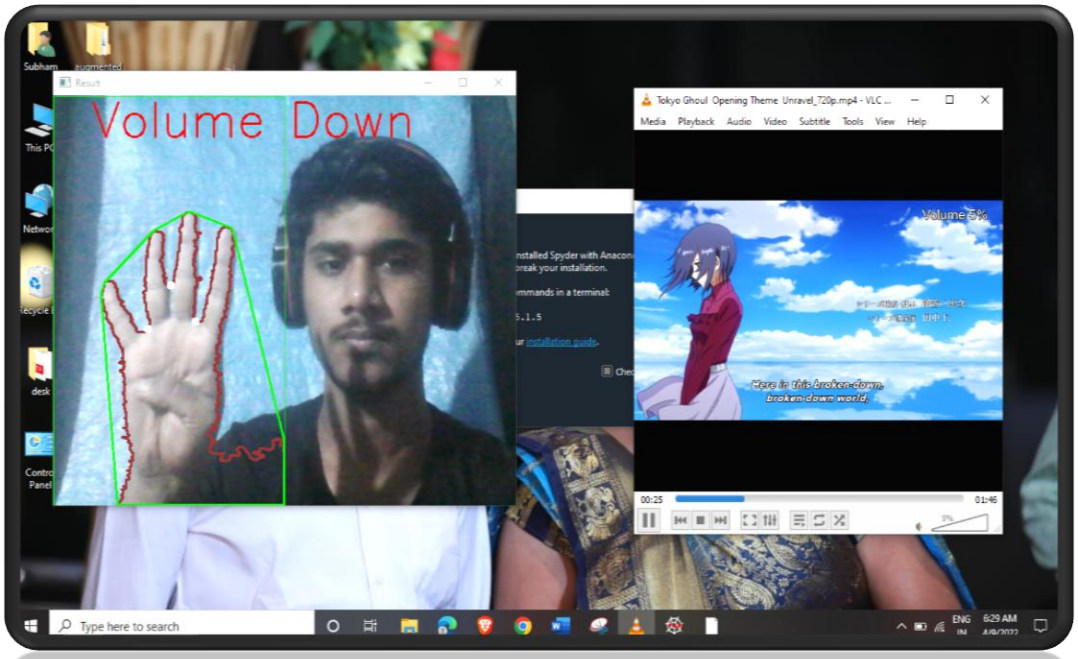
Topic :-

Hand Gesture to Control Volume, Presentation, And Document



RAMNIRANJAN JHUNJHUNWALA COLLEGE

Department OF INFORMATION TECHNOLOGY

Ghatkopar (West), Mumbai - 86

Project Report On

Hand Gesture to Control Volume, Presentation, And Document.

By

Mr. ShubhamKumar Shyambihari Yadav

M.Sc. Information Technology
University of Mumbai
2021-2022

Project Guide

Mrs. BHARATI BHOLE

RAMNIRANJAN JHUNJHUNWALA COLLEGE

Department of Information Technology

Ghatkopar (West), Mumbai – 86

2021-2022

Project Report
on

Hand Gesture to Control Volume, Presentation, And Document.

In partial fulfillment of M.Sc. (INFORMATION TECHNOLOGY)

By

Mr. ShubhamKumar Shyambihari Yadav

Project Guide

Prof. Bharati Bhole

RAMNIRANJAN JHUNJHUNWALA COLLEGE (AUTONOMOUS)

(Affiliated to University of Mumbai)

GHATKOPAR(WEST), 400086.

Certificate

This is to certify that the Project entitled, “Hand Gesture to Control Volume, Presentation, And Documents” is bonafide work of Mr. Shubhamkumar Shyambihari Yadav bearing Seat No: - 25 submitted in partial fulfilment of the requirements for the award of Degree Master of Science in Information Technology.

Signature of Internal Guide

Sign of Co-Ordinator

Examiner

Date:

College Seal

ACKNOWLEDGEMENT

I have a great pleasure in representing this project report entitled "Hand Gesture to Control Volume, Presentation, And Documents" and I grab this opportunity to convey my immense regards towards all the distinguished people who have their valuable contribution in the hour of need.

The four things that go on to make a successful endeavor are dedication, hard work, patience and correct guidance. Able and timely guidance not only helps in making an effort fruitful but also transforms the whole process of learning and implementing into an enjoyable experience.

In particular, I would like to thank our principal **Dr. Himanshu Dawda** & R.J college.

I would like to give a very special honor and respect to our teacher, **Prof. BHARATI BHOLE** who took keen interest in checking the minute details of the project work and guided us throughout the same. A sincere quote of thanks to the non-teaching staff for providing us software & their time.

DECLARATION

I hereby declare that the project entitled, "**Hand Gesture to Control Volume, Presentation, And Documents"** done at **place where the project is done**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTERS OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

ShubhamKumar Shyambihari Yadav

Signature of the Student

INDEX

ABSTRACT

In this Project we try to control our media player using hand gestures with the help of OpenCV and Python. Computer applications require interaction between human and computer. This interaction needs to be unrestricted and it has made it challenging to traditional input devices such as keyboard, mouse, pen etc.

Hand gesture is an important component of body languages in linguistics. Human computer interaction becomes easy with the use of the hand as a device. Use of hand gestures to operate machines would make interaction interesting. Gesture recognition has gained a lot of importance. Hand gestures are used to control various applications like windows media player, Presentation, gaming etc. Use of gesture makes interaction easy, convenient and does not require any extra device.

INTRODUCTION

Hand gesture recognition is very significant for human-computer interaction. In this project, I present a novel real-time method for hand gesture recognition. In our framework, the hand region is extracted from the background with the background subtraction method. Then, the palm and fingers are segmented so as to detect and recognize the fingers. Finally, a rule classifier is applied to predict the labels of hand gestures. The experiments on the set of 10 images/slides has done to show that our method performs well and is highly efficient.

Gesture is a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. It falls into two categories: static gesture and dynamic gesture. For the former, the posture of the body or the gesture of the hand denotes a sign. For the latter, the movement of the body or the hand conveys some messages. Gesture can be used as a tool of communication between computer and human. It is greatly different from the traditional hardware based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts.

OBJECTIVES/PURPOSE

In this project, I have created a Volume Controlled by Hand Gesture based system. We will be able to increase and decrease volume of any Video as well as pause and Fast Forward And move the slides back and forth . And to make it more usable we will add an erasing gesture as well.

From the business point of view hand gesture recognition has been providing positive outputs in various fields such as a

- **Vehicle Industry**

Hand gestures are powerful human to human communication channel which convey a major part of information transfer in our everyday life. Hand gestures are the natural way of interactions when one person is communicating with one another and therefore hand movements can be treated as a nonverbal form of communication. Hand gesture recognition is a process of understanding and classifying meaningful movements by the human hands. Nowadays vehicles launched from the industry offers an increasing number of infotainment systems as well as comfort functions that can be controlled by the driver. Though they are feature rich which demands more attention of the driver and degrade the driving performance and thereby reducing the safety. The gestural interaction is a promising means to cover the full range of driver's operational needs while minimizing the visual workload. Hand gesture recognition is of great importance for human computer interaction (HCI) because of its extensive applications in virtual reality and sign language recognition etc. Human hand is very smaller with very complex articulations comparing with the entire human body and therefore errors can be easily affected. It is thus a very challenging problem to recognize hand gestures. This paper comprises of the existing methods in detecting and recognizing hand gestures and a detailed study on their performances, accuracy, convenience, operational range and design challenges etc.

- **Teaching Industry**

Gesturing is a powerful tool in the classroom. It helps teachers explain and assess, and helps students recall information and solve problems more efficiently. The great news is that you don't need special training to make better use of gestures for learning, you just need to know that each effect exists.

- **Healthcare**

Emergency rooms and operating rooms may be chaotic, with lots of noise from personnel and machines. In such environments, voice commands are less effective than gestures. Touchscreens are not an option either, since there's a strict boundary between what is and is not sterile. But accessing information and imaging during surgery or another manipulation is possible with HGR tech, as proven by Microsoft. Gesture provides doctors with the ability to check MRI, CT, and other imagery with simple gestures without scrubbing out.

- **Virtual Reality**

A hand tracking application from ManoMotion recognizes gestures in three dimensions using a smartphone camera (on both Android and iOS) and can be applied in AR and VR environments. The use cases for this technology include gaming, IoT devices, consumer electronics, and robots.

SURVEY OF TECHNOLOGY

REQUIREMENT ANALYSIS

Problem definition

With the development of ubiquitous computing, current user interaction approaches with keyboard, mouse and pen are not sufficient. Due to the limitation of these devices the useable command set is also limited. Direct use of hands can be used as an input device for providing natural interaction.

Requirements Specification

i)Programming Language: Python

ii)Hardware: - Min. 3 GB - Intel ® i3® processor with 1.6 Hz

- 64-bit Operating System
- Webcam

iii)Platform: - spyder

SOFTWARE :-

Spyder

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection and beautiful visualization capabilities of a scientific package.n Furthermore, Spyder offers built-in integration with many popular scientific packages, including NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and more.n Beyond its many built-in features, Spyder can be extended even further via third-party plugins.n Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console or advanced editor, in your own software.

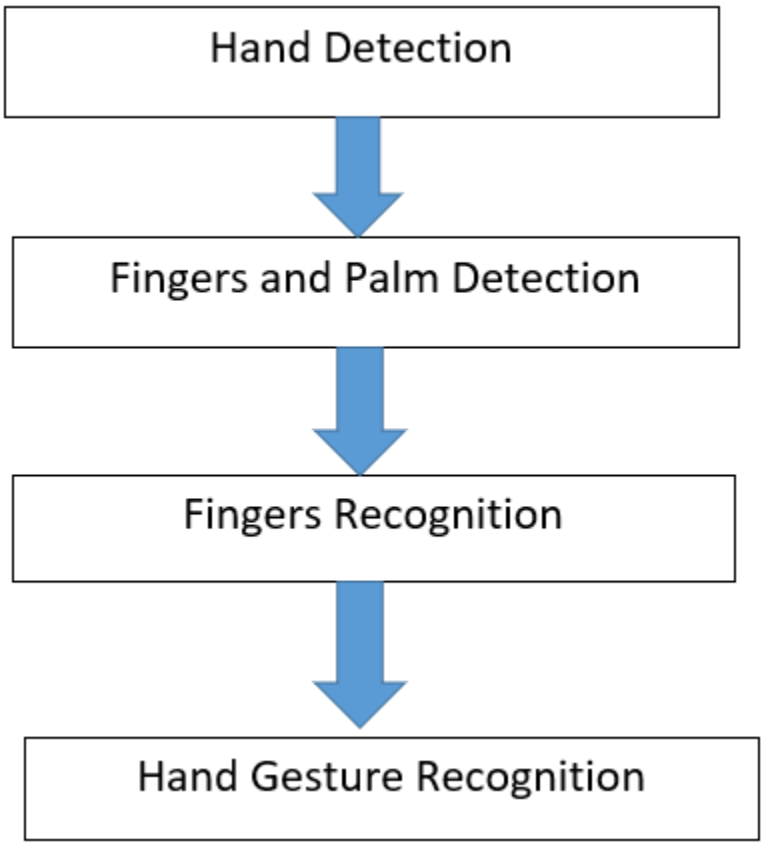
Computer Vision

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information.

Python3

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its 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.

SYSTEM DESIGN :-



IMPLEMENTATION

Implementation Approaches

Phased Implementation

A crucial phase in system development is the successful implementation of the new system design. Implementation includes all those activities that take place to convert from the old system to the new system. The new system may be completely new replacing an existing manual or automated system or it may be major modification to an existing system.

When small parts of the new system gradually replace small parts of the old system, the implementation method is said to be **phased**.

The system has been developed with the phased implementation approaches.

Steps

Coding Details:-

First install the following:-

```
!pip install opencv-python
!pip install pyautogui
```

#Step - 1 -Import Libraries and capture camera.

```
import cv2
import numpy as np
import math
import pyautogui as p # with the help of this library we can controll our keyboard with our gesture
import time as t



#Read Camera
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW) # 0 for your laptop camera we can pass 1 for any additional camera or
```

Import libraries used in this project.

Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

OpenCV

OpenCV (Open Source Computer Vision Library) is **an open source computer vision and machine learning software library**. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

Math

The Python Math Library **provides us access to some common math functions and constants in Python**, which we can use throughout our code for more complex mathematical computations. The library is a built-in Python module, therefore you don't have to do any installation to use it.

Time

The Python time module **provides many ways of representing time in code, such as objects, numbers, and strings**. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

Pyautogui

Pyautogui is a library that **allows you to control the mouse and keyboard to do various things**. It is a cross-platform GUI automation Python module for human beings. As it is a third party library, we need to install it.

Creating Windows For Trackbar and Result Screen.

```
def nothing(x):
    pass

#window name
cv2.namedWindow("Color Adjustments",cv2.WINDOW_NORMAL)
cv2.resizeWindow("Color Adjustments", (300, 300))
cv2.createTrackbar("Thresh", "Color Adjustments", 0, 255, nothing)

#COlor Detection Track

cv2.createTrackbar("Lower_H", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Lower_S", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Lower_V", "Color Adjustments", 0, 255, nothing)
cv2.createTrackbar("Upper_H", "Color Adjustments", 255, 255, nothing)
cv2.createTrackbar("Upper_S", "Color Adjustments", 255, 255, nothing)
cv2.createTrackbar("Upper_V", "Color Adjustments", 255, 255, nothing)


while True:
    _,frame = cap.read()
    frame = cv2.flip(frame,2)
    frame = cv2.resize(frame,(600,500))
    # Get hand data from the rectangle sub window
    cv2.rectangle(frame, (0,1), (300,500), (255, 0, 0), 0)
    crop_image = frame[1:500, 0:300]
```

#Step - 2 -Convert frames Into hsv (hue saturated value)

```
#Step -2
hsv = cv2.cvtColor(crop_image, cv2.COLOR_BGR2HSV)
#detecting hand
l_h = cv2.getTrackbarPos("Lower_H", "Color Adjustments")
l_s = cv2.getTrackbarPos("Lower_S", "Color Adjustments")
l_v = cv2.getTrackbarPos("Lower_V", "Color Adjustments")

u_h = cv2.getTrackbarPos("Upper_H", "Color Adjustments")
u_s = cv2.getTrackbarPos("Upper_S", "Color Adjustments")
u_v = cv2.getTrackbarPos("Upper_V", "Color Adjustments")
```

#Step - 3 -Track hand on color basis

```
lower_bound = np.array([l_h, l_s, l_v])
upper_bound = np.array([u_h, u_s, u_v])
```

#Step - 4 -Create mask on the basis of color and filter actual color

```
#Step - 4
#Creating Mask
mask = cv2.inRange(hsv, lower_bound, upper_bound)
#filter mask with image
filtr = cv2.bitwise_and(crop_image, crop_image, mask=mask)
```

#Step - 5 -Invert pixel value and then enchance the result for better output

```
#Step - 5
mask1  = cv2.bitwise_not(mask)
m_g = cv2.getTrackbarPos("Thresh", "Color Adjustments") #getting track bar value
ret,thresh = cv2.threshold(mask1,m_g,255,cv2.THRESH_BINARY)
dilata = cv2.dilate(thresh,(3,3),iterations = 6)
```

#Step - 6 -Find Contours for specific colored object

#Step - 7 -Find Max area contour and draw it on live feed

```
#Step -6
#findcontour(img,contour_retrival_mode,method)
cnts,hier = cv2.findContours(thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)


try:
    #print("try")
     #Step -7
     # Find contour with maximum area
    cm = max(cnts, key=lambda x: cv2.contourArea(x))
    #print("C==",cnts)
    epsilon = 0.0005*cv2.arcLength(cm,True)
    data= cv2.approxPolyDP(cm,epsilon,True)

    hull = cv2.convexHull(cm)

    cv2.drawContours(crop_image, [cm], -1, (50, 50, 150), 2)
    cv2.drawContours(crop_image, [hull], -1, (0, 255, 0), 2)
```

#Step - 8 -Find Convexity detect for counting Values and Apply Cosin method

```
#Step - 8
# Find convexity defects
hull = cv2.convexHull(cm, returnPoints=False)
defects = cv2.convexityDefects(cm, hull)
count_defects = 0
#print("Area==",cv2.contourArea(hull) - cv2.contourArea(cm))
for i in range(defects.shape[0]):
    s,e,f,d = defects[i,0]

    start = tuple(cm[s][0])
    end = tuple(cm[e][0])
    far = tuple(cm[f][0])
    #Cosin Rule
    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)
    angle = (math.acos((b ** 2 + c ** 2 - a ** 2) / (2 * b * c)) * 180) / 3.14
    #print(angle)
    # if angle <= 50 draw a circle at the far point
    if angle <= 50:
        count_defects += 1
        cv2.circle(crop_image,far,5,[255,255,255],-1)

print("count==",count_defects)
```

#Step - 9 -Bind hand gestures with keyboard keys.

```
        #Step - 9
        # Print number of fingers
        if count_defects == 0:

            cv2.putText(frame, " ", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255),2)
        elif count_defects == 1:

            p.press("space")
            cv2.putText(frame, "Play/Pause", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
        elif count_defects == 2:
            p.press("up")

            cv2.putText(frame, "Volume UP", (5, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
        elif count_defects == 3:
            p.press("down")

            cv2.putText(frame, "Volume Down", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
        elif count_defects == 4:
            p.press("right")

            cv2.putText(frame, "Forward", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2,(0,0,255), 2)
        else:
            pass

    except:
        pass
```

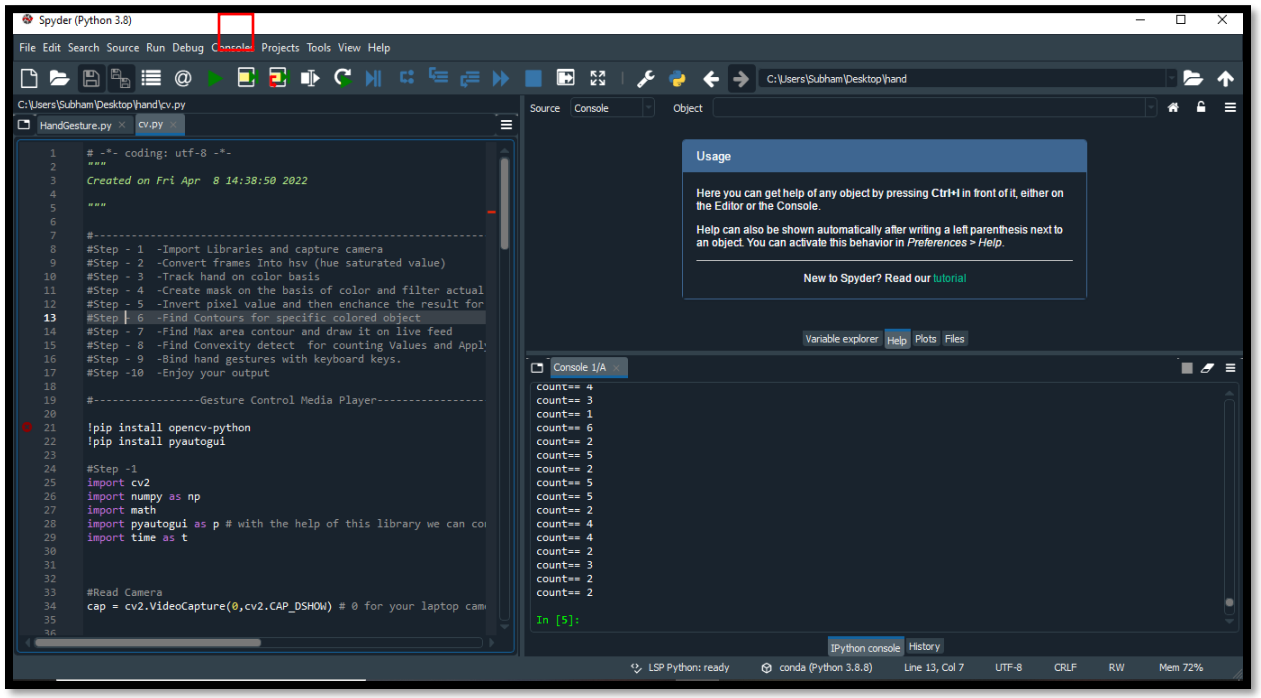
#Step -10 -Enjoy your output

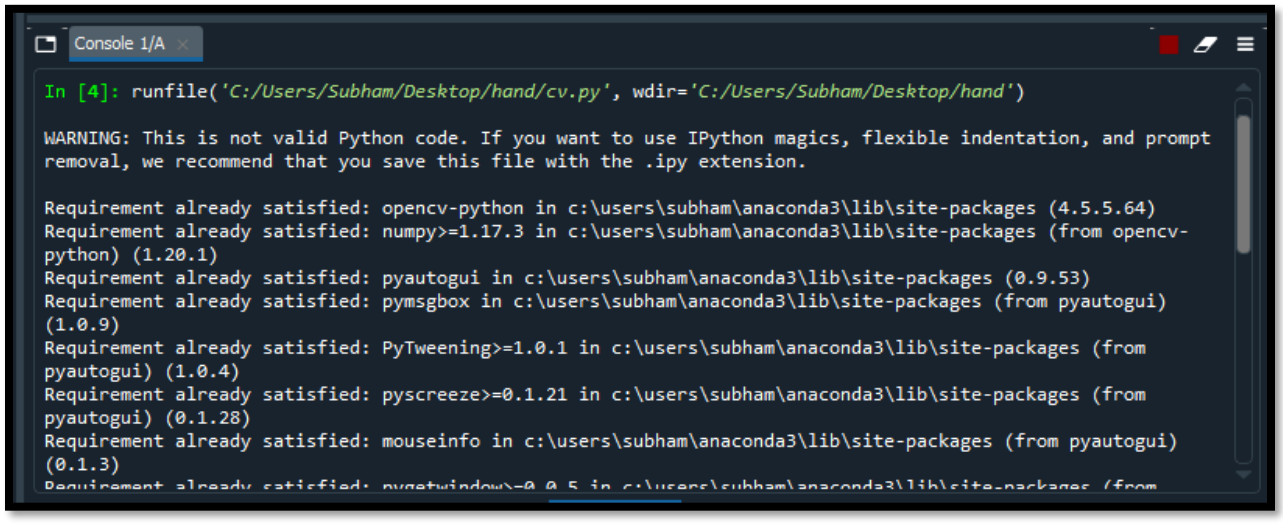
```
            pass
        #step -10
        cv2.imshow("Thresh", thresh)
        #cv2.imshow("mask==",mask)
        cv2.imshow("filter==",filtr)
        cv2.imshow("Result", frame)

        key = cv2.waitKey(25) &0xFF
        if key == 27:
            break
cap.release()
cv2.destroyAllWindows()
```

RESULTS

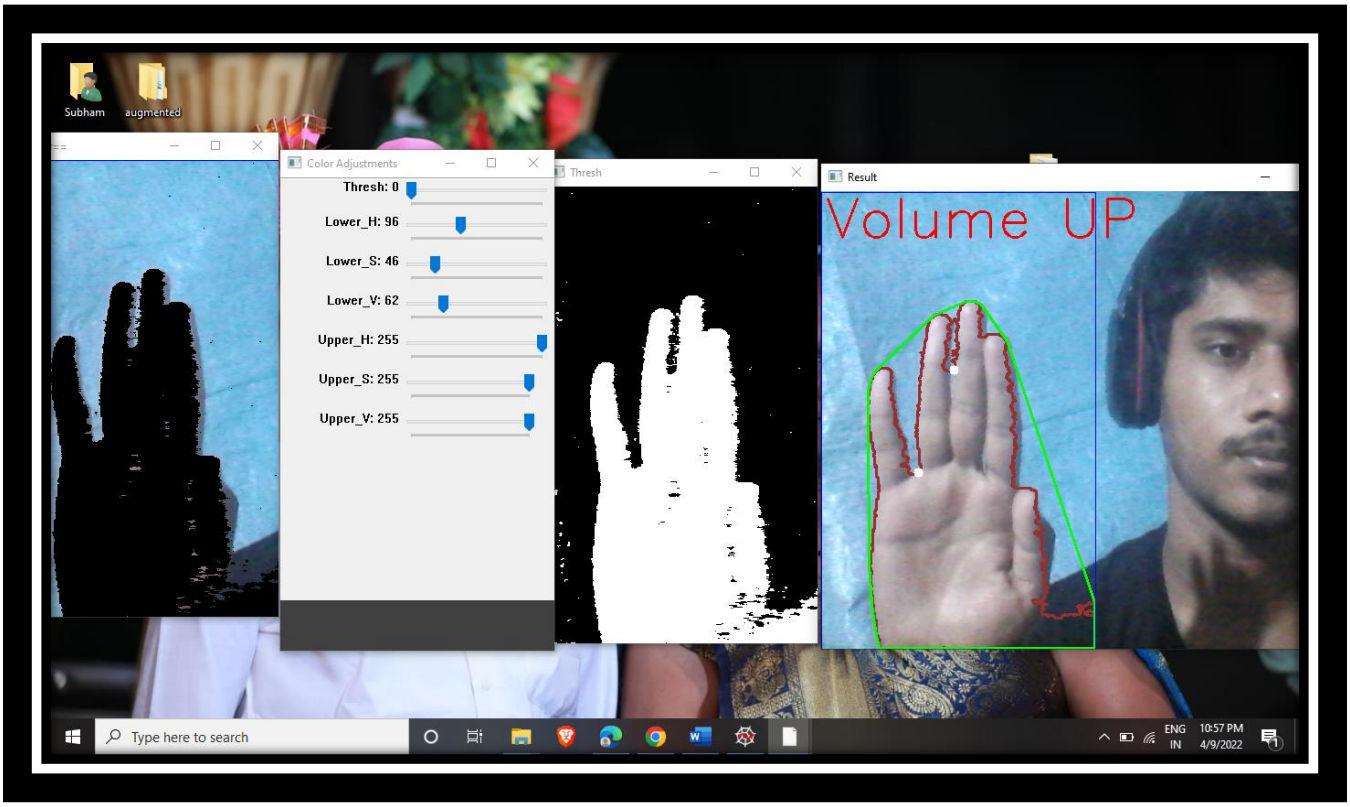
To Run the Project Just Click on **Run** Tab OR **F5**





After Running Project 4 Screen will appear

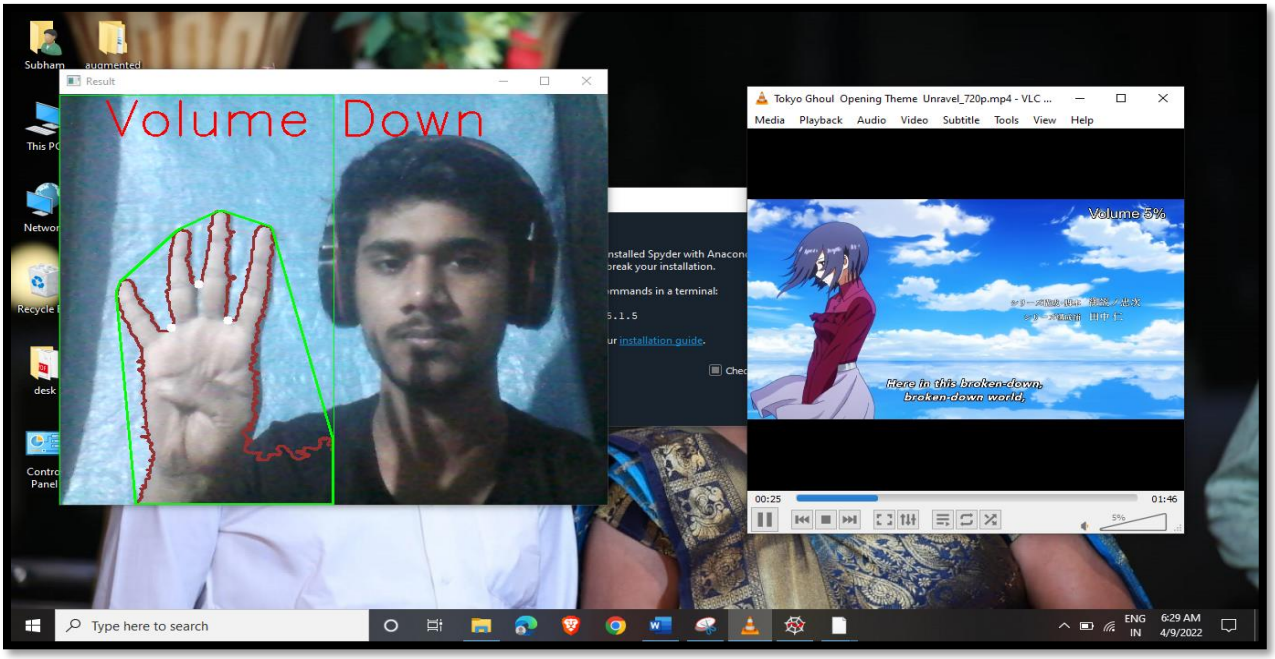
1- Result
2- Thresh
3- Color Adjustments
4- Filters

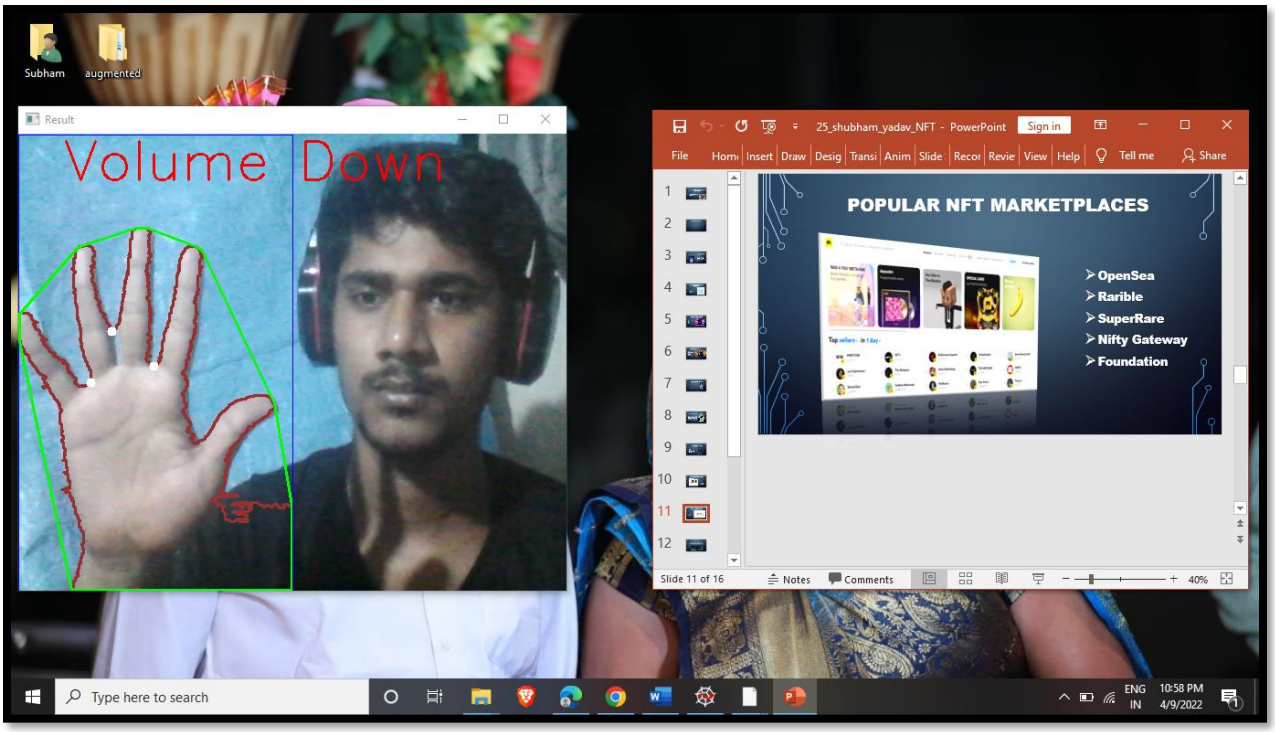


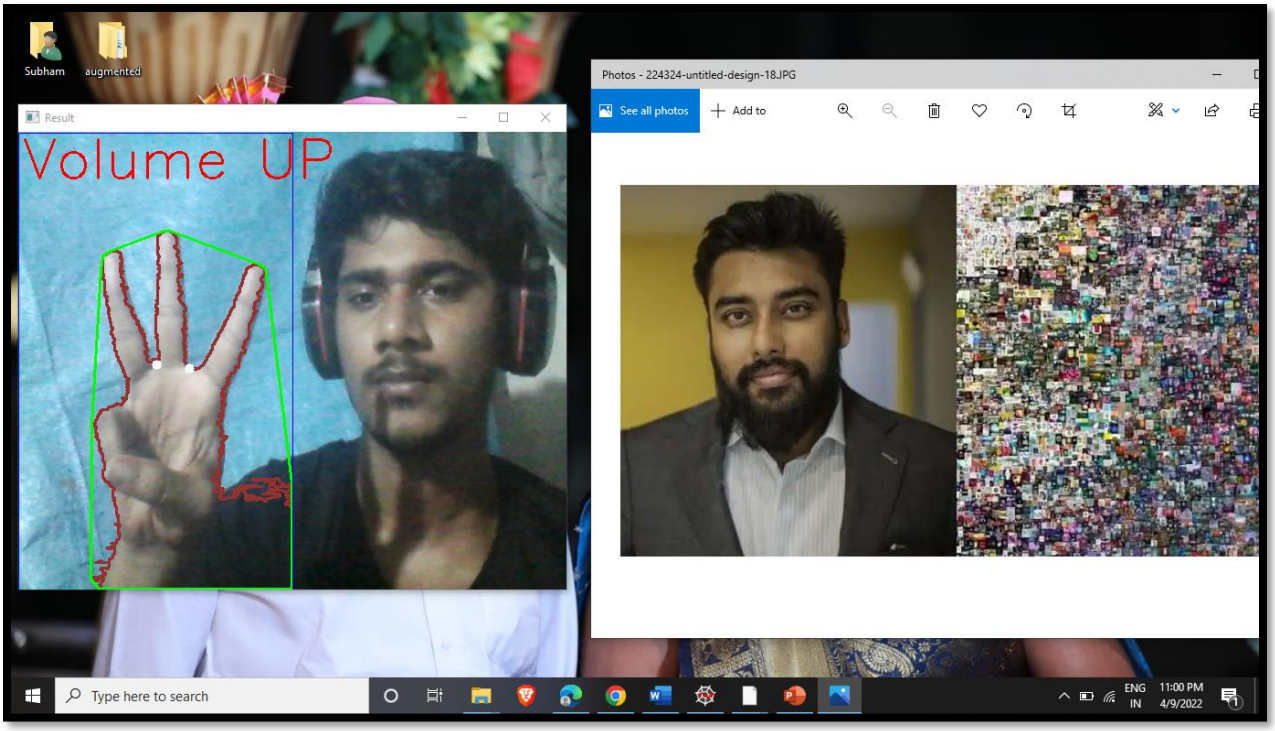
Adjust the Trackbar in Color Adjustment Screen

- ✓ **Lower_H :96**
- ✓ **Lower_S :46**
- ✓ **Lower_V :62**
- ✓ **And other remain Same**

- ✓ **The Red Line is called Contours.**
- ✓ **The Green Line is Called Convex.**
- ✓ **And White dots are called convexity Defects.**







CONCLUSION

1. This is to conclude that the hand gesture recognize system is recognizing the gestures appropriately.

2. After successfully recognizing the hand gestures it is giving expected results as output.

3. Currently i have introduced gestures for increase and decrease of volume in video and next and previous slides and up and down word file, etc.

4. Limitation of this hand gesture recognize system is that it requires good processor to take and give response to the hand gestures quickly.

5. In future there is a whole new world for this hand gesture recognize system to evolve where the main key elements will be virtual reality/augmented reality.