International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 06 | June - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Capture Your Emotions

Mohammad Sufiyaan Khan
B. Tech
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

M. Vaishnav
B. Tech
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

R. Vaishnavi
B. Tech
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

B. Vaishnavi
B. Tech
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

P. Vaishnavi
B. Tech
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

G. Vaishnavi
B. Tech
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

Guide: Siva Kumar
Asst Professor
School of Engineering
Computer Science – AI&ML
Malla Reddy University, India.

Abstract: In computer vision, one essential problem we are trying to figure out is to go automatically detect objects in an image without human intervention. Face detection can be thought of as a problem where we detect human faces in an image. In the field of Artificial Intelligence, Computer Vision is one of the most interesting and Challenging tasks. OpenCV is an open- source library. It is supported by various programming languages such as R, and Python. It runs on most platforms such as Windows, Linux, and MacOS. It is an image and video processing library and is used for image and video analysis. This project utilizes OpenCV Library to make Real-Time Face Detection using your webcam as a primary camera. The requirements are Python 2.7, OpenCV, NumPy, and Haar Cascade Frontal face classifiers. This project uses LBPH (Local Binary Patterns Histograms) Algorithm to detect faces. It labels an image's pixels by thresholding each pixel's neighborhood. The model built is trained with the faces with tags given to them, and later on, the machine is given test data and the machine decides the correct label for it. Perhaps, we will develop a real-time human face recognition model with python and OpenCV.

Keywords: Python, OpenCV, Haar Cascades, Numpy.

I. INTRODUCTION

Stepping into the realm of face emotion analysis, where our expressions unveil the depths of our feelings. Through advanced technology and AI algorithms, we decode the language of facial expressions, mapping them to a rich spectrum of emotions. This powerful tool opens doors to mental health assessments, customer feedback analysis, and more, revolutionizing human connection. Join us as we unravel the mysteries behind our smiles and tears, forging a future where empathy and understanding thrive.

While Facial Recognitions can be convenient and useful in many situations, they also have somelimitations, including:

a. Bias: Facial Recognition can be biased against certain groups, particularly people of color, women, and individuals with disabilities. This can lead to higher rates of misidentification and false arrests, particularly in law enforcement applications.

b. Ethical Concerns: The use of facial recognition technology raises ethical concerns,particularly in law enforcement and security applications. There are concerns about the potential for abuse, particularly when it comes to identifying individuals based on their political beliefs or other personal characteristics.

II. REQUIRED TOOLS

a) Software Requirements

- Programming Language: Programming language Python is used to build a Facial Recognition code.
- Integrated development Learning environment: An IDE such as Android Studio, or PyCharm may be used to write, debug, and test the code.
- Data storage Tools: Tools such as SQL databases or cloud storage services may be used to store and retrieve data for Face recognition.
- OpenCV: OpenCV (Open-Source Computer Vision Library) is a software library, which as the name suggests is open-sourced and uses machine learning capabilities to solve computer vision problems as

previously explained.

- Dlib: The dlib library contains an implementation of "deep metric learning" used to create face embeddings used in the actual recognition process.
- Haar Cascade Frontal, Face classifiers, Data set, Deep face, Tensor flow.

b) Hardware Requirements

- CPU: A modern multicore processor (e.g., Intel Core i5 or above) is typically sufficient for running face emotion recognition algorithms.
- GPU: Graphics Processing Units (GPUs) can significantly accelerate the performance of face emotion recognition tasks, especially when utilizing deep learning algorithms.
- RAM: Sufficient memory is crucial for efficient processing.
- Camera: A high-resolution camera capable of capturing clear facial images is essential for accurate face emotion recognition.
- Storage: Sufficient storage space is required to store datasets, models, and application data.
- Operating System: The face emotion recognition application can be developed and run on popular operating systems such as Windows, macOS, or Linux.

III. MODULES

There three major modules for the processing of face emotion recognition.

a) Localization Module

It is not possible to use motion information to locate a person. For this reason, we decided to use facial color as a first step in personal localization. Because skin color is notvery clearly defined in the RGB color space, we use the YUV color space. A nontypical imageis used for better illustration: non-facial skin color can be found as well. It is possible to eliminate "cold" colors. The resulting pixels are used as hypotheses for further calculations.

b) Classification Module

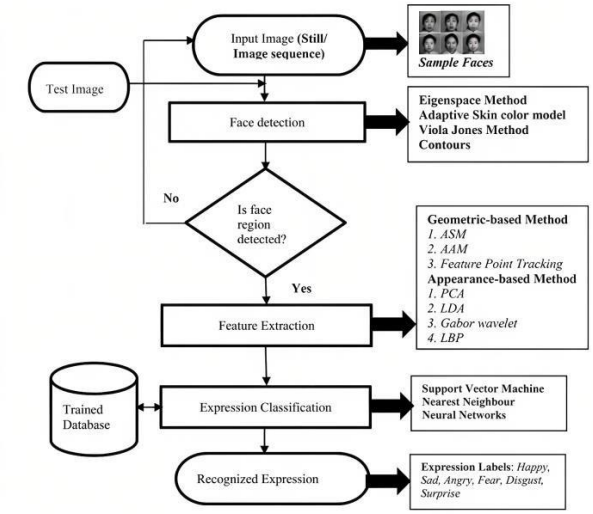
After the position of the face/the person is localized, the user state of this person can be classified. The classification is done by holistic classifiers as well. The method proposed by us for the recognition of facial expressions is a modification of a standard eigenspace classification for user identification. Eigenspace methods are well-known in the field of face recognition.

c) Combination and Sending Module

The classification step produces results for every capture frame, but a facial expression has a longer duration than a single frame. Normally, it starts with a neutral expression, then the expression of the user state evolves and goes back to the neutral expression. Therefore, the single classification results are combined in order to send fewer messages to the Pool thus after the expression is detected.

IV. ARCHITECTURE

Facial recognition technology uses a combination of computer vision and machine learning algorithms to identify and verify individuals based on their facial features.

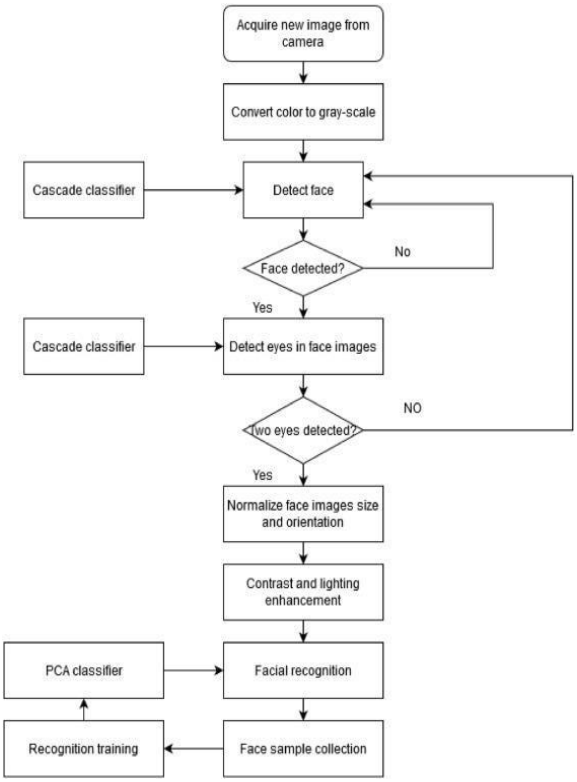


Here are the main components of the architecture of a facial recognition system:

a) Face detection: The system uses computer vision algorithms to detect and extract facesfrom images or video frames. This involves identifying regions of an image that are likely to contain faces and applying filters to isolate them.

b) Face alignment: Once a face is detected, the system may use algorithms to adjust the image so that the face is centered and aligned in a standard position. This makes it easier for subsequent processing steps to compare the features of

different faces.



c) Feature extraction: The system uses machine learning algorithms to extract unique features from each face, such as the distance between the eyes, the shape of the nose, and the contour of the jawline. These features are then used to create a mathematical representation of each face, often referred to as a "faceprint."

d) Matching: The system compares the faceprint of a new image to a database of known faceprints to find a match. This involves using algorithms to calculate the similarity between two faceprints and determining a threshold for a match. If the similarity scoreexceeds the threshold, the system identifies the person in the new image as a match.

e) Verification or Identification: Depending on the use case, the system can either verify the identity of a person by comparing their faceprint to a single reference image, or it can identify a person by searching a database of faceprints for a match.

f) Decision: Based on the output of the previous step, the system can then make a decision, such as allowing or denying access to a secured area or triggering an alert if an unauthorized person is detected.

Overall, the architecture of a facial recognition system involves a combination of computer vision and machine learning techniques to detect, align, extract features, match, and identify or verify individuals based on their facial features.

V. DESIGN

DFD Diagram

Fig. DFD for Face Emotion Recognition

Designing a facial recognition system involves several key steps, including:

a) Identifying the purpose of the system: Before designing a facial recognition system, it'simportant to identify the specific use case and goals of the system. For example, is it intended for security purposes, or to streamline the customer check-in process at a hotel or airport? Understanding the purpose of the system will help determine the necessaryfeatures and functionality.

b) Selecting appropriate hardware: The hardware used in the system, such as cameras andprocessing units, must be capable of capturing high-quality images and processing large amounts of data in real-time. The selection of hardware will depend on factors such asthe environment in which the system will be deployed, the required level of accuracy, and the intended usage.

c) Choosing facial recognition algorithms: The system's accuracy and performance rely heavily on the choice of facial recognition algorithms. The system should use reliable algorithms that are optimized for the specific use case, taking into account factors suchas lighting conditions, variations in facial expressions, and potential changes in appearance.

d) Collecting and storing data: The system needs a database of facial images to compare against. It is important to ensure that the data collected is representative of the population and that the data is stored securely and ethically.

e) Establishing a user interface: The user interface should be intuitive and easy to use, allowing users to interact with the system seamlessly. This may include designing a system for user enrollment, training, and monitoring.

f) Testing and validating the system: Before deploying the system, it's important to test and validate it to ensure it

performs accurately and reliably. Testing should involve a range of scenarios and use cases to ensure the system can handle real-world conditions.

g) Addressing ethical and privacy concerns: Facial recognition systems raise ethical and privacy concerns that must be addressed during the design process. Considerations should include obtaining consent from individuals whose data is collected, ensuring that data is kept secure, and implementing mechanisms to prevent bias and discrimination.

VI. RESULTS

Executing the above code after importing the necessary modules and packages we get thefollowing output:

This is a screen shot of the output of the emotion recognition showing the emotions of four pictures taken from google.

VII. CONCLUSION

In conclusion, the emotion classifier project aimed to develop a system capable of accurately identifying and classifying human emotions based on input such as facial expressions. Throughout the project, we implemented and evaluated different algorithms and techniques to achieve this goal.

We started by collecting a diverse and representative dataset of emotions, which served as the foundation for training and evaluating our models. We explored various approaches, including machine learning algorithms such as support vector machines (SVM), random forests, or deep learning architectures like convolutional neural networks (CNN) or recurrent neural networks (RNN).

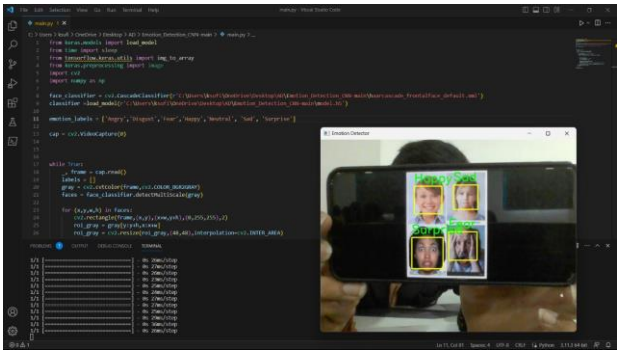
During the model development and evaluation phase, we carefully considered the input sources and their associated features. We evaluated the performance of our models using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, or confusion matrix. We also compared the results against baseline models or existing approaches to assess the effectiveness and improvement of our classifiers.

Based on the evaluation results, we observed that our emotion classifiers demonstrated promising performance in accurately predicting and classifying emotions across different inputmodalities. We achieved high accuracies and effectively captured various emotional states, allowing for better understanding and analysis of human emotions.

In conclusion, the emotion classifier project has made significant strides in accurately classifying human emotions across different input modalities.

ACKNOWLEDGMENT

We would like to express our gratitude to the project guide Prof.



Siva Kumar for their valuable guidance, support, and mentorship throughout the development ofthis project. Their vast knowledge and expertise, as well as their willingness to go the extra mile to ensure that we understood the concepts, have been instrumental in shaping our understanding of the subject matter and ensuring thesuccessful completion of this project.

Furthermore, We would like to acknowledge the Head of the Department, Dr. Thayyaba Khatoon, for their continuous encouragement and support throughout the project. Their constant motivation and willingness to provide resources have been pivotal in ensuring the success of this project.

REFERENCES

[1]. Code learnt and extracted from Github from a repository named Emotion_Detection_CNN by akmadan.
Github link: https://github.com/akmadan/Emotion_Detection_CNN

[2]. Machine learning code was executed in an online platform named Kaggle.

Kaggle link: https://www.kaggle.com/code/sufiyaankhan/face-expression-evaluation

[3]. Data set is taken from Kaggle.

Data Set Link - https://www.kaggle.com/jonathanoheix/face-expression- recognition-dataset

[4]. We used haarcascade_frontalface_default.xml file which was downloaded from github.

[5]. "Realtime Face Emotion Recognition" by DeeoLearning_by_PhDScholar

(https://www.youtube.com/watch?v=fkgpvkqcoJc&t=989s)

[6]. "Tensorflow Tutorial for Python in 10 Minutes" by Nicholas Renotte

(https://www.youtube.com/watch?v=6_2hzRopPbQ)