its resolution, the process needs to be repeated a few times to obtain a perfect result that at the end of the process we obtain a frame similar to the original one but with a decreased resolution and an increased smoothing level. To detect the objects in images in the field of image processing, HOG is a feature descriptor, a technique that counts occurrences of gradient orientation in a localized portion of an image. The main objective of using this technique is to describe the face within the image with a set of distributions of intensity gradients. The linear classification step is the last in the face detection process. They used the linear classifier instead of the SVM to decrease the computational time that the classification process will take and so to guarantee a faster face detection operation. The limitation of this system is not able to record all types of emotions or detect the emotion correctly because of the less use of the training dataset. The Overall Architecture Diagram and Flow Diagram are represented in Figures 2.5 and 2.6.

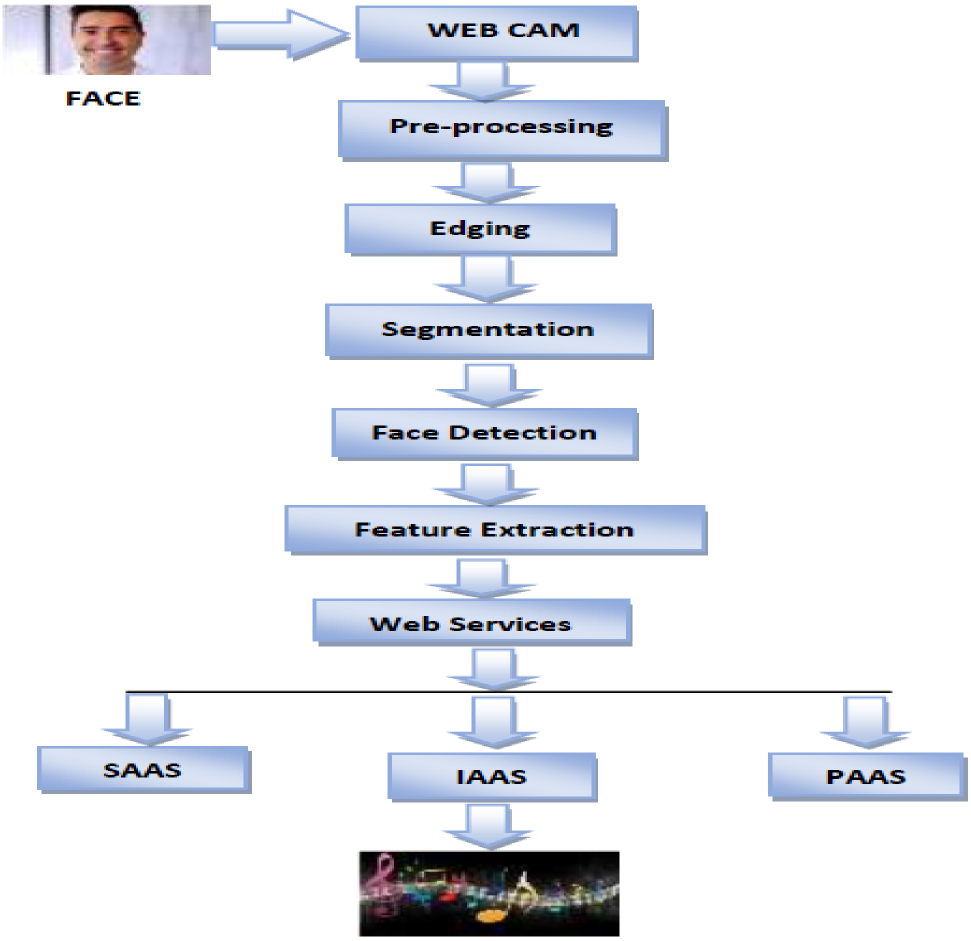


Figure 2.5. Overall Architecture Diagram

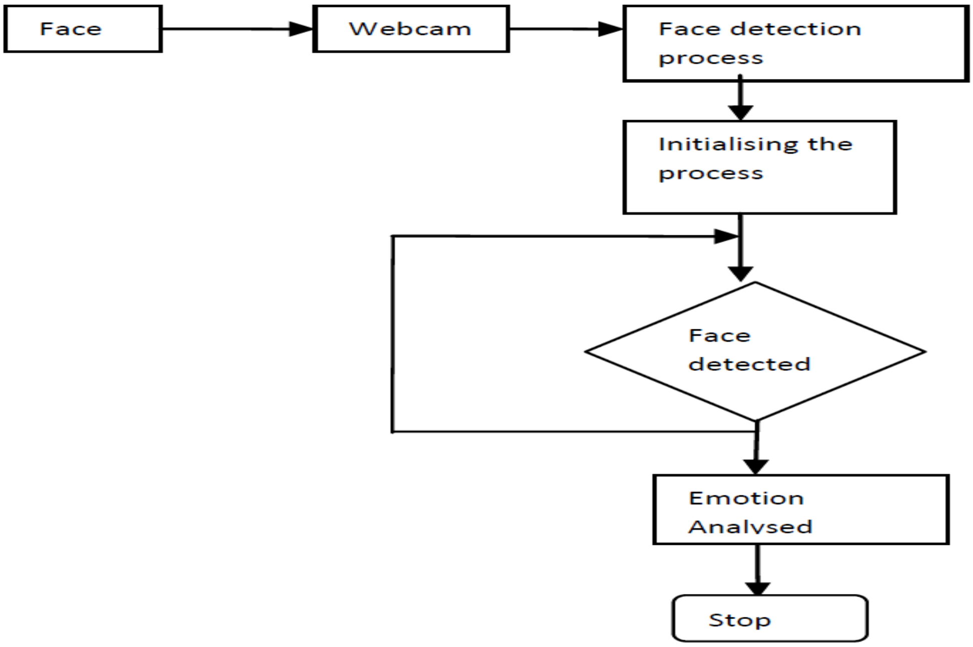


Figure 2.6. Flow Diagram

2.5 REAL-TIME FACE DETECTION AND RECOGNITION IN COMPLEX BACKGROUND [5]

This paper provides efficient and robust real-time face detection and recognition algorithms in complex backgrounds. The algorithms are implemented using a series of signal processing methods including Ada Boost, cascade classifier, Local Binary Pattern (LBP), Haar-like feature, facial image pre-processing, and Principal Component Analysis (PCA). The Ada Boost algorithm is implemented in a cascade classifier to train the face and eye detectors with robust detection accuracy. The LBP descriptor is utilized to extract facial features for fast face detection. The eye detection algorithm reduces the false face detection rate. The detected facial image is then processed to correct the orientation and increase the contrast, maintaining high facial recognition accuracy. Finally, the PCA algorithm is used to recognize faces efficiently. Large databases with face and non-face images are used to train and validate face detection and facial recognition algorithms. This algorithm can detect and recognize faces with high accuracy in real-time. It has a faster detection speed compared to other detection methods. Eye detection is used to increase face detection accuracy. The facial recognition performances are also greatly improved by using facial components alignment, contrast enhancement, and image smoothing. Images of faces are collected as training samples in real-time and recognized under various conditions including other faces. The limitation is that the less facial recognition in a wider range of facial orientations. The Flowchart of Facial Recognition in represented in Figure 2.7.

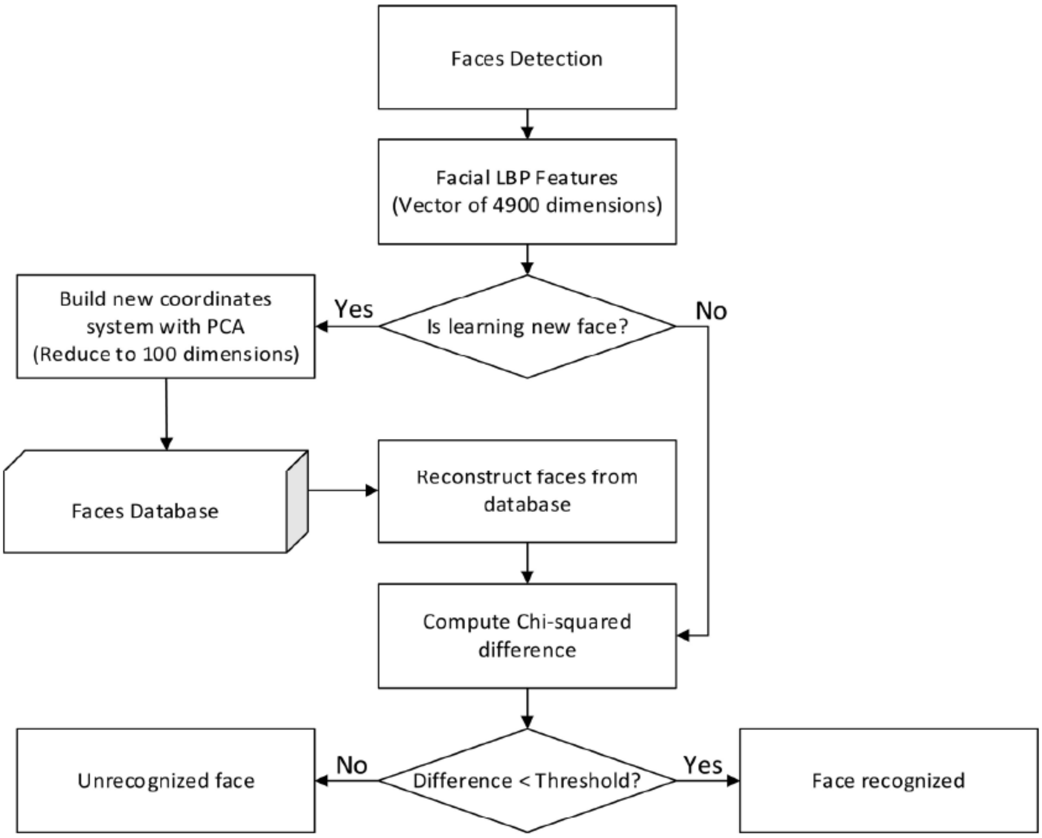

Faces Detection
Facial LBP Features (Vector of 4900 dimensions)
Is learning new face?
Yes
No
Build new coordinates system with PCA (Reduce to 100 dimensions)
Faces Database
Reconstruct faces from database
Compute Chi-squared difference
Difference < Threshold?
No
Yes
Unrecognized face
Face recognized

Figure. 2.7. Flowchart of Facial Recognition

CHAPTER - 3

REQUIRED STUDY AND ANALYSIS

System Analysis is a systematic process of gathering, recording, and interpreting facts. It also includes studying the problems encountered in the present systems and planning the development of a system that could overcome the problems.

3.1 EXISTING SYSTEM

As per the papers referenced, there exist various systems that could predict the song by their emotions. Human emotion has played a vital role in recent times emotion is based on human feelings which can be both expressed or not. Emotion represents human behavior which can be in different forms. Extraction of the emotion states a human's individual state of behavior. The objective of this project is to extract features from the human face and detect emotions to play music according to the emotion detected. However, many existing techniques use previous data to suggest music and the other algorithms used are normally slow, usually, they are less accurate and even require additional hardware like EEG or physiological sensors. Facial expressions are captured by a local capturing device or an inbuilt camera. Here we use algorithms like Fisher Face and Haar-cascade for the recognition of the feature from the captured image. Thus, the proposed system is based on the facial expression captured, and will music will be played automatically. This project was designed to make better interaction between the music system and the user. because Music is helpful in chain help changes and for some people, it is a stress reliever. Recent development shows a wide perspective on the development of the emotion-based music recommendation system. Thus, the present system presents a Face(expressions) based recognition system so that it could detect emotions and music will be played accordingly.

3.2 PROBLEMS IN THE EXISTING SYSTEM

The need for a more efficient emotion-based music system has arisen in the current scenario since there exists an emotion-based music system so that it could detect emotions and music will be played accordingly. A user who wants to detect their emotion correctly and play the suitable song for their mood is what a user looking for so the drawback of this system is they only have four types of emotions ‘angry, happy, sad, and neutral’. If a user uses this system but they have a different type of emotion than the above mentioned the users didn't get satisfied and they get stressed by the system so the user eventually gets rid of the system. And also, this system only has a limited number of songs which also may be a problem for some users that are not listening to repeated songs.

3.3 PROPOSED SYSTEM

Humans tend to show their emotions unknowingly mainly they reflect on their faces. The proposed system helps us to provide an interaction between the user and the music system. This project mainly focuses on the user’s preferred music that is recommended due to emotional awareness. In the initial stage of the proposed system, we have given a capture button by clicking that button the system camera capture the face of the user after capturing the face of the user we detect the emotion from the face by using the Haar-cascade algorithm there are 7 types of emotion are taking here like angry, disgust, fear, happy, neutral, sad, and surprise. Then compare the detected emotion with the trained and tested data set using the CNN algorithm. Then the user gets the types of songs that suites their emotion and the user can select the song that they want to listen to. Users can listen to any this system doesn’t have e destination on songs because this system uses the YouTube platform for playing songs. The users can happily listen to the song and enjoy the mood of their bodies.

CHAPTER - 4

IMPLEMENTATION

Implementation is the execution or practice of a plan, a method, or any design, idea, model, specification, standard, or policy for doing something. As such, implementation is the action that must follow any preliminary thinking for something to happen.

4.1 METHODOLOGY

The music is recommended by the emotion that we are displaying. For the recognition and detection, the use of the CNN algorithm and the Haar-cascade algorithm is used. In this system Haar-cascade, the is used for This algorithm is used for Object Detection which is used to identify faces in an image or a real-time video. The CNN algorithm is used for image recognition and processing that is specifically designed to process pixel data. By testing and training, the system accuracy of 80 percent is obtained. The total emotion used was seven typed happy, sad, angry, disgust, neutral, surprise, and fear. All the emotions have been detected with less error in detection. The process in the system is represented in Figure 4.1.

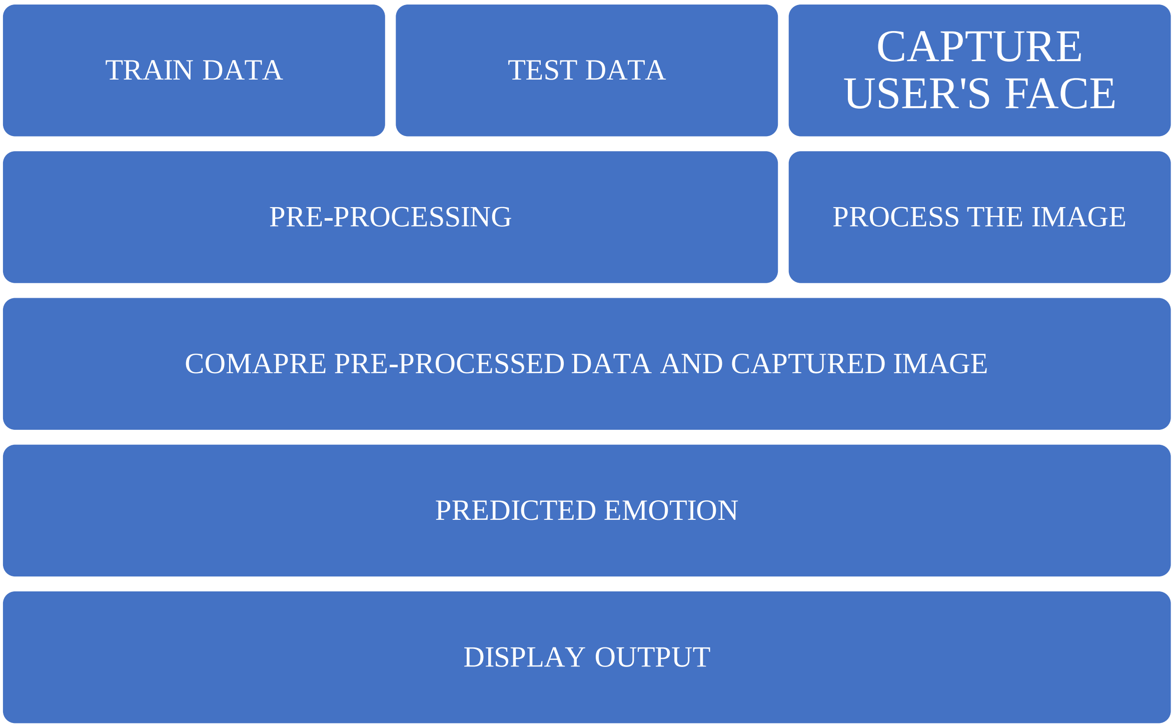


Figure 4.1. Process In the System

4.2 STEPS

STEP 1: Acquiring Initial Dataset

To start with, the collecting of the dataset comes first. For that the need for two types of datasets one for training and the other for testing the system that is implemented. In this project, the dataset has been selected from Kaggle. In this system, the data set was used for the training and testing of the system with a directory containing 35,887 files and 28,709 files for training, and 7,178 files for testing the system.

STEP 2: Selecting the Algorithm

The algorithm selected was the CNN algorithm and the Haar cascade algorithm. The CNN is used for image processing of the data set that has been given, and the training and testing process is done with the help CNN algorithm. The Haar-cascade is used for object detection which is used to detect the emotion of the user. It also processes the image.

STEP 3: Pre-processed Data

The CNN algorithm helps use the train and test the images in the directory. The image is being processed and is saved to a huge array file which can contain a huge amount of data. The model that has been trained is saved to an HDF5 file in the name of ‘model.h5’. In this process, the image in the dataset is resized and greyscaled the image for edge detection of the image.

STEP 4: Image Capture and Process the Image

The Haar-cascade algorithm is used for the real-time capture of the image of the user’s face. The captured image is converted to a greyscaled image and is resized to the desired size. The image processing is compared with the previously trained model and detects if there is any common feature in the image. This helps in the detection of emotion.

STEP 5: Implement Stage

The protected emotion is redirected to YouTube for recommending the music. The user will only be seeing the user interface in the system.

The capturing interface and the recommendation interface are visible to the user. The user is provided with a satisfying output.

4.3 SYSTEM REQUIREMENTS

4.3.1 HARDWARE REQUIREMENTS

- Processor – Intel i3 and above, AMD Ryzen 5 and above (64 bits)
- Ram – 4 GB or more
- Hard disk Space – 2 GB
- Graphics card - Intel HD Graphics 510 2GB, NVidia gtx 660 2GB

4.3.2 SOFTWARE REQUIREMENTS

- Operating System – Windows 10 and above, Ubuntu
- Web Browsers - MS Edge, Google Chrome, Mozilla Firefox
- Internet Bandwidth – 1 Mbps and more
- Framework - Flask

4.4 MODULES

4.4.1 USER MODULE

The main module interacts with the user through the website. A well-defined, simple to use, and user-friendly interface design for interaction. The user only has to click on the button ’capture’ to capture their image. Then YouTube will site will be opened for the music selection, and the song predicted will be emotion based on current emotion.

4.4.2 IMAGE MODULE

This module is for viewing the captured image when we click on the capture button the image will be displayed and the current emotion will also be mentioned in this module

CHAPTER - 5

SYSTEM DESIGN AND DEVELOPMENT

System design is the process of developing a candidate system that meets the criteria established in system analysis. User requirements are translated into system characteristics during system design. System design involves firstly the logical design and then the physical construction of a new system. The logical design describes the detailed specifications for the new system, the input/output, files and databases, and procedures, all in a manner that must be project requirements. Physical construction, the activity following design produces software files and a working system.

5.1 INPUT DESIGN

Input design is a part of the overall system design, which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and the people involved. If the data going into the system is incorrect, then the processing and output will magnify the errors. Thus, the clear objectives of input design are:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable to and understood by the user.

5.2 OUTPUT DESIGN

Effective output design will improve the clarity and performance of outputs. The output design phase of the system is concerned with the convergence of information to the end user-friendly manner. The output design should be efficient, intelligible so that the system relationship with the end user is improved and thereby enhancing the process of decision making. Outputs from the processing systems are required primarily to communicate the results of the processing to the users. The output design should be made with extra care since it depicts the overall effectiveness of the system for the purpose for which it is designed. There should not be any ambiguities among the end users in the output results obtained.

5.3 SYSTEM ARCHITECTURE

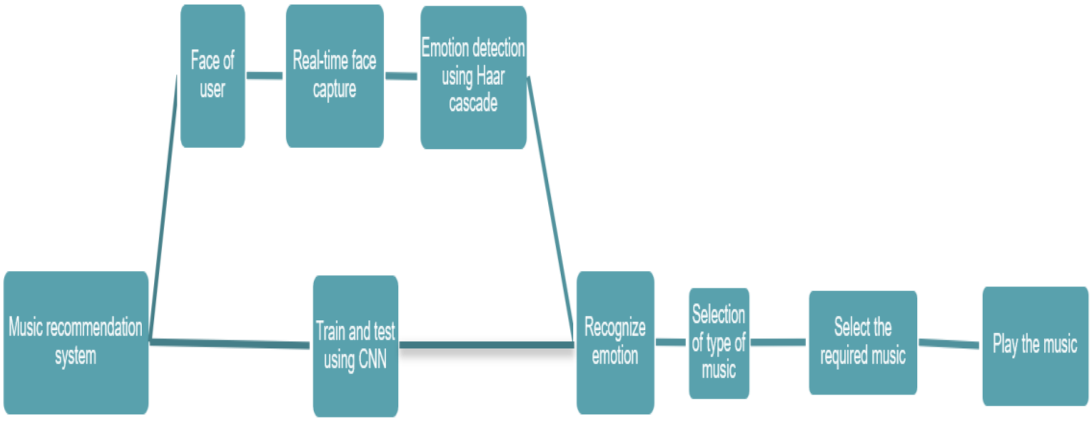

Face of user
Real-time face capture
Emotion detection using Haar cascade
Music recommendation system
Train and test using CNN
Recognize emotion
Selection of type of music
Select the required music
Play the music

Figure 5.1 System Architecture

5.4 TESTING

Software testing is used for checking flaws and errors in the software. It helps the system to develop further without any flaws. In this system, there are two tubes of testing. They are:

- The testing with the user's real face to get the emotion of the user. This system is mainly developed for the detection of the emotions of the user, and recommending the song they required.
- The other testing is done by the artificial image capturing. In this testing, the face is detected by using an external source such as a photograph image of a person. The emotion will be detected by the system same in the way as it does with the real-time capture

CHAPTER - 6

TOOLS AND TECHNOLOGY

6.1 FRONT END

The front-end of a system is the part where users interact with everything. HTML and CSS are used here to create and style the webpage Bootstrap is also used to make the website more interactive and responsive. XML is used to structure data from storage and transport it HDF5 files are used for storing a large amount of data it is an open-source file that comes in handy to store large amounts of data.

6.1.1 BOOTSTRAP

Bootstrap is the most popular HTML, CSS, and CSS-based JavaScript framework for developing a responsive and friendly website. It is free to download and use. It is a front-end framework used for easier and faster web development. It includes HTML and CSS-based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and many others can also use JavaScript plug-ins to facilitate your to create responsive designs.

6.1.2 HTML

The Hypertext Mark-up Language or HTML is the standard mark-up language for documents designed to be displayed in a web browser. Web browsers receive HTML documents from a web server or local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

6.1.3 CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML. CSS is a cornerstone technology of the World Wide Web used to alter the font, color, size, and spacing of your content split it into multiple columns, or add animations and other decorative features. A webpage created using HTML can provide the functionality but the user interface becomes attractive by embedding CSS to it.

6.1.4 XML

An XML file is an Extensible Markup Language file. They are plain text files that don't do anything in and of themselves except describe the transportation, structure, and storage of data.

6.1.5 HDF5

HDF5 file stands for Hierarchical Data Format 5. It is an open-source file that comes in handy to store a large amount of data. As the name suggests, it stores data in a hierarchical structure within a single file. So, if we want to quickly access a particular part of the file rather than the whole file, we can easily do that using HDF5. This functionality is not seen in normal text files hence HDF5 is becoming seemingly popular being a new concept. To use HDF5, NumPy needs to be imported. One important feature is that it can attach a meta set to every data in the file thus providing powerful searching and accessing.

6.2 BACK END

Python is used here for an application that includes prediction, data retrieval, storage, querying, etc. FLASK is also used in the back end It is a framework that is mainly used for machine learning applications.

6.2.1 PYTHON

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. Features of Python:

- Easy to code: Python is a high-level programming language
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is a Portable language
- Python is an integrated language
- Expressive
- Embeddable
- Large and Standard Library
- Dynamically Typed

6.2.2 FLASK

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects. The Flask framework uses Werkzeg as one of its bases. jinja2 is a popular template engine for Python. A web template system combines a template with a specific data source to render a dynamic web page. Flask is often referred to as a micro-framework.

CHAPTER - 7
RESULT AND DISCUSSION

The emotion-based music recommendation system is successfully finished. The result that the user gets is music according to their emotions. This system is successfully implemented with the help of CNN and Haar-Cascade algorithms. The user can detect their emotion with this system as well as recommend music according to their emotion. the music recommendation model is based on the emotions that are captured in real-time images of the user. The present system presents a Face(expressions) based recognition system so that it could detect emotions and music will be played accordingly. In this project first, the data set given is read and then trained and tested creating a huge data of about the process. this data is used as the comparison file for the next step. the real-time capture of the face is done. then it is greyscaled and then resized to the desired size. then the next process is to recognize the emotion to recommend the music. the picture that is captured is compared with the trained model to set the desired output. the output is successfully obtained. the photo is captured and the music is recommended on YouTube

The code for this system is in Appendix A and the image related to the capturing of the image and recommendation of the music is given in Appendix B

CHAPTER - 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

In this project, the music recommendation model is based on the emotions that are captured in real-time images of the user. This project is designed to make better interaction between the music system and the user. because Music helps change the mood of the user and for some people, it is a stress reliever. Recent development shows a wide perspective in developing the emotion-based music recommendation system. Thus, the present system presents a Face(expressions) based recognition system so that it could detect emotions and music will be played accordingly.

8.2 FUTURE SCOPE

The music player that we are using can be used locally and nowadays everything became portable and efficient to carry but the emotion of a person can be taken by different wearable sensors and easy to use rather than the whole manual work it would be possible using GSR (galvanic skin response) and PPG (plethysmography physiological sensors). that would give us enough data to predict the mood of the customer accurately. This system with enhanced will be able to benefit and the system with advanced features needs to be constantly upgraded. The methodology that enhances the automatic playing of songs is done by detection. The facial expressions are detected with the help of a programming interface that is present in the local machine. An alternative method is based on the additional emotions which are being excluded from our system.

REFERENCES

[1] Manoj Kumar M V, Prashanth B S, Syeda Sarah, Md. Kashif, Sneha H R Department of Information Science and Engineering Nitte Meenakshi Institute of Technology, Bengaluru.

[2] Dr. Sunil Bhutada, Assistant Professor, Information Technology Department, Sreenidhi Institute of Science and Technology, Yamnampet, Hyderabad. CH. sadhvika, Gutta.A bigna, P. Srinivas reddy, Bachelor of Technology IVth year, IT Department, Sreenidhi Institute of Science and Technology, Yamnampet, Hyderabad.

[3] Divyansh Mulchandani, SCOPE, Vellore Institute of Technology, Vellore; Ankush Rauniyar SITE, Vellore Institute of Technology, Vellore.

[4] H. Immanuel James, J. James Anto Arnold, J. Maria Masilla Ruban, M. Tamilarasan, R. Saranya Dept. of Information Technology, Valliammai Engineering College, Chennai, Tamil Nadu; Asst.Prof, Dept. of Information Technology, Valliammai Engineering College, Chennai, Tamil Nadu.

[5] Xin Zhang, Thomas Gonnoy, and Jafar Saniie, (2017) Real-Time Face Detection and Recognition in Complex Background. Journal of Signal and Information Processing

[6] Viola-Jones object detection framework is a machine learning object detection framework proposed in 2001 by Paul Viola and Michael Jones (https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework#:~:text=The%20Viola%E2%80%93Jones%20object%20detection,detection%20of%20other%20object%20classes.)

[7] Pinker, S. How the Mind Works (Norton,1997) https://en.wikipedia.org/wiki/How_the_Mind_Works

APPENDIX A

CODES

PYTHON CODE FOR TRAINING AND TESTING OF THE SYSTEM

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.layers import Dense, Dropout, Flatten

from tensorflow.keras.models import Sequential

#from google.colab.patches import cv2_imshow

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.layers import Conv2D

from tensorflow.keras.layers import MaxPooling2D

num_train = 28707

num_val = 7174

batch = 32

num_of_epoch = 20

data_train = ImageDataGenerator(rescale=1./255)

val_datagen = ImageDataGenerator(rescale=1./255)

valid_gen = val_datagen.flow_from_directory(

                                                     'data/test',

                                                      batch_size=batch,

                                                      target_size=(48,48),

                                                    color_mode="grayscale",

                                                    class_mode='categorical')
```

```
training_gen = data_train.flow_from_directory(
                                                'data/train',
                                                 batch_size=batch,
                                                 target_size=(48,48),
                                                 color_mode="grayscale",
                                                 class_mode='categorical')
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001),metrics=['accuracy'])
model_info = model.fit_generator(
training_gen,
validation_data=valid_gen,

epochs=num_of_epoch,

validation_steps =num_val // batch)

model.save_weights('model.h5')
```

PYTHON CODE FOR REAL-TIME IMAGE CAPTURE AND PROCESS IMAGE

```
from flask import Flask, render_template, request

import numpy as np

import cv2

from keras.models import load_model

import webbrowser

app = Flask(__name__)

app.config['SEND_FILE_MAX_AGE_DEFAULT'] = 1

info = {}

haarcascade = "haarcascade_frontalface_default.xml"

label_map = ['angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'suprise']

print("+"*50, "loadin gmmodel")

model = load_model('model.h5')

cascade = cv2.CascadeClassifier(haarcascade)

@app.route('/')

def index():

    return render_template('index.html')

@app.route('/emotion_detect', methods=["POST"])
```

```
def emotion_detect():
    found = False
    cap = cv2.VideoCapture(0)
    while not(found):
        _, frm = cap.read()
        gray = cv2.cvtColor(frm,cv2.COLOR_BGR2GRAY)
        faces = cascade.detectMultiScale(gray, 1.4, 1)
        for x,y,w,h in faces:
            found = True
            roi = gray[y:y+h, x:x+w]
            cv2.imwrite("static/face.jpg", roi)
    roi = cv2.resize(roi, (48,48))
    roi = roi/255.0
    roi = np.reshape(roi, (1,48,48,1))
    prediction = model.predict(roi)
    print(prediction)
    prediction = np.argmax(prediction)
    prediction = label_map[prediction]
    cap.release()
    link=f"https://www.youtube.com/results?search_query={prediction
     }+song"
    webbrowser.open(link)
    return render_template("emotion_detect.html", data=prediction,
     link=link)
```

```
if __name__ == "__main__":
        app.run(debug=True)
```

HTML CODE FOR WEB STRUCTURE

```
<!DOCTYPE html>
<html>
<head>
<title>music recommend</title>
<link rel="stylesheet" type="text/css" href="{{url_for('static', filename='style.css')}}">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"integrity="sha384eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0
Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
</head>
<body style="background-color: #06283D;">
<form action="{{url_for('emotion_detect')}}" method="POST">
 <br>
<input type="submit" name="btn" value="CAPTURE" class="btn btn-outline-success btn-
lg">
<br>
</form>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0beta3/dist/js/bootstrap.bundle.min.js"int
egrity="sha384-
JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ek
f" crossorigin="anonymous"></script>
```

```
</body>

</html>
```

HTML CODE FOR IMAGE CAPUTRE

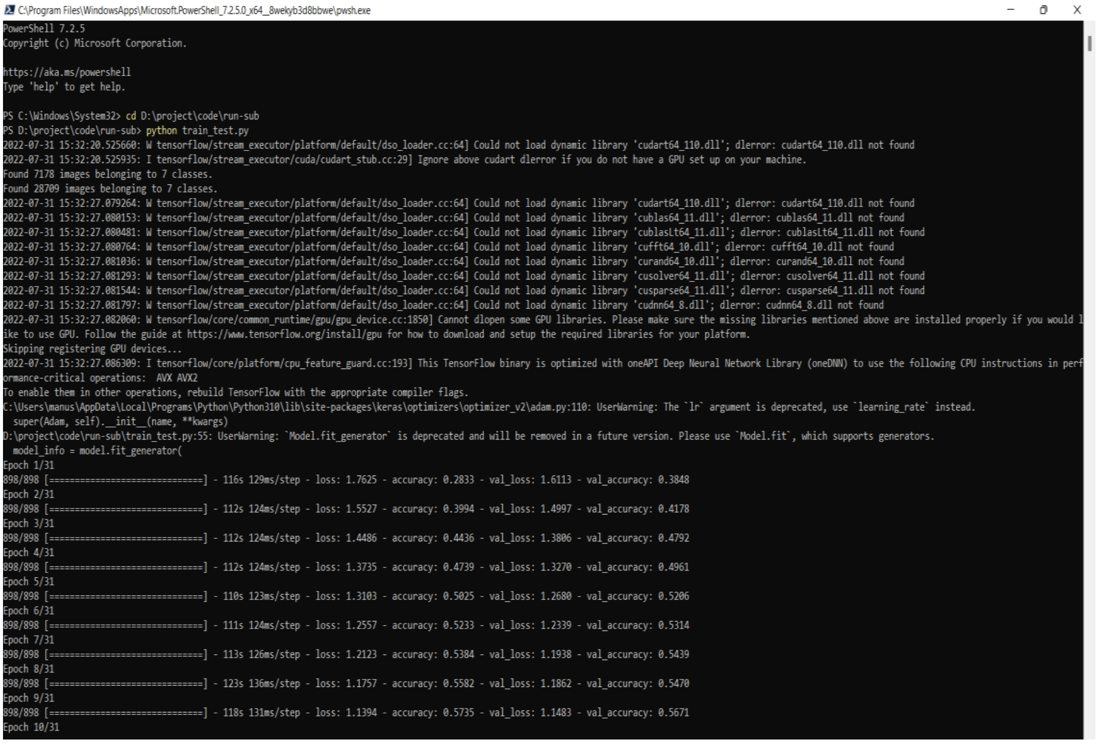
```
<!DOCTYPE html>

<html>

<head>

<title>detecting emotion</title>

<link rel="stylesheet" type="text/css" href="{{url_for('static', filename='style.css')}}">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

</head>

<body style="background-color: #06283D;">

<div class="title">

<h1 class="display-4">you looks like {{data}}</h1>

<h1><a href="{{link}}" title="">go to here</a></h1>

</div>

<img src="{{url_for('static', filename='face.jpg')}}">

</body>

</html>
```

CSS CODE FOR STYLING THE WEB PAGE

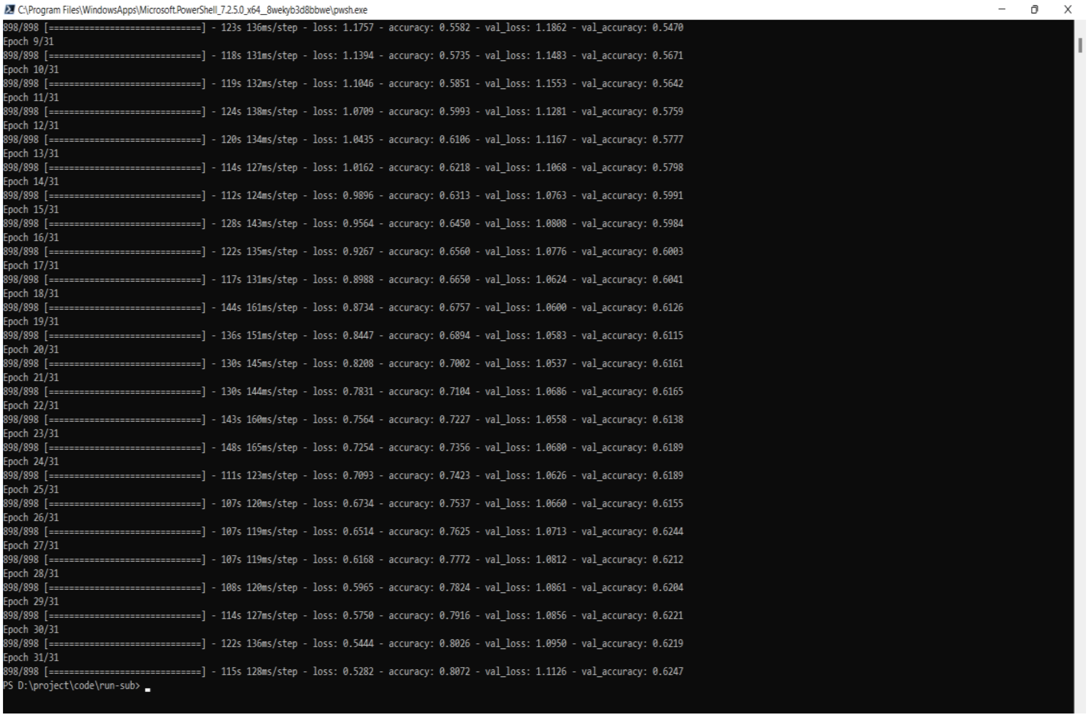
```
 body{
        text-align: center;
}
h1{
        font-family: sans-serif;
        color: #F6F6F6;
}
.title{
        margin-top: 20px;
}
.btn{
        width: 300px;
        height: 60px;
        background-color: #B2C8DF;
        }
img{
        margin-top: 50px;
        height: 400px;
        border : 2px solid green;
        border-radius: 30px;
}
```

APPENDIX B

SCREENSHOTS

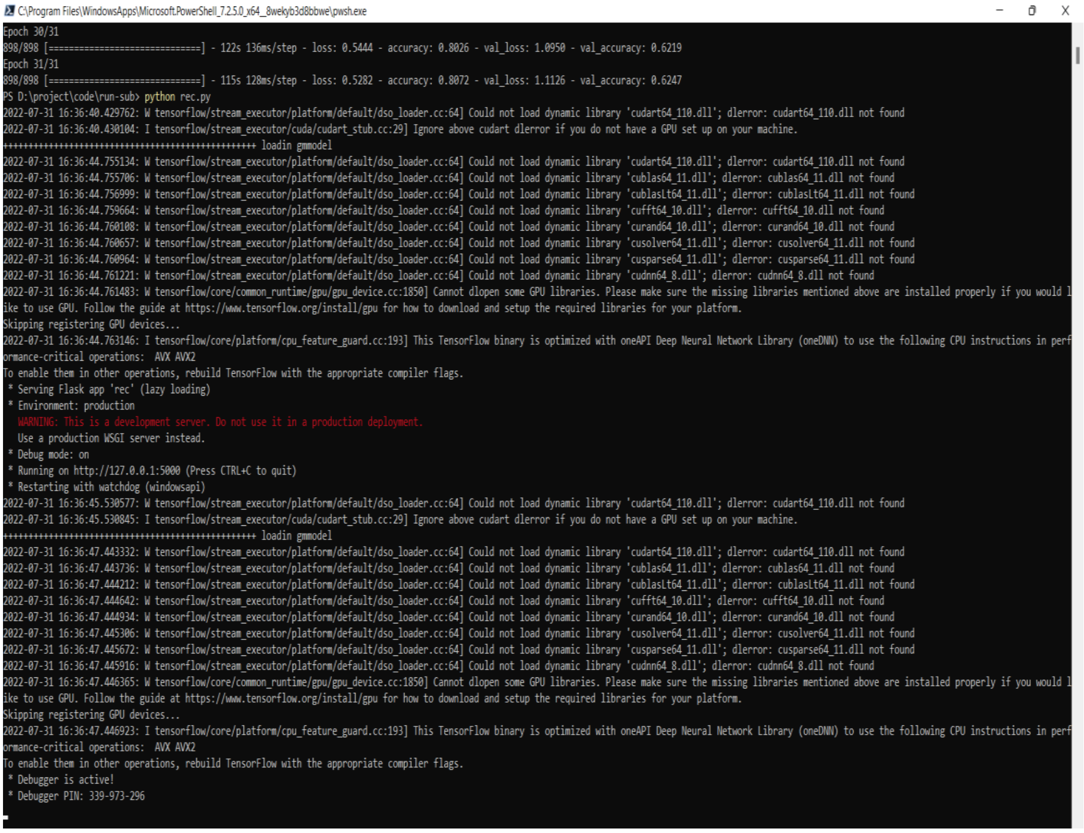

C:\Program Files\WindowsApps\Microsoft.PowerShell_7.2.5.0_x64__8wekyb3d8bbwe\pwsh.exe

```
PowerShell 7.2.5
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Windows\System32> cd D:\project\code\run-sub
PS D:\project\code\run-sub> python train_test.py
2022-07-31 15:32:20.525660: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-07-31 15:32:20.525935: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
Found 7178 images belonging to 7 classes.
Found 28709 images belonging to 7 classes.
2022-07-31 15:32:27.079264: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-07-31 15:32:27.080153: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2022-07-31 15:32:27.080481: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2022-07-31 15:32:27.080764: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2022-07-31 15:32:27.081036: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2022-07-31 15:32:27.081293: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2022-07-31 15:32:27.081544: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2022-07-31 15:32:27.081797: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2022-07-31 15:32:27.082060: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2022-07-31 15:32:27.086309: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
C:\Users\manus\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\optimizers\optimizer_v2\adam.py:110: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(Adam, self).__init__(name, **kwargs)
D:\project\code\run-sub\train_test.py:55: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model_info = model.fit_generator(
Epoch 1/31
898/898 [==============================] - 116s 129ms/step - loss: 1.7625 - accuracy: 0.2833 - val_loss: 1.6113 - val_accuracy: 0.3848
Epoch 2/31
898/898 [==============================] - 112s 124ms/step - loss: 1.5527 - accuracy: 0.3994 - val_loss: 1.4997 - val_accuracy: 0.4178
Epoch 3/31
898/898 [==============================] - 112s 124ms/step - loss: 1.4486 - accuracy: 0.4436 - val_loss: 1.3806 - val_accuracy: 0.4792
Epoch 4/31
898/898 [==============================] - 112s 124ms/step - loss: 1.3735 - accuracy: 0.4739 - val_loss: 1.3270 - val_accuracy: 0.4961
Epoch 5/31
898/898 [==============================] - 110s 123ms/step - loss: 1.3103 - accuracy: 0.5025 - val_loss: 1.2680 - val_accuracy: 0.5206
Epoch 6/31
898/898 [==============================] - 111s 124ms/step - loss: 1.2557 - accuracy: 0.5233 - val_loss: 1.2339 - val_accuracy: 0.5314
Epoch 7/31
898/898 [==============================] - 113s 126ms/step - loss: 1.2123 - accuracy: 0.5384 - val_loss: 1.1938 - val_accuracy: 0.5439
Epoch 8/31
898/898 [==============================] - 123s 136ms/step - loss: 1.1757 - accuracy: 0.5582 - val_loss: 1.1862 - val_accuracy: 0.5470
Epoch 9/31
898/898 [==============================] - 118s 131ms/step - loss: 1.1394 - accuracy: 0.5735 - val_loss: 1.1483 - val_accuracy: 0.5671
Epoch 10/31
```


C:\Program Files\WindowsApps\Microsoft.PowerShell_7.2.5.0_x64__8wekyb3d8bbwe\pwsh.exe

```
898/898 [==============================] - 123s 136ms/step - loss: 1.1757 - accuracy: 0.5582 - val_loss: 1.1862 - val_accuracy: 0.5470
Epoch 9/31
898/898 [==============================] - 118s 131ms/step - loss: 1.1394 - accuracy: 0.5735 - val_loss: 1.1483 - val_accuracy: 0.5671
Epoch 10/31
898/898 [==============================] - 119s 132ms/step - loss: 1.1046 - accuracy: 0.5851 - val_loss: 1.1553 - val_accuracy: 0.5642
Epoch 11/31
898/898 [==============================] - 124s 138ms/step - loss: 1.0709 - accuracy: 0.5993 - val_loss: 1.1281 - val_accuracy: 0.5759
Epoch 12/31
898/898 [==============================] - 120s 134ms/step - loss: 1.0435 - accuracy: 0.6106 - val_loss: 1.1167 - val_accuracy: 0.5777
Epoch 13/31
898/898 [==============================] - 114s 127ms/step - loss: 1.0162 - accuracy: 0.6218 - val_loss: 1.1068 - val_accuracy: 0.5798
Epoch 14/31
898/898 [==============================] - 112s 124ms/step - loss: 0.9896 - accuracy: 0.6313 - val_loss: 1.0763 - val_accuracy: 0.5991
Epoch 15/31
898/898 [==============================] - 128s 143ms/step - loss: 0.9564 - accuracy: 0.6450 - val_loss: 1.0808 - val_accuracy: 0.5984
Epoch 16/31
898/898 [==============================] - 122s 135ms/step - loss: 0.9267 - accuracy: 0.6560 - val_loss: 1.0776 - val_accuracy: 0.6003
Epoch 17/31
898/898 [==============================] - 117s 131ms/step - loss: 0.8988 - accuracy: 0.6650 - val_loss: 1.0624 - val_accuracy: 0.6041
Epoch 18/31
898/898 [==============================] - 144s 161ms/step - loss: 0.8734 - accuracy: 0.6757 - val_loss: 1.0600 - val_accuracy: 0.6126
Epoch 19/31
898/898 [==============================] - 136s 151ms/step - loss: 0.8447 - accuracy: 0.6894 - val_loss: 1.0583 - val_accuracy: 0.6115
Epoch 20/31
898/898 [==============================] - 130s 145ms/step - loss: 0.8208 - accuracy: 0.7002 - val_loss: 1.0537 - val_accuracy: 0.6161
Epoch 21/31
898/898 [==============================] - 130s 144ms/step - loss: 0.7831 - accuracy: 0.7104 - val_loss: 1.0686 - val_accuracy: 0.6165
Epoch 22/31
898/898 [==============================] - 143s 160ms/step - loss: 0.7564 - accuracy: 0.7227 - val_loss: 1.0558 - val_accuracy: 0.6138
Epoch 23/31
898/898 [==============================] - 148s 165ms/step - loss: 0.7254 - accuracy: 0.7356 - val_loss: 1.0680 - val_accuracy: 0.6189
Epoch 24/31
898/898 [==============================] - 111s 123ms/step - loss: 0.7093 - accuracy: 0.7423 - val_loss: 1.0626 - val_accuracy: 0.6189
Epoch 25/31
898/898 [==============================] - 107s 120ms/step - loss: 0.6734 - accuracy: 0.7537 - val_loss: 1.0660 - val_accuracy: 0.6155
Epoch 26/31
898/898 [==============================] - 107s 119ms/step - loss: 0.6514 - accuracy: 0.7625 - val_loss: 1.0713 - val_accuracy: 0.6244
Epoch 27/31
898/898 [==============================] - 107s 119ms/step - loss: 0.6168 - accuracy: 0.7772 - val_loss: 1.0812 - val_accuracy: 0.6212
Epoch 28/31
898/898 [==============================] - 108s 120ms/step - loss: 0.5965 - accuracy: 0.7824 - val_loss: 1.0861 - val_accuracy: 0.6204
Epoch 29/31
898/898 [==============================] - 114s 127ms/step - loss: 0.5750 - accuracy: 0.7916 - val_loss: 1.0856 - val_accuracy: 0.6221
Epoch 30/31
898/898 [==============================] - 122s 136ms/step - loss: 0.5444 - accuracy: 0.8026 - val_loss: 1.0950 - val_accuracy: 0.6219
Epoch 31/31
898/898 [==============================] - 115s 128ms/step - loss: 0.5282 - accuracy: 0.8072 - val_loss: 1.1126 - val_accuracy: 0.6247
PS D:\project\code\run-sub>
```

Training and testing of system Images


C:\Program Files\WindowsApps\Microsoft.PowerShell_7.2.5.0_x64__8wekyb3d8bbwe\pwsh.exe

```
Epoch 30/31
898/898 [==============================] - 122s 136ms/step - loss: 0.5444 - accuracy: 0.8026 - val_loss: 1.0950 - val_accuracy: 0.6219
Epoch 31/31
898/898 [==============================] - 115s 128ms/step - loss: 0.5282 - accuracy: 0.8072 - val_loss: 1.1126 - val_accuracy: 0.6247
PS D:\project\code\run-sub> python rec.py
2022-07-31 16:36:40.429762: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-07-31 16:36:40.430104: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
+++++++++++++++++++++++++++++++++++++++++++++++++++ loadin gmmodel
2022-07-31 16:36:44.755134: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-07-31 16:36:44.755706: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2022-07-31 16:36:44.756999: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2022-07-31 16:36:44.759664: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2022-07-31 16:36:44.760108: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2022-07-31 16:36:44.760657: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2022-07-31 16:36:44.760964: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2022-07-31 16:36:44.761221: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2022-07-31 16:36:44.761483: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2022-07-31 16:36:44.763146: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
 * Serving Flask app 'rec' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
 * Restarting with watchdog (windowsapi)
2022-07-31 16:36:45.530577: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-07-31 16:36:45.530845: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
+++++++++++++++++++++++++++++++++++++++++++++++++++ loadin gmmodel
2022-07-31 16:36:47.443332: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll not found
2022-07-31 16:36:47.443736: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2022-07-31 16:36:47.444212: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2022-07-31 16:36:47.444642: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2022-07-31 16:36:47.444934: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'curand64_10.dll'; dlerror: curand64_10.dll not found
2022-07-31 16:36:47.445306: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2022-07-31 16:36:47.445672: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2022-07-31 16:36:47.445916: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2022-07-31 16:36:47.446365: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would l
ike to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2022-07-31 16:36:47.446923: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in perf
ormance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
 * Debugger is active!
 * Debugger PIN: 339-973-296
```

Server Log Feedings Images

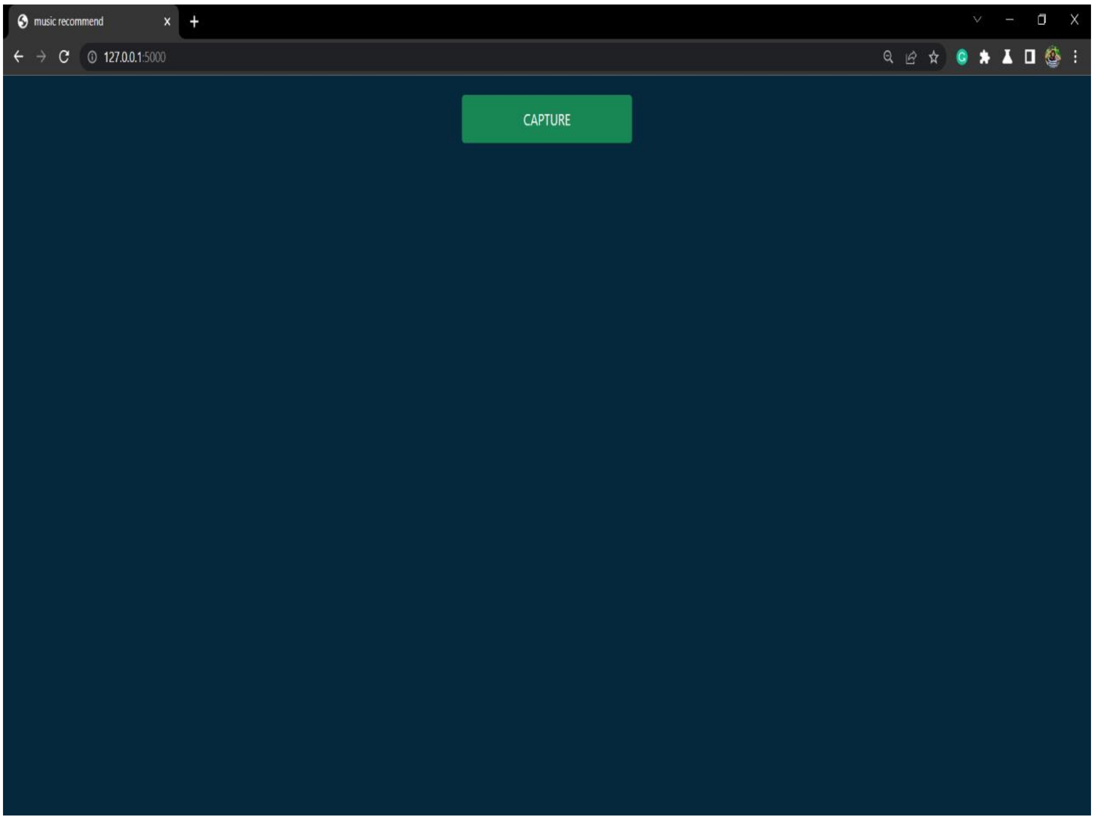
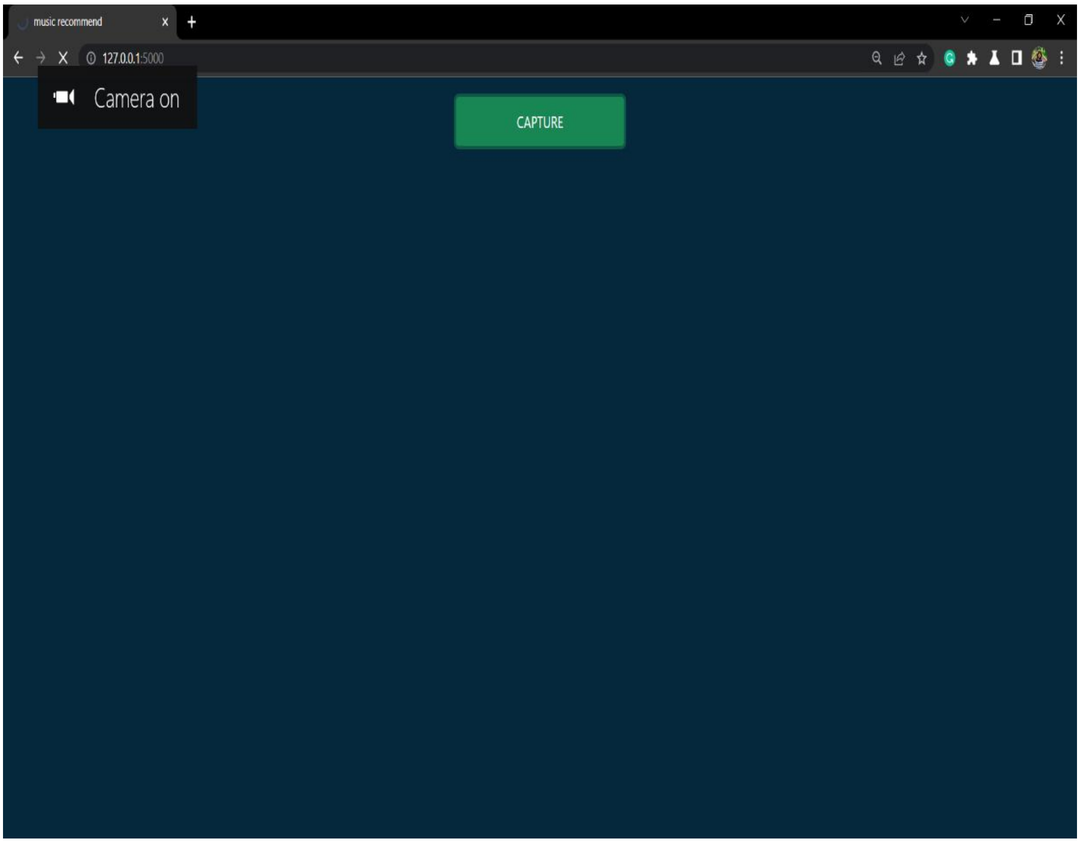
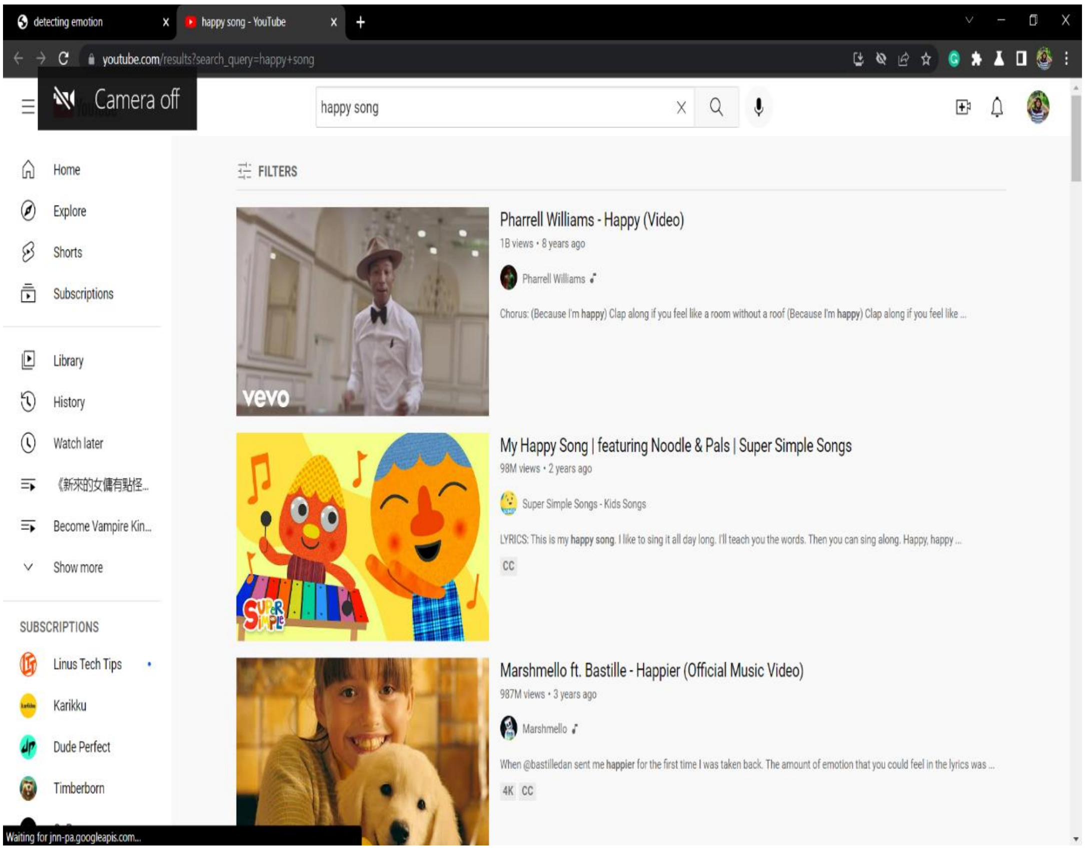


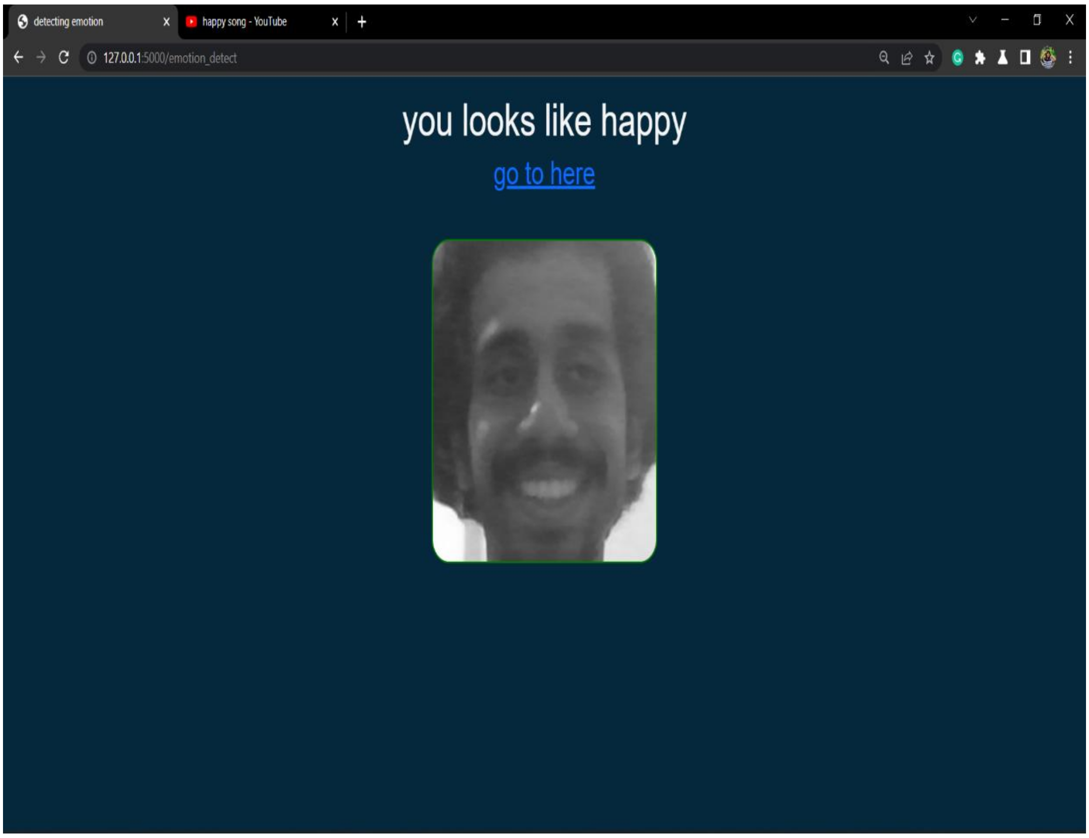
Image Capture Interface Images



Camera On Image


detecting emotion
happy song - YouTube
youtube.com/results?search_query=happy+song
Camera off
happy song
Home
Explore
Shorts
Subscriptions
Library
History
Watch later
《新來的女傭有點怪…
Become Vampire Kin...
Show more
SUBSCRIPTIONS
Linus Tech Tips
Karikku
Dude Perfect
Timberborn
FILTERS
Pharrell Williams - Happy (Video)
1B views • 8 years ago
Pharrell Williams
Chorus: (Because I'm happy) Clap along if you feel like a room without a roof (Because I'm happy) Clap along if you feel like ...
vevo
My Happy Song | featuring Noodle & Pals | Super Simple Songs
98M views • 2 years ago
Super Simple Songs - Kids Songs
LYRICS: This is my happy song. I like to sing it all day long. I'll teach you the words. Then you can sing along. Happy, happy ...
CC
Marshmello ft. Bastille - Happier (Official Music Video)
987M views • 3 years ago
Marshmello
When @bastilledan sent me happier for the first time I was taken back. The amount of emotion that you could feel in the lyrics was ...
4K CC
Waiting for jnn-pa.googleapis.com...

Final Output and Camera Off Image



you looks like happy

go to here

Captured Image and Detected Emotion