**PROJECT REPORT**

*On*

**Hand Gesture Identification**

(CST VI Semester Mini Project)

Submitted in partial fulfillment of the requirement for the VI semester

**Bachelor of Computer Science and Technology**

A picture containing text, scene, room, gambling house

Description automatically generated

***Guided By: Submitted By:***

**Mr. Ankit Tomer** Suhani Thapliyal

(Resource Person) Roll No.: 2015397

CST-VI-SEM

Session: 2021-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**Text

Description automatically generated with medium confidence**

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the dissertation entitled **“Hand gesture identification”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Technology, submitted in the Department of Computer Science and Engineering of the Graphic Era Deemed to be University, Dehradun is an authentic record of my own work carried out under the supervision of Mr. Ankit Tomer, Assistant Professor, Department of Computer Science and Engineering of the Graphic Era Deemed to be University, Dehradun (Uttarakhand).

The matter presented in this dissertation has not been submitted by me for the award of any other degree of this or any other Institute or University.

**Name- Suhani Thapliyal**

**Univ. Roll no.- 2015397**

**CST (VI Semester)**

**PROBLEM STATEMENT:**

Hand Gesture Identification

**SOFTWARES AND TOOLS/MODULES USED:**

* Language used- Python (version-3.8.2)
* OpenCv (version- 4.5.3)
* Numpy (version- 1.19.2)
* Math (version- 3.5)
* IDE used - Visual Studio Code

**MOTIVATION:**

The idea behind developing this project was to have experience with “gesture recognition and image processing”. Processing of an image consist of identifying an object in the image, identify the dimensions, number of objects, changing the images to blur effect and such effects are highly appreciated in modern era of media and communication. Hand gesture recognition is one of the system that can detect the gesture of hand in a real time video. The gesture of hand is classify within a certain area of interest.

Gesture recognition has been a very interesting problem in Computer Vision community for a long time. This is particularly due to the fact that segmentation of foreground object from a cluttered background is a challenging problem in real-time. The most obvious reason is because of the semantic gap involved when a human looks at an image and a computer looking at the same image. Humans can easily figure out what’s in an image but for a computer, images are just 3-dimensional matrices. It is because of this, computer vision problems remains a challenge.

My desire to learn the Python programming language and its related libraries served as my secondary motivation. The world's job market is driven by Python programming. It is the fastest-growing significant programming language without a doubt. As a result of its extensive library and module collection, it may be used for coding, app development, data science, machine learning, and other areas.

**METHODOLOGY AND ABOUT PROJECT:**

The project is developed in **Python** language. Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python consistently ranks as one of the most popular programming languages.

The term Computer Vision (CV) is used and heard very often in artificial intelligence (AI) and machine learning applications. The term essentially means giving a computer the ability to see the world as we humans do.

**Computer Vision** is a field of study which enables computers to replicate the human visual system. It’s a subset of artificial intelligence which collects information from digital images or videos and processes them to define the attributes. The entire process involves image acquiring, screening, analysing, identifying and extracting information. This extensive processing helps computers to understand any visual content and act on it accordingly.

Algorithms have been developed based on computer vision methods to detect hands using different types of cameras. The algorithms attempt to segment and detect hand features such as skin color, appearance, motion, skeleton, depth, 3D model, deep learn detection and more.  **Skin color detection** is one of the most popular methods for hand segmentation and is used in a wide range of applications, such as object classification, degraded photograph recovery, person movement tracking, video observation, HCI applications, facial recognition, hand segmentation and gesture identification.

**OpenCV** is an open-source library for the computer vision. It provides the facility to the machine to recognize objects. OpenCVis one of the most popular computer vision libraries. OpenCV was created to provide a shared infrastructure for applications for computer vision and to speed up the use of machine perception in consumer products. There are some predefined packages and libraries that make our life simple and OpenCV is one of them It was officially launched in 1999 by Intel. It was written in C/C++ in the early stage, but now it is commonly used in Python for the computer vision as well. The library has more than 2500 optimised algorithms, including an extensive collection of computer vision and machine learning algorithms. Using OpenCV it becomes easy to do complex tasks such as identify and recognise faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D object models, generate 3D point clouds from stereo cameras, stitch images together to generate an entire scene with a high resolution image and many more. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

There are many ways in which you can install OpenCV on your computer. One is by installing it via Anaconda and the other is to directly use pip command in command prompt of windows. I have used the latter.

OpenCV can be directly downloaded and installed with the use of pip (package manager). To install OpenCV, just go to the command-line and type the following command:

pip install opencv-python

**NumPy**, which stands for Numerical Python, is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for high-level mathematical functions to operate on these arrays. It is written partially in Python, but most of the parts that require fast computation are written in C or C++.

It contains various features including A powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities. Numpy is also used by other libraries such as OpenCV

NumPy provides data structures used to deploy OpenCV with Python.

The **math** module of python provides access to the mathematical functions defined by the C standard. Sometimes when working with some kind of financial or scientific projects it becomes necessary to implement mathematical calculations in the project. Python provides the **math module**to deal with such calculations. Math module provides functions to deal with both basic operations such as addition(+), subtraction(-), multiplication(\*), division(/) and advance operations like trigonometric, logarithmic, exponential functions. Math module also provides various the value of various constants like pi, tau.

**A brief understanding of my code is as follows:**

Hand gesture recognition system has developed excessively in the recent years, reason being its ability to cooperate with machine successfully. Gestures are considered as the most natural way for communication among human and PCs in virtual framework. We often use hand gestures to convey something as it is non-verbal communication which is free of expression. In this project, our PC's camera records a live video, from which a preview is taken with the assistance of its functionalities or activities.

First we capture the video and then define the region of interest. In the region of interest it converts all the BGR colors to HSV colors.

hsv = cv2.cvtColor(roi, cv2.COLOR\_BGR2HSV)

The **HSV** (which stands for **Hue Saturation Value**) scale provides a numerical readout of your image that corresponds to the color names contained therein. Hue is measured in degrees from 0 to 360. The Hue in HSV represents the color, Saturation in HSV represents the greyness, and Value in HSV represents the brightness. Whenever we want to solve problems related to object detection, it is necessary to use HSV and find the range of HSV. The Hue, Saturation, and Value in HSV have their own range of values. The Hue range in HSV is [0,179], the Saturation range in HSV is [0,255] and the Value range in HSV is [0,255]. There is also an upper bound and lower bound range for a range of each color in HSV. The HSV or Hue, Saturation, and value of a given object provide better performance when compared to RGB or Red, Blue, and Green color space and hence it is used widely in the area of computer vision.

After that I have defined the skin color range in HSV anything that is of skin color will be taken as one or white and any other color will be taken as zero or black. Then I have just dilated and blurred the image to decrease any amount of noise present in the image.

      lower\_skin = np.array([0,20,70], dtype=np.uint8)

      upper\_skin = np.array([20,255,255], dtype=np.uint8)

After that I have found **contours** in that image. **Contours** are just the outlines of any area that is being shown that region of interest. **Contours** are defined as the line joining all the points along the boundary of an image that are having the same intensity. Contours come handy in shape analysis, finding the size of the object of interest, and object detection. OpenCV has findContour() function that helps in extracting the contours from the image.

contours,hierarchy=cv2.findContours(mask,cv2.RETR\_TREE,cv2.CHAIN\_APPROX\_SIMPLE)

Then I have defined the **convex hull**. A convex hull of an object is the minimum boundary that can completely enclose or wrap the object(or contour of that object). Here, Convex hull is this green outline around my hand.

Then I have taken the area ratio, every gesture will have its own unique area ratio that is the basis on which gestures are differentiated.

After that **defects** in convex hull are to be found. Defects are regions that are not covered by the hand in the convex hull. Any deviation of the contour from the convex hull is known as convexity defect. OpenCV comes with a ready-made function to find this, cv.convexityDefects(). It takes as input the contour and corresponding hull indices and returns an array containing convexity defects as output. It returns an array where each row contains these values - [ start point, end point, farthest point, approximate distance to farthest point ].

hull = cv2.convexHull(approx, returnPoints=False)

defects = cv2.convexityDefects(approx, hull)

All the angles between our fingers are less than 90 degrees and the sides ones are greater than 90, so we count the no. of defects that are less than 90 degrees. With the help of cosine we find angle between defects and count the defects thereon.

angle = math.acos((b\*\*2 + c\*\*2 - a\*\*2)/(2\*b\*c)) \* 57

if angle <= 90 and d>30:

   l += 1

   cv2.circle(roi, far, 3, [255,0,0], -1)

So now I have the information for the number of defects in the convex hull and

also the area ratio. Now I have to just display what is in the box so number of defects plus 1 is the number of fingers in my region of interest.

 #print corresponding gestures which are in their ranges

        font = cv2.FONT\_HERSHEY\_SIMPLEX

        if l==1:

            if areacnt<2000:

                cv2.putText(frame,'Put your hand in the box',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

            else:

                if arearatio<12:

                    cv2.putText(frame,'Gesture - ZERO',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

                elif arearatio<17.5:

                    cv2.putText(frame,'Gesture - Best of luck',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

                else:

                    cv2.putText(frame,'Gesture - ONE',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

        elif l==2:

            cv2.putText(frame,'Gesture - TWO',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

        elif l==3:

              if arearatio<27:

                    cv2.putText(frame,'Gesture -THREE',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

              else:

                    cv2.putText(frame,'Gesture - OKAY',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

        elif l==4:

            cv2.putText(frame,'Gesture - FOUR',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

        elif l==5:

            cv2.putText(frame,'Gesture - FIVE',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

        elif l==6:

            cv2.putText(frame,'please re-position the hand',(0,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

        else :

            cv2.putText(frame,'please re-position the hand',(10,50), font, 2, (0,0,255), 3, cv2.LINE\_AA)

If there is nothing in the box or the area covered by hand is too less then I am displaying “put hand in the box”. We can differentiate three on the basis of area ratio covered to “okay” or “three”. If there are more than 6 defects then reposition the hand. This may be due to some error or lightning effects of the room.

Then I have just displayed both the python boxes and by pressing escape key the code execution ends.

**Screenshots of after successful compilation:**

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, website

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

**CONCLUSION:**

Developing recognition system which is efficient of working under different conditions is tough, but it is more possible because these hurdles exist in real-world environment. Hand gesture recognition is very significant for human-computer interaction. Gesture recognition appears in many real-life applications, e.g. gesture based gaming control, medical applications, controlling home appliances, in IoT devices, sign language translations and many more.

This project helped me learn new concepts and gain knowledge about them. Also I took assistance from the internet to help understand new topics. The knowledge of Python and its libraries which the project provided, would now enable me to quickly and efficiently write codes and help me in future.

**THANKYOU**