AI Virtual Mouse

A PROJECT REPORT
Submitted by

Prakhyat Nandan Srivastava (20MIM10036)
Aditya Baraiya (20MIM10004)
Manan Patel (20MIM10045)
Anas Khan (20MIM10001)
Anam Rehman (20MIM10031)

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
in
CSE WITH SPECIALIZATION IN AIML

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

APRIL,2022

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE
MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **“AI VIRTUAL MOUSE”** is the bonafide work of “**Prakhyat Nandan Srivastava (20MIM10036), Aditya Barariya (20MIM10004), Manan Patel (20MIM10045), Anas Khan (20MIM10001), Anam Rehman (20MIM10031)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. A. V. R . Mayuri, Senior Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Rakshita Mall,<< Designation>>
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held in April 2022.

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to **Dr.MAYURI AVR**, **PROGRAM CHAIR INTEGRATED M.TECH AIML , School of Computer Science and Engineering** for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide **Dr.Rakshita Mall,** for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Aeronautical Science, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

1. AI = Artificial Intelligence
2. HCI = Human Computer Interaction
3. HSV = Hue Saturation Value
4. GB = GigaByte
5. RAM = Random Access Memory
6. OS = Operating System
7. FPS = Frames Per Second
8. Webcam = Web Camera

LIST OF FIGURES AND GRAPHS

ABSTRACT

This project promotes an approach for the Human Computer Interaction (HCI) where cursor movement can be controlled using a real-time camera, it is an alternative to the current methods including manual input of buttons or changing the positions of a physical computer mouse. Instead, it utilizes a camera and computer vision technology to control various mouse events and is capable of performing every task that the physical computer mouse can.

The Virtual Mouse color recognition program will constantly acquire real-time images where the images will undergo a series of filtration and conversion. Whenever the process is complete, the program will apply the image processing technique to obtain the coordinates of the targeted colors position from the converted frames. After that, it will proceed to compare the existing colors within the frames with a list of color combinations, where different combinations consist of different mouse functions. If the current colors combination found a match, the program will execute the mouse function, which will be translated into an actual mouse function to the users' machine.

PURPOSE

The main objective of the proposed AI virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions, and this can be achieved with the help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse function such as left click, right click, and scrolling function.

TABLE OF CONTENTS

CHAPTER-1:
PROJECT DESCRIPTION AND OUTLINE

1.1 INTRODUCTION:

In this topic we are going to learn about how we got the motivation for this project along with the Introduction to the project including their techniques. Also we are going to learn about the problems that the initial developed techniques were facing as well as the objectives of our work.

1.2 MOTIVATION OF THE WORK:

It is fair to say that the Virtual Mouse will soon to be substituting the traditional physical mouse in the near future, as people are aiming towards the lifestyle where that every technological devices can be controlled and interacted remotely without using any peripheral devices such as the remote, keyboards, etc. it doesn't just provides convenience, but it's cost effective as well.

1.2.1. Convenient

It is known in order to interact with the computer system, users are required to use an actual physical mouse, which also requires a certain area of surface to operate, not to mention that it suffers from cable length limitations. Virtual Mouse requires none of it, as it only a webcam to allow image capturing of user's hand position in order to determine the position of the pointers that the user want it to be. For example, the user will be able to remotely control and interact the computer system by just facing the webcam or any other image capturing devices and moving your fingers, thus eliminating the need to manually move the physical mouse, while able to interact with the computer system from few feet away.

1.2.2. Cost Effective

A quality physical mouse is normally cost from the range of 2$ to a 100$, depending on their functionality and features. Since the Virtual Mouse requires only a webcam, a physical mouse are no longer required, thus eliminating the need to purchase one, as a single webcam is sufficient enough to allow users to interact with the computer system through it, while some other portable computer system such as the laptop, are already supplied with a built-in webcam, could simply utilize the Virtual Mouse software without having any concerns about purchasing any external peripheral devices.

1.3 PROBLEM STATEMENT:

It's no surprise that every technological device has its own limitations, especially when it comes to computer devices. After the review of various types of the physical mouse, the problems are identified and generalized. The following describes the general problem that the current physical mouse suffers:

- Physical mouse is subjected to mechanical wear and tear.
- Physical mouse requires special hardware and surface to operate.
- Physical mouse is not easily adaptable to different environments and its performance varies depending on the environment.
- Mouse has limited functions even in present operational environments.
- All wired mouse and wireless mouse have their own lifespan.

1.4 OBJECTIVE OF THE WORK:

The purpose of this project is to develop a Virtual Mouse application that targets a few aspects of significant development. For starters, this project aims to eliminate the needs of having a physical mouse while able to interact with the computer system through webcam by using various image processing techniques. Other than that, this project aims to develop a Virtual Mouse application that can be operational on all kind of surfaces and environment.The following describes the overall objectives of this project:

- To design to operate with the help of a webcam.
- The Virtual Mouse application will be operational with the help of a webcam, as the webcam is responsible to capture the images in real time. The application would not work if there were no webcam detected.
- To design a virtual input that can operate on all surface.
- The Virtual Mouse application will be operational on all surface and indoor environment, as long the users are facing the webcam while doing the motion gesture.
- To program the camera to continuously capturing the images, which the images will be analysed, by using various image processing techniques.
- As stated above, the Virtual Mouse application will be continuously capturing the images in real time, where the images will be undergo a series of process, this includes HSV conversion, Binary Image conversion, salt and pepper noise filtering, and more.

- To convert hand gesture/motion into mouse input that will be set to a particular screen position.
- The Virtual Mouse application will be programmed to detect the position of the defined colours where it will be set as the position of the mouse pointers. Furthermore, a combination of different colours may result in triggering different types of mouse events, such as the right/left clicks, scroll up/down, and more.

1.5 ORGANIZATION OF THE PROJECT:

We the group members firstly sat together to come on the same topic. After coming on the same topic we then had a research on our topic and each member was assigned with a respective topic. After the research on the respective topic was over we made a presentation and then we started to code accordingly and then we attached each module into a common program which was our final project.

1.6 SUMMARY:

In this Chapter we learned about the model basics that is what was the reason to select this particular topic. We also learned the objectives of our topic and the problems that we overcome in our model. In the last we talked about how we organized our project to complete it.

CHAPTER-2:

RELATED WORK INVESTIGATION

2.1 INTRODUCTION:

In this chapter we are going to learn about what is the core area of our project, and we are also going to discuss various methods/approaches for the existing models as well as their pros and cons of that model . At the end we will see the Issues/observations from our investigation.

2.2 CORE AREA OF THE PROJECT:

The core area of our project is that we can trace/track our hand without wearing any gloves or without coloring our hands with respective colors. Our model is trained in such a way that it can identify the functions done with our hand such as scroll, left click, right click, etc. These functions are carried out with the help of a media pipe library which sees the hand gesture points that are assigned to our hands.the figure given below.

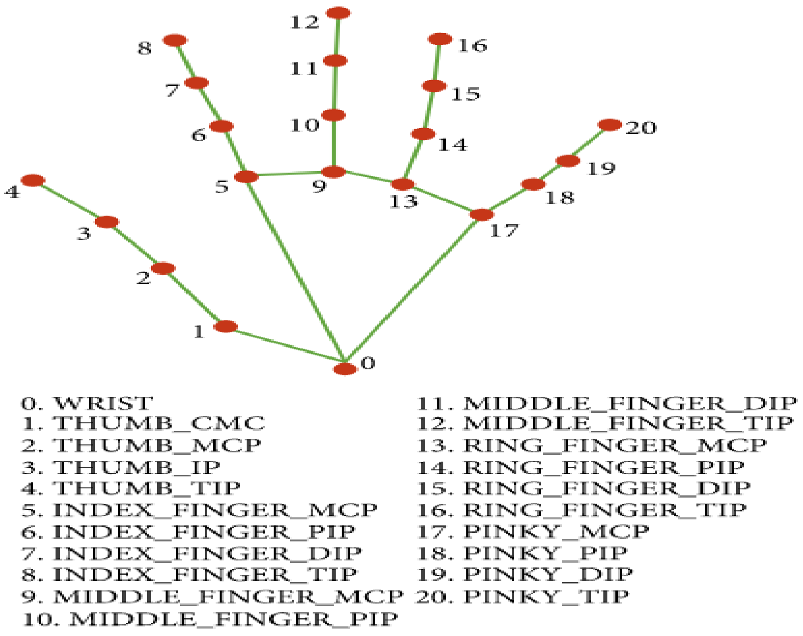


Figure 2.1: Coordinates of the Land-marks on the hand

2.3 EXISTING APPROACHES/METHODS:

2.3.1 Approach 1 (Using Hand Gloves):

There are some related works carried out on virtual mouse using hand gesture detection by wearing a glove in the hand and also using color tips in the hands for gesture recognition, but they are no more accurate in mouse functions. The recognition is not so accurate because of wearing gloves; also, the gloves are also not suited for some users, and in some cases, the recognition is not so accurate because of the failure of detection of color tips. Some efforts have been made for camera-based detection of the hand gesture interface.

2.3.2 Approach 2 (Using different colors):

Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013 proposed a study on “Cursor Control System Using Hand Gesture Recognition.” In this work, the limitation is stored frames are needed to be processed for hand segmentation and skin pixel detection.

Vinay Kr. Pasi, Saurabh Singh, and Pooja Kumari in 2016 proposed “Cursor Control using Hand Gestures” in the IJCA Journal. The system proposes the different bands to perform different functions of the mouse. The limitation is it depends on various colors to perform mouse functions.

Chaithanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 proposed “Virtual Mouse Using Hand Gesture” where the model detection is based on colors. But, only few mouse functions are performed.

2.4 PRO’S and CON’S:

Here are the following Pros of Using Gloves and Colors for detection:

- ➢ Using gloves and Colors Computer was able to detect the hands
- ➢ Different colors used were easily identified by the computer model.
- ➢ On the gloves white section was easily recognisable.

Here are the following Cons of Using Gloves and Colors for detection:

- ➢ Dependent only on specific colors for recognition. If a different color rather than assigned color is used it will not detect the hand and no functions will be done.
- ➢ If there are no gloves on the hand then there will not be any type of functions performed.
- ➢ For this specific type of gloves are needed.

2.5 OBSERVATION FROM INVESTIGATION:

From the above work we come to know that there are various types of limitations such as :

- It depends on various colors to perform mouse functions.
- Stored frames are needed to be processed for hand segmentation and skin pixel detection.
- The recognition is not so accurate because of the failure of detection of color tips in Gloves method.

there were various other problems such as the accuracy level was not more accurate as expected, etc.

2.6 Related Work:

There are some related works carried out on virtual mouse using hand gesture detection by wearing a glove in the hand and also using color tips in the hands for gesture recognition, but they are no more accurate in mouse functions. The recognition is not so accurate because of wearing gloves; also, the gloves are also not suited for some users, and in some cases, the recognition is not so accurate because of the failure of detection of color tips. Some efforts have been made for camera-based detection of the hand gesture interface.

In 1990, Quam introduced an early hardware-based system; in this system, the user should wear a DataGlove [2]. The proposed system by Quam although gives results of higher accuracy, but it is difficult to perform some of the gesture controls using the system.

Dung-Hua Liou, ChenChiung Hsieh, and David Lee in 2010 [3] proposed a study on “A Real-Time Hand Gesture Recognition System Using Motion History Image.” The main limitation of this model is more complicated hand gestures.

Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013 [4] proposed a study on “Cursor Control System Using Hand Gesture Recognition.” In this work, the limitation is stored frames are needed to be processed for hand segmentation and skin pixel detection.

Vinay Kr. Pasi, Saurabh Singh, and Pooja Kumari in 2016 [5] proposed “Cursor Control using Hand Gestures” in the IJCA Journal. The system proposes the different bands to perform different functions of the mouse. The limitation is it depends on various colors to perform mouse functions.

Chaithanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 [6] proposed “Virtual Mouse Using Hand Gesture” where the model detection is based on colors. But, only few mouse functions are performed.

2.7 SUMMARY:

In this chapter we come to know what were the difficulties faced by the previous models. We were able to overcome these problems in our model and the accuracy level was also well maintained.

CHAPTER-3:
REQUIREMENT ARTIFACTS

3.1 INTRODUCTION:

In this chapter we will learn about the hardware and software required for our model along with their specifications. We will also learn about data requirements, functional requirements, security requirements, etc.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS:

3.2.1 HARDWARE:

The following describes the hardware needed in order to execute and develop the Virtual Mouse application:

System will be using

- Processor : Core of any generation
- Main Memory : 4GB RAM
- Hard Disk : 128 GB and more
- Display : 14" Monitor
- Webcam (720P)

3.2.2 SOFTWARE:

The following describes the software needed in-order to develop the Virtual Mouse application:

- OS : Window 10 64-bit
- Language : Python 3.8.0
- Python Library : Open CV, AutoPy , Mediapipe (0.8.3.1)

3.3 SPECIFIC DATA REQUIREMENTS:

For doing our project we require different types of information which are listed below:

3.3.1 DATA REQUIREMENT:

For this model we do not require any type of data as we are using a webcam to detect our model.

For doing our coding we had referred some data from the websites and youtube as well which is listed in the reference 1,2.

3.3.2 FUNCTION REQUIREMENT:

We need different functions to perform our coding and run our model. After developing this application, users should be able to access their system through Motion Tracker Application. Calculations required for this application are all related to motion detect operations & Data processing is done by using different python libraries like NumPy & media-pipe.

3.3.3 PERFORMANCE AND SECURITY:

Talking about the performance if we use the above Hardware and Software there will not be any issue in running our model and talking about the security we don't need any high level security for the model since it is a virtual system only we have to see that there is no any kind of malware function or virus which could damage our pc or folder or model.

3.4 SUMMARY:

In this chapter, we come to know what are the various system requirements, especially talking about the Hardware and Software requirements. If we use the proper function as listed above we will be able to perform our task very smoothly.

CHAPTER-4:

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 METHODOLOGY AND GOAL:

AI virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions, and this can be achieved with the help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse function such as left click, right click, and scrolling function.

The various functions and conditions used in the system are explained in the flowchart of the real-time AI virtual mouse system in Figure

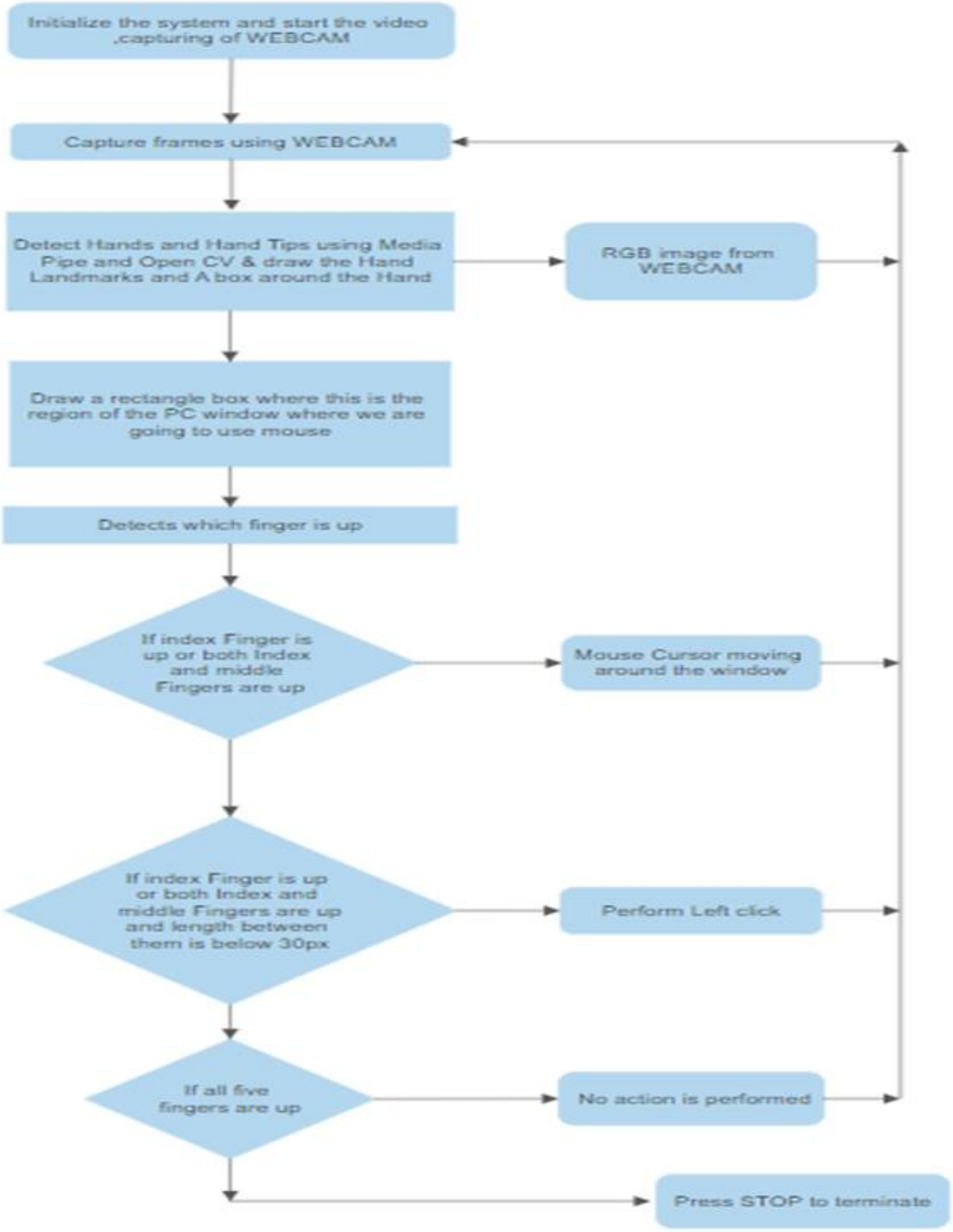
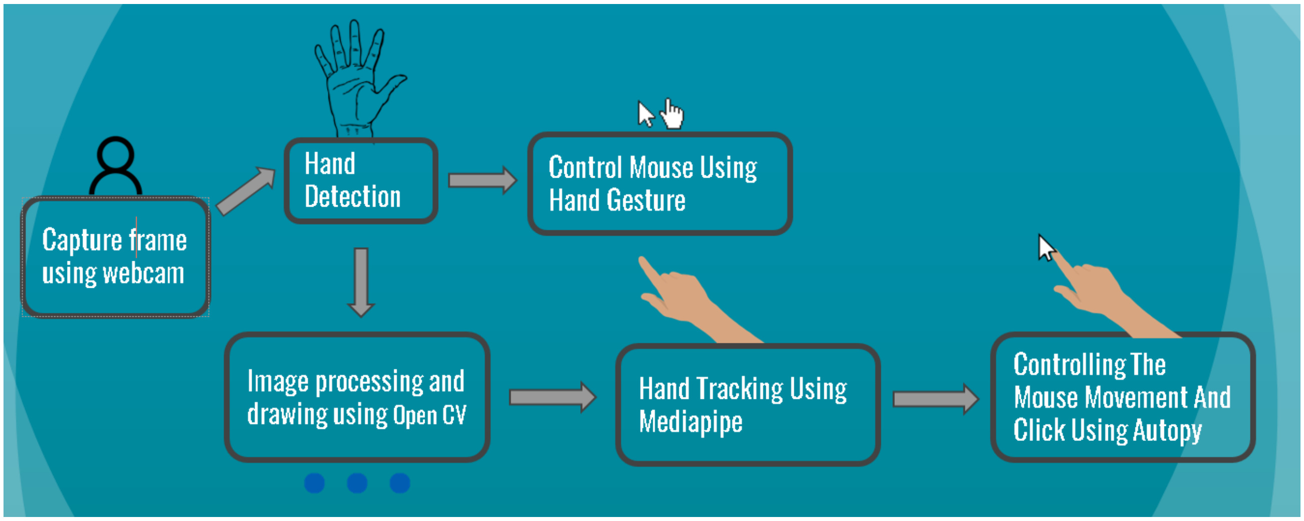

Initialize the system and start the video ,capturing of WEBCAM
Capture frames using WEBCAM
Detect Hands and Hand Tips using Media Pipe and Open CV & draw the Hand Landmarks and A box around the Hand
RGB image from WEBCAM
Draw a rectangle box where this is the region of the PC window where we are going to use mouse
Detects which finger is up
If index Finger is up or both Index and middle Fingers are up
Mouse Cursor moving around the window
If index Finger is up or both Index and middle Fingers are up and length between them is below 30px
Perform Left click
If all five fingers are up
No action is performed
Press STOP to terminate

Figure 4.1 :- FlowChart

4.2 FUNCTIONAL MODULES DESIGN AND ANALYSIS:

From the above given flowchart we have designed our model and the model has given the best output as compared to the previous models.

4.3 SOFTWARE ARCHITECTURAL DESIGN:


Capture frame using webcam

Hand Detection

Control Mouse Using Hand Gesture

Image processing and drawing using Open CV

Hand Tracking Using Mediapipe

Controlling The Mouse Movement And Click Using Autopy

Figure 4.2: Overall System Arhitecture

4.4 USER INTERFACE DESIGN:

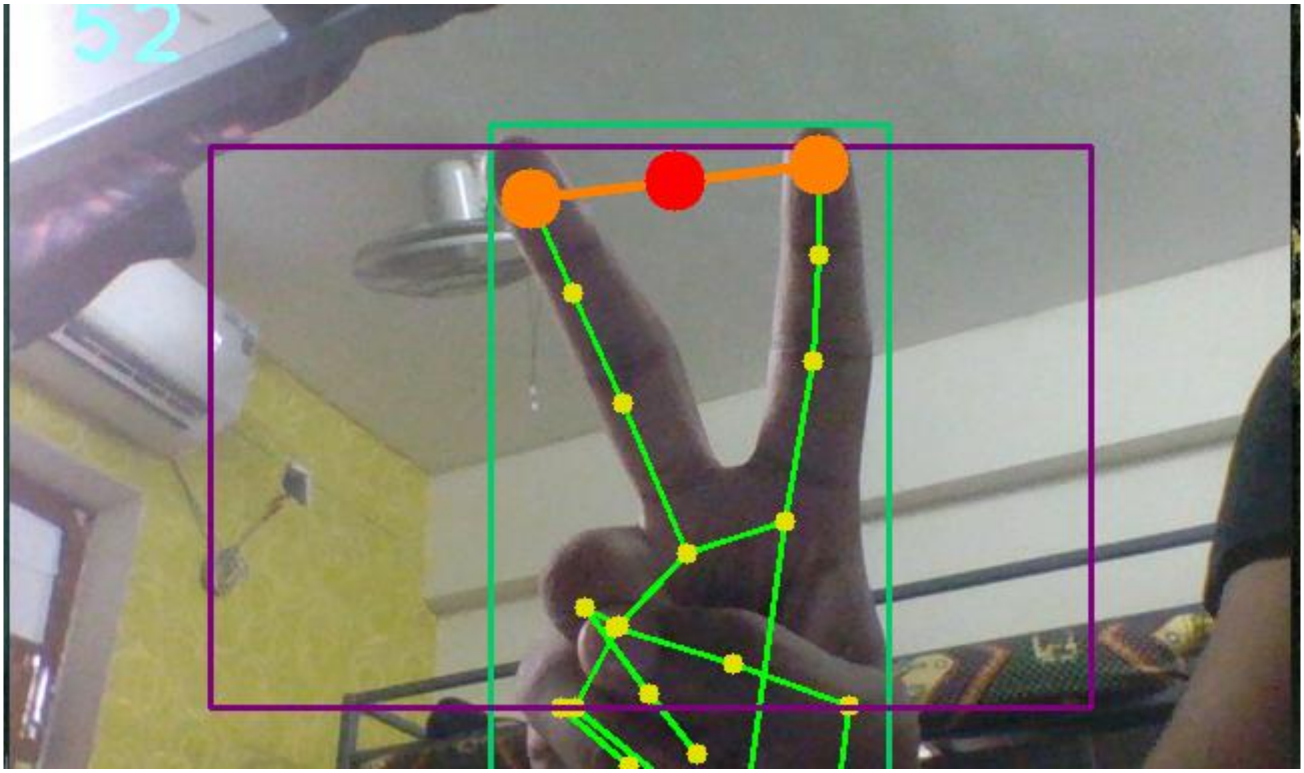


Figure 4.3: Interface Design

In the above fig.4.4, we can see the Interface Design in which both the hands are shown together but the only one hand is captured where there is a rectangular box. We have designed our Program in such a way that if there are more than two or more hands it will identify any one hand left hand since we have programmed our model in such a way. If the FPS goes below 10 then our program will not work properly or may be closed. So to run the model on any desktop it is recommended to have a 30 or above frame rate webcam.

4.5 SUMMARY:

In this chapter we learned about the method which we had used to prepare our model i.e. Agile method. We also saw the flowchart diagram components for preparing our model along with the software architecture which tell about how our model will work, and in the last we saw the User Interface of our model.

CHAPTER-5:
TECHNICAL IMPLEMENTATION & ANALYSIS

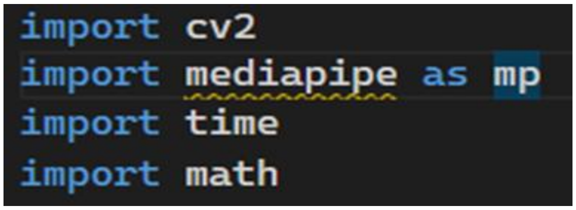
5.1 OUTLINE:

The overall coding work related to the project is shown here. We will show the various codes, functions etc.

5.2 TECHNICAL CODING AND CODE SOLUTION:

For making the AI Virtual Mouse we have to make our hand tracking module and then implement it to the main code for virtual mouse.

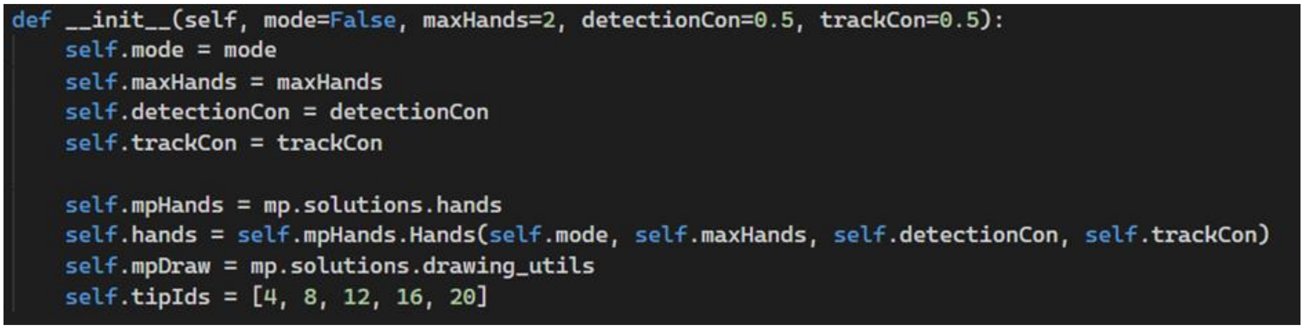
- Libraries that we are using in our Hand Tracking Module



```
import cv2
import mediapipe as mp
import time
import math
```

- Function for hand detection

We are using hand tracking module from opencv in this module we are using this objects



```
def __init__(self, mode=False, maxHands=2, detectionCon=0.5, trackCon=0.5):
    self.mode = mode
    self.maxHands = maxHands
    self.detectionCon = detectionCon
    self.trackCon = trackCon

    self.mpHands = mp.solutions.hands
    self.hands = self.mpHands.Hands(self.mode, self.maxHands, self.detectionCon, self.trackCon)
    self.mpDraw = mp.solutions.drawing_utils
    self.tipIds = [4, 8, 12, 16, 20]
```

- For hand tracking we are going to make four different classes :

class handDetector()

def findHands(self, img, draw=True):

def findPosition(self, img, handNo=0, draw=True):

def findDistance(self, p1, p2, img, draw=True, r=15, t=3):

def main():

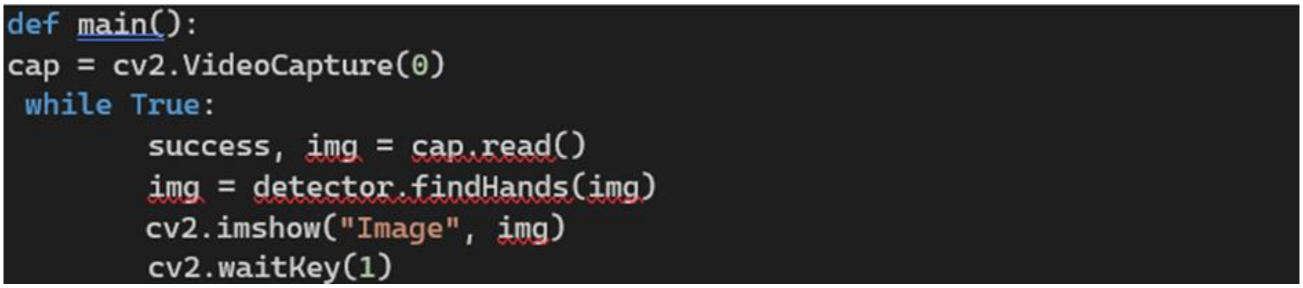
First one is the find position

Second one is the find hand

Third one is the find distance

Fourth one is to run our code main function

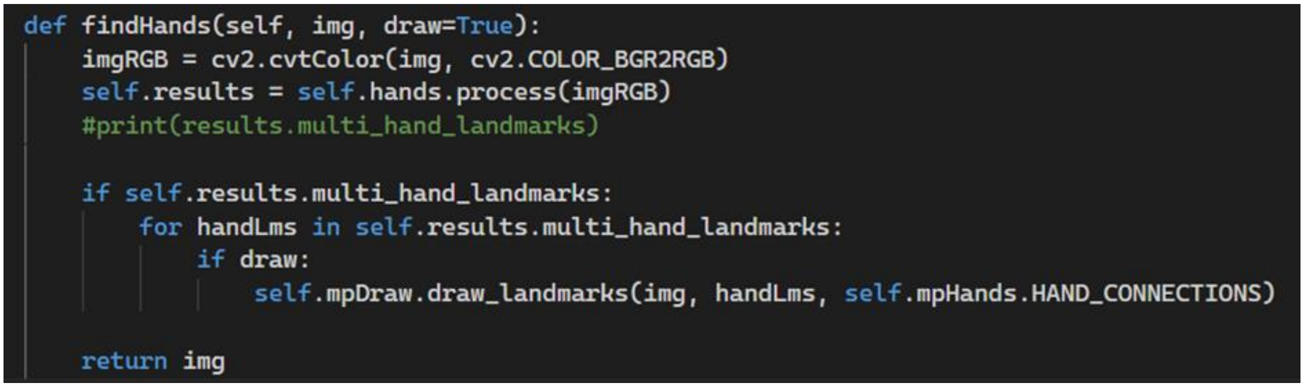
And in the main function we are going to connect a webcam to the pc so it can detect and track the hand for that we have to do



```
def main():
cap = cv2.VideoCapture(0)
 while True:
        success, img = cap.read()
        img = detector.findHands(img)
        cv2.imshow("Image", img)
        cv2.waitKey(1)
```

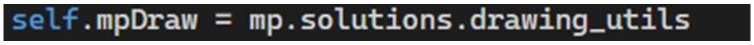
13

- So till now we are able to see our hand now we have to detect our hand, for that



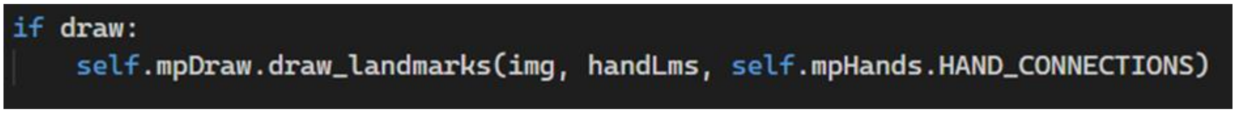
```
def findHands(self, img, draw=True):
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    self.results = self.hands.process(imgRGB)
    #print(results.multi_hand_landmarks)

    if self.results.multi_hand_landmarks:
        for handLms in self.results.multi_hand_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, handLms, self.mpHands.HAND_CONNECTIONS)

    return img
```

This mpDraw function will detect all landmark on our hand



```
self.mpDraw = mp.solutions.drawing_utils
```

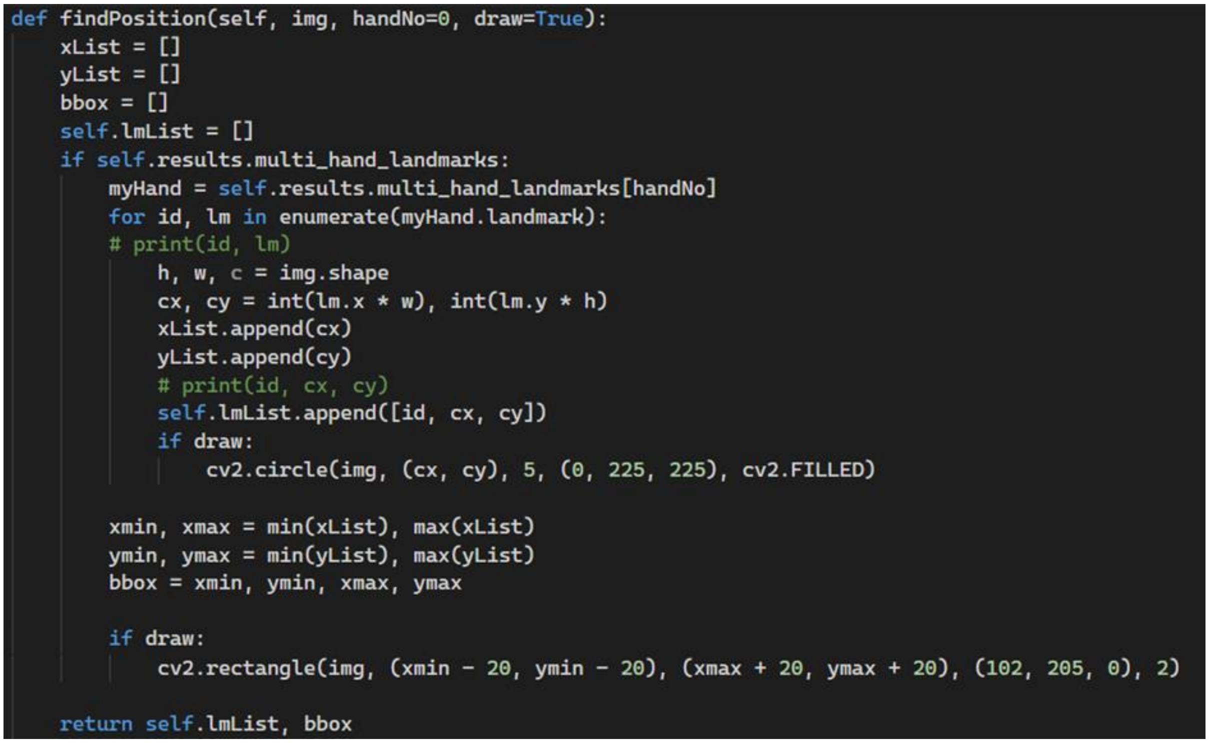
Now we want to track our hand gesture for that we will use this statement



```
if draw:
    self.mpDraw.draw_landmarks(img, handLms, self.mpHands.HAND_CONNECTIONS)
```

Now we get all the values of our hand landmarks but we still don’t know how to use it for tracking

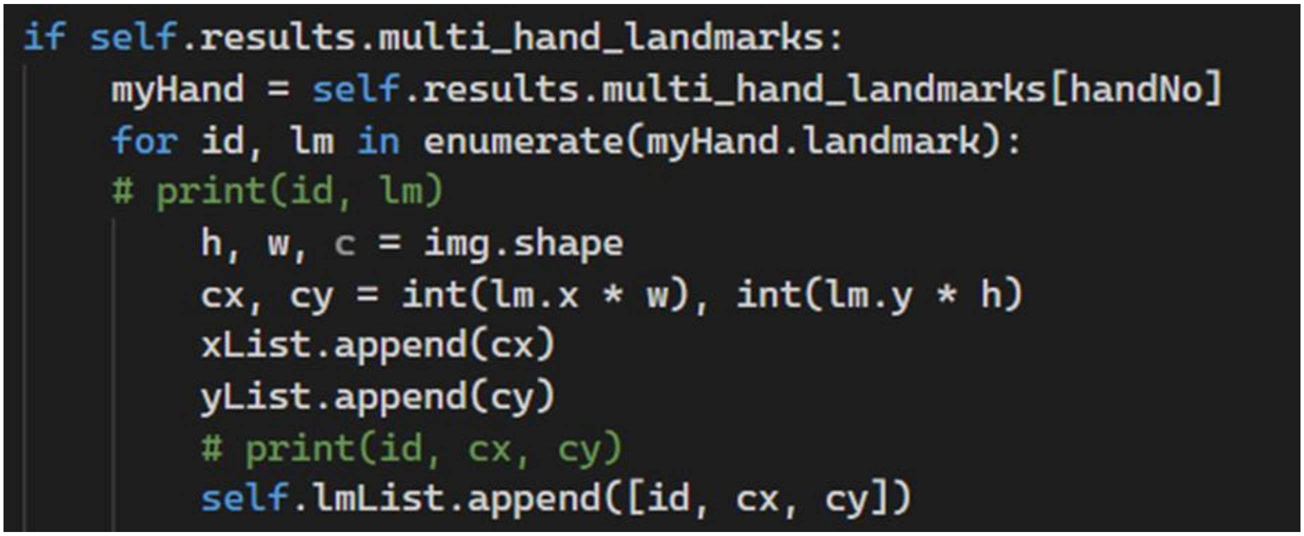
For that we have to make another function for that we called as find position



```
def findPosition(self, img, handNo=0, draw=True):
    xList = []
    yList = []
    bbox = []
    self.lmList = []
    if self.results.multi_hand_landmarks:
        myHand = self.results.multi_hand_landmarks[handNo]
        for id, lm in enumerate(myHand.landmark):
        # print(id, lm)
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            xList.append(cx)
            yList.append(cy)
            # print(id, cx, cy)
            self.lmList.append([id, cx, cy])
            if draw:
                cv2.circle(img, (cx, cy), 5, (0, 225, 225), cv2.FILLED)

        xmin, xmax = min(xList), max(xList)
        ymin, ymax = min(yList), max(yList)
        bbox = xmin, ymin, xmax, ymax

        if draw:
            cv2.rectangle(img, (xmin - 20, ymin - 20), (xmax + 20, ymax + 20), (102, 205, 0), 2)

    return self.lmList, bbox
```

This is the whole function in this function we are finding our hand’s landmarks position to take help for our purpose.

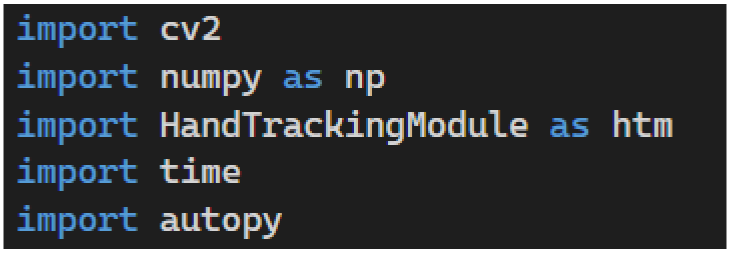
For that we are writing this statement



```
if self.results.multi_hand_landmarks:
    myHand = self.results.multi_hand_landmarks[handNo]
    for id, lm in enumerate(myHand.landmark):
    # print(id, lm)
        h, w, c = img.shape
        cx, cy = int(lm.x * w), int(lm.y * h)
        xList.append(cx)
        yList.append(cy)
        # print(id, cx, cy)
        self.lmList.append([id, cx, cy])
```

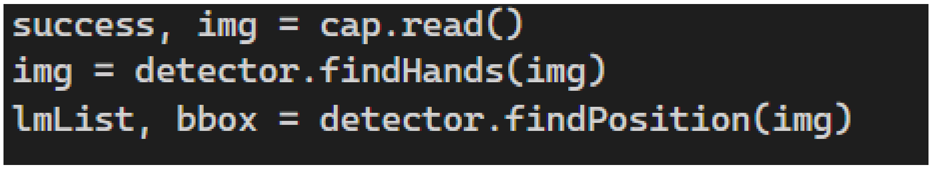
Till now we made our Hand Tracking module now we are going to write main code for virtual mouse.

- Ai virtual mouse now we import some libraries and our hand tracking module also



```
import cv2
import numpy as np
import HandTrackingModule as htm
import time
import autopy
```

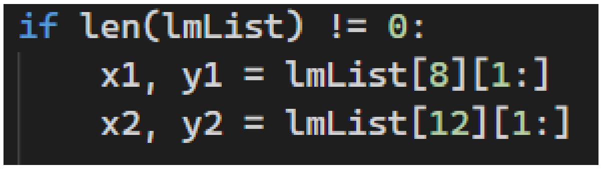
- For detecting our hand we take the help of the hand tracking module



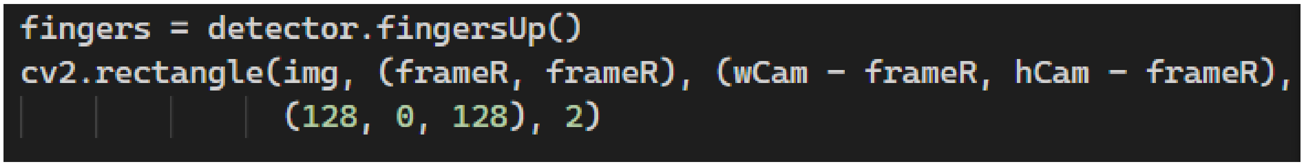
```
success, img = cap.read()
img = detector.findHands(img)
lmList, bbox = detector.findPosition(img)
```

Now we have to define the mouse actions for that we are going to define some sign or taking reference to recognize the function to do mouse actions.

- For that lets recognize our index finger and middle finger’s landmarks

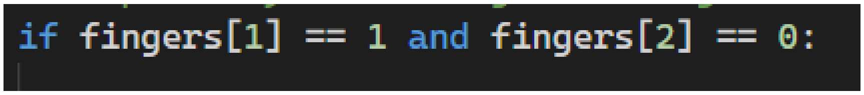


```
if len(lmList) != 0:
    x1, y1 = lmList[8][1:]
    x2, y2 = lmList[12][1:]
```

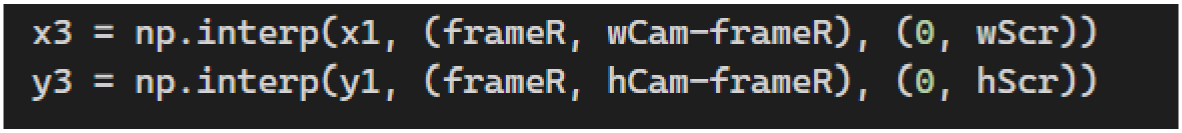


```
fingers = detector.fingersUp()
cv2.rectangle(img, (frameR, frameR), (wCam - frameR, hCam - frameR),
                (128, 0, 128), 2)
```

- Now if only index finger is up then we define the cursor movement

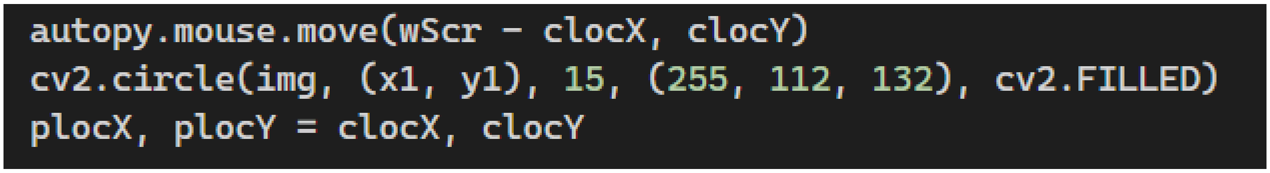


```
if fingers[1] == 1 and fingers[2] == 0:
```



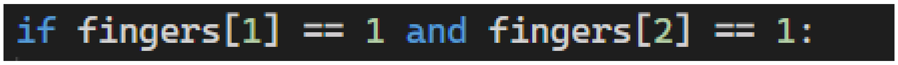
```
x3 = np.interp(x1, (frameR, wCam-frameR), (0, wScr))
y3 = np.interp(y1, (frameR, hCam-frameR), (0, hScr))
```

- to show what function are running we input different types of colors



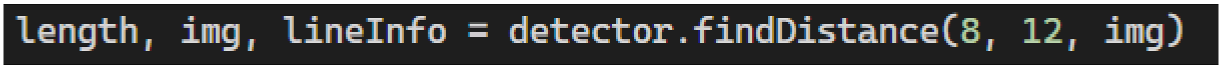
```
autopy.mouse.move(wScr - clocX, clocY)
cv2.circle(img, (x1, y1), 15, (255, 112, 132), cv2.FILLED)
plocX, plocY = clocX, clocY
```

- if both the fingers are up then consider it as clicking mode

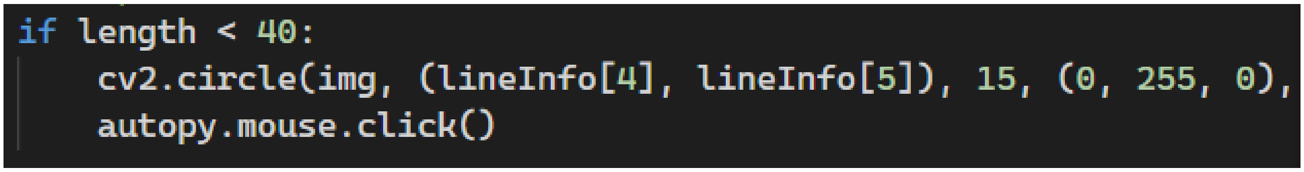


```
if fingers[1] == 1 and fingers[2] == 1:
```

- now if the distance between the two fingers is 0 then it will run the clicking while we are clicking the cursor will not be able to move

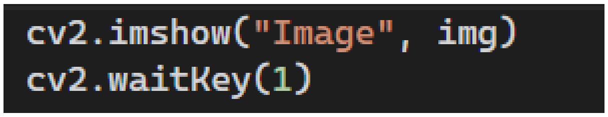


```
length, img, lineInfo = detector.findDistance(8, 12, img)
```



```
if length < 40:
    cv2.circle(img, (lineInfo[4], lineInfo[5]), 15, (0, 255, 0),
    autopy.mouse.click()
```

- now we display the result of our whole code



```
cv2.imshow("Image", img)
cv2.waitKey(1)
```

5.3 WORKING LAYOUT OF FORMS:

- Accessing the camera

Figure 5.1: Access Camera

- detecting all landmark

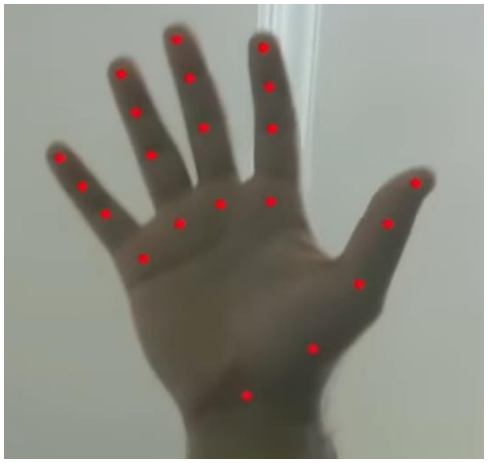
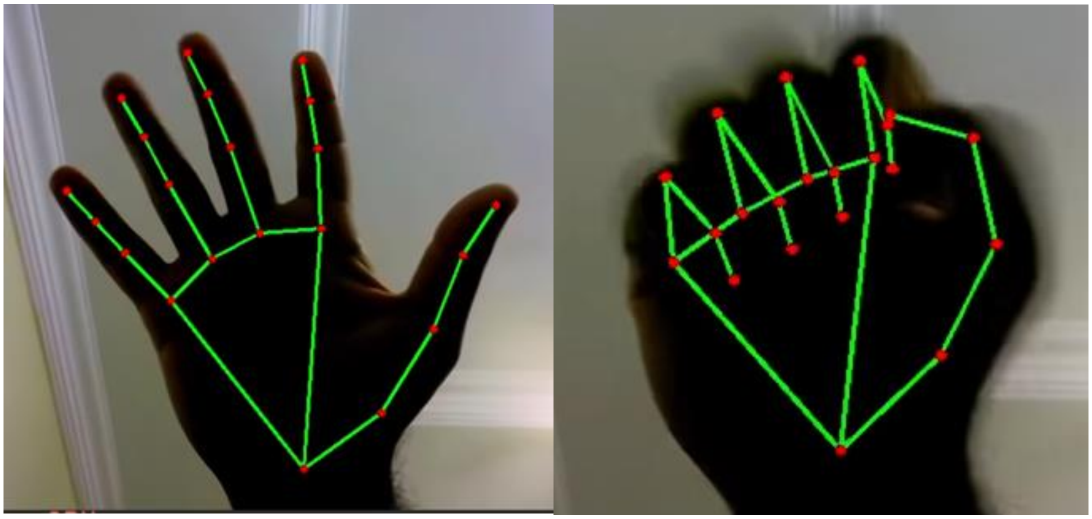


Figure 5.2: Landmark Detection

- track our hand gesture and getting all the values of our hand landmarks

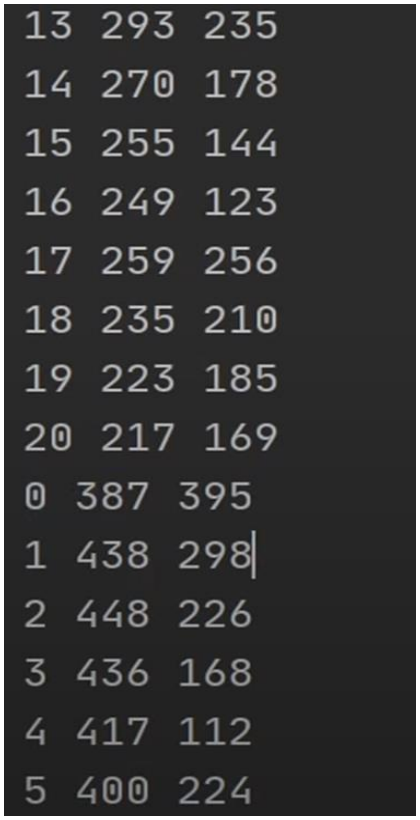


(a) (b)

Figure 5.3 (a & b): Tracking hand gestures

5.4 TEST AND VALIDATION:

- finding our hand's landmarks position to take help for our purpose



```
13 293 235
14 270 178
15 255 144
16 249 123
17 259 256
18 235 210
19 223 185
20 217 169
0 387 395
1 438 298
2 448 226
3 436 168
4 417 112
5 400 224
```

Figure 5.4: Positions of landmarks

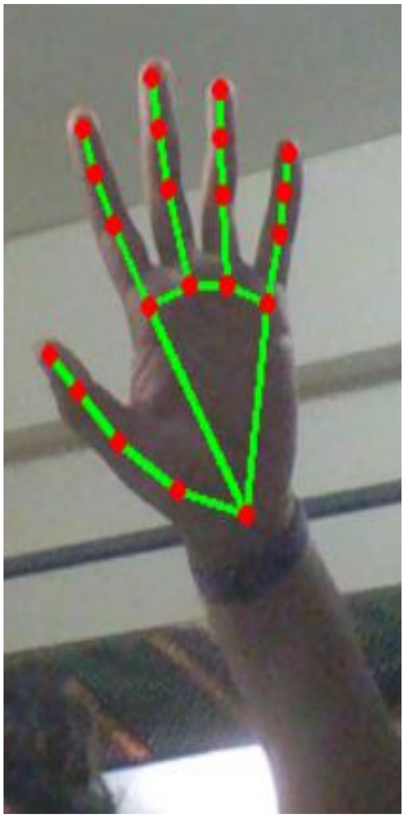


Figure 5.5: Landmarks of our hand

- the cursor moving process

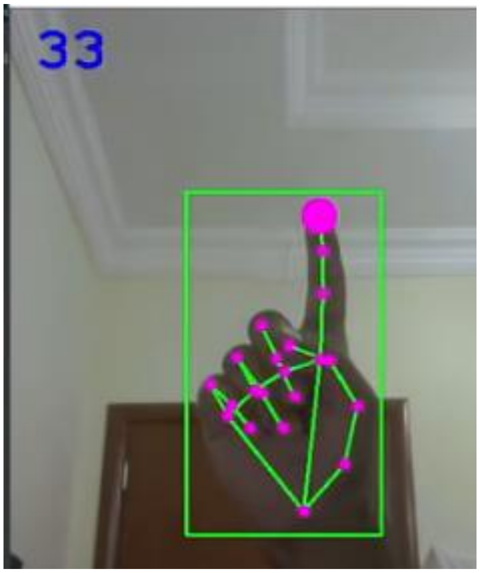


Figure 5.6: Cursor Moving

- the clicking process
- the freezing process while clicking

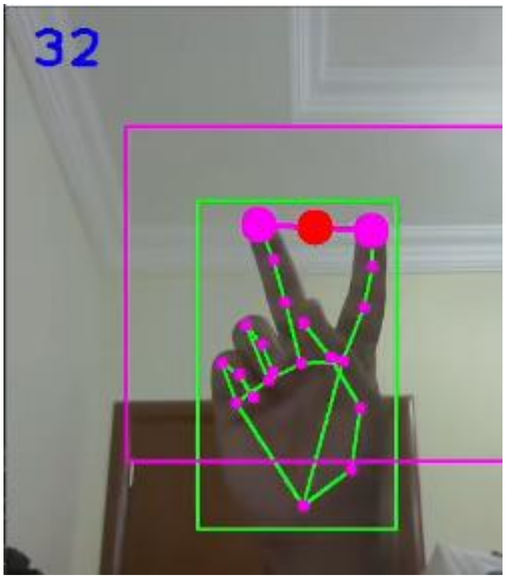


Figure 5.7: Freeze Point

- clicking process when the distance between fingers is zero

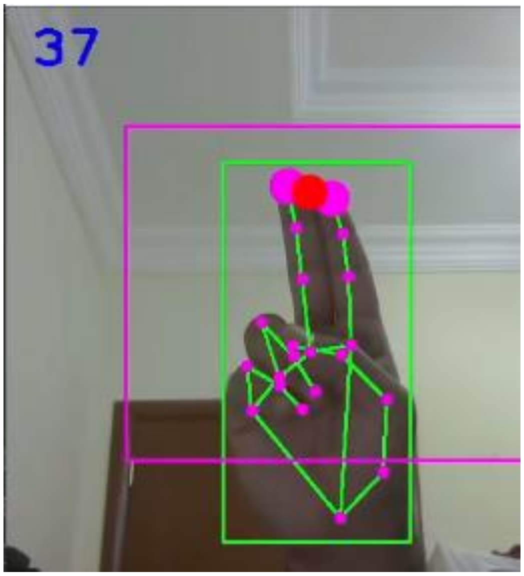


Figure 5.8: Clicking Process

5.5 Performance Analysis(Graphs/Charts):

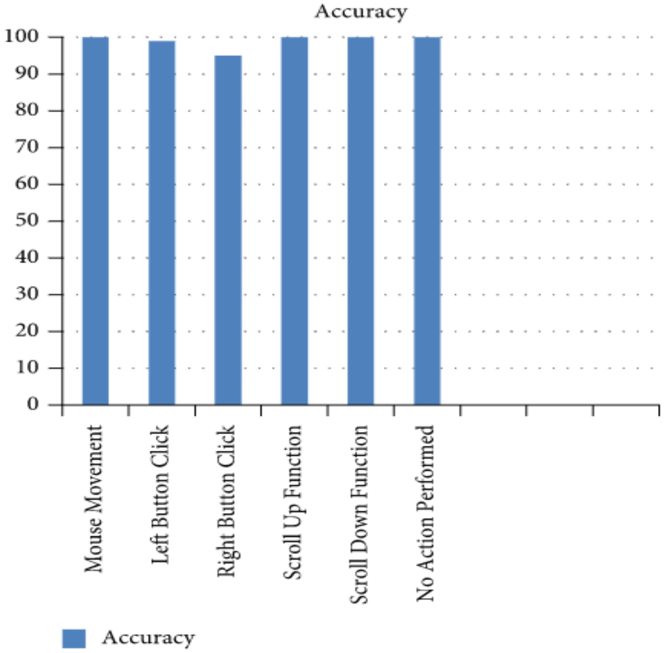


Table 5.9: Accuracy

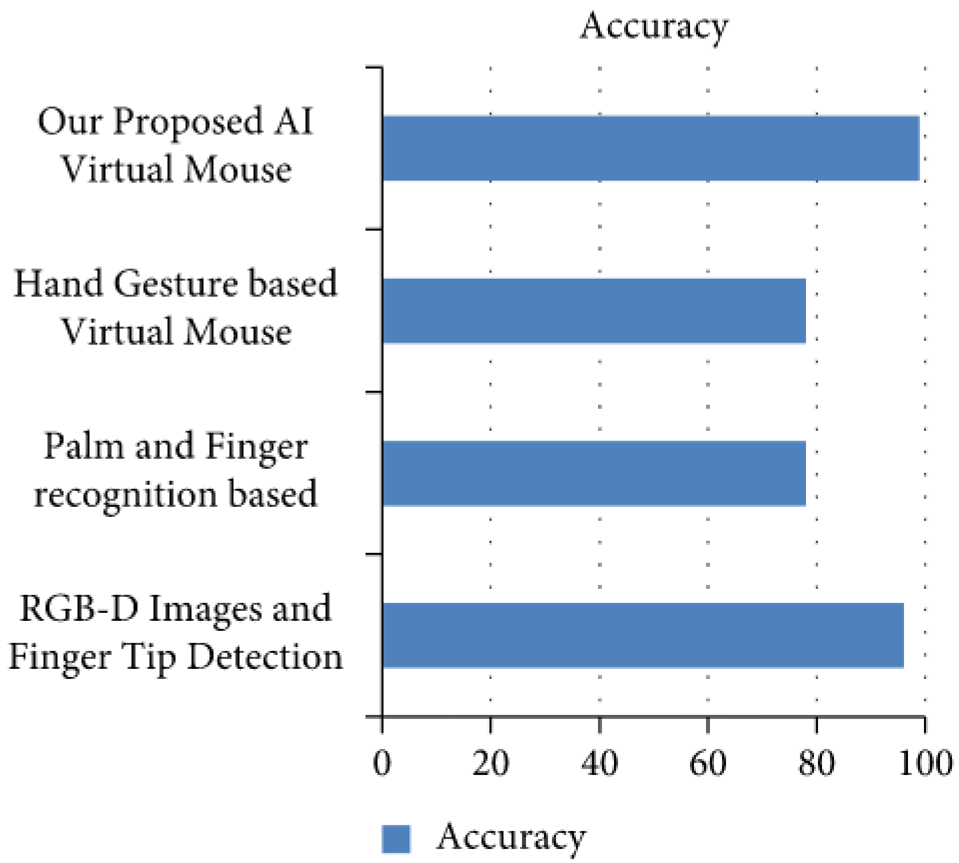


Table 5.10: Accuracy Comparison

5.6 Summary:

In this Chapter we learned about the coding used for our model and their particular outputs. We have achieved accuracy more efficiently. We would further like to achieve a few more mouse functions.

CHAPTER-6:
PROJECT OUTCOME AND APPLICABILITY

6.1 Outline:

In this chapter we are going to learn about our project outcome and their applicability. There are various ways in which we can achieve our objectives and accuracy using different types of methods.

6.2 Key implementations outlines of the System:

Taking the help of Opencv, multimedia, and autopy we made a hand tracking module after making it, now we have to make our virtual mouse.

Our AI virtual mouse works on some steps so that we all can easily understand

Steps for our main implementation of Ai virtual mouse.

Step1: Find the landmarks
Step2: Get the tip of the index and middle finger
Step3: Check which fingers are up
Step4: Only Index Finger: Moving Mode
Step5: Convert the coordinates
Step6: Smooth Values
Step7: Move Mouse
Step8: Both Index and middle are up: Clicking Mode
Step9: Find distance between fingers
Step10: Click mouse if distance short
Step11: Frame rate
Step12: Display

6.3 Significant project outcomes:

The AI virtual mouse system is useful for many applications; it can be used to reduce the space for using the physical mouse, and it can be used in situations where we cannot use the physical mouse. The system eliminates the usage of devices, and it improves the human-computer interaction.

6.4 Project applicability on Real-world applications:

- ❖ Amidst the COVID-19 situation, it is not safe to use the devices by touching them because it may result in a possible situation of spread of the virus by touching the devices, so the proposed AI virtual mouse can be used to control the PC mouse functions without using the physical mouse
- ❖ The system can be used to control robots and automation systems without the usage of devices
- ❖ 2D and 3D images can be drawn using the AI virtual system using the hand gestures
- ❖ AI virtual mouse can be used to play virtual reality- and augmented reality-based games without the wireless or wired mouse devices.
- ❖ Persons with problems in their hands can use this system to control the mouse functions in the computer

6.5 Inference:

We can say that our mouse can be used for many applications, so that it can reduce the space of using the physical mouse. Also during covid-19 situations it would be more preferable so that we can reduce the risk of getting infected.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1 Outline:

In this we are going to talk about the conclusion drawn from our model along with the recommendations. We will also talk about the limitations where our model is restricted to some consequences as well as the future enhancement of our model.

7.2 Limitation/Constraints of the System:

In earlier models , the user had to wear special gloves in which fingers and palm were differentiated using different colors which made it easy for the program to understand the gestures.In contrary to this our model does not require any special gloves to understand the gestures.It can differentiate between the fingers and palm on its own.

Our virtual mouse has achieved 70% accuracy and we are trying to reach 95% accuracy in future.

7.3 Future Enhancements:

The main objective of the proposed AI virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions, and this can be achieved with the help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse function such as left click, right click, and scrolling function.

7.4 Inference:

We came to know about the limitations of our model as well as the future enhancement of our model.

The main objective of our project is to develop an alternative model to the regular and traditional mouse system to perform and control the mouse functions.

REFERENCES

1. https://www.hindawi.com/journals/jhe/2021/8133076/#related-articles

From here we have taken the reference for the motivation and existing work with limitations .

2. https://www.irjet.net/archives/V8/i7/IRJET-V8I7695.pdf

For here we have referred to the objectives for our model.

3. https://www.youtube.com/watch?v=8gPONnGIPgw
4. D. L. Quam, "Gesture recognition with a DataGlove," IEEE Conference on Aerospace and Electronics, vol. 2, pp. 755–760, 1990.
5. D.-H. Liou, D. Lee, and C.-C. Hsieh, "A real time hand gesture recognition system using motion history image," in Proceedings of the 2010 2nd International Conference on Signal Processing Systems, IEEE, Dalian, China, July 2010.
6. K. P. Vinay, "Cursor control using hand gestures," International Journal of Critical Accounting, vol. 0975–8887, 2016.
7. L. Thomas, "Virtual mouse using hand gesture," International Research Journal of Engineering and Technology (IRJET, vol. 5, no. 4, 2018.
8. Google,MP, https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html.