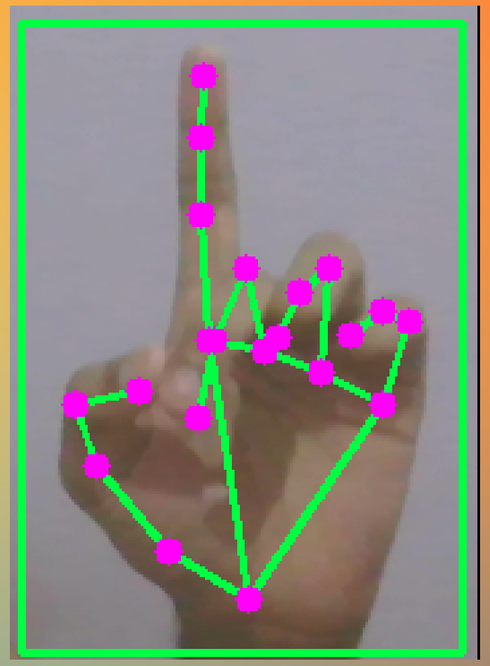
AI VIRTUAL MOUSE
and
VOLUME CONTROLLER



By
Subham Subhasish Panda
19BIT0093

Contents

Here's what we'll cover:

Introduction

- As technology is developing people are having smaller and smaller devices
- The importance of Human Computer Interaction (HCI) and in particular vision based gesture and object recognition
- In this project, the approach of using a video device to control the mouse system and tasks like click, double click and scrolling function of a mouse to control volume of the system is suggested

Objective

- Most laptops have a physical touch pad or desktop use a physical device such as mouse or joystick to control the cursor on the screen
- Devices using touch pad or mouse using scrolling function of the mouse to control the volume of the system.
- The aim is to manage the application of a mouse by using gestures and simple hand movement
- The aim is to remove the requirement of having a physical device
- This could reduce cost of hardware required to build a computer

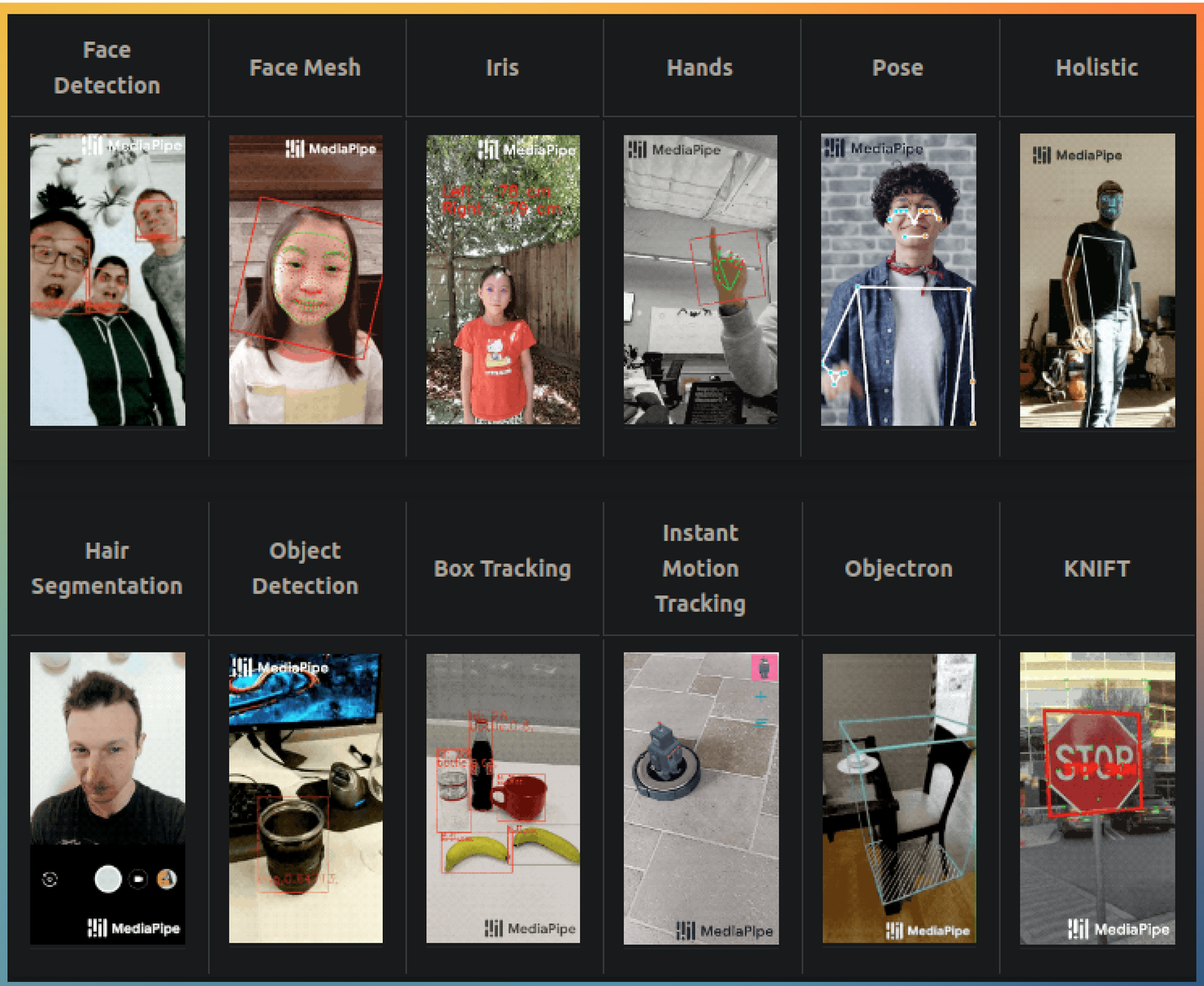
Libraries used for the Project

The following python based libraries were use for the project:

- MediaPipe
- Autopy
- Numpy
- OpenCV-Python

Steps Involved in Processing

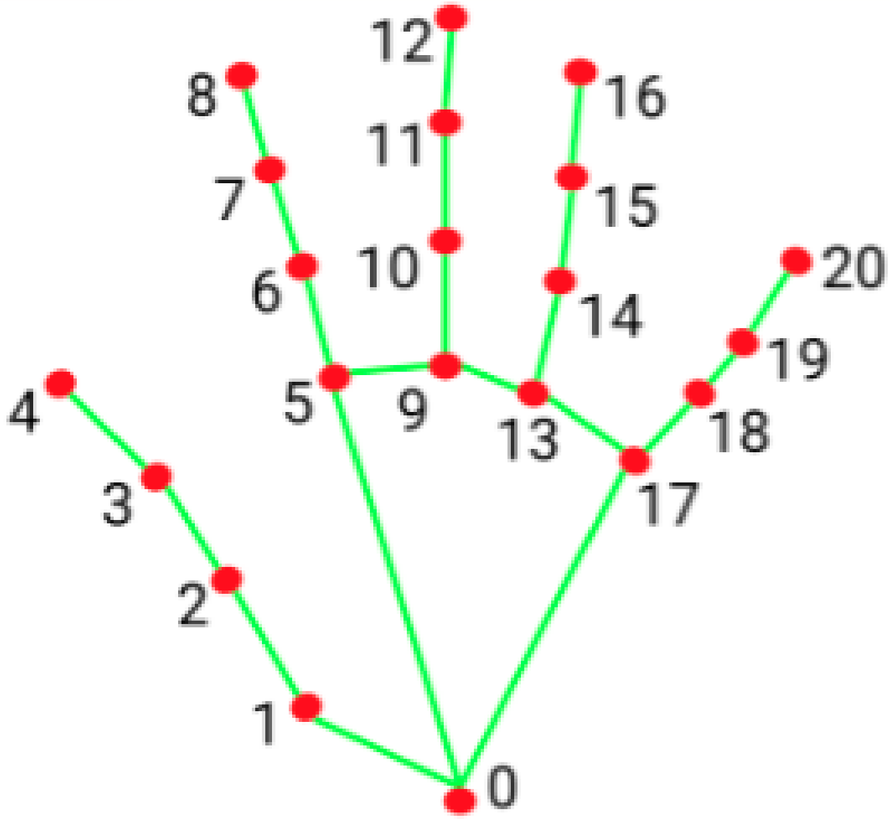
- The library of MediaPipe has been use in the project
- MediaPipe is a framework by google thats provides solutions to live media.



Steps Involved in Processing

Hand landmarks in MediaPipe:

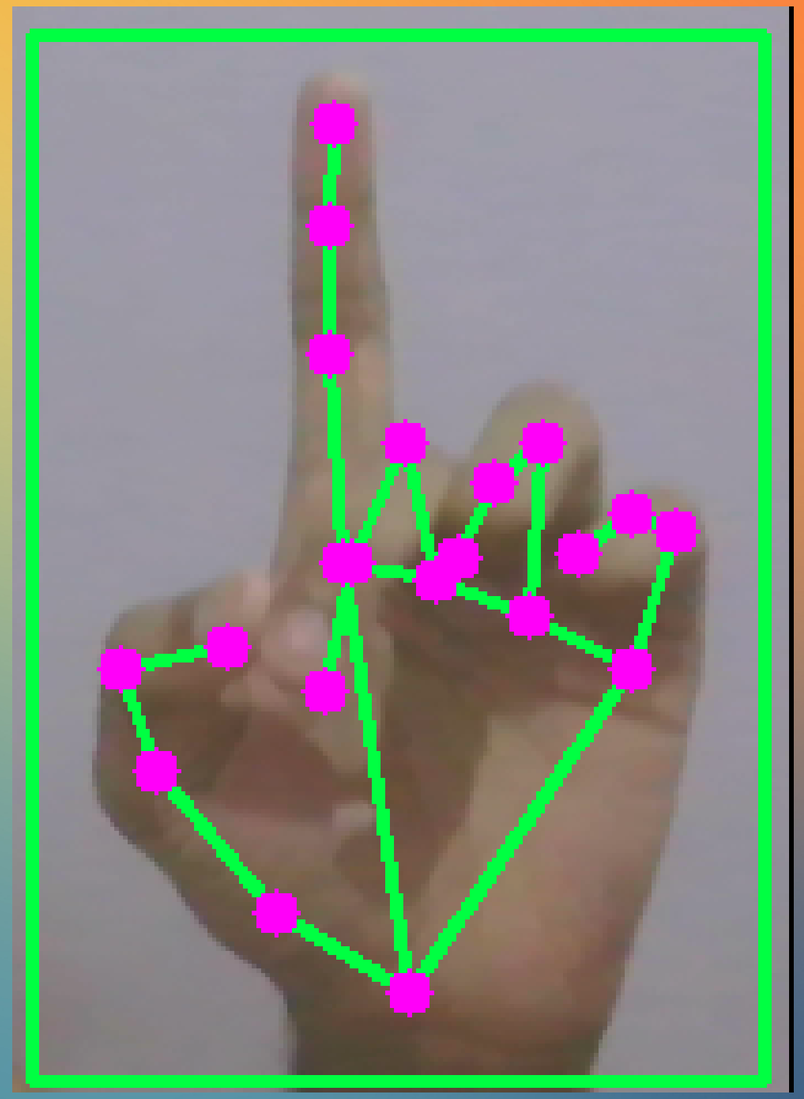
- Mediapipe has over 50,000 images trained in their models
- Media pipe recognized hands when images are provided to the library and returns landmarks which are points designated to identify various points on the hand and they have been given a number



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

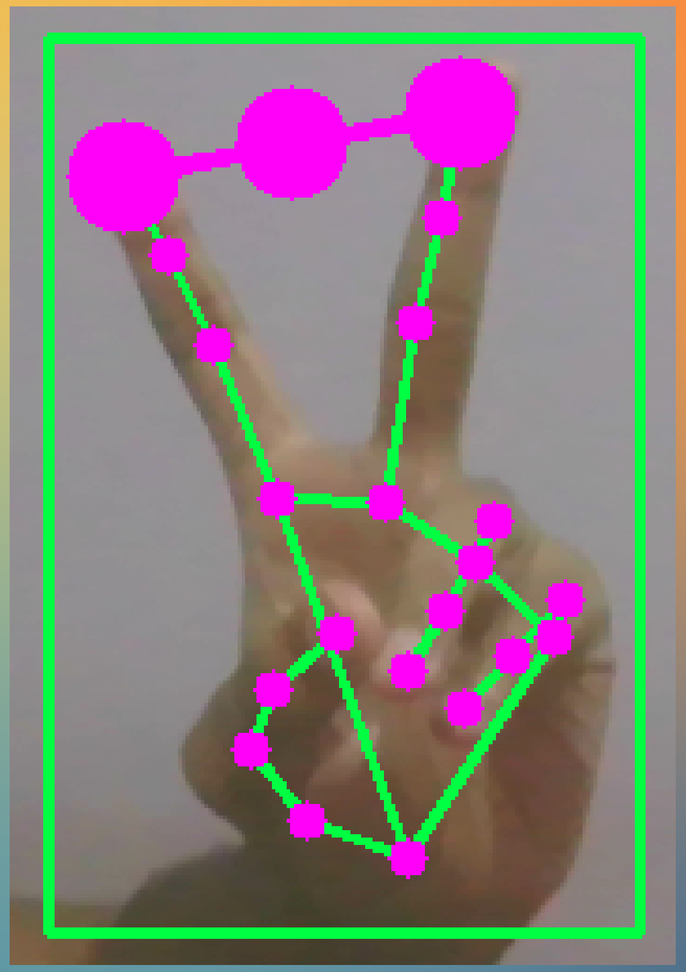
Steps Involved in Processing

- We feed the image data of each moment from the live cam to mediapipe which recognises the hands if present and then returns the coordinates of the various 20 points set by mediapipe.
- We then use those coordinates to tract the position of the index finger and move the mouse to the calculated coordinate on the screen
- To move the mouse we use the autopy library which is a GUI automation library in python



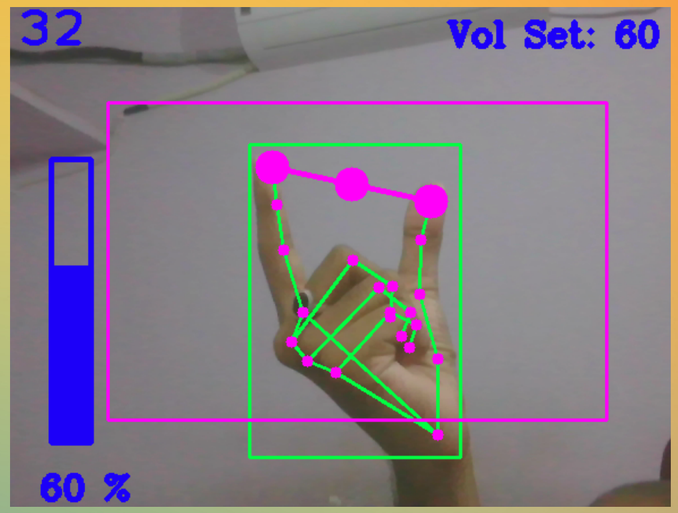
Steps Involved in Processing

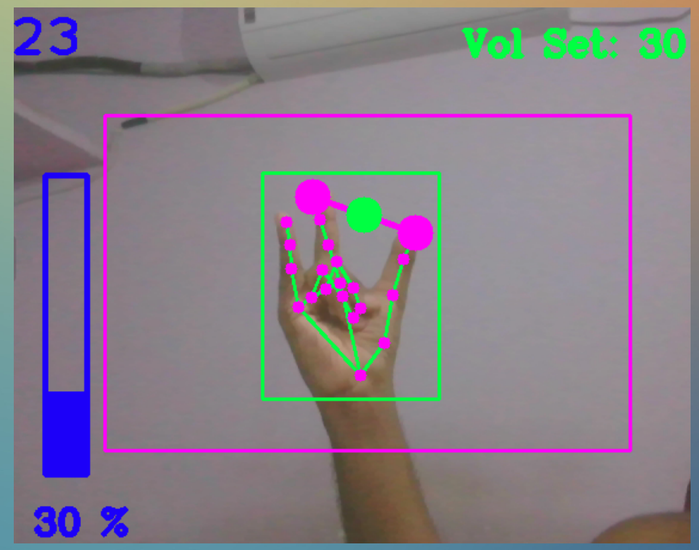
- Next we find the distance between the points on the tips of the index finder and middle finger and draw a line between them.
- We then see if the length is smaller than a particular value, we perform a click.
- This allows us to click on anything by snapping the raised index finger and middle finger



Steps Involved in Processing

- We check if the pinky finger is down and the index fingers and thumb is up, then we, calculate volume percentage using the distance between the tip of the index finger and thumb.
- Depending on the distance between the two fingers we calculate a volume percentage.
- Now when the user raise his pinky finger we set the volume percentage which was calculated by the distance between the index finger and thumb





Conclusion

- The virtual mouse has pretty good accuracy, but fails sometimes when the background color is indistinguishable with the skin color.
- In such cases sometimes the coordinates of points is miscalculated
- This can be improved by training more and more images in various lighting conditions various backgrounds etc

- Future scope of this project includes adding more and more gestures and maybe even use palm for various gestures
- We can let the user create his own set of gestures for various commands