A

MINI PROJECT REPORT

ON

"CURSOR CONTROLLER"

Submitted in the partial fulfillment of the requirements in the 3th semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

Aditya Purswani-[1NH19IS190]

FOR

COURSE NAME :MINI PROJECT

COURSE CODE : 19ISE391

Under the guidance of,

Mrs. Divya K V

Sr. Asst. Professor,

Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NAAC with 'A' Grade & NBA)Ring Road, Bellandur Post, Near Marathahalli, Bengaluru-560103, INDIA www.newhorizonindia.edu

NEW HORIZON
COLLEGE OF ENGINEERING

CERTIFICATE

Certified that the project work entitled ……………………………………… carried out by Mr./Mrs ……………………….……………….., bearing USN………………..………, a bonafide student of 6th semester in partial fulfillment for the award of Bachelor of Engineering in Information Science & Engineering of the Visveswaraiah Technological University, Belagavi during the year 2018-19. It is certified that all corrections, suggestions indicated for Internal Assessment have been incorporated.The project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.

Name & Signature of Guide	Name & Signature of HOD	Name & Signature of Princip l
Mr.Gangadhar Immadi	Dr.Anandhii R.J.	Dr.Manjunatha

Examiners :

Name	Signature
1. ……………………………………….	………………………………..
2. ……………………………………….	…………………………………

DECLARATION

I hereby declare that I have followed the guidelines provided by the Institution in preparing the project report and presented report of the project titled “Cursor Controller”, and is uniquely prepared by us after the compilation of the project work. I also confirm that the report is only prepared for my academic requirement and the results embodied in this report have not been submitted to any other University or Institution for the award of any degree.

Signature of the Student:

Name: ADITYA PURSWANI
USN: 1NH19IS190

ABSTRACT

How it Works?
The program first tries to detect/captures the image or frame of the hand and tries to delete all possible noises in the background for better detection of hand(Also if there is a clear and dark background with suitable lighting condition its benifitial for the detection).

Then using the data aquired from the frame of hand to create a hull(convex hull) around the hand and contours functions to get the points along the boundary of the hand and using the hulls we find the defects(i.e. the gaps or valleys formed between the fingers) to calculate the number of fingers.

Then we use the brightest point on the screen for the movement of the cursor on the screen using the automation library of python (PyAutoGUI)

Some Tips:

1. Keep the background clear to when the masked image is created because we need the brightest image in order the move the cursor.

2. Show 5 fingers to start or stop the program.

3. To terminate the program press ctrl+C on your keyboard

.

Technology and Programming Languages: Python

ACKNOWLEDGEMENT

Any project is a task of great enormity and it cannot be accomplished by an individual without support and guidance. I am grateful to a number of individuals whose professional guidance and encouragement has made this project completion a reality.

I have a great pleasure in expressing my deep sense of gratitude to the beloved Chairman Dr. Mohan Manghnani for having provided me with a great infrastructure and well-furnished labs.

I take this opportunity to express my profound gratitude to the Principal Dr. Manjunatha for his constant support and management.

I am grateful to Dr. R J Anandhi, Professor and Head of Department of ISE, New Horizon College of Engineering, Bengaluru for her strong enforcement on perfection and quality during the course of my project work.

I would like to express my thanks to the guide Mrs. Divya K V, Senior Assistant Professor, Department of ISE, New Horizon College of Engineering, Bengaluru who has always guided me in detailed technical aspects throughout my project.

I would like to mention special thanks to all the Teaching and Non-Teaching staff members of Information Science and Engineering Department, New Horizon College of Engineering, Bengaluru for their invaluable support and guidance.

NAME: ADITYA PURSWANI
USN: 1NH19IS190

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Motivation of Project:

Images and videos are an essential part for understanding anything or everything, Every day we see things that are saved in our subconcious memory as images. Nowadays manipulation of images and videos is a very big market(be it graphics , editing, photoshop etc.) images/videos are an important part of everything.

Images are part of data that are used in software fields that are at a boom like Augmented reality/Virtual reality, Machine learning, Artificial Intelligence all of these are used as one or the other fragments of each of these.

The main motivation was that most of the population with a screen with camera(whether educated or uneducated) can use this as a useful object as everyone can understand and remember images.

But manipulating the image content is not that easy as we need to be well versed with the image manipulating algorithms.
Here in this project we are working on an application/subset of machine learning (i.e. Computer Vision) from which we are capturing data from the laptop as an image and manipulate that data so we can use it to control the cursor.

Till the program is running we are collecting the data of each and every moment(that are contours(points along the boundary of the hand) and convex hull and the defects in that hull.

These defects are used to calculate the number of fingers and also calculate the area of the hand shown and the angles formed in between the fingers.

Through this different number of fingers are assigned to different functions that we are going to perform on the screen by the cursor
.
This automation is done by another python library that is PyAutoGUI(also known as python's automation library)

We learn basics of image manipulation by this project.

1.2 Problem Statement:

The problem in front of us requires the following:

- Successful detection of the hand.
- Use of implementation of convex hull and contours to map the hand around the boundary.
- Ability to manipulate data using the computer vision library
- Triggering the program when we show 5 fingers.
- Different number of fingers are assigned with different functions that can be changed in the program depending on person to person using automation library.

Chapter 2: System Requirement

2.1 Software and Hardware used:

2.1.1 Software used:

- The Python 3 programming language was used for building this project.
- It is required that Python 3 is installed on the computer for the program to work.(VS Code is used)

2.1.2 Hardware used:

- Windows 10 (64-bit) - Intel i7 (8th gen)
- 1TB HDD and 256 SSD
- 8 GB RAM

Chapter 3: System Design

3.1Modules:

1. Computer Vision Library (Opencv/cv2):

An open source computer vision framework that allows us to process data and manipulate data.

Some Opencv functions used in the project:

1. cv2.VideoCapture(0): Captures the video frame when the program is triggered to true.
2. cv2.createBackgroundSubtractorMOG2(): Eliminates the noises from the background using MOG/KNN algorithms.
3. cv2.cvtColor(input_image, flag): For converting the colour of input image from RGB to some other form.
4. cv2.dilate(mask, kern, iterations=1): If in the matrix(kernel) if any one pixel is 1 the pixel surrounding it also changes to 1 increasing the surface area.
5. cv2.erode(mask, kern, iterations=1): A pixel in the original image will be considered as on if all the pixels in the matrix are 1.
6. cv2.GaussianBlur(mask, (5, 5), 100): A way to manipulate data for better detection of objects by blurring the background.
7. cv2.convexHull(cnt): Creates the convex hull around the hand by joining the end points with convex curves.
8. cv2.contourArea(hull): Area of the object according to the convex hull.
9. cv2.contourArea(cnt): Area of the object according to the contour functions. (Area along the complete boundary of the object.)
10. cv2.convexityDefects(approx, hull): Calculates the defects in the object with convex hull(here important to calculate the number of images)
11.cv2.findContours(mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE): To get_the contours(that are the points along the boundary of the object.)
12. cv2.destroyAllWindows(): Vanishes/Stops the working of the program.

2. The Automation Library(PyAutoGUI):

A collection of functions that allows us to automate the mouse, keyboard, screen etc all within this library.

Some PyAutoGUI(pag) functions used in the project:

1. pag.click(): Used to left click on the bars/apps to open or close them.
2. pag.moveTo(): Used to move the cursor according to the broghtest pixel on the screen.
3. pag.scroll(-100): To scroll (negative value means scroll down).
4. pag.scroll(100): To scroll (positive value means scroll up).

3.Numpy:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

4. Time:

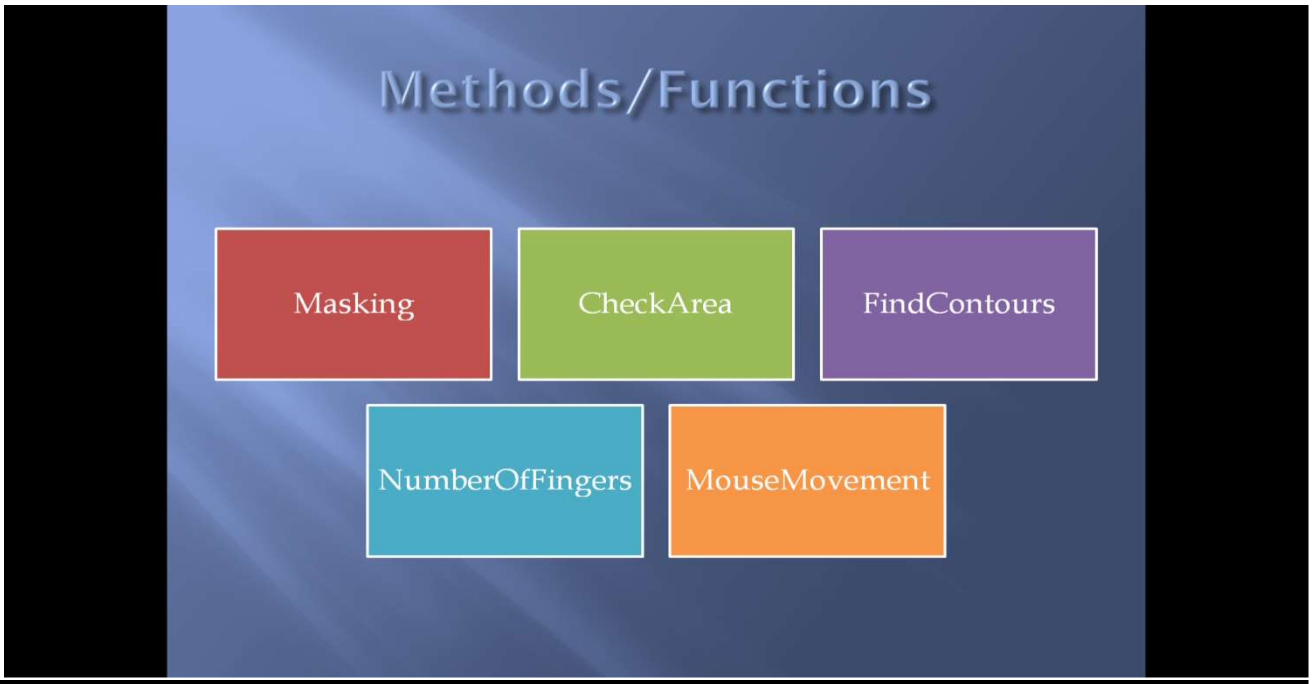
To know for how much time we are using the camera to capture the object. To pause the program between the execution of two lines.

Fig 3.1.1- Modules Logo

Fi

3.2 Architechture:

3.2.1 Program classes and methods:



Methods/Functions

Masking

CheckArea

FindContours

NumberOfFingers

MouseMovement

Fig3.2.1 Methods

The program contains the following functions:

1. Masking
2. CheckArea
3. FindContours
4. NumberOfFingers
5. MouseMovement

1.**Masking:** The image captured by the camera needed to be masked suitably for better detection of the object and also to reduce the noises in the background that can be anything(be it a person standing behind or it is a mere wallpaper.

This can be done by using the following functions:

1. cv2.cvtColor(croppedCam, cv2.COLOR_BGR2HSV)
2. cv2.inRange(hsv, LB, UB)
3. cv2.dilate(mask, kern, iterations=1)
4. cv2.erode(mask, kern, Iterations=1) cv2.GaussianBlur(mask, (5, 5), 100)
5. cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kern)

The resulting image of every function is used as the input image for the upcoming function.

2. CheckArea: When we place our hand in front of the camera and it aquires the data we need to calculate the area of the hand as the area will not be constant throughout and will be fluctuating so we calculate the area in order to find the brightest point on the screen and also the find if we show the fist and 1 finger.

3.FindContours: We find the contours for starting point, end point and far points along the whole boundary of the object(hand). Through these points on hand we calculate the angles between two fingers, area of the defects, and we also get the number of defects.

4.NumberOfFingers: Here we calculate the number of fingers from calculating the number of defects, angles and the surface length and we calculate the fingers by hit and trial of different angles and surface area through which we get the defects and also the fingers.
No. of fingers = No. of defects + 1

5. MouseMovement: Here we assign different functions to different number of fingers that are as follows:

0 Finger	: To LeftClick at a specific position on the screen
1 Finger	: To control the postition of the cursor according to brightest point
2 Fingers	: Can be assigned to the users need
3 Fingers	: To Scroll Down
4 Fingers	: To Scroll Up
5 Fingers	: To Start or Stop the Program

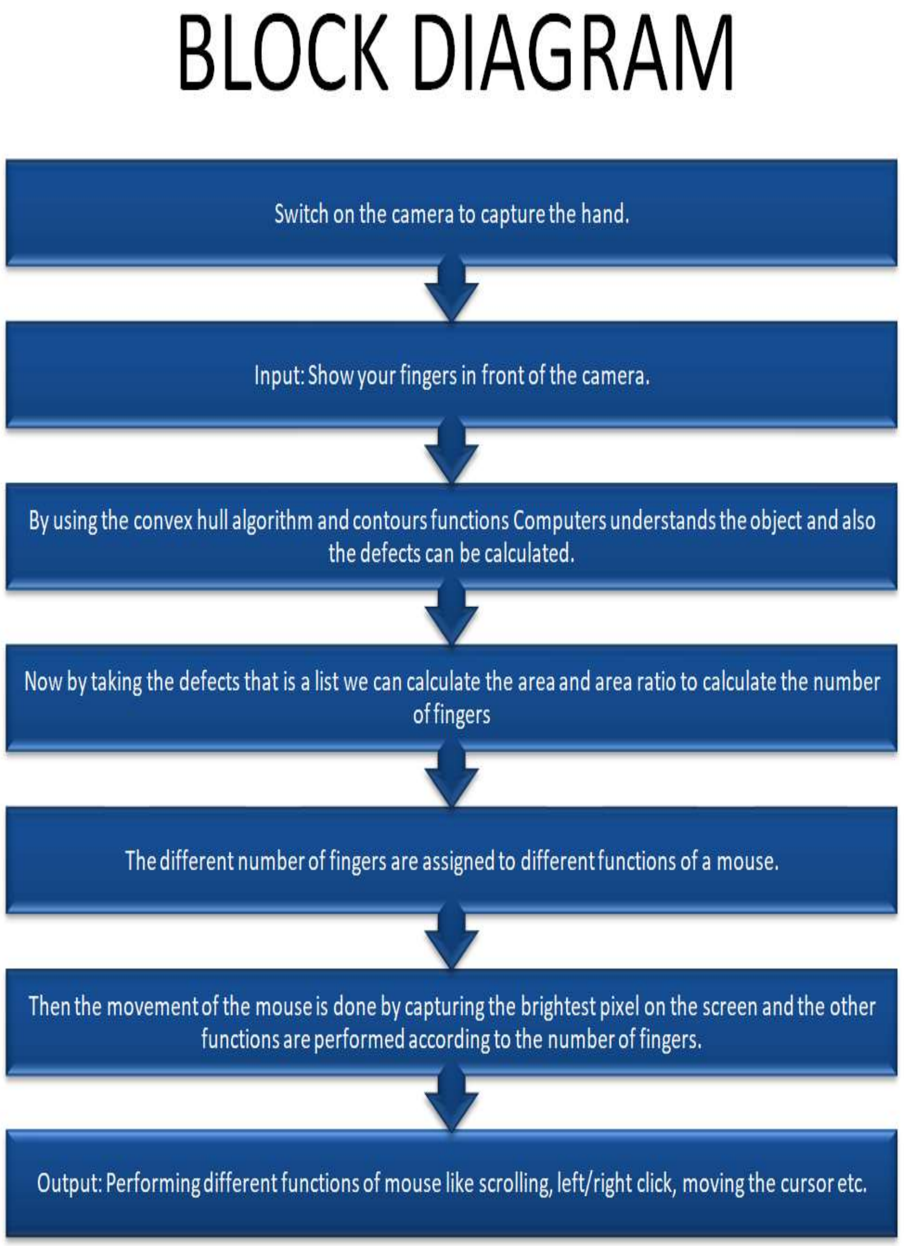

BLOCK DIAGRAM
Switch on the camera to capture the hand.
Input: Show your fingers in front of the camera.
By using the convex hull algorithm and contours functions Computers understands the object and also the defects can be calculated.
Now by taking the defects that is a list we can calculate the area and area ratio to calculate the number of fingers
The different number of fingers are assigned to different functions of a mouse.
Then the movement of the mouse is done by capturing the brightest pixel on the screen and the other functions are performed according to the number of fingers.
Output: Performing different functions of mouse like scrolling, left/right click, moving the cursor etc.

Fig 3.2.2 Work Flow

3.2.2 Methodology:

Aim: To minimize the use of hardware in daily life and to see how we can implement the data captured by the camera into one of the various ways.

Benefit: One of the benifit is that it minimizes the use of hardware in the day to day life. And people who dont know how to use computer can also get comfortable with using the gestures and can be able to use it.
Also we are learning manipulating images using computer vision

In the proposed project, an approach for Human and Computer Interaction (HCI) is presented, where an attempt is made to control the mouse cursor movement and click events of the mouse using hand gestures. Hand gestures were acquired using a camera based on colour detection technique. This method mainly focuses on the use of a Web Camera to develop a virtual human computer interaction device in a cost effective manner. The system is interacting with human's hand for performing mouse operations. The system serves as a block between Users and computer. It will capture the required feature with a webcam and it will monitor all its action in order to translate it to some events that will try to communicate or interact with the computer. In this proposed work a webcam is used that affords an average resolution and frame rate as the capturing device in order to make the ability of using the program affordable for all individuals.

The geometrical operation and integral image processes are performed on the input. It will check the file when there is movement detected. System will compare the coordinates captured from the frame of the webcam and perform the action i.e. it will allow the user to control the mouse operations using webcam based on the number of fingers shown in front of the web camera and performs the operations accordingly.

Capturing And Manipulating/Processing the image:

..Capturing

1. We capture the images through the webcam and if there is no movement in the webcam the frame remains the same.
2. Frame of the image changes when we move our hand continously in front of the camera or change the shape of our hand.
3. This image/frame is first converted into a 3 dimensional matrix in which the dimensions are height, width, depth of the colour.
4. The image that is converted into a matrix hepls us manipulate the image and reduce noises by erosion, dilation etc..
5. This matrix also helps us to find the brightest point in the frame as it contains the depth of the image as a parameter(also known as the intensity of the colour).

..Manipulating

>> The image is then manipulated in various ways:

1. By creating the convex hull around the object.
2. By getting the cordinates to calculate the area and number of fingers by the find contours functions.
3. By converting the image from RGB to greyscale.
4. Calculalting the number of defects.

..Assigning the Mouse Controls:

Through the puautogui library we can automate the processes like mouse controls, keyboard controls:

1. >>>Different number of fingers are assigned to different mouse controls that can be altered accordingly to the users preferences.
2. >>>Here if we show 5 fingers its use to trigger the program(i.e. first time if we show 5 fingers the program starts and if we show them again it stops)
3. So it works as a trigger for the program.

..Output:

>> Till the program is running in the background we can show our hand in the camera and perform the mouse controls as they are assigned by the programmer.

3.3 Algorithm Used:

3.3.1 Convex Hull Algorithms:

1. Graham Scan Algorithm:

- Find the point with smallest y-coordinate. If we have more than one coordinate with same minimum y coordinate,go on check the minimum x coordinate.

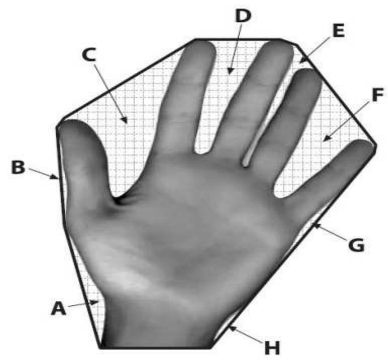
- Now sort the points according to there angles made by points with the x axis. While performing the sorting algorithm we dont calculate the angles, instead we calculate the orientation of the points in order to find out which point makes the larger angle. Sort the points with the angles made between points and x axis.

- The sorting algorithm has a time complexity of O(nlogn).

- After performing sorting, check for the collinear points, now from the collinear points the farthest points is taken in the convex hulls and the rest are ignored.

- The first two points on the sorted list are always considered in the convex hull.now we maintain the stack and push the two points in the stack Now we take two more points from the sorted list and check for the point that is on the leftmost from the line(i.e. making the largest angle) and the other point is popped out. Repeat the process until we have covered all the points in the list.

- We check if the point is towards the left or right of the line.If its on the left we push the point into stack and hat becomes a part of the convex hull, if its on the right then we moveto the next point/ coordinate on the list.



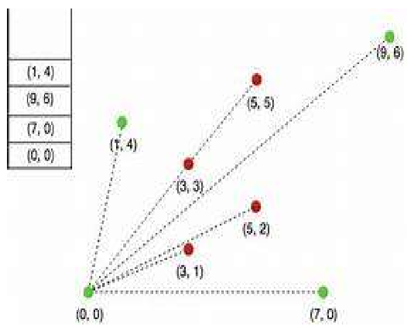


Fig 3.3.1 Convex Hull

2. Quickhull Algorithm:

Under some circumstances the algorithm works very well, but the program execution usually becomes slow in cases if there is more symmetry or points are lying on the edge of the circle. The algorithm can be written down to the following process:

1. The points with maximum/ minimum x coordinates will always be inthe convex hull but if there are many coordinates with the same x coordinates then look out for maximum/ minimum y coordinates.

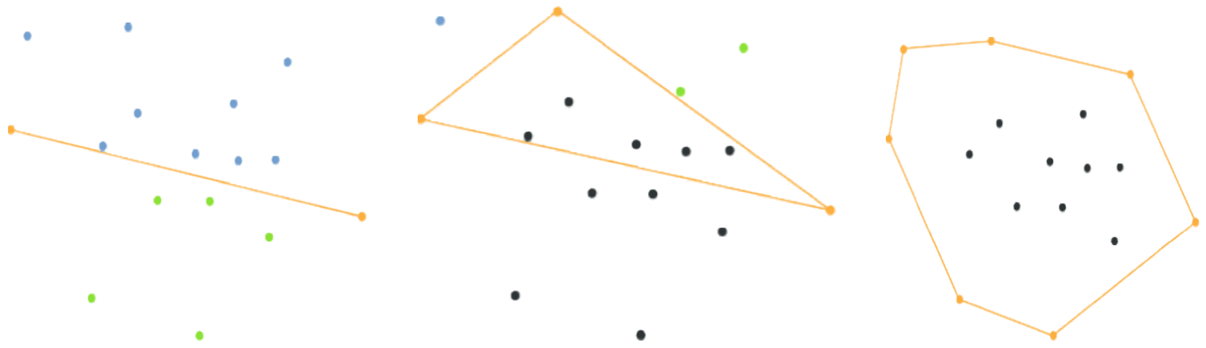
2. The line generated by the two points is used to divide the plane in two halfes which then will be processed recursively.

3. Now the next point that will be the part of the hull will be at the maximum distance from the line forming a triangle with the line.

4. Now the points that are lying in that triangle will never be considered in the convex hull and hence are ignored.

5. Now Repeat the 3rd and 4th steps on the other two lines of the triangle

6. Repeat the entire process until no more points are left to be used and recursive process is completed and we finally get a convex hull.



Step 1,2 Step 3 to 5 Step 6

Fig 3.3.2 Convex Hull

3.3.1 Program Algorithm:

1. Read and capture the image from the webcam.

2. Process the image to reduce the background noises by creating suitable masks

3. Image is converted into a n dimensional matrix.

4. Using this n dimentional matrix we get the coordinates along the boundary of the hand by contour functions.

5. From these coordinates we create the convex hull around the object using one of the convex hull algorithm (Here we use built in functions in computer vision library).

6. From the coordinates of convex hull and contours we detect the defects (i.e. the concavity defects along the hand.

7. Through these defects we can calculate the number of fingers by calculating the area and angles between thw fingers.

8. Once we know the number of fingers we then assign the mouse controls to them.

9. After we have assigned the mouse controls we can execute the program.

10. In the output we have to show our hands at a specific position on the camera and we can perform the mouse controls

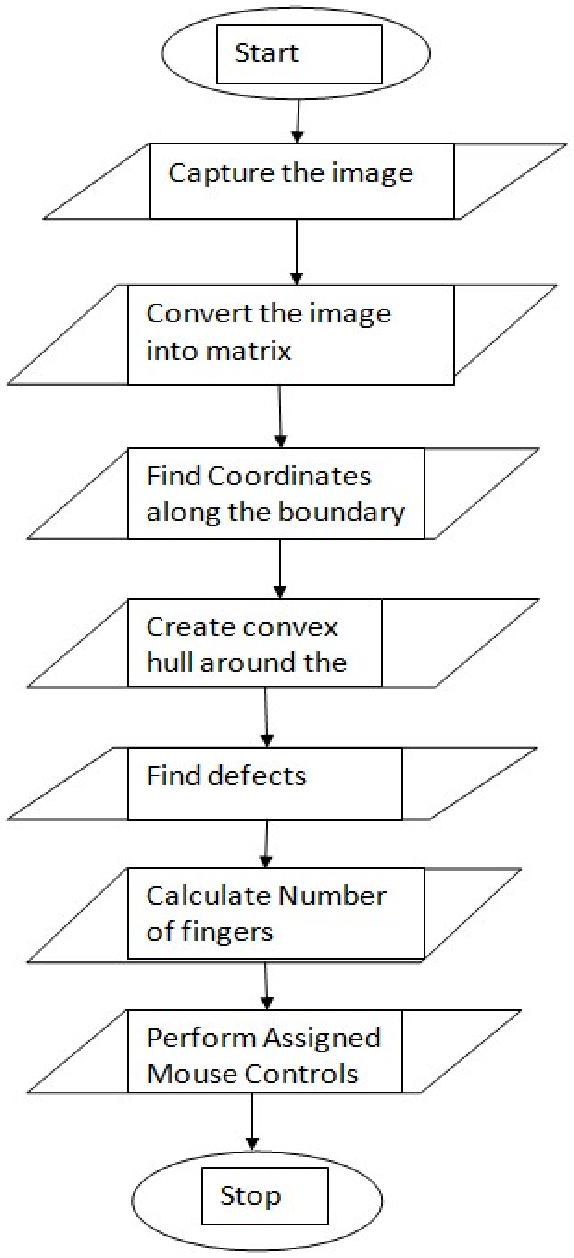

Start
Capture the image
Convert the image into matrix
Find Coordinates along the boundary
Create convex hull around the
Find defects
Calculate Number of fingers
Perform Assigned Mouse Controls
Stop

Fig 3.3.3 Flowchart

Controls Assignment flowchart:

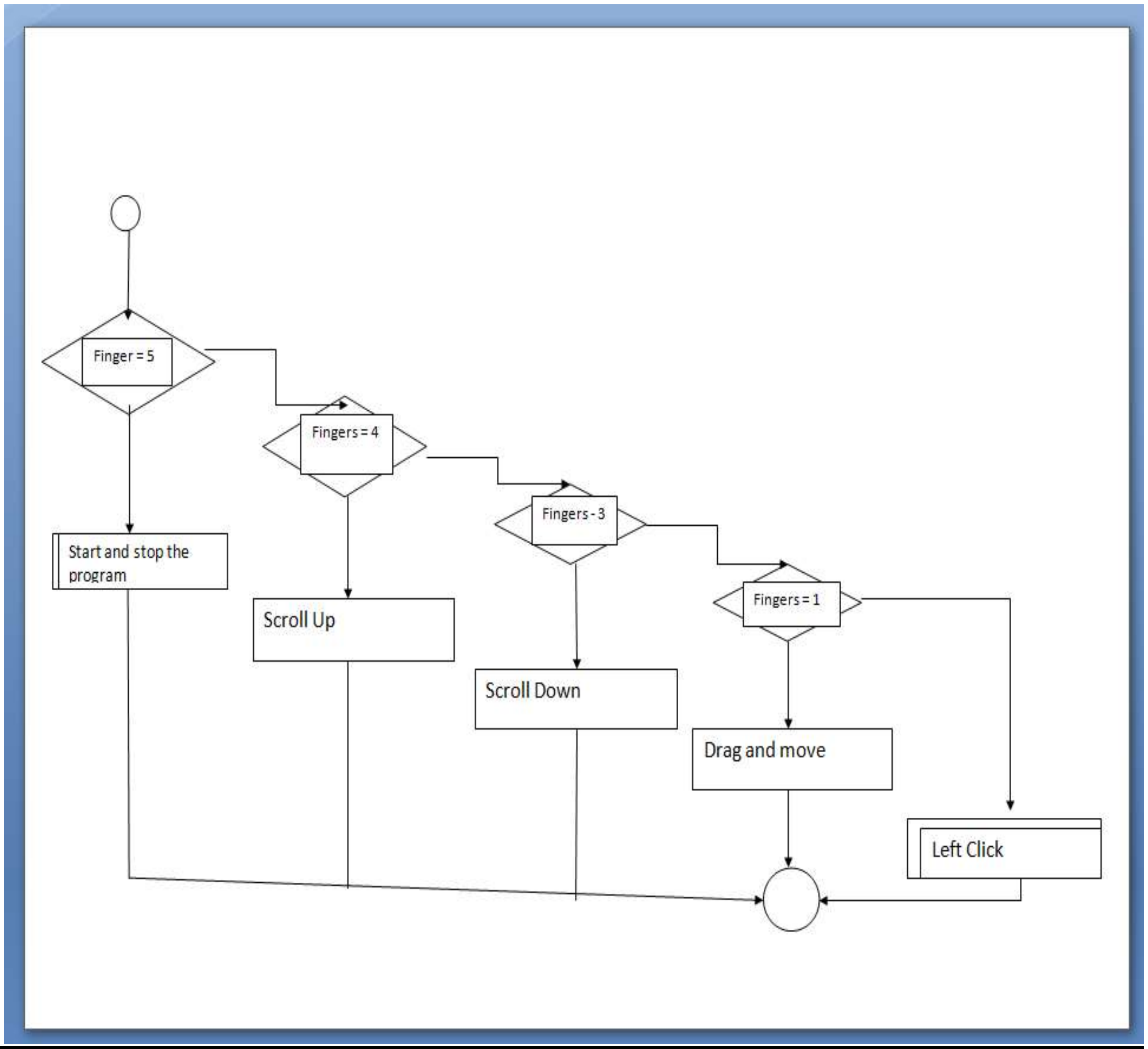
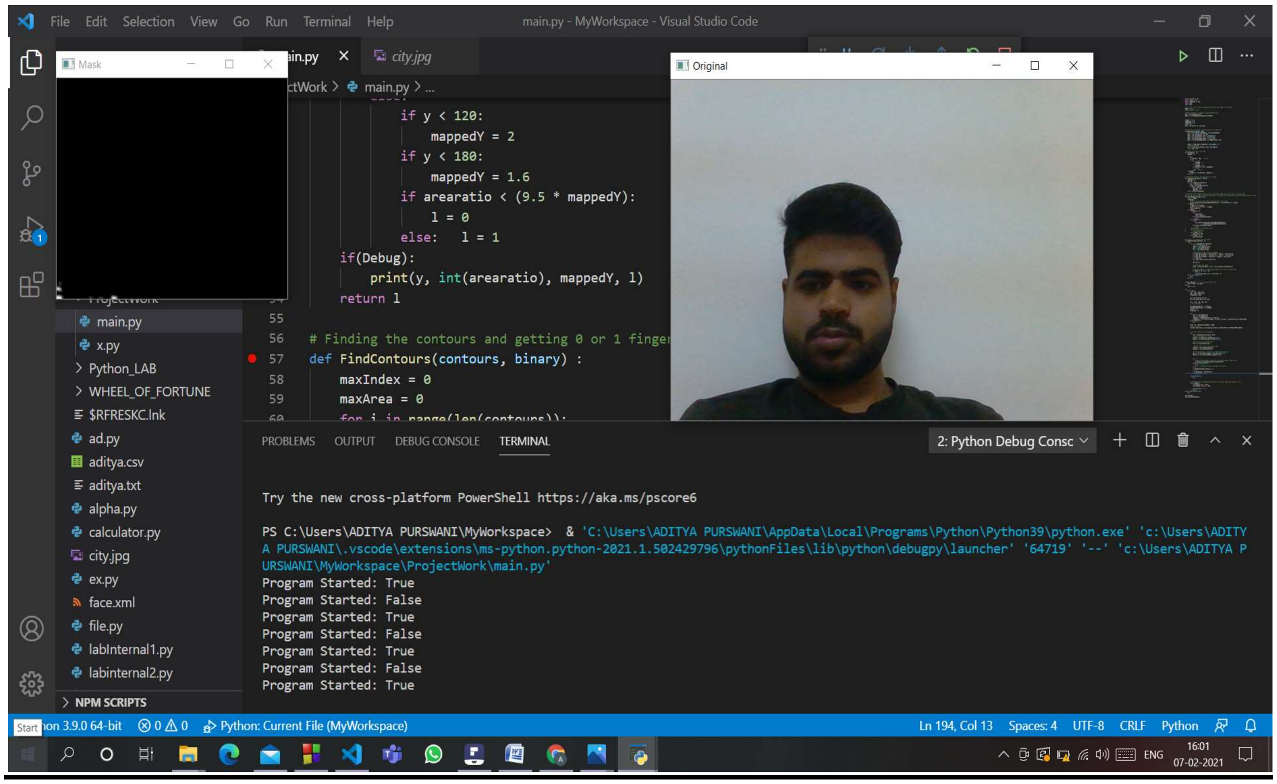


Fig 3.3.4 Flowchart

3.3.2 Example on Working of Algorithm:

Capturing the image:



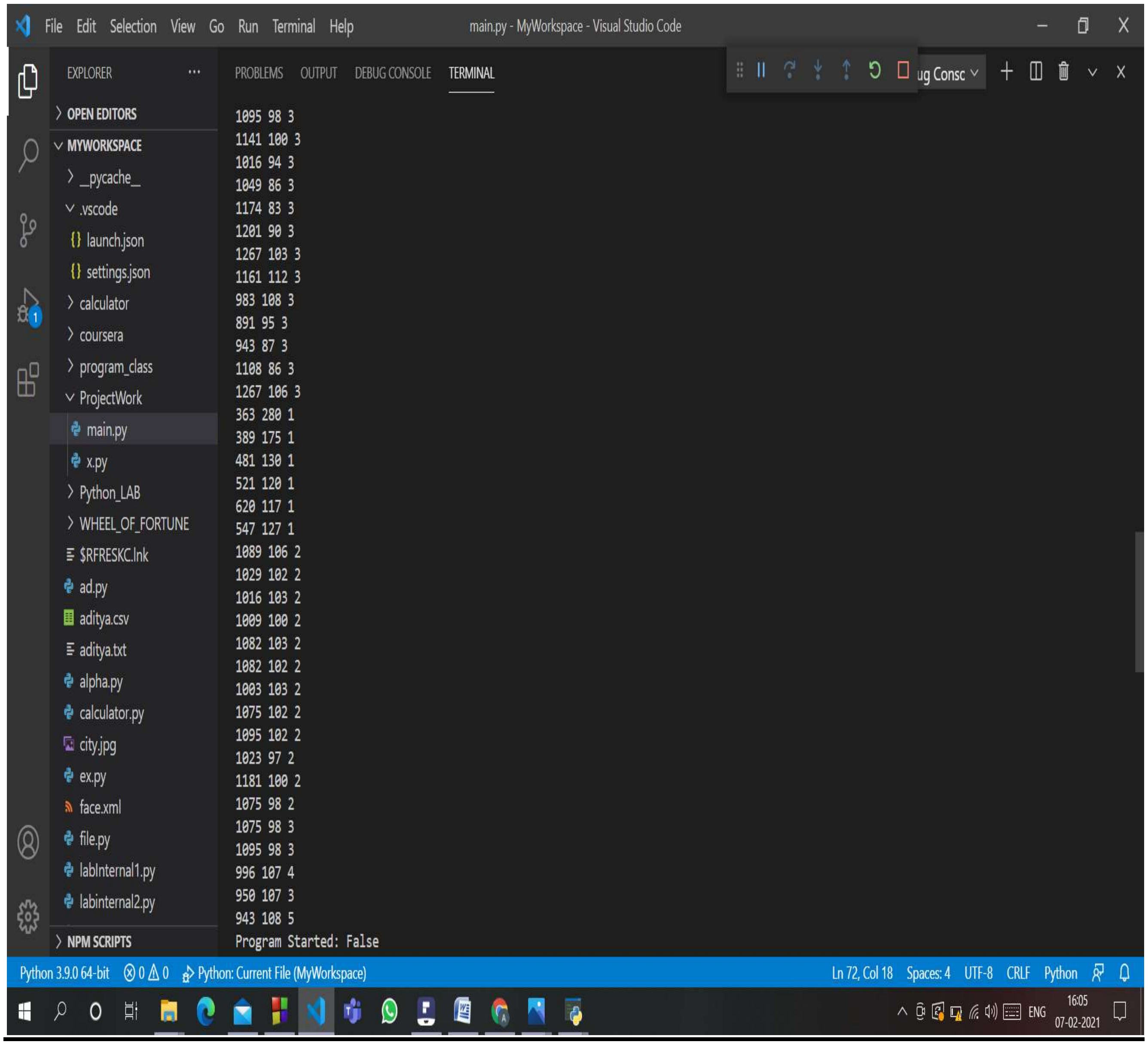
```
                if y < 120:
                    mappedY = 2
                if y < 180:
                    mappedY = 1.6
                if arearatio < (9.5 * mappedY):
                    l = 0
                else:   l = 1
        if(Debug):
            print(y, int(arearatio), mappedY, l)
        return l

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '64719' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
```

Fig 3.3.5(a) Example

Capturing the contours:

Once we have got to know the number of fingers we can then perform the operations as assigned



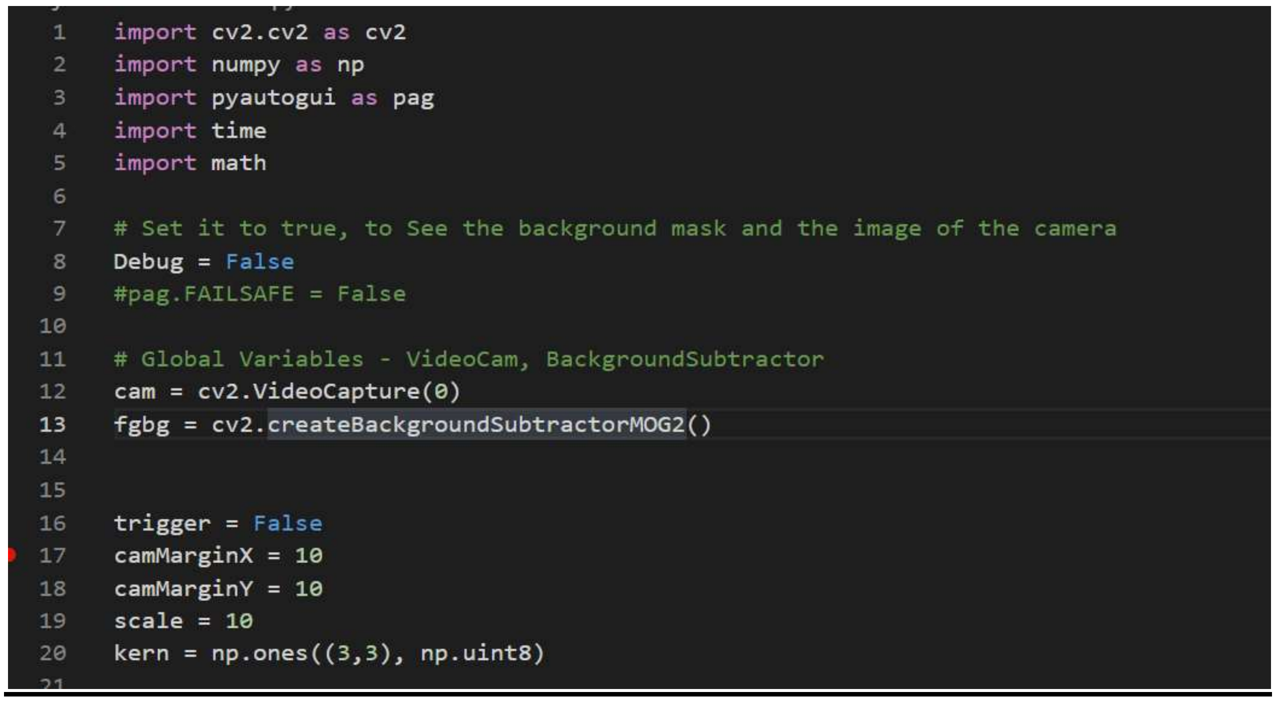
```
1095 98 3
1141 100 3
1016 94 3
1049 86 3
1174 83 3
1201 90 3
1267 103 3
1161 112 3
983 108 3
891 95 3
943 87 3
1108 86 3
1267 106 3
363 280 1
389 175 1
481 130 1
521 120 1
620 117 1
547 127 1
1089 106 2
1029 102 2
1016 103 2
1009 100 2
1082 103 2
1082 102 2
1003 103 2
1075 102 2
1095 102 2
1023 97 2
1181 100 2
1075 98 2
1075 98 3
1095 98 3
996 107 4
950 107 3
943 108 5
Program Started: False
```

Fig3.3.5(b) Example

- Here the third column of the list shows the number of fingers
- First column shows pixel that is the brightest.
- Second column shows the least brightest pixel.

3.4 Code And Implementation

1. Modules Used:

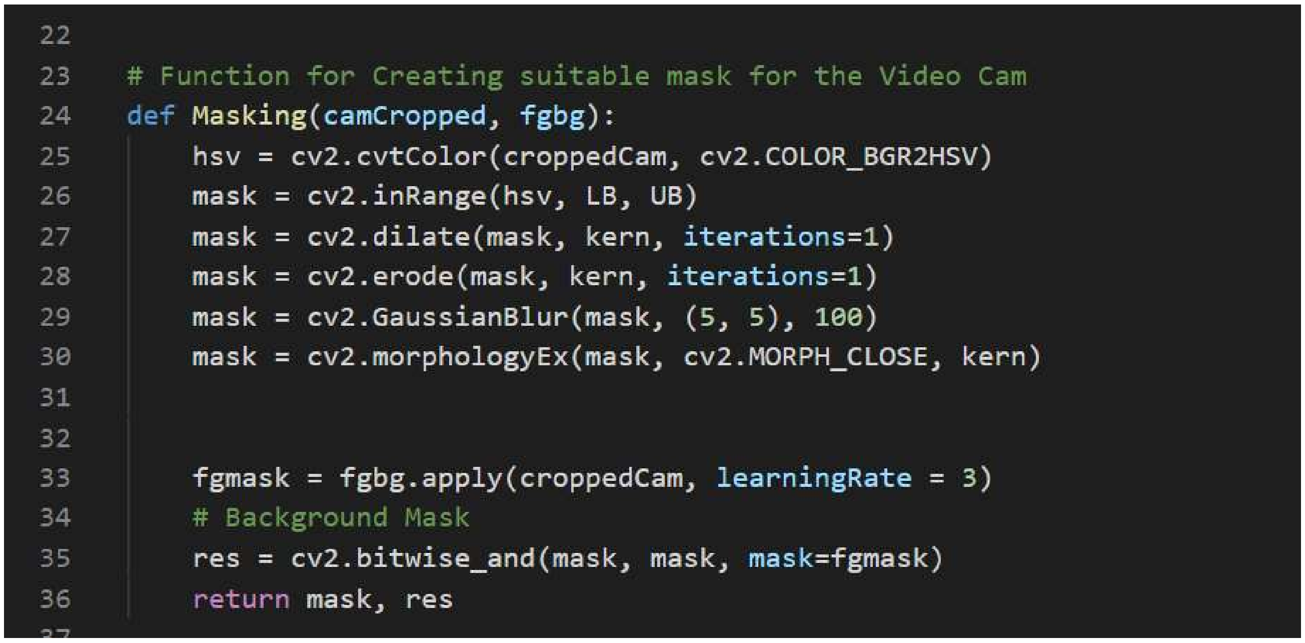


```python
import cv2.cv2 as cv2
import numpy as np
import pyautogui as pag
import time
import math

# Set it to true, to See the background mask and the image of the camera
Debug = False
#pag.FAILSAFE = False

# Global Variables - VideoCam, BackgroundSubtractor
cam = cv2.VideoCapture(0)
fgbg = cv2.createBackgroundSubtractorMOG2()


trigger = False
camMarginX = 10
camMarginY = 10
scale = 10
kern = np.ones((3,3), np.uint8)
```

Initializing/Importing the libraries(modules) that will come into use.

Also Initializing some variables that will be helpful in execution.

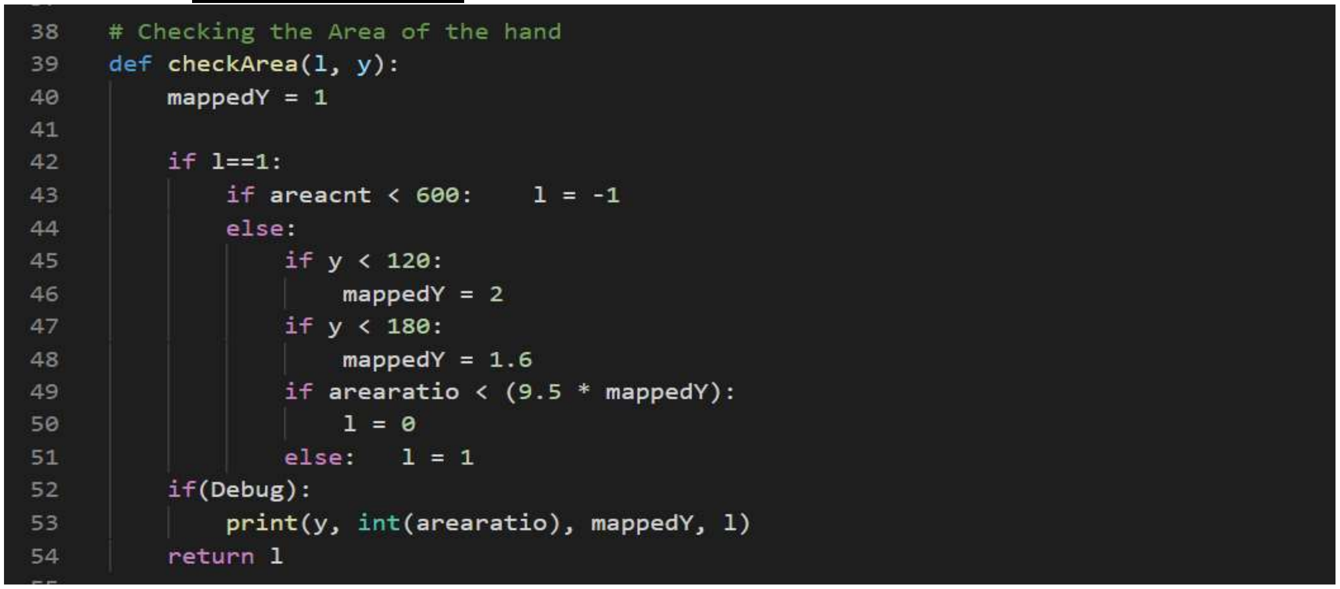
2.Creaing Suitable masks (Masking):



```python

# Function for Creating suitable mask for the Video Cam
def Masking(camCropped, fgbg):
    hsv = cv2.cvtColor(croppedCam, cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv, LB, UB)
    mask = cv2.dilate(mask, kern, iterations=1)
    mask = cv2.erode(mask, kern, iterations=1)
    mask = cv2.GaussianBlur(mask, (5, 5), 100)
    mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kern)


    fgmask = fgbg.apply(croppedCam, learningRate = 3)
    # Background Mask
    res = cv2.bitwise_and(mask, mask, mask=fgmask)
    return mask, res
```

Here we craete suitable masks for better detection of hand so that we face the least amount of noises in the background.

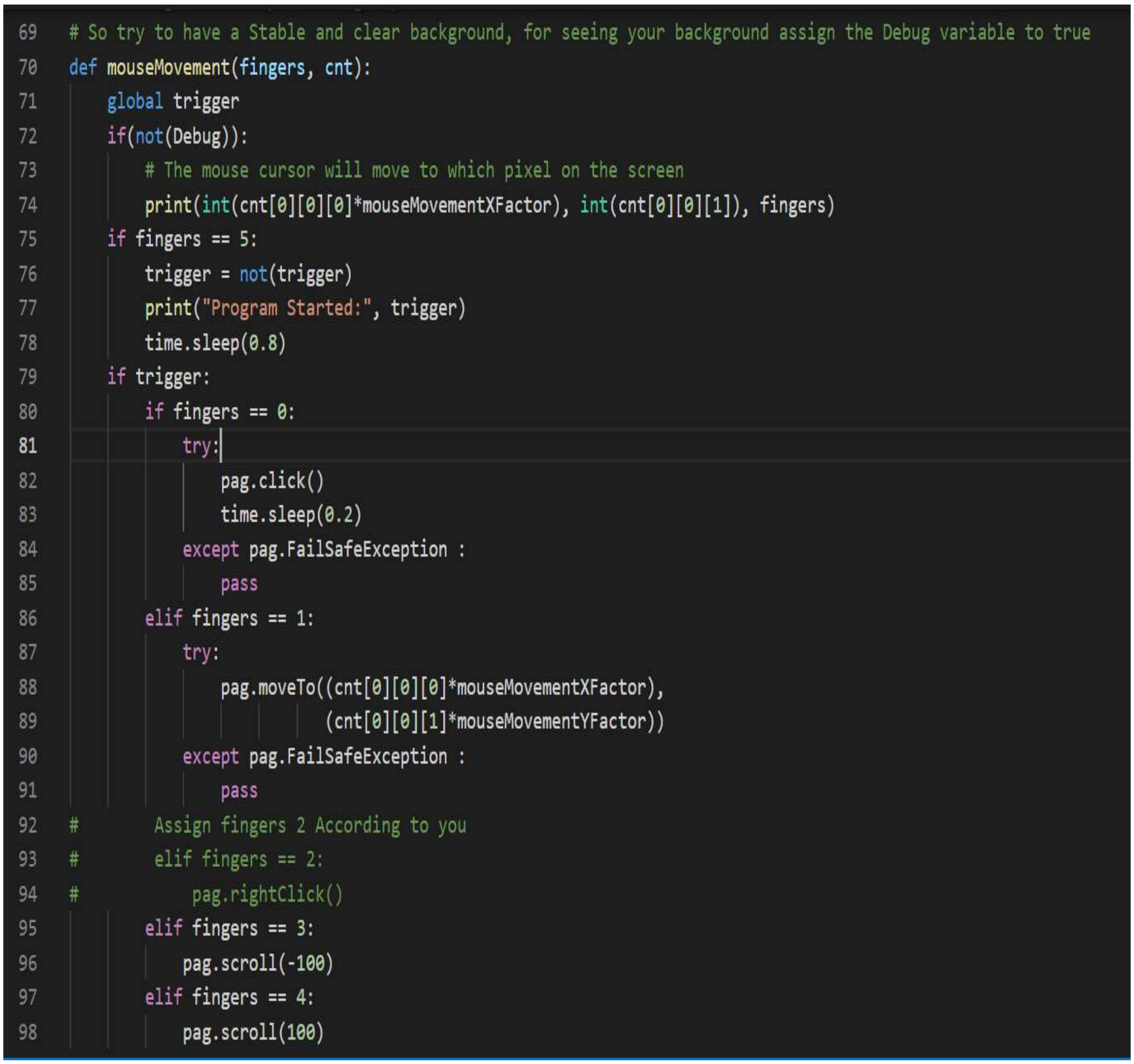
3.CheckArea:



```python
# Checking the Area of the hand
def checkArea(l, y):
    mappedY = 1

    if l==1:
        if areacnt < 600:    l = -1
        else:
            if y < 120:
                mappedY = 2
            if y < 180:
                mappedY = 1.6
            if arearatio < (9.5 * mappedY):
                l = 0
            else:   l = 1
    if(Debug):
        print(y, int(arearatio), mappedY, l)
    return l
```

Here we check and calculate the area of hand.

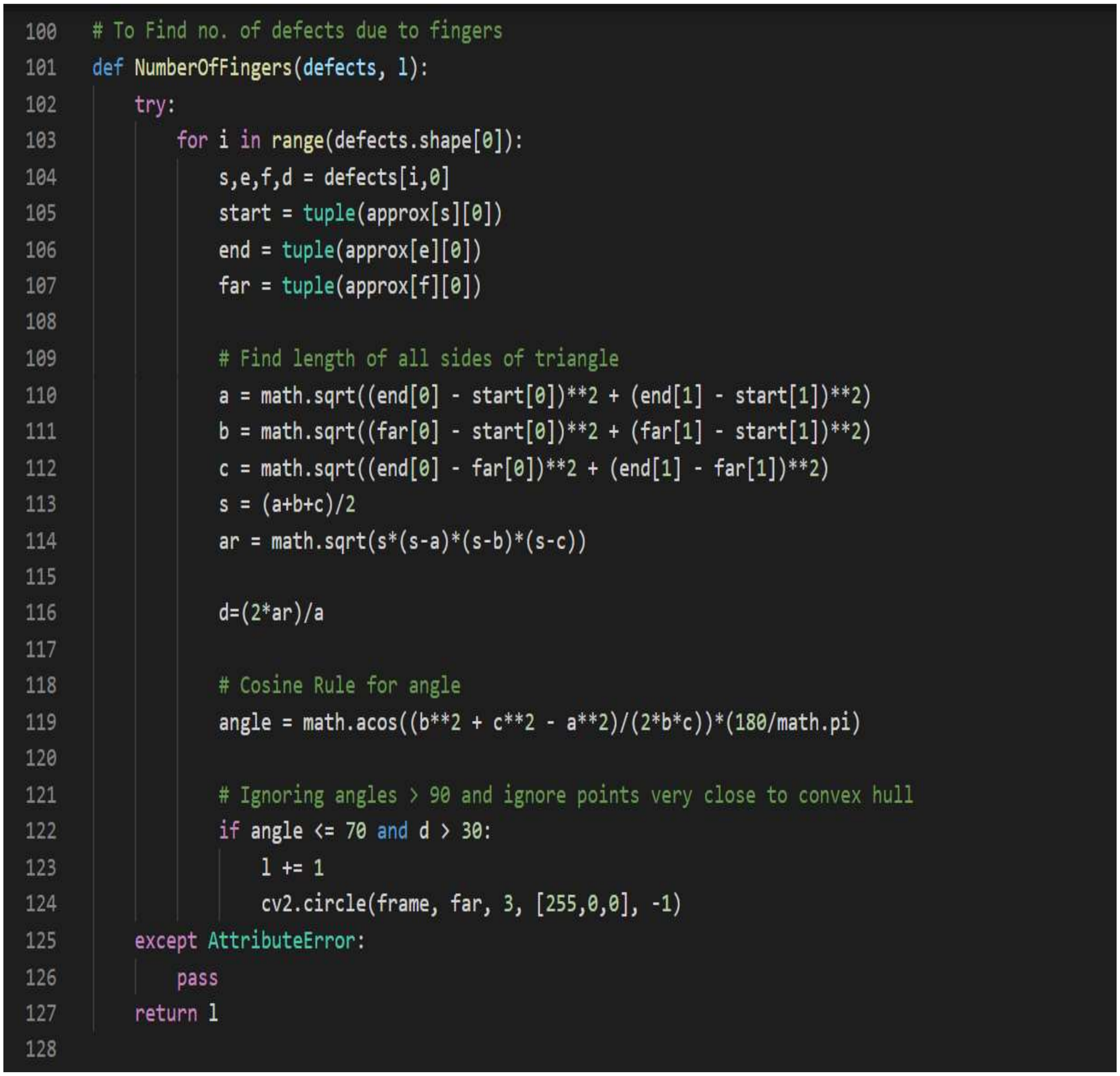
4.MouseMovement:



```python
# So try to have a Stable and clear background, for seeing your background assign the Debug variable to true
def mouseMovement(fingers, cnt):
    global trigger
    if(not(Debug)):
        # The mouse cursor will move to which pixel on the screen
        print(int(cnt[0][0][0]*mouseMovementXFactor), int(cnt[0][0][1]), fingers)
    if fingers == 5:
        trigger = not(trigger)
        print("Program Started:", trigger)
        time.sleep(0.8)
    if trigger:
        if fingers == 0:
            try:
                pag.click()
                time.sleep(0.2)
            except pag.FailSafeException :
                pass
        elif fingers == 1:
            try:
                pag.moveTo((cnt[0][0][0]*mouseMovementXFactor),
                           (cnt[0][0][1]*mouseMovementYFactor))
            except pag.FailSafeException :
                pass
#         Assign fingers 2 According to you
#         elif fingers == 2:
#             pag.rightClick()
        elif fingers == 3:
            pag.scroll(-100)
        elif fingers == 4:
            pag.scroll(100)
```

1. Here we assign the mouse controls to the fingers.

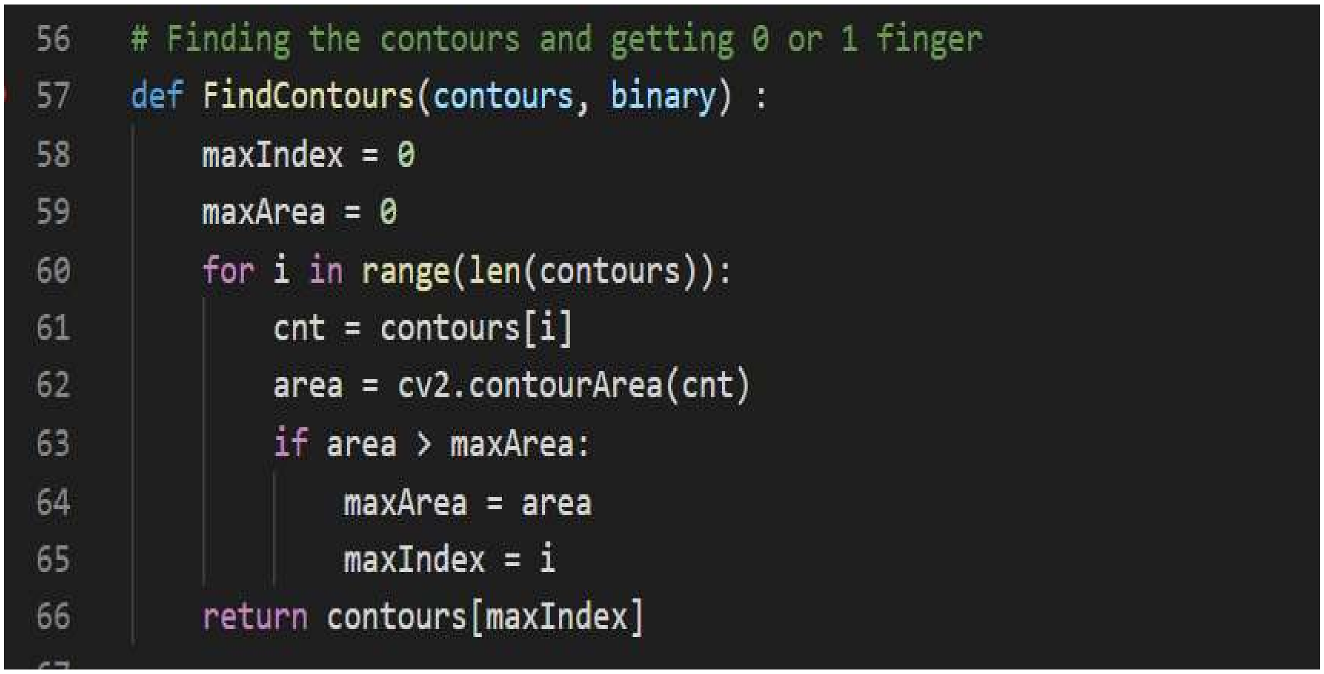
5. **NumberOfFingers:**



```python
# To Find no. of defects due to fingers
def NumberOfFingers(defects, l):
    try:
        for i in range(defects.shape[0]):
            s,e,f,d = defects[i,0]
            start = tuple(approx[s][0])
            end = tuple(approx[e][0])
            far = tuple(approx[f][0])

            # Find length of all sides of triangle
            a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
            b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
            c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
            s = (a+b+c)/2
            ar = math.sqrt(s*(s-a)*(s-b)*(s-c))

            d=(2*ar)/a

            # Cosine Rule for angle
            angle = math.acos((b**2 + c**2 - a**2)/(2*b*c))*(180/math.pi)

            # Ignoring angles > 90 and ignore points very close to convex hull
            if angle <= 70 and d > 30:
                l += 1
                cv2.circle(frame, far, 3, [255,0,0], -1)
    except AttributeError:
        pass
    return l
```

We calculate the number of fingers according to the angles and area between the fingers.

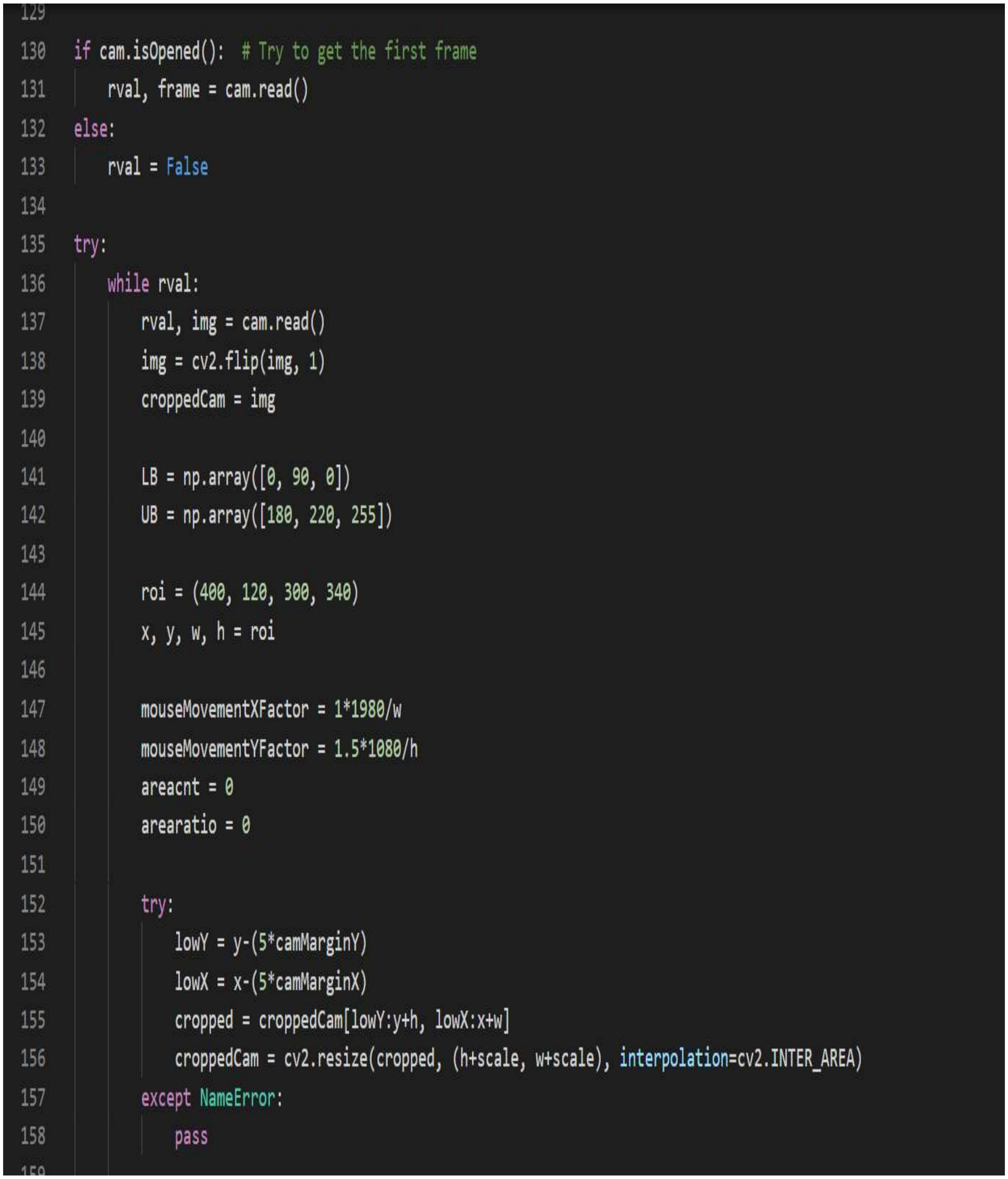
6.FindContours:



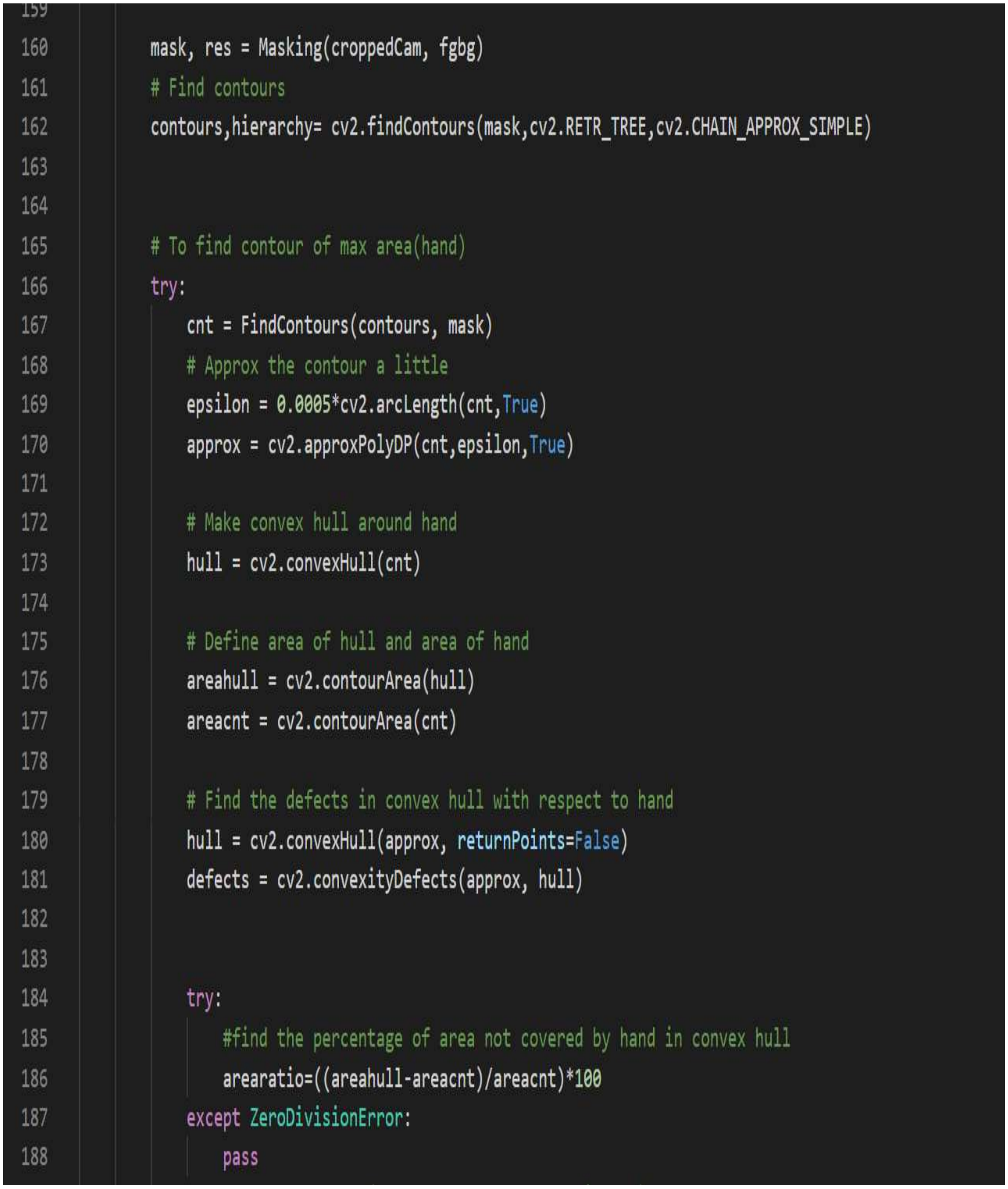
```python
# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]
```

Here we find te coordinates along the boundary of the hands

7.Code Outside any Function:



```
if cam.isOpened():  # Try to get the first frame
    rval, frame = cam.read()
else:
    rval = False

try:
    while rval:
        rval, img = cam.read()
        img = cv2.flip(img, 1)
        croppedCam = img

        LB = np.array([0, 90, 0])
        UB = np.array([180, 220, 255])

        roi = (400, 120, 300, 340)
        x, y, w, h = roi

        mouseMovementXFactor = 1*1980/w
        mouseMovementYFactor = 1.5*1080/h
        areacnt = 0
        arearatio = 0

        try:
            lowY = y-(5*camMarginY)
            lowX = x-(5*camMarginX)
            cropped = croppedCam[lowY:y+h, lowX:x+w]
            croppedCam = cv2.resize(cropped, (h+scale, w+scale), interpolation=cv2.INTER_AREA)
        except NameError:
            pass
```

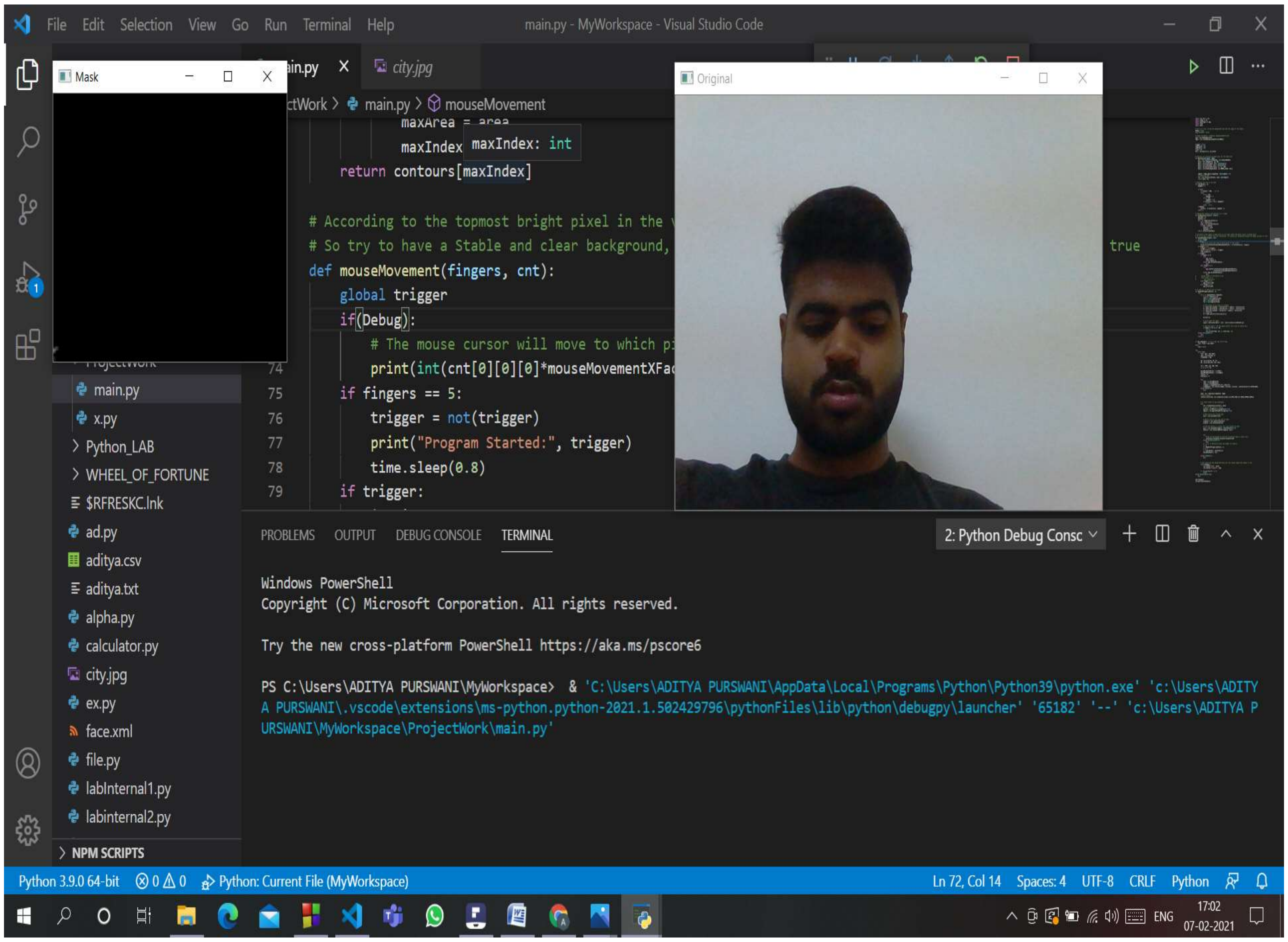


```python
        mask, res = Masking(croppedCam, fgbg)
        # Find contours
        contours,hierarchy= cv2.findContours(mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)


        # To find contour of max area(hand)
        try:
            cnt = FindContours(contours, mask)
            # Approx the contour a little
            epsilon = 0.0005*cv2.arcLength(cnt,True)
            approx = cv2.approxPolyDP(cnt,epsilon,True)

            # Make convex hull around hand
            hull = cv2.convexHull(cnt)

            # Define area of hull and area of hand
            areahull = cv2.contourArea(hull)
            areacnt = cv2.contourArea(cnt)

            # Find the defects in convex hull with respect to hand
            hull = cv2.convexHull(approx, returnPoints=False)
            defects = cv2.convexityDefects(approx, hull)


            try:
                #find the percentage of area not covered by hand in convex hull
                arearatio=((areahull-areacnt)/areacnt)*100
            except ZeroDivisionError:
                pass
```

<u>**Chapter 4: Results And Discussion:**</u>

Output 1: Showing nothing:

Here we show no fingers so nothing happens in the program.



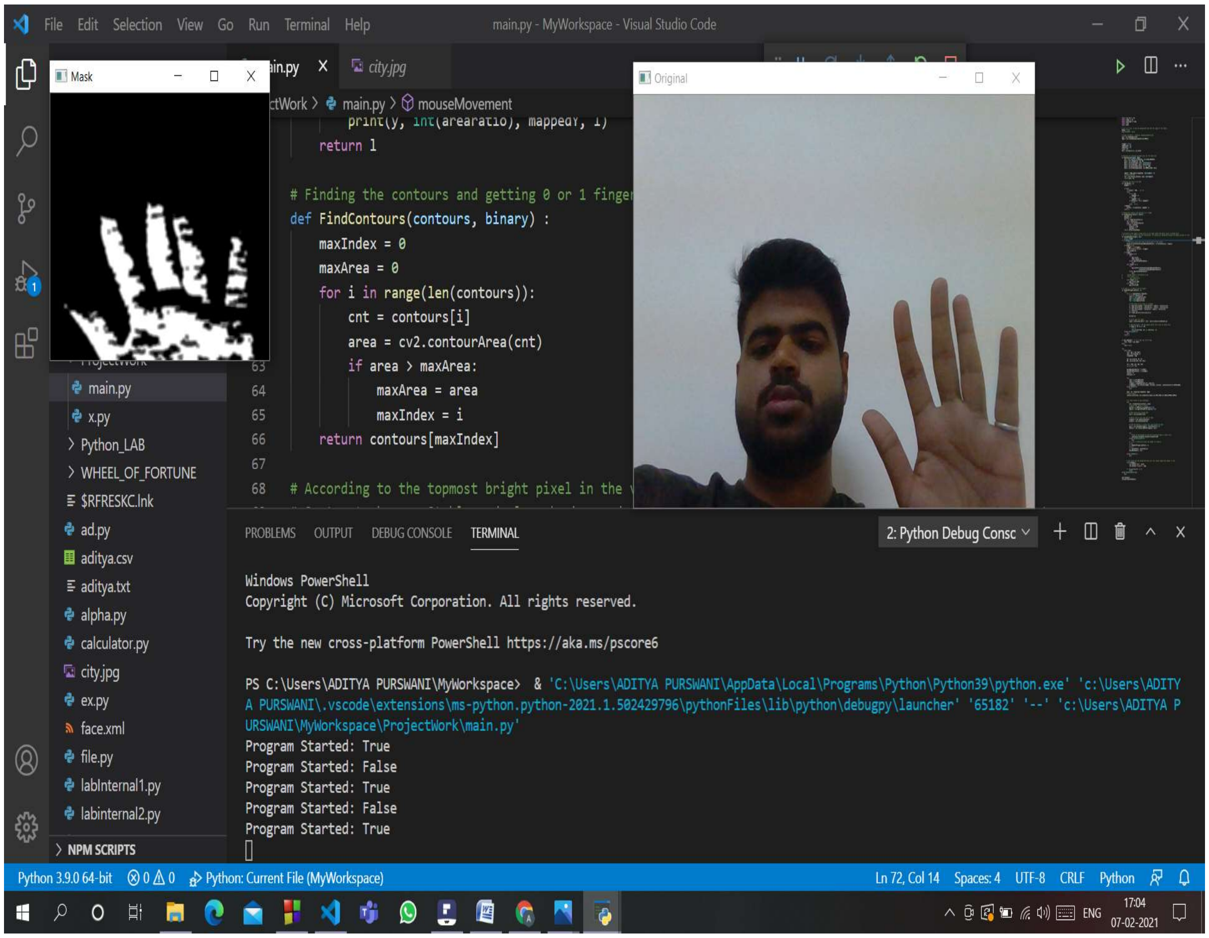
```
            maxArea = area
            maxIndex maxIndex: int
    return contours[maxIndex]

# According to the topmost bright pixel in the v
# So try to have a Stable and clear background,                    true
def mouseMovement(fingers, cnt):
    global trigger
    if(Debug):
        # The mouse cursor will move to which pi
        print(int(cnt[0][0][0]*mouseMovementXFac
    if fingers == 5:
        trigger = not(trigger)
        print("Program Started:", trigger)
        time.sleep(0.8)
    if trigger:
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
```

Output 2: Showing 5 fingers:

Here when we show 5 fingers the program execution triggers:

1. If the program is in execution then it pauses the execution until the next trigger.
2. If the program is paused then the program execution is resumed.

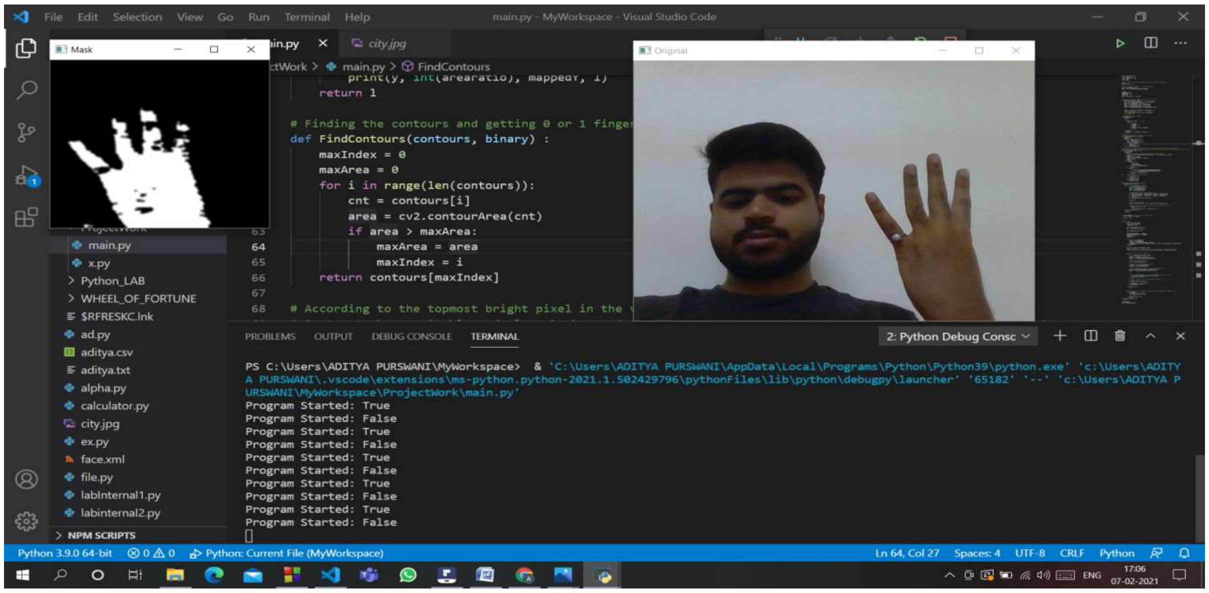

main.py - MyWorkspace - Visual Studio Code
Mask
Original
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Python Debug Consc

```
        print(y, int(arearatio), mappedY, 1)
    return 1

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]

# According to the topmost bright pixel in the
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
```

Output 3: Showing 4 fingers:

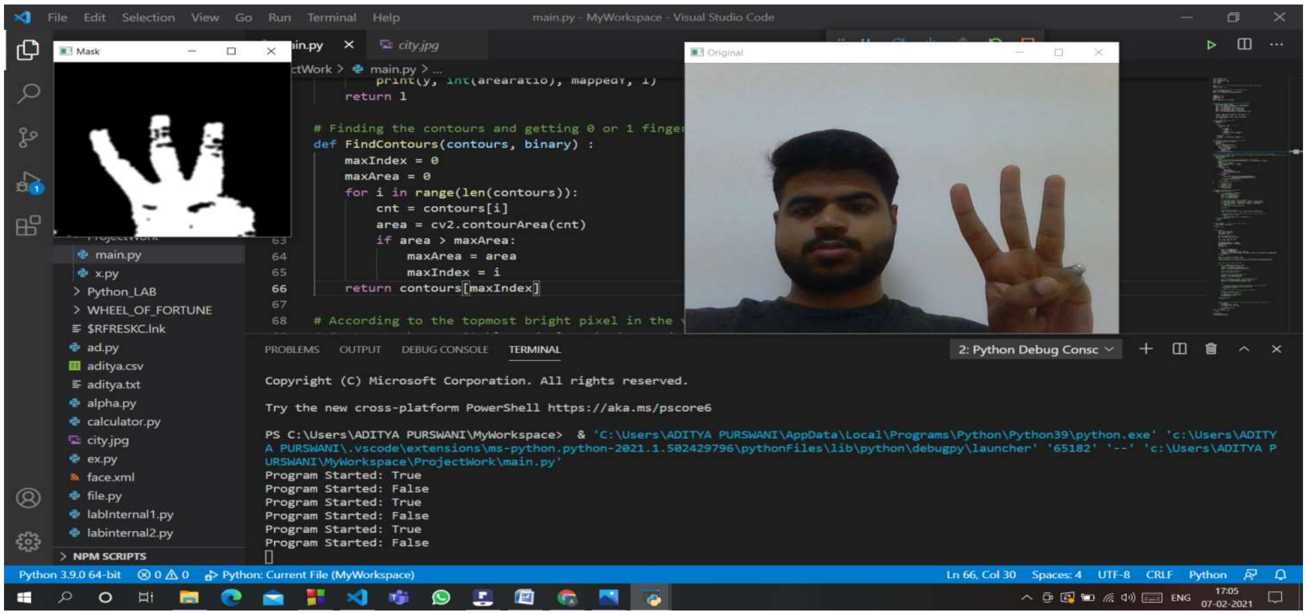
We scroll up when we show 4 fingers.


```
        print(y, int(arearatio), mappedY, 1)
    return 1

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]

# According to the topmost bright pixel in the
```

```
PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
```

Output 4: Showing 3 fingers:

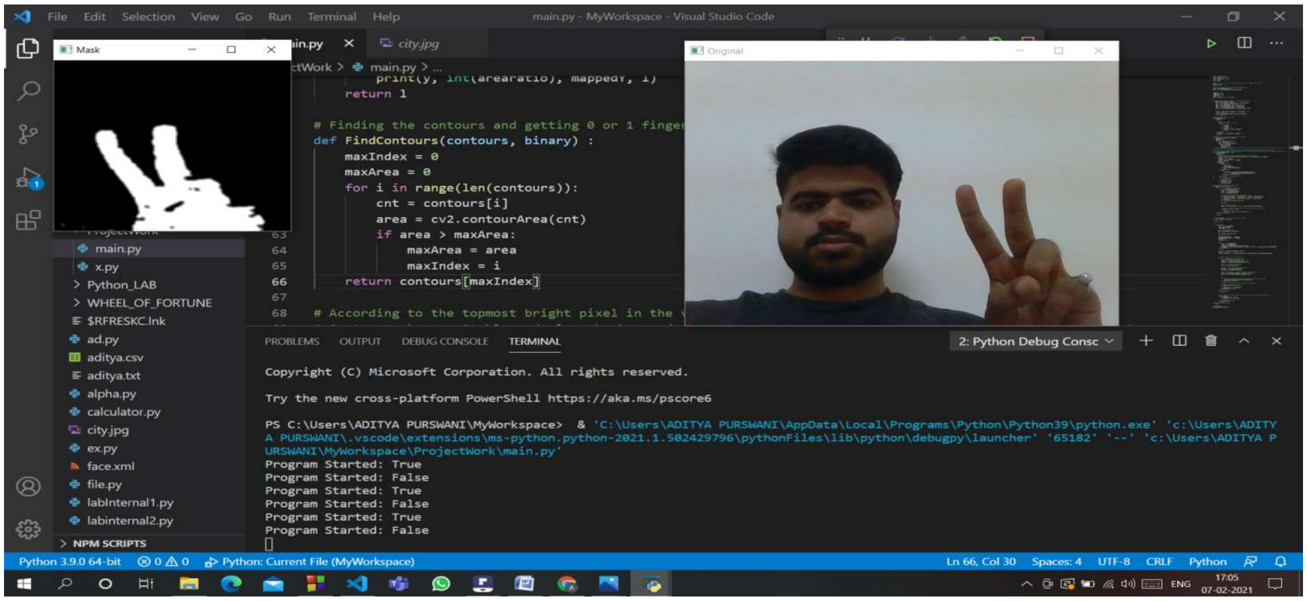
We scroll down when we show 3 fingers


```
        print(y, int(arearatio), mappedY, 1)
    return 1

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]

# According to the topmost bright pixel in the
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
```

Output 5: Showing 2 fingers:

Here it won't do anything but we assign it to any of the operation

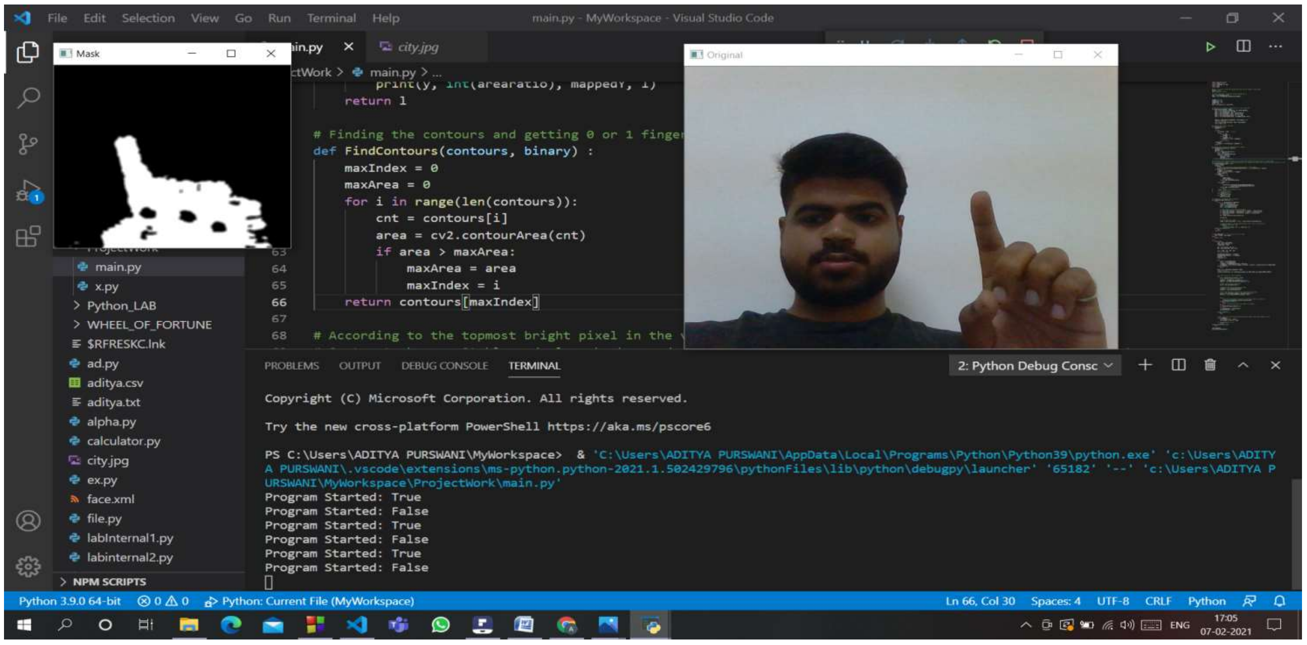


```
    print(y, int(arearatio), mappedY, 1)
    return 1

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]

# According to the topmost bright pixel in the
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
```

Output 6: Showing 1 finger:

We move the mouse cursor on the screen according to the brightest pixel

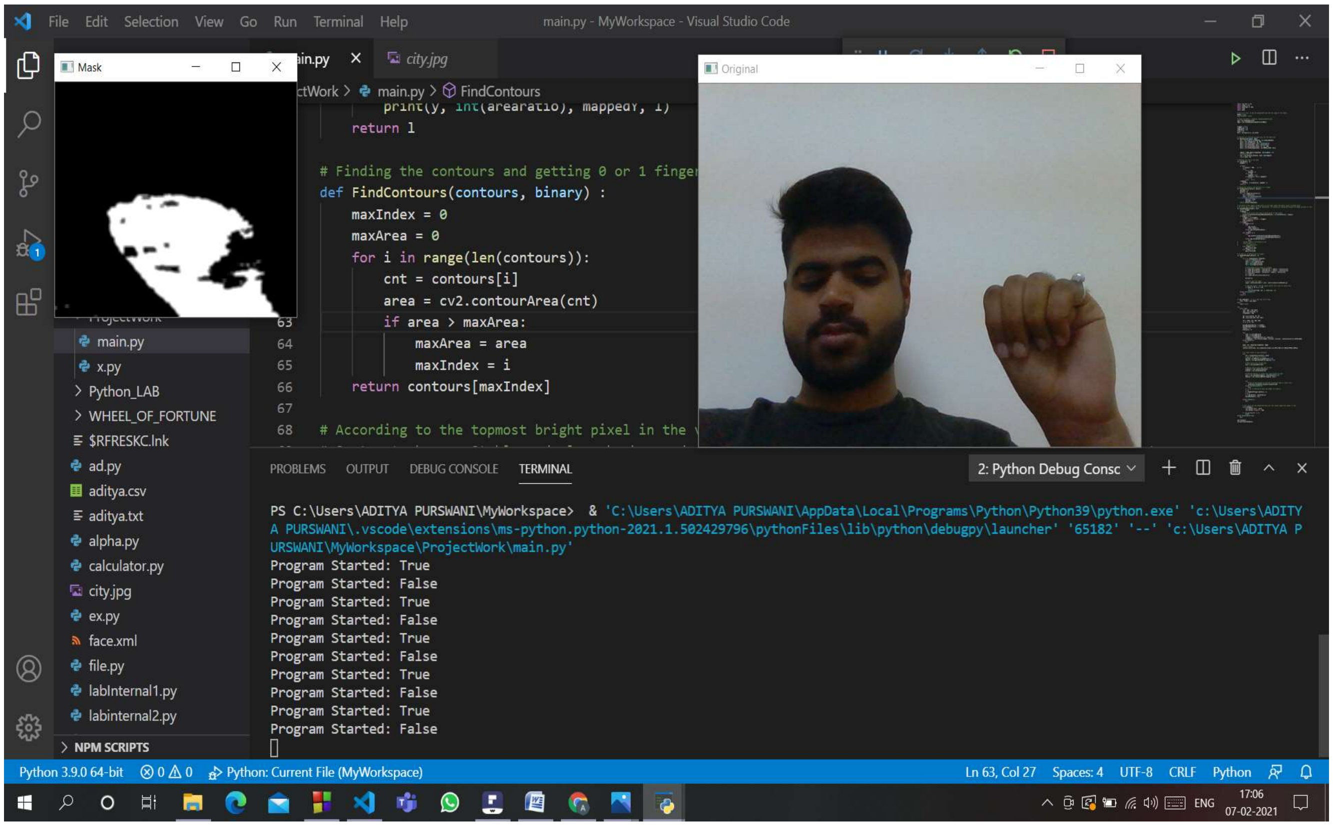


```
    print(y, int(arearatio), mappedY, 1)
    return 1

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]

# According to the topmost bright pixel in the
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
```

Output 7: Showing fist / 0 fingers:

We can left click on the screen and on the apps to open them



```
        print(y, int(arearatio), mappedY, 1)
    return 1

# Finding the contours and getting 0 or 1 finger
def FindContours(contours, binary) :
    maxIndex = 0
    maxArea = 0
    for i in range(len(contours)):
        cnt = contours[i]
        area = cv2.contourArea(cnt)
        if area > maxArea:
            maxArea = area
            maxIndex = i
    return contours[maxIndex]

# According to the topmost bright pixel in the
```

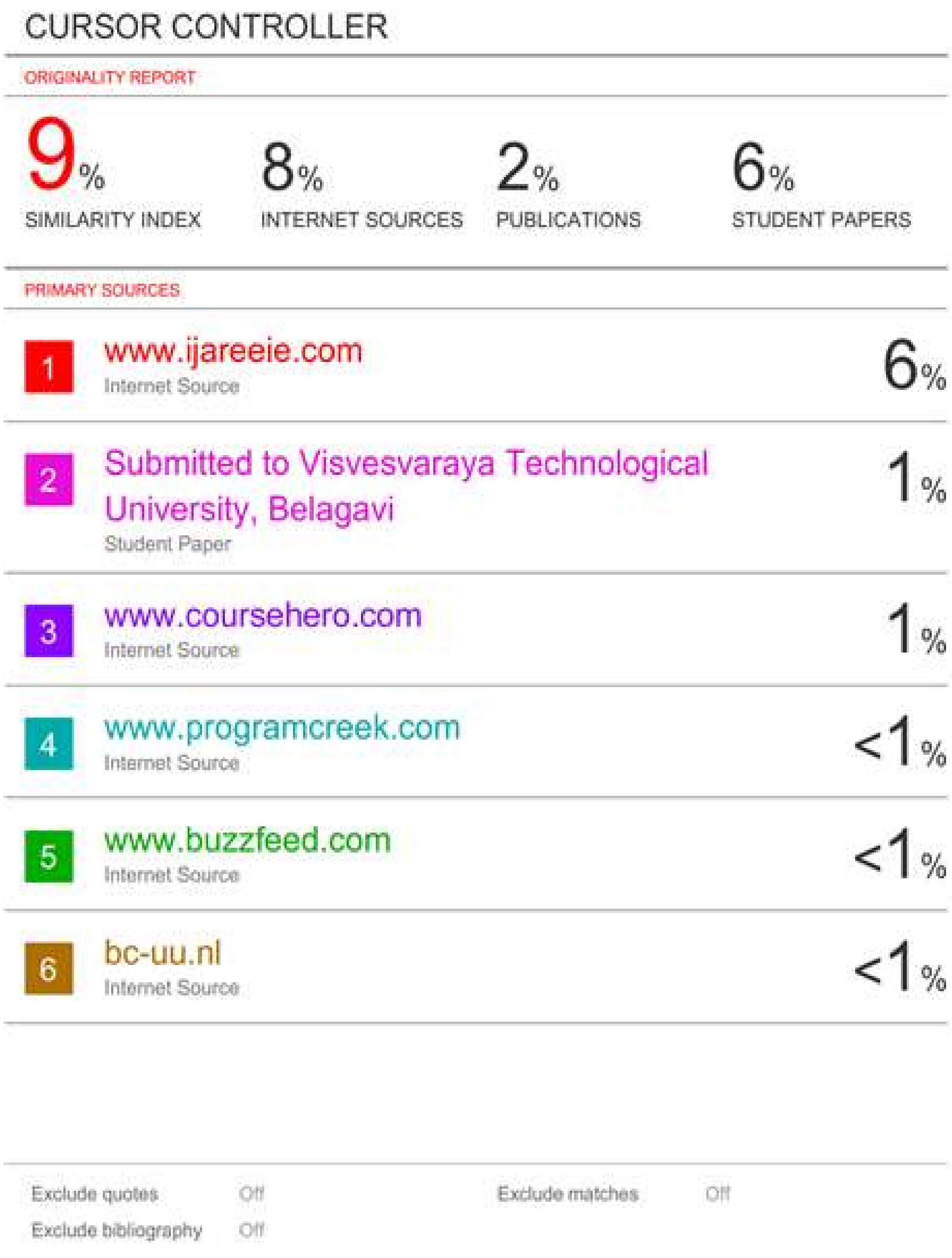
```
PS C:\Users\ADITYA PURSWANI\MyWorkspace>  & 'C:\Users\ADITYA PURSWANI\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ADITY
A PURSWANI\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '65182' '--' 'c:\Users\ADITYA P
URSWANI\MyWorkspace\ProjectWork\main.py'
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
Program Started: True
Program Started: False
```

Conclusion:

- This can be an interpretation of how future technologies look like, with minimum use of hardware and most things performed manually.
- Helps us understand the importance of image processing and manipulation in today's world.
- Now we can control the cursor through hand gestures.

Refrences:

1. **Wikipedia: For Convex Hull Algorithms**
2. **GeeksForGeeks**
3. **GoogleImages: For all the Images.**



CURSOR CONTROLLER

ORIGINALITY REPORT

9%	8%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijareeie.com Internet Source	6%
2	Submitted to Visvesvaraya Technological University, Belagavi Student Paper	1%
3	www.coursehero.com Internet Source	1%
4	www.programcreek.com Internet Source	<1%
5	www.buzzfeed.com Internet Source	<1%
6	bc-uu.nl Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		