Performing Facial Expression For Command Execution Using OpenCV

VEDANT RAVAL

LABDHI GATHANI

CHAVDA MIHIR

Department of Information & Technology

V.V.P. Engineering College , Rajkot , India

Abstract: In human-computer interaction, sign language interpretation plays a vital role, and face detection become predominant in computer vision research. The primary goal of this proposed system is to create a system, which can identify facial detection to perform a particular comma Even though that there are numerous con-trolled weapon vehicles using commands from user or self-controlled that uses GPS and sensors, the requirement for gesture controlled weapon vehicles are other ascent for military purposes us wartime ,which is called as Unmanned ground vehicles (UGVs) . So we have designed and developed facial gesture recognition system in which live capture image is pre-processed for enhancing the face object in image from which specific feature need to be expected for speedy & accurate recognition of the facial gesture.

Keywords **: Python library (**Open CV, Tensor flow, Keras, Pandas, NumPy, Matplotlib) Python, Facial gesture, Pre – Processing Command execution.

INTRODUCTION

Nowadays performing basic commands on a system has become one of the fundamental parts of our daily lives and it is used by anyone and everyone. Generally functions like Shut down, restart, save, print, screen lock, and screen unlock. Facial gestures are termed as an artificial process in which gesture is executed with facial muscles and it differs from a facial expression which is a natural process depending on a situation. This gesture-based interaction performs commands through a web camera in which the camera detects the face and the developed system tries to extract facial gestures from the image rather than recognizing the face which would be different terminology. We will be achieving this through Image Processing and Keras library.

The goal of this research is to provide an easier human-machine interaction routine in which users can able to execute the command using facial gestures. The system technology dependencies are Open CV, Keras, and TensorFlow which would be libraries used for the efficient and speedy working of The system. With the aid of a regular web camera, a machine can detect and recognize a facial gesture.

The objective of this research is to provide an algorithm for the development of a system that can detect a facial gesture and perform a corresponding command. This algorithm must provide at least a 95% successful recognition rate, out of which less than 3% of the detected faces are false positives.

SYSTEM BLOCK DIAGRAM

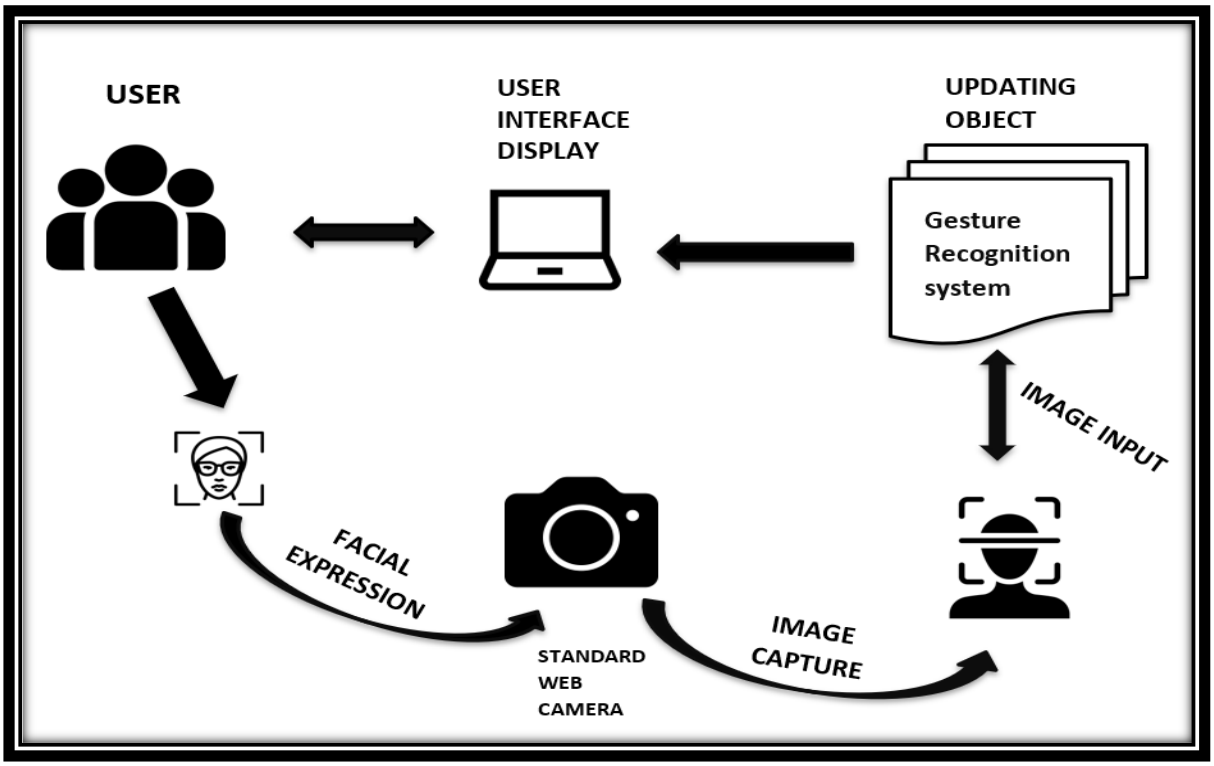
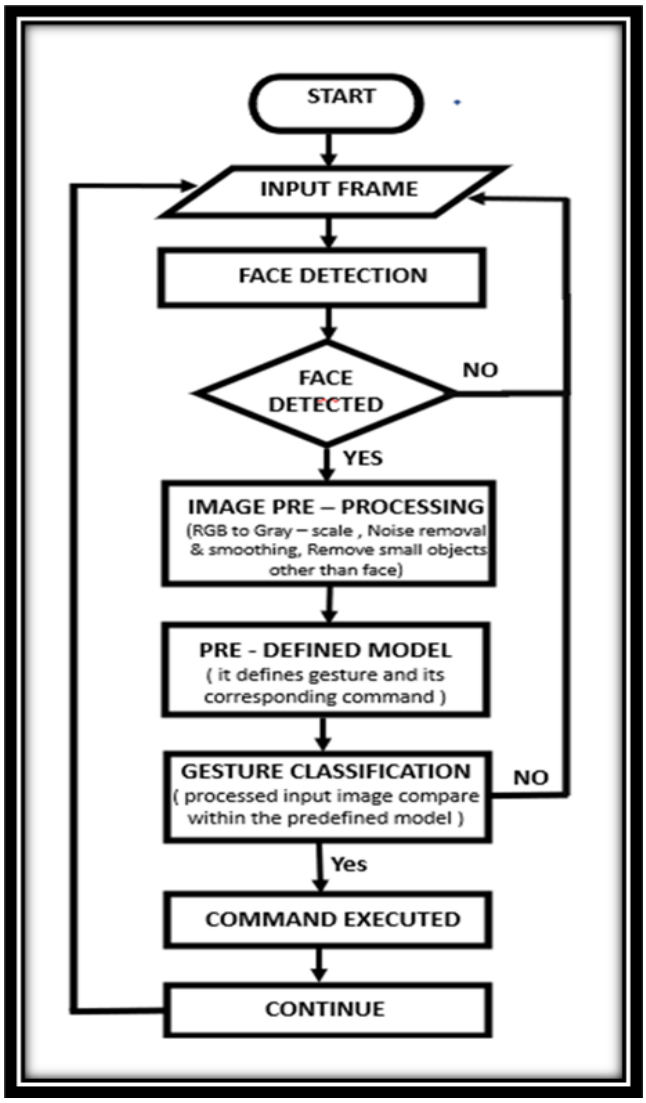


Fig. 1.1: SYSTEM BLOCK DIAGRAM

Figure 1.1 (System block diagram) represents the basic detection of the face, extraction of facial gestures performed by the user and the execution of the required commands. Initially, the user performs a facial gesture that is being recognized by the input frame (Camera of the system), now after recognition of the facial gesture by the input frame, this captured input is being processed by the system in which the image is converted from color band to grayscale, further carried out with processes such as binarizing, noise removal, smoothing and region filling. Eventually, if there is a defined command for the user given facial gesture then it will be directly executed by the system, so the user can experience the output of his/her defined commands through the user interface display. In the end, if the requirements for the execution are not matched, the process will be terminated and will start the process from the beginning.



ALGORITHM: Algorithm for…

1: Start

2: The input frame is activated and takes the input.

3: The system tries to detect the face.

4: If the face is detected then continue the process else start from step 2.

5: Image is pre-processed by the System.

6: The system compares the processed image with the Pre-trained data model.

7: If there exists a corresponding command for given input then the process is continued else start from step 2.

8: Command executed.

Fig 1.2 FLOWCHART

MATERIALS AND METHODS

The materials are as follows :

Open CV (Library): OpenCV (open-source computer vision library) is a library of programming functions mainly aimed at real-time computer vision originally developed by intel.

Tensor flow (Library): TensorFlow is an open-source library for fast numerical computing.

Keras (Library): Keras is one of the most powerful and easy-to-use python libraries.

Pandas (Library): Pandas is a speedy, strong, malleable and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib (Library): Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

NumPy (Library): NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices.

Python: Python is a powerful programming language with efficient high-level data structures and an approach to object-oriented programming.

PyCharm: PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

The methods are as follows :

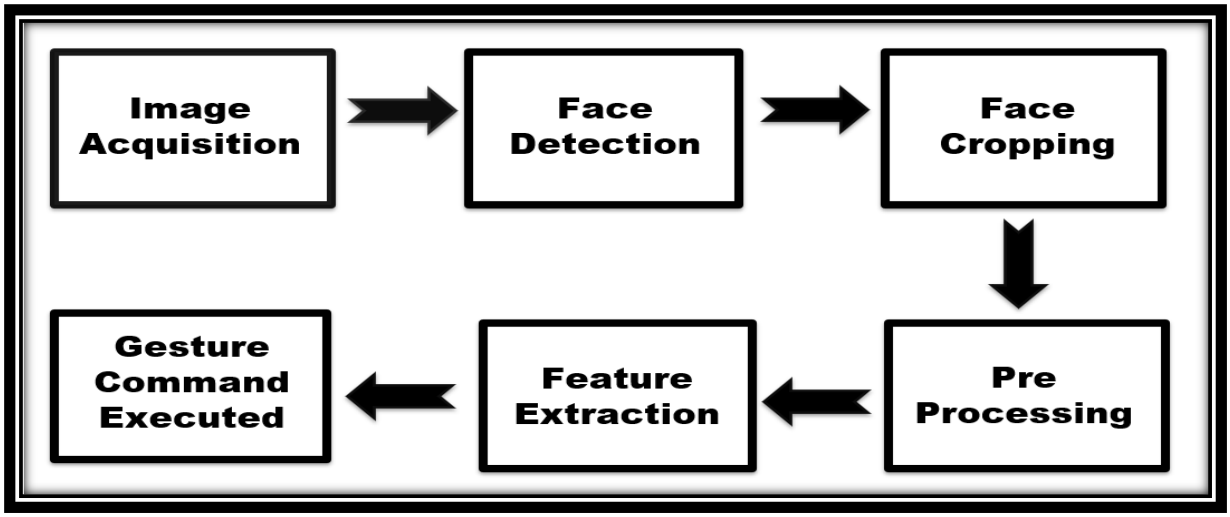


Fig 2.1 WORKING

(1) Image acquisition :

In this step, a GUI(Graphical User Interface), is made which shows the video stream of the scene. From that GUI when the system detects a face with some gesture it clicks the image directly. The problem is that this scene includes the whole body and other unwanted objects as well.

(2) Face Detection :

First of all colored image is read which is captured in the image acquisition step. Once we get the image, the dimensions of the image are calculated. The number of color bands should be on Then it will be processed further to the next step of crop face.

(3) Face Cropping :

Once the portion of the face is separated from the image, then the background face image is cropped out, for this certain threshold is used. Actually, in binarizing the image a threshold value is used, which only gives out the portion of the image with a cropped background, and then we can crop out the face. This image of the face is then stored and passed to the next phase.

(4) Pre–Processing :

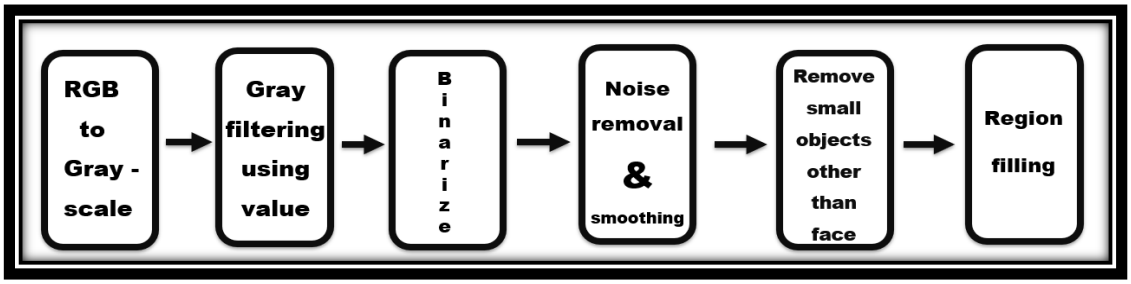


Fig 2.2 Pre -Processing

(5) Feature Extraction :

In this step features of each image are extracted using the method of the HU set of invariant moments and store the result for each image of the training set in a file so that during the classification step it need not be done again. The file contains a matrix having descriptor values of each image from the training dataset and its classifier class. It saves time and makes classification robust because the most time-consuming operation among most of these is training.

(6) Gesture command executed :

As per the above reference of process if every step is successfully performed then the corresponding gesture command gets executed.

Firstly, we will start with importing the required libraries (TensorFlow, Keras, Pandas, NumPy, Matplotlib, OpenCV). Now we insert the dataset(our dataset has 7 categories of expressions) into our python notebook on which our model will be trained and will validate how good or bad our model has performed on the particular dataset so that we can improve the accuracy score. Now after acquiring the dataset, we have to declare the path for the train folder, for that we will use the OS library of python. Now we will generate training and validation batches so that the model could be trained and validated on our test data, this will help us to get accuracy. After that, we have used an image data generator from the Keras module which will generate batches of image data in the dataset while comparing it with real-time data augmentation. Now after the verification and testing of the model we will run our VideoTester.py code in PyCharm. then we can finally detect the emotions through our webcam and we will be able to run the commands respective to the emotions.

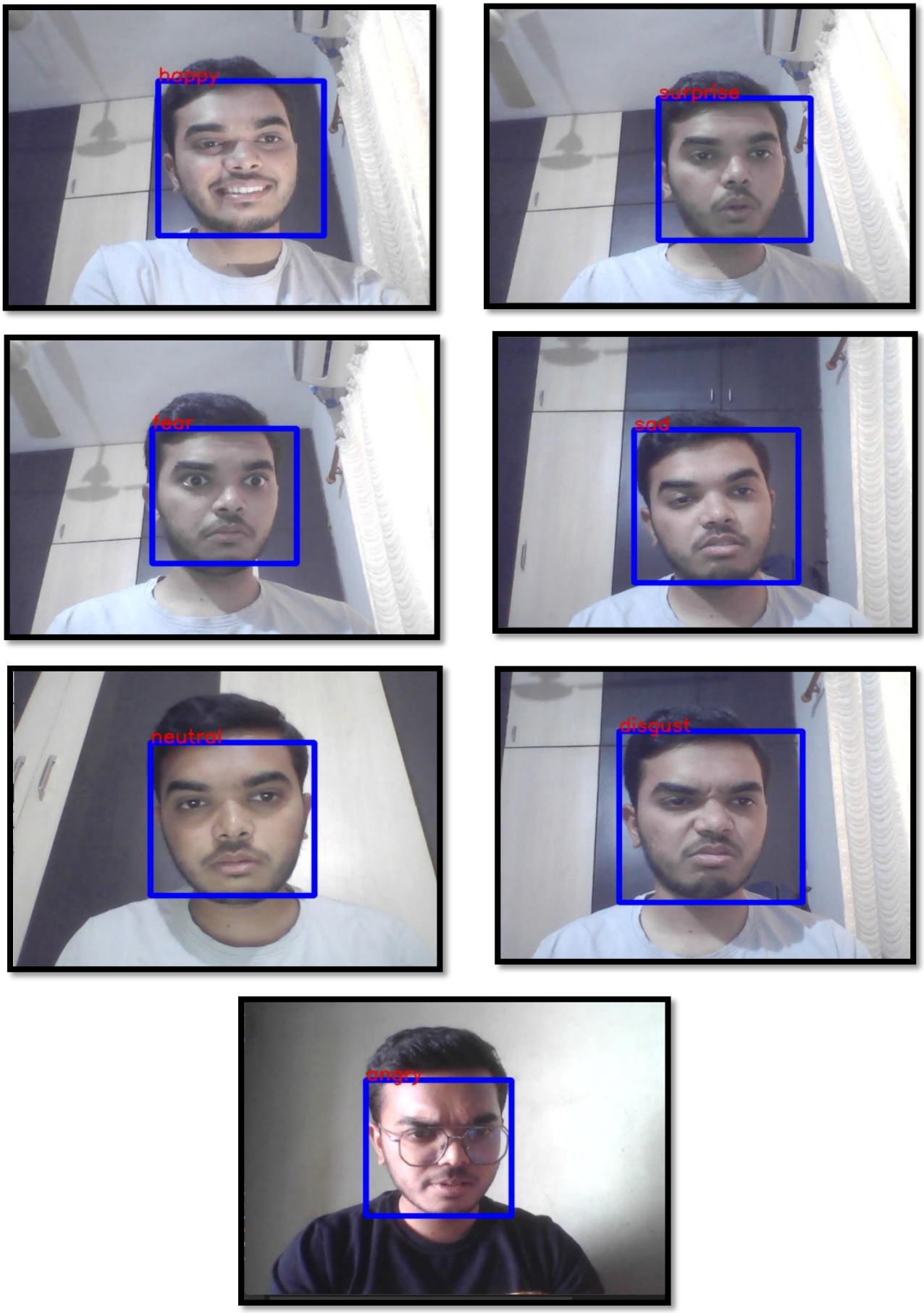


Fig 2.3 Facial Expression

CONCLUSION

The proposed system is to perform various command such as start, shut down , save etc. through facial gesture and face detection. This project is more efficient for situation where keyboard is unable such as in emergency or at LOC during war times and also use for non technical person to perform various basic technical commands very easily. This algorithm must provide at least a 95% successful recognition rate, out of which less than 3% of the detected faces are false positives. System would be speedy and sufficient reliable for good performance with complex background. The system successfully recognized static and dynamic gestures.
This system is applicable for wartime, deaf and dump people, non-technical person, military purposes, giving feedback, reducing accident by monitering the driver and monitering patient's emotions in hospitals.

REFERENCES

[1] Shervin Emami, Rotating or Resizing an Image in OpenCV, http://shervinemami.info/imageTransforms.html.

[2] Face Detection and Recognition using OpenCV, Article, http://shervinemami.info/faceRecognition.html, Published by Shervin Emami, 2010

[3]. ABBES Zeineb and CHIBANI Chaala "Hand Gesture Recognition System", International Journal of Computer Science and Information Security (IJCSIS). Vol. 14, No. 6, June 2016.