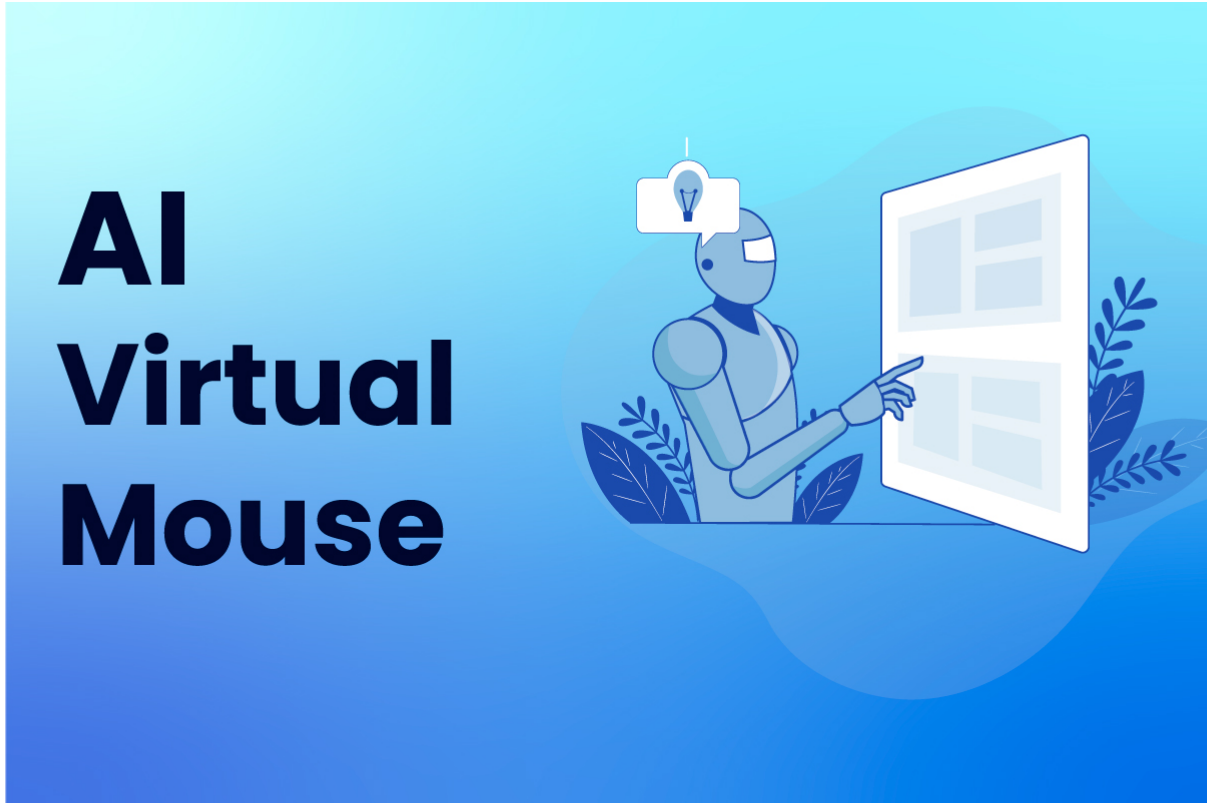
SESSION 8 - Project Virtual Mouse Cursor

TOC :

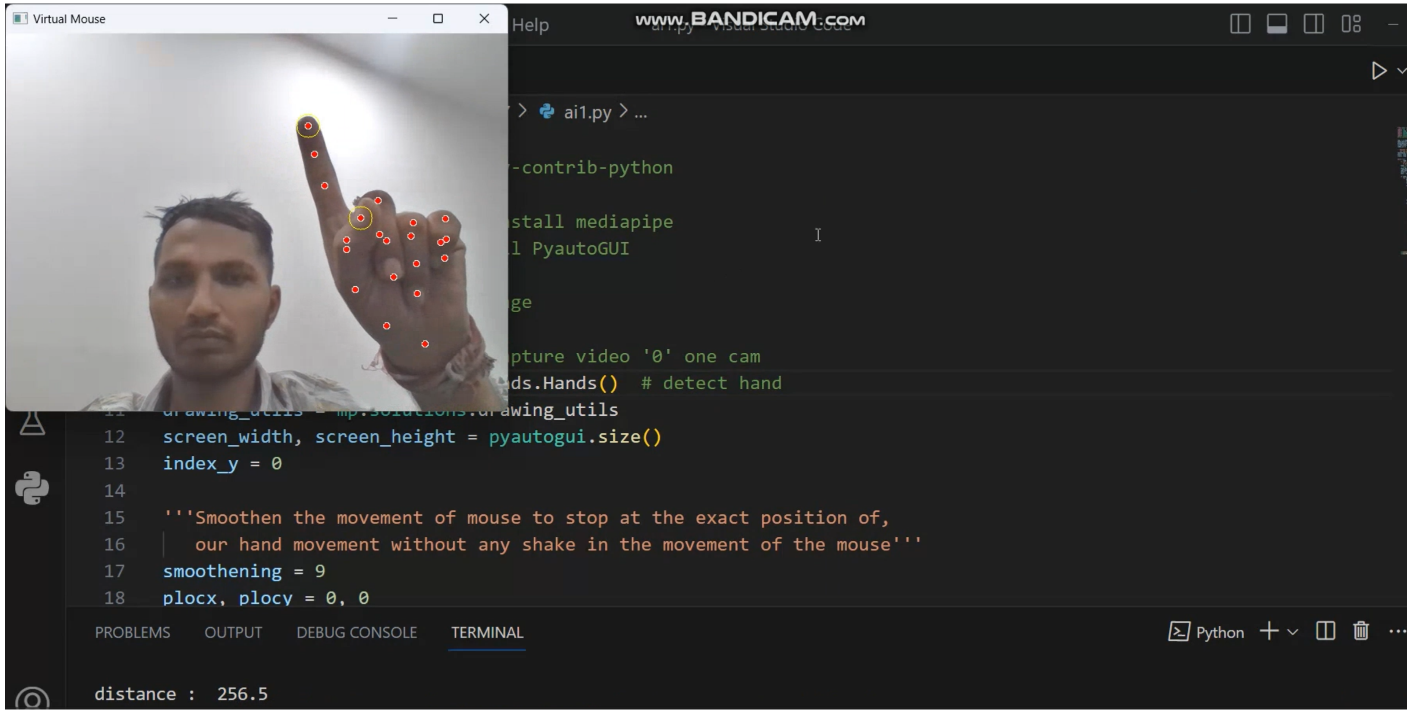
1. Import the necessary packages and Convention



1. Introduction

The mouse is the one of the wonderfull inventions of the Human Computer Interaction(HCI) Technology.

Currently we are using wired or may be wireless mouses,In real time cases some computers may not support for a physical mouse or may some users may be dealth with some hand problems or handicap and cannot use physical mouse, so this Hand controlled AI Virtual Mouse can be used to overcome this problem. Making a user to control the mouse by reducing the computer human interaction.



```
-contrib-python

stall mediapipe
l PyautoGUI

ge

pture video '0' one cam
ds.Hands()  # detect hand
awing_utils
screen_width, screen_height = pyautogui.size()
index_y = 0

'''Smoothen the movement of mouse to stop at the exact position of,
   our hand movement without any shake in the movement of the mouse'''
smoothening = 9
plocx, plocy = 0, 0
```

```
distance :  256.5
```

FUNCTIONALITIES:

- This was built using the openCV-python and mediapipe for Detecting and processing the image and mediapipe an open source cross-platform developed by google for media processing and ready-to-use ML solutions for computer vision tasks.
- Our Hand Controlled Virtual mouse can able to move the mouse anywhere on the screen and can able to perform the click operation.
- Our Index finger can be used to move the mouse over the screen.
- When Our Index finger and Thumb come close to each other or touch each other then it performs the click operation.
- And the PyautoGUI for programmatically control the mouse and keyboard.

Install the following necessary pip

```
- pip install opencv-python
- pip install numpy
- pip install matplotlib
- pip install mediapipe
- pip install PyautoGUI
- pip install PIL
```

3. Project Implementation

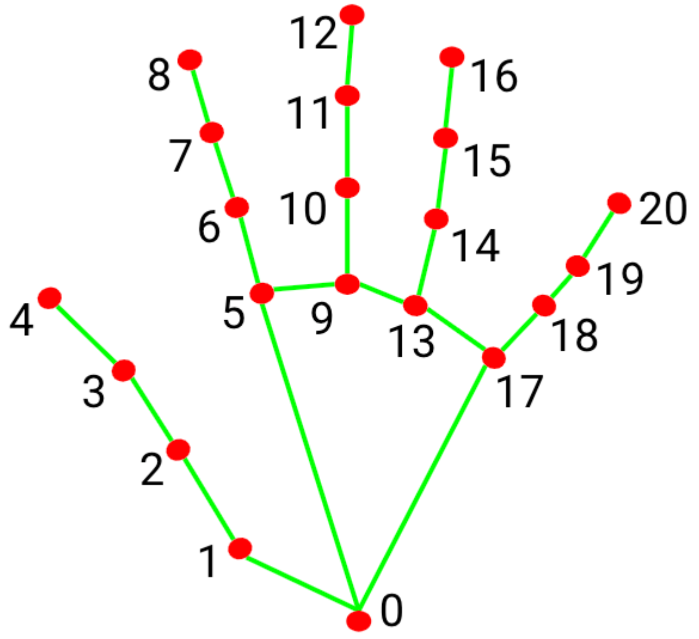
In [ ]:

```python
from turtle import ht
import cv2  # pip install opencv-contrib-python
import numpy as np
import mediapipe as mp  # pip install mediapipe
import pyautogui    # pip install PyautoGUI
import matplotlib.pyplot as plt
from IPython.display import Image
```

In [ ]:

```python
points = Image(url="images/hand_landmarks.png")
points
# img=cv2.imread("hand_landmarks.png")
# RGB_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
# plt.imshow(img)
```

Out[ ]:



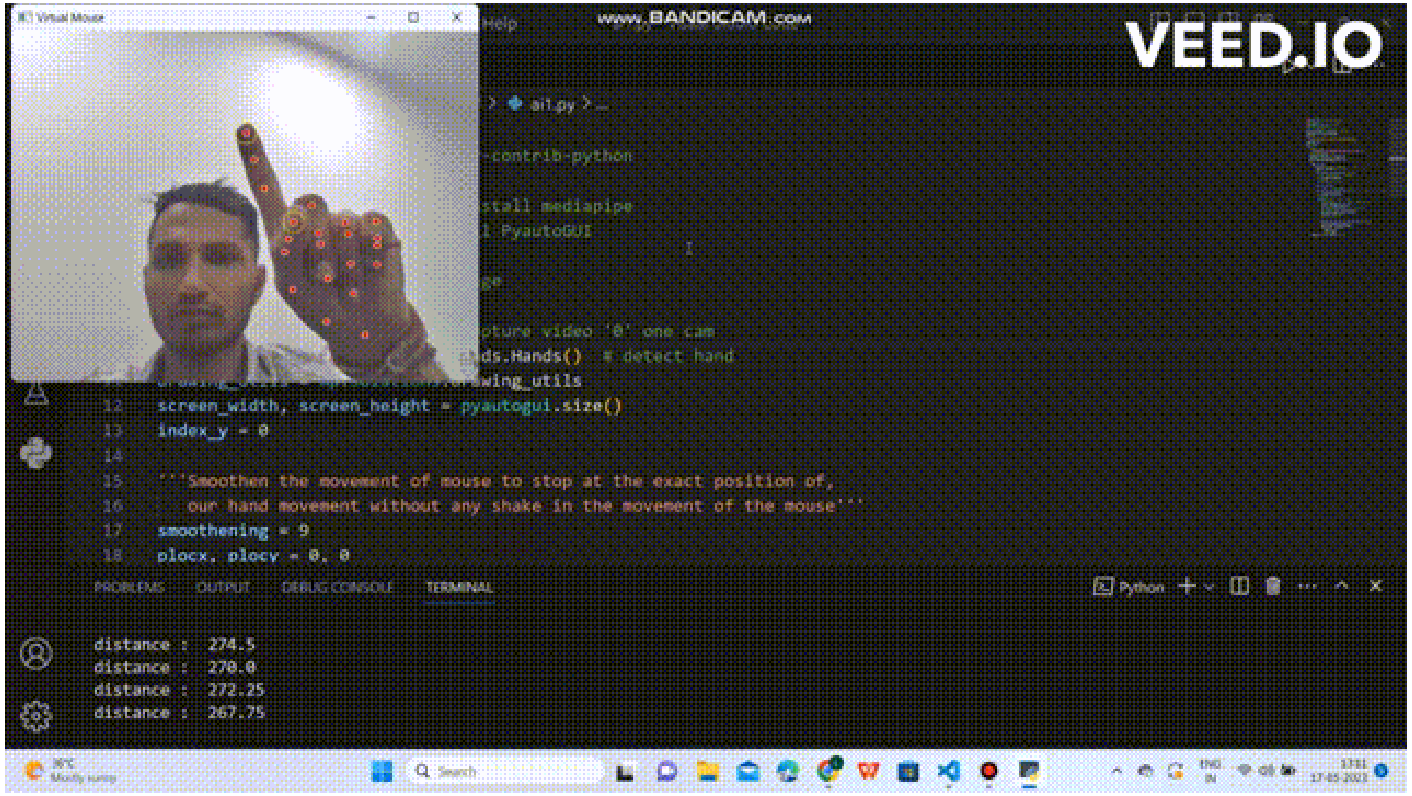
0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP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

In [ ]:
```python
cap = cv2.VideoCapture(0)   # capture video '0' one cam
hand_detector = mp.solutions.hands.Hands()  # detect hand
drawing_utils = mp.solutions.drawing_utils
screen_width, screen_height = pyautogui.size()
# print(screen_width, screen_height)
index_y = 0
```

In [ ]:
```python
'''Smoothen the movement of mouse to stop at the exact position of,
   our hand movement without any shake in the movement of the mouse'''
smoothening = 9
plocx, plocy = 0, 0
clocx, clocy = 0, 0
x1,y1 = 0,0
```

In [ ]:
```python
while True:
    _, frame = cap.read()   # read data from cap
    '''Flip the frame or screen since the camera shows the mirror image,
       of our hand and moves in opposite direction so we need to flip the screen'''
    frame = cv2.flip(frame, 1)
     # shape gives frame height and width using shape
    frame_height, frame_width, _ = frame.shape
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)  # detect on rgb frame color
    output = hand_detector.process(rgb_frame)
    hands = output.multi_hand_landmarks # hand landmark

    if hands:
        for hand in hands:
            drawing_utils.draw_landmarks(frame, hand)   # see landmarks on frame
            landmarks = hand.landmark
            # we use our index finger tip move the mouse

            for id, landmark in enumerate(landmarks):   # add counter
                # show the landmarks on kernel in x and y axis
                # x and y axis is multiplies by the height and width to get the x and y axis on the frames
                x = int(landmark.x*frame_width)
                y = int(landmark.y*frame_height)
                # Index finger tip point number is 8
                # and draw a boundary to the point a circle
                if id == 8:
                    cv2.circle(img=frame, center=(x,y), radius=15, color=(0, 255, 255))
                    # pyautogui.moveTo(x,y)
                    index_x = (screen_width/frame_width)*x
                    index_y = (screen_height/frame_height)*y
                    # co-ordinates need to be changed
                    # smoothining varies with the change in the smoothening factor
                    clocx = plocx + ( index_x - plocx ) / smoothening
                    clocy = plocy + ( index_y - plocy ) / smoothening
                    pyautogui.moveTo(clocx, clocy)
                    plocx, plocy, x1, y1 = clocx, clocy, landmark.x, landmark.y

                # thumb tip point number is 4

                if id == 4:
                    cv2.circle(img=frame, center=(x,y), radius=15, color=(0, 255, 255))
                    thumb_x = (screen_width/frame_width)*x
                    thumb_y = (screen_height/frame_height)*y
                    print('distance : ', abs(index_y - thumb_y))
                    if abs(index_y - thumb_y) < 70:
                        print('click')
                        pyautogui.click()
                        pyautogui.sleep(1)
    cv2.imshow('Virtual Mouse', frame)  # show image
    cv2.waitKey(1)  # waits for key infinitely
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Checkout the Demo Below :



```
-contrib-python

stall mediapipe
l PyautoGUI

ge

pture video '0' one cam
ds.Hands()  # detect hand
wing_utils
screen_width, screen_height = pyautogui.size()
index_y = 0

'''Smoothen the movement of mouse to stop at the exact position of,
   our hand movement without any shake in the movement of the mouse'''
smoothening = 9
plocx, plocy = 0, 0
```

```
distance :  274.5
distance :  270.0
distance :  272.25
distance :  267.75
```