CURSOR CONTROL USING HAND GESTURE RECOGNITION

Research paper

Name: Anushree Abhay Bhuskute

Class: TY BSc (CS) Division: B

Roll no: 228811

INTRODUCTION:

Computer technology has grown tremendously over the past few years and has now become an integral part of everyday life. The devices that help in Human Computer Interaction (HCI) are keyboard, mouse, etc. These devices are not suitable for Human Robot Interaction (HRI). There has been research for replacing mouse for Human Computer Interaction. The most viable and natural option for replacing mouse is by using hand gestures. This project thus focuses on developing cursor control using hand gesture recognition.

PROBLEM STATEMENT:

The project is completely done for the purpose of learning the direct interaction of humans with computer. This takes the user experience to next level.

The proposed system is vision based, which uses image processing techniques and input from webcam of computer. The input frame would be captured from the webcam and then processed. The hand processing would be done by landmarking the parts of the hand using the mediapipe library. Hand tracking would be used to navigate the computer cursor and hand gestures would be used to perform mouse functions such as right click, left click and double click.

The scope of the project would therefore be to design a vision-based cursor control system, which can perform the previously stated mouse functions.

FUNCTIONAL REQUIREMENTS:

The system must support the following functionalities:

- The software must trace the video within range.
- Process the video into a series of frames(snaps)
- Find out the landmarks on the hand detected.
- Compare the gestured pattern with a set of defined gestures.
- Process the gesture and do the functionalities as described by the developer.

NON-FUNCTIONAL REQUIREMENTS:

- Speed: Response time should be less
- Number of hands simultaneously detected: 2

TECHNICAL REQUIREMENTS:

- A computer with inbuilt or external webcam.
- A computer having 8GB ram
- Programming language used: Python
- All the required (listed later) libraries must be installed.

FINAL DELIVERABLES:

System has a gesture recognition system which translates image to user hand gesture which can perform the following functions of mouse

- Neutral gesture
- Cursor movement
- Right click
- Left click
- Double click

PROGRAM FLOW:

Step 1: Import Libraries

The required python libraries include:

i. Open CV (cv2): used for computer vision, Machine Learning, and Image Processing
ii. Mediapipe: used for processing time series data like video, audio, etc.
iii. Pyautogui: used to automate gui and for keyboard and mouse control.
iv. Enum: used to count the number of iterations
v. Ctypes: allows C compatible datatypes
vi. Comtypes: Component Object Model based on Ctypes

Step 2: Initialise gesture control attributes like camera height, width, etc.

Step 3: Set the hand landmark points using mediapipe library

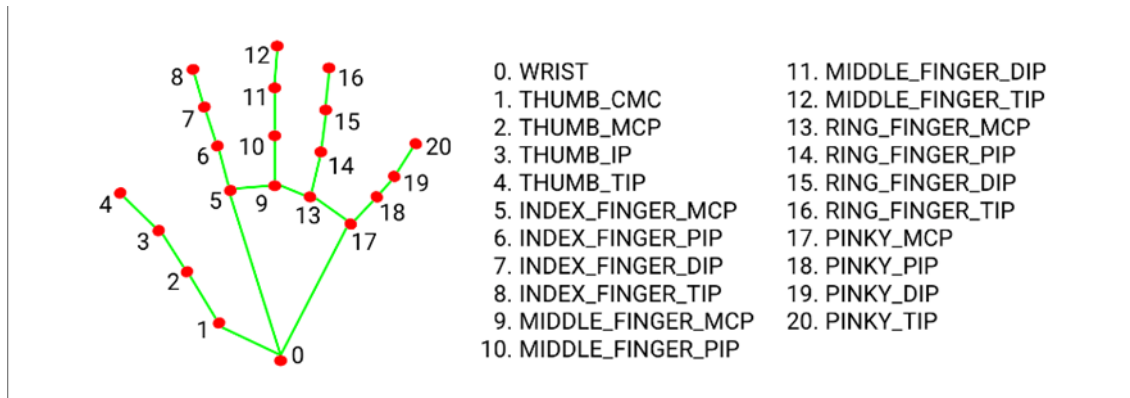The standard landmarks provided by mediapipe library are shown below.

0. WRIST	11. MIDDLE_FINGER_DIP
1. THUMB_CMC	12. MIDDLE_FINGER_TIP
2. THUMB_MCP	13. RING_FINGER_MCP
3. THUMB_IP	14. RING_FINGER_PIP
4. THUMB_TIP	15. RING_FINGER_DIP
5. INDEX_FINGER_MCP	16. RING_FINGER_TIP
6. INDEX_FINGER_PIP	17. PINKY_MCP
7. INDEX_FINGER_DIP	18. PINKY_PIP
8. INDEX_FINGER_TIP	19. PINKY_DIP
9. MIDDLE_FINGER_MCP	20. PINKY_TIP
10. MIDDLE_FINGER_PIP	

Fig. 1 Image source: mediapipe

Step 4: Convert the inputted image from BGR to RGB colour format

Step 5: Setting up multihanded mode with one major and one other minor hand

Step 6: Displaying the video to the user by using open cv imshow() function along with landmarks drawn on the hand by mp_drawing function of mediapipe library.

Step 7: Locate the hand to get the cursor position

Step 8: Calling the get_gesture() function to detect the gesture.

Step 9: Analysing the gesture done by the user

    i.   If palm is detected set gesture to neutral gesture

    ii.   If the ratio between tip of first two fingers and base of first two fingers is greater than 1.7, then set gesture to V gesture

    iii.   Else if the distance between tip of first two fingers is less than 0.1, then set gesture to Two fingers closed.

    iv.   Else set gesture to middle finger up

Step 10: Set the frames and increment the frame count.

Step 11: See the gesture and set control as per requirement using pyautogui library functions.

    i.   If 'V gesture' is detected function set is 'Cursor movement' pyautogui.moveTo() function

    ii.   If 'Middle finger' gesture is detected function is 'Left Click' pyautogui.click()

    iii.   If 'Index finger' gesture is detected, function set is 'Right Click' pyautogui.click(button= 'right')

    iv.   If 'Two fingers closed' gesture is detected, function set is 'Double Click' pyautogui.doubleclick()

    v.   If 'palm' is detected, function set is 'Neutral gesture'

HOW DOES PYTHON LIBRARIES WORK?

1. **Open CV** :

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human.

In this project, opencv is used to capture video input from the webcam and display the output.

cv2.videocapture(0) is used to capture the video.

When we capture the video, we need to set the camera height and width. This is done using: GestureController.CAM_HEIGHT = GestureController.cap.get(cv2.CAP_PROP_FRAME_HEIGHT)

GestureController.CAM_WIDTH = GestureController.cap.get(cv2.CAP_PROP_FRAME_WIDTH)   where GestureController is the programmer defined class which handles the initial set up of the system.

The video captured from opencv is in bgr colour format which we need to convert into rgb colour format for further processing. This is done using the following command: image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)

Finally, the image is displayed on the users computer screen using imshow() function as follows: cv2.imshow('Gesture Controller', image)

2. **Mediapipe:**

MediaPipe offers cross-platform, customizable ML solutions for live and streaming media. Mediapipe depends on OpenCV for video data handling.

We use mediapipe to detect hand landmarks and draw the landmarks on the output window. Predefined set of hand landmarks are already set in mediapipe library. Shown in fig 1. To perform the actions, we use the following steps:

1.mp_drawing = mp.solutions.drawing_utils : to set up the drawing function of hands landmarks on the image

2: mp_hands = mp.solutions.hands : initializing the hand class

3: mp_hands.Hands(max_num_hands = 2,min_detection_confidence=0.5, min_tracking_confidence=0.5) : TO hold the landmark points

4: mp_drawing.draw_landmarks(image, hand_landmarks,
        mp_hands.HAND_CONNECTIONS): to draw the landmarks on output screen.

3. **Pyautogui:**

Python pyautogui library is an automation library that allows mouse and keyboard control. Or we can say that it facilitates us to automate the movement of the mouse and keyboard to establish the interaction with the other application using the Python script.

In this project, pyautogui is used to set the functions of the gesture recognised.

1. pyautogui.moveTo() function is used for the movement of cursor.
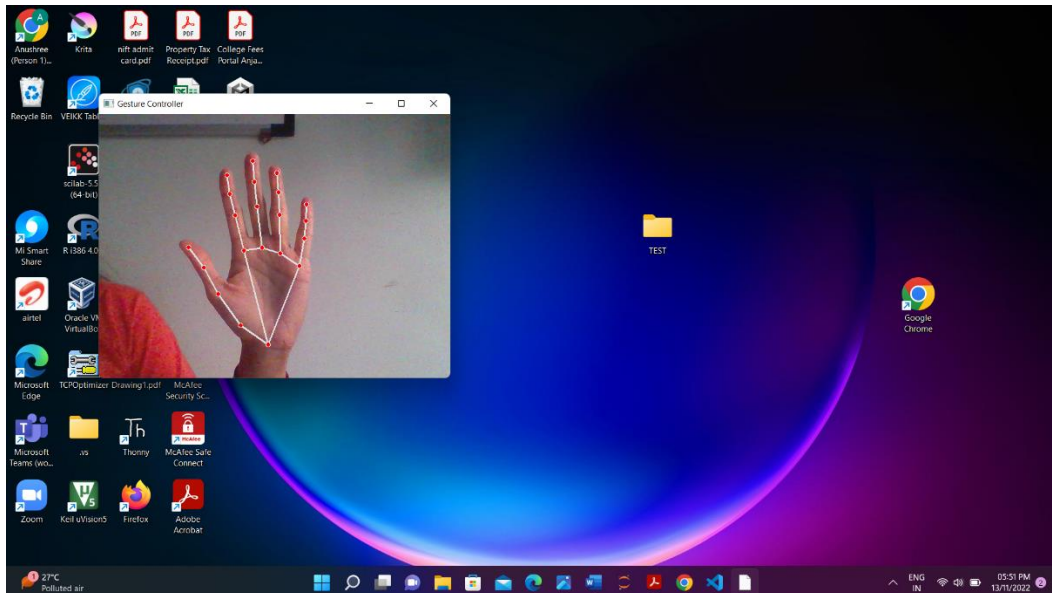2. pyautogui.click() is used to set the left click function.
3. pyautogui.click(button='right') is used to set the right click function of the cursor.

4. pyautogui.doubleclick() is used to set the double click function of the cursor.
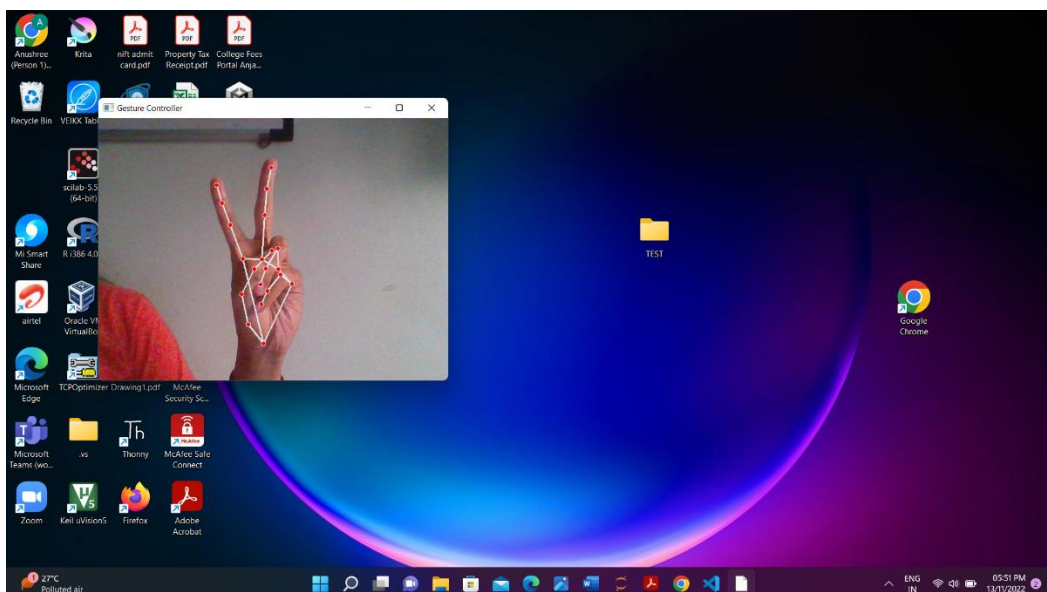
WORKING OF THE PROGRAM:

1. Neutral Gesture:
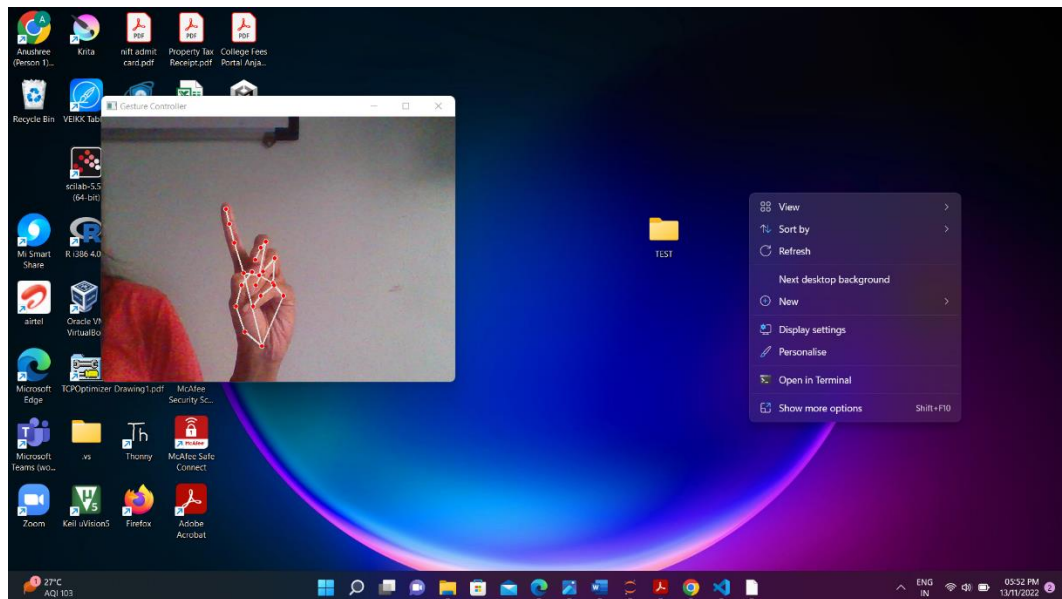   When open palm is detected, it is considered as a neutral gesture.

2. V gesture (Cursor movement):
   When the ratio of distance between tip of first two fingers and the distance between base of first two fingers is greater than 1.7, V gesture is detected.
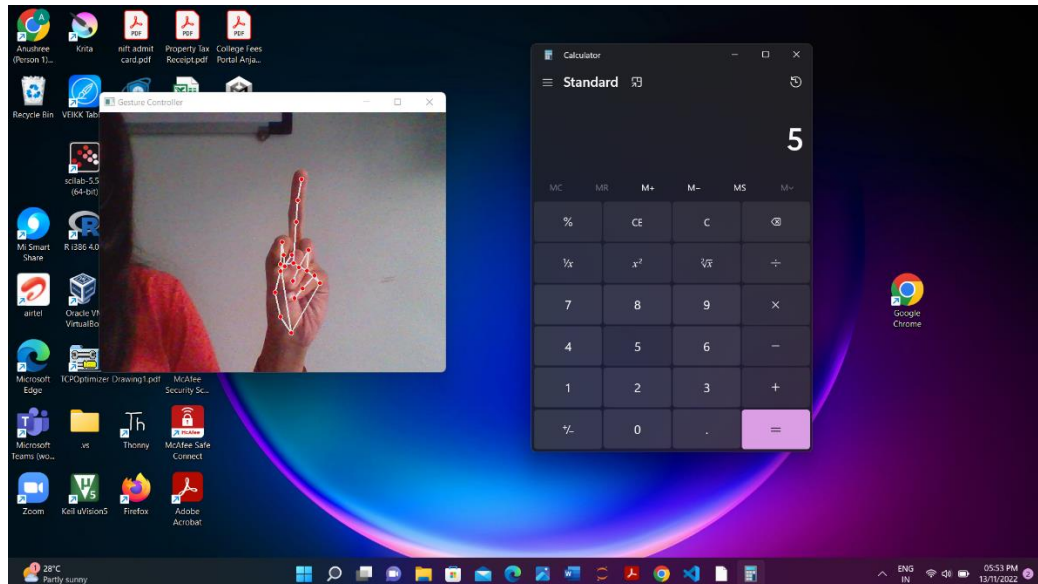
3. Right click:
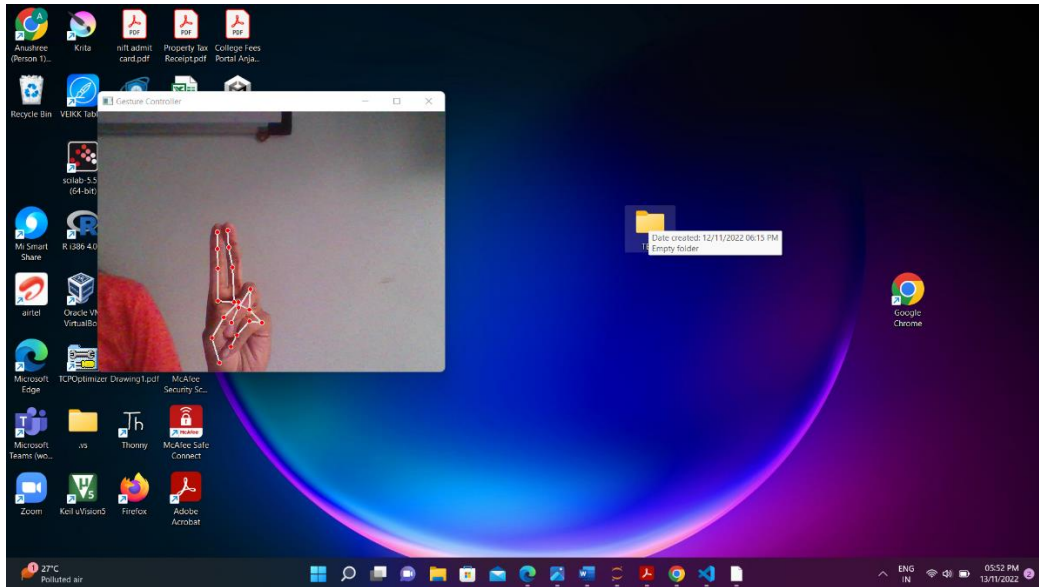   When the index finger is detected, right click gesture is detected.

4. Left click:
   When the middle finger is detected, left click gesture is detected.

5. Double click
   When the distance between tip of first two fingers is less than 0.1, gesture detected is double click.

CONCLUSION:

The system provides user to control the computer cursor using hand movements.

In this project, gesture recognizer is made with the help of mainly OpenCV toolkit and mediapipe library. Other python libraries like pyautogui are also used to automate the Graphical User Interface and control the cursor movements.

The system currently has the following functions which are:

    i.   Cursor movement
    ii.   Left click
    iii.   Right click
    iv.   Double click

The system function can be further expanded as per the requirements of the user. We can further make necessary improvements in the system such that it will make interaction of differently abled people with computers easier. For the Social use, this system is very helpful for detecting hand gesture in real time video capture.