

Registration Number: 19BIT0093

Name: Subham Subhasish Panda

PROJECT REPORT

(COMPUTERS IN SOCIETY 2 CREDIT COURSE)

AI VIRTUAL MOUSE

And

VOLUME CONTROLLER

With

HAND GESTURES

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:

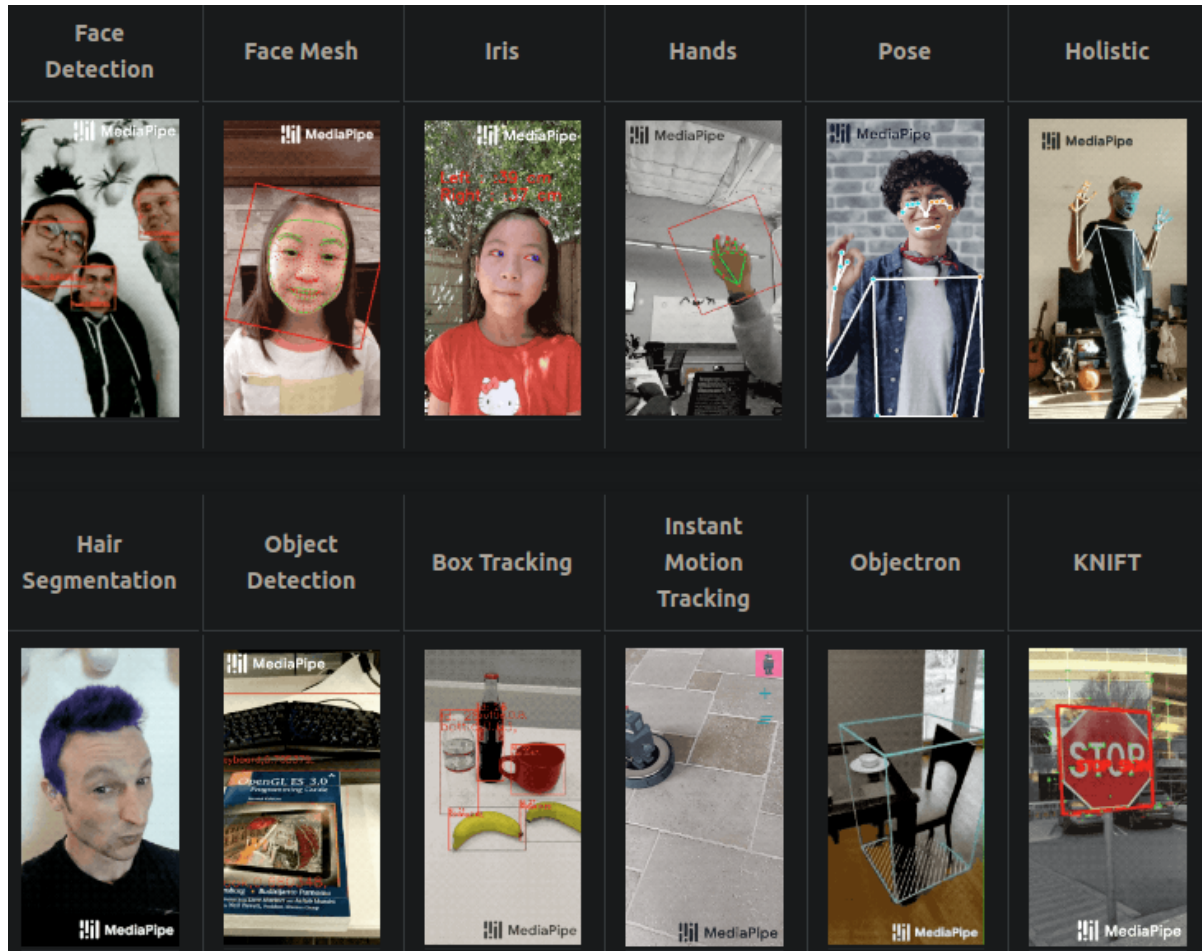
- Most laptops have a physical touch pad or desktop use a physical device such as mouse or joystick to control the cursor on the screen
- Devices using touch pad or mouse using the scrolling function of the mouse to control the volume of the system.
- The aim is to manage the application of a mouse by using gestures and simple hand movement
- The ai 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:

- MediaPipe
- Autopy
- Numpy
- OpenCV-Python

Steps involved in processing:

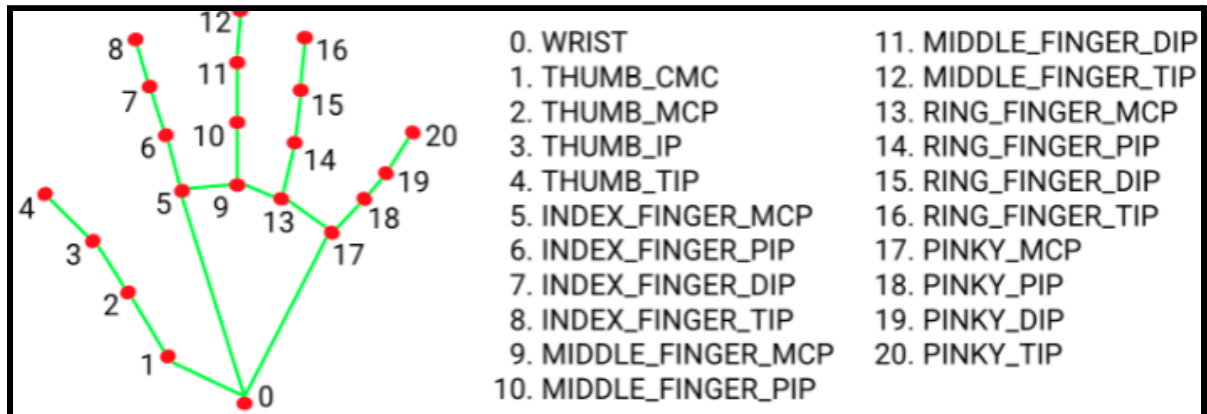
- The library of MediaPipe has been use in the project
- MediaPipe is a framework by google thats provides solutions to live media.



Some of the Major models provided by MediaPipe

Hand landmarks in mediapipe:

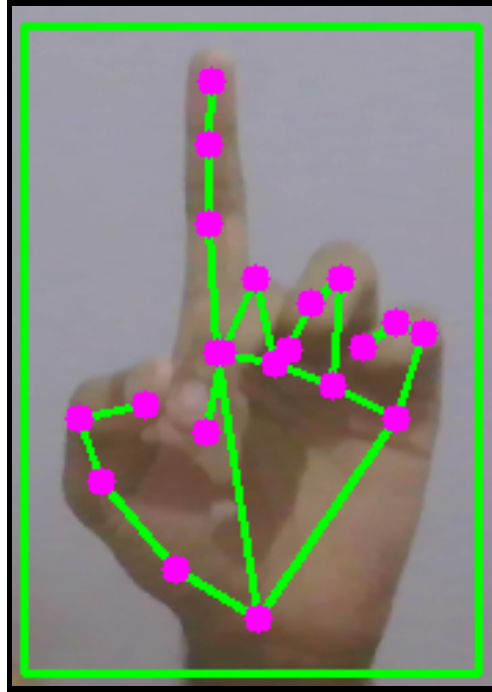
- Mediapipe has over 50,000 images trained in their models
- Media pipe recognized hands when images are provided to the library and returns landmarks which are points designated to identify various points on the hand and they have been given a number



The figure shows the 20 points of a hand as recognised by mediapipe and the position of each of those 20 points

Steps for Mouse Cursor Movement:

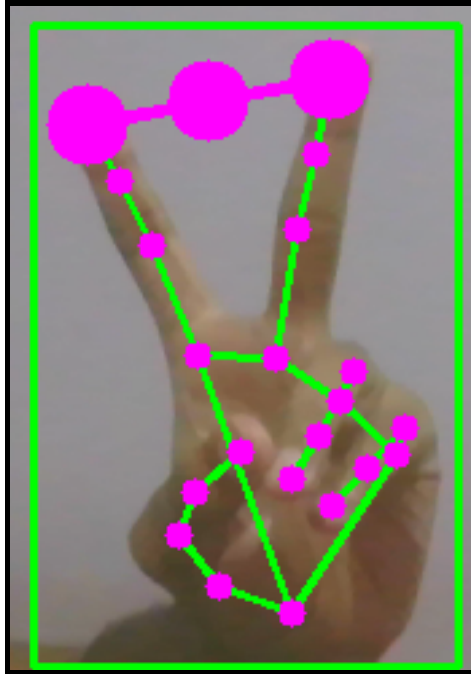
- We feed the image data of each moment from the live cam to mediapipe which recognises the hands if present and then returns the coordinates of the various 20 points set by mediapipe.
- We then use those coordinates to track the position of the index finger and move the mouse to the calculated coordinate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



The figure shows the 20 points on the hand recognized when hand image is provided to mediapipe. We then move the raised index finger to move the mouse

Steps for Mouse Click on the Screen:

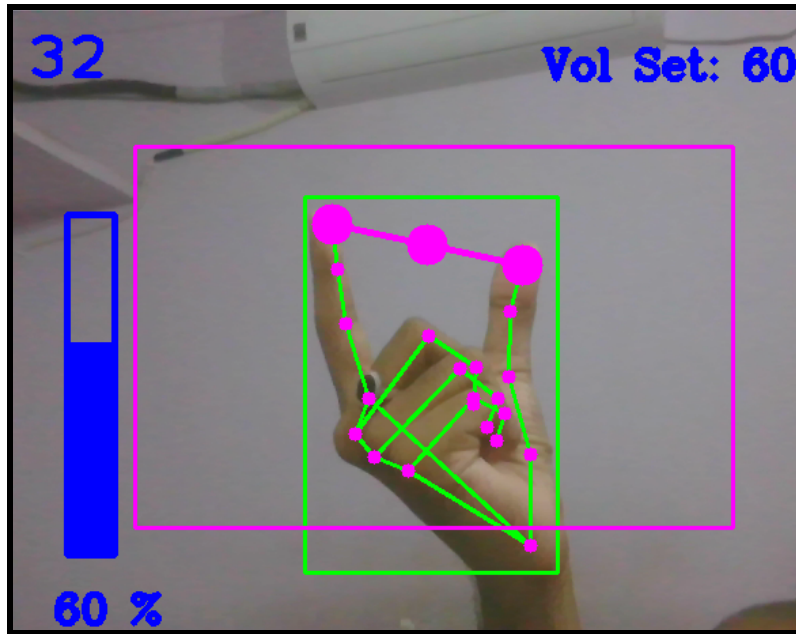
- Next we find the distance between the points on the tips of the index finger and middle finger and draw a line between them.
- We then see if the length is smaller than a particular value, we perform a click.
- This allows us to click on anything by snapping the raised index finger and middle finger



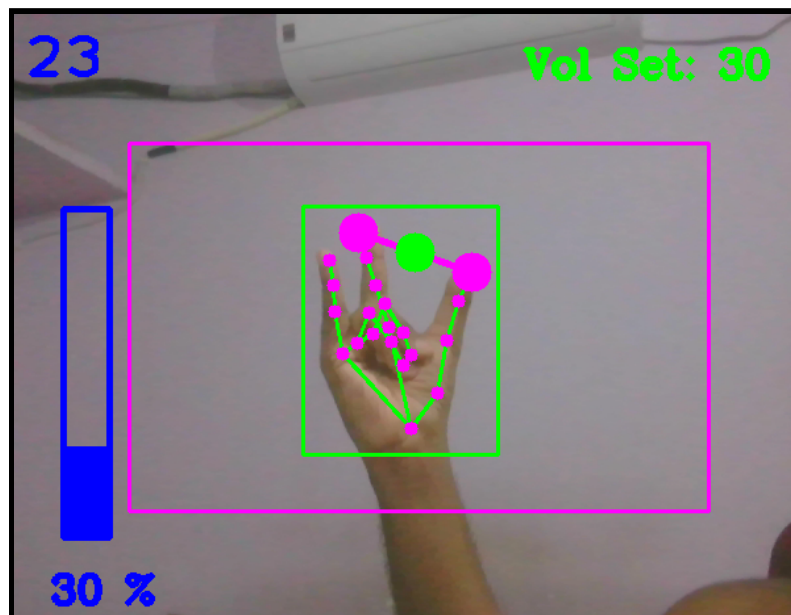
The figure shows the raised index and middle finger being recognised and the line joining their tips and showing its midpoint. When the length of this connected line is less than a given length (length of 25 units used in the project code), i.e when the fingers are almost joined, a click occurs on the screen

Steps for Volume control using Gesture:

- 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



The figure shows the raised index and thumb finger being recognised and the line drawn between their tips whose distance is used to calculate the volume percentage



The figure shows the raised index, thumb, and pinky finger being recognised. When the pinky finger is up, the volume percentage calculated is set to the system, and hence the color of the volume on the top right turns green.

Software and Hardware Requirements:

- A web camera is required to track the live feed of the hand moving
- Python 3 or above must be available on the system
- The system must have pip to install the require dependencies

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
-

Code:

The code for the project can be found on my github at

<https://github.com/Subham-Panda/VIT-J-Components/tree/master/CSI%20-%202%20cr%20edit%20course/python%20virtual%20mouse/AiVirtualMouse>

To run the code:

- Clone the folder
- Navigate inside the folder
- To install the dependencies execute *pip install -r requirements.txt*
- To run the AiVirtualMouse run the AiVirtualMouse.py using the command in your terminal - *python AiVirtualMouse.py*