A Thesis/Project/Dissertation Report

on

AI Virtual Gesture Mouse

Submitted in partial fulfillment of the

requirement for the award of the degree of

B.TECH IN

COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. Ship Kumar Verma
(Professor)

Submitted By

Joy Guha 18SCSE1010040

Shreya Kumari 19SCSE1010489

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the **"AI Virtual Mouse Using Hand Gesture Recognition"** inpartial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of **Aug, 2021 to Dec, 2021**, under the supervision of **Dr. Shiv kumar Verma**, Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Joy Guha, 18SCSE1010040

Shreya Kumari, 19SCSE1010489

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Shiv Kumar verma

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **Joy Guha 18SCSE1010040, Shreya Kumari 19SCSE1010489** has been held on_____and his/her work is recommended for the award of Bachelor of Technology

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:
Place: Greater Noida

ACKNOWLEDGEMENT

First and foremost, I am profoundly thankful to _____ **(Head Of Department, Computer Science and Engineering)**, **Galgotias University, Greater Noida India**, for allowing this paper to be my own work.

I would also like to thank our project supervisor **Dr. Shiv Kumar Verma** for guiding me and sharing their precious time and knowledge with patience. Without their participation and contribution, this project would not have been successful.

Finally I want to express my immense gratitude to my friends and my parents for their endless encouragement and support through all these years. Without whom this project would have been a distant reality.

Thank you.

ABSTRACT

The computer mouse is one of the wondrous inventions of humans in the field of Human-Computer Interaction (HCI) technology. In new generation of technology, wireless mouse or a contact less mouse still uses devices and isn't freed from devices fully, since it uses power from the device or may be from external power sources like battery and acquire space and electric power, also during COVID pandemic it is advised to make social distancing and avoid to touch things which is used by different peoples.

Within the projected AI virtual mouse using hand gesture system, this limitation may be resolve by using digital camera or constitutional camera for recognize the hand gestures and fingers detection mistreatment computer machine vision. The algorithmic rule utilized in the system makes use of the artificial intelligence and machine learning algorithmic rule. supported the hand gestures, the device may be controlled just about and might doing left click, right click, scrolling functions, and computer device pointer perform while not the utilization of the physical mouse. The algorithmic rule is predicated on deep learning for analyse the gesture of hands. Hence, the is project can avoid COVID-19 unfold by reducing the human intervention and dependency of devices to regulate the computer machine, this vision are very use full during COVID for maintain social distance & untouchability.

The main objective of the AI virtual mouse system is to Control the mouse pointer functions by victimization the hand gestures instead of using a physical mouse. The projected system could also be achieved by using a web or external camera or a integral camera that detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions.

For the aim of detection of hand gestures, the Media Pipe framework is employed, and OpenCV library is employed for computer machine vision the rule makes use of the machine learning ideas to trace and acknowledge the hand gestures and hand tip with the python package like Pyauto, Numpy.

CONTRIBUTION OF INDIVIDUAL TEAM MEMBER

Name of the Student(s)	Enrollment/Admission No.	Contributions
Joy Guha	19SCSE1010040	Developer, Analysis
Shreya Kumari	19SCSE1010489	Testing, Documentation

List of Figures

Table of Contents

CHAPTER-1

INTRODUCTION

In the new modern technologies at intervals the areas of exaggerated reality and devices that we tend to tend to use in our existence, these devices have gotten compact at intervals the sort of Bluetooth or wireless technologies. This paper proposes Associate in Nursing AI virtual mouse system that produces use of the hand gestures and hand tip detection for performing arts mouse functions at intervals the portable computer exploitation portable computer vision. The most objective of the projected system is to perform device pointer operates and scroll perform employing an internet camera or an inherent camera at intervals the portable computer instead of employing an ancient mouse device. Hand gesture and hand tip detection by exploitation portable computer vision is used as an HCI [with the laptop. With the use of the AI virtual mouse system, we are going to track the tip of the hand gesture by employing an inherent camera or net camera and perform the mouse pointer operations and scrolling operate and jointly move the pointer with it.

While using a wireless or a Bluetooth mouse, some devices just like the mouse, the device to connect to the pc, and also, battery to power the mouse to control a used, but throughout this paper, the user uses his/her inherent camera or a photographic camera and uses his/her hand gestures to manage the laptop mouse operations. Within the projected system, information superhighway camera captures therefore process the frames that square measure captured therefore acknowledges the various hand gestures and hand tip gestures so performs the particular mouse operate.

Python programming language is used for developing the AI virtual mouse system, and also, OpenCV that's that the library for portable computer vision is used at intervals the AI virtual mouse system. Within the projected AI virtual mouse using hand gesture, the model makes use of the python Mediapipe package for the chase of the hands and for chase of the tip of the hands, and also, Numpy, Autopy, and PyAuto GUI packages were used for on the move the screen of the laptop for performing arts functions like left click, right click, and scrolling functions. The results of the projected model showed really high accuracy level, and thus the projected model can work very well in real-world application with the use of a processor whereas not the use of computer GPU.

1.1 FORMULATION OF PROBLEM

The projected AI virtual mouse using hand gesture system could also be accustomed to overcome problems inside the place like things where there is not any space to use a physical mouse and put together for the persons who have issues in their hands and do not appear to be able to manage a physical mouse. Also, the COVID state of affairs, it isn't safe to use the devices by touching them as a results of it's aiming to result in a possible state of affairs of unfold of the virus by touching the devices, that the projected AI virtual mouse could also be accustomed overcome these problems since hand gesture and hand Tip detection is used to manage the device mouse functions by using a camera or an inherent camera like web cam.

The main objective of the planned AI virtual mouse system is to develop alternate to the regular and ancient mouse system to perform and management the mouse functions, and this could be achieved with the help of an interior net camera that captures the hand gestures and hand tip then processes these frames to perform the particular mouse performs like left click, right click, and scrolling perform. Hence, the projected system will avoid COVID-19 unfold by eliminating the human intervention and dependency of devices to control the pc, this vision at terribly use full throughout covid for maintain social distance & untouchability.
Virtual Mouse exploitation Hand gesture recognition permits users to manage mouse with the assistance of hand gestures. System's digital web camera is employed for following hand gestures. pc vision techniques were used for gesture recognition. OpenCV python package consists as video capture that is employed to capture information from a live video, main issue we want to spot the applications the model goes to develop so the development of the mouse movement without touching or using of the mouse.

1.1.1 Tools and Technology Used

- ➢ Python 3.8.10
- ➢ Visual Code Text Editor
- ➢ OpenCV
- ➢ MediaPipe
- ➢ Pyauto
- ➢ Numpy

CHAPTER-2

LITERATURE SURVEY/ PROJECT DESIGN

2.1 Preface

Multi-point Interactive Whiteboards are available using the Wiimote [4]. The components used are IR pen, computer with Windows XP (installed with Microsoft .NET framework, the Wiimote Connect program and the Wiimote Whiteboard software), wiimote controller, a beamer capable of a 1024 x 786 pixel resolution. Here the wiimote controller tracks the infra-red source on the white board and sends info to PC via Bluetooth. The teaching platform comprises of a Wii-mote based multi-touch teaching station, a Wii-mote-based interactive whiteboard and a Wii-mote-based stylus input conversion tool [5]. According to the literature survey, most people have used the Wii-mote to configure it as a virtual marker.

There are traditional approaches for virtual keyboard and mouse systems which are usually based on eye gestures. Our literature review focuses on the research works on virtual keyboard and virtual mouse which were published in Elsevier, Springer, ACM Digital Library, IEEE Digital Library etc. We discussed about few related works on virtual keyboard and virtual mouse in the following two subsections.

A lot of virtual markers are available in market, but it can only function as a marker alone. A high level processor is used to process the data and used as a virtual marker, but it can do even more function like mouse functions. Its capability is not being utilized fully to its ability to function and hence the product becomes under-loaded and hence costlier in terms of market price versus function.

In 2016, S. Shetty et al. constructed a virtual mouse system using color detection.They used webcam for detecting mouse cursor movement and click events using OpenCV built-in functions. A mouse driver, written in java, is required as well. This system fails to perform well in rough background. P. C. Shindhe et al. expanded a method for mouse free cursor control where mouse cursor operations are controlled by using hand fingers. They have collected hand gestures via webcam using color detection principles. The built-in function of Image Processing Toolbox in MATLAB and a mouse driver, written in java, used in this approach. The pointer was not too efficient on the air as the cursor was very sensitive to the motion. G. Sahu et al. built a system for controlling mouse pointer using webcam which control volume of media player, PowerPoint slides and can make or end a call. They used RGB color tapes to recognize user's finger. In 2019, K. Hassan et al. presented a system to design and develop a hand gesture based virtual mouse. They captured different gestures via webcam and performed mouse functions according to the gestures. This system achieved 78%-90% accuracy. The system does not work efficiently in the complex or rough background. As we can see from the reviewed literature, previous systems includes either virtual keyboard or virtual mouse. Those systems can't fully eliminate the need of mouse and

keyboard completely. This work aims to build an interactive computer system which can be operated without any physical mouse and keyboard.

2.2 Existing System

There are many AR VR devices available nowadays. Some are designed by either linguistic or machine learning models. Different AR VR have different works depending upon the companies and their requirements. Many researchers in computer science and human computer interaction developed various technologies related to virtual keyboard and mouse. However all of them used different techniques. Approaches related to keyboard, where in Eckert. Ddeveloped for the persons with physical impairments with presenting a new middleware for mapping gestures, obtained by a motion sensing camera device. Another approach was developed by Zhang, Yunzhou introduced a method by the use of infrared laser module, keyboard pattern projector, embedded system and a single image sensor where every keystroke can be determine accurately by image processing including morphology principle and ellipse fitting Approach related to mouse. One approach, by Erdem to, controls the motion of the mouse by fingertip tracking. A click of the mouse button was implemented on the screen such that a click occurred when a user's hand passed over the region [3, 4]. Another approach was developed by ChuFeng Lien . He controls the mouse cursor and clicking event by using the finger-tips movement. His clicking method was based on image density, and required the user to hold the mouse cursor on the desired spot for a short period of time. Paul et al, used some another method to click. He used the motion of the thumb from a 'thumbs-up' position to a fist to mark a clicking event of thumb. By

Making a special hand sign moved the mouse pointer. Our project was inspired by a paper of Jun Hu.They developed bare-finger touch interaction on regular planar surfaces for e.g. walls or tables, with only one standard camera and one projector. The touching information of finger tips is recovered just from the 2-D image captured by the camera. We used the concept of camera and image processing but without the help of projector and laser light a simple keyboard is drawn on the paper and the movement of typing is captured by camera same for the mouse the eye movement is captured.

Here unit some connected works assigned on virtual mouse victimization hand gesture detection by sporting a glove among the hand and in addition victimization color tips among the hands for gesture recognition, but they are now further correct in mouse functions. The recognition is not thus correct as result of sporting gloves; in addition, the gloves don't seem to be fitted to some users, and in some cases, the recognition is not thus correct as results of the failure of detection of color tips. Some efforts square measure created for camera-based detection of the hand gesture interface. In 1990, Quam introduced associate early hardware-based system;

throughout this technique, the user got to wear a knowledge Glove. The projected system by Quam tho' provides results of higher accuracy; but it's powerful to perform variety of the gesture controls victimization the system. Dung-Hua Liou, ChenChiung Hsieh, and David Lee in 2010 projected a study on "A period of time Hand Gesture Recognition System victimization Motion History Image." the foremost limitation of this model is further subtle hand gestures. Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013 projected a study on "Cursor system victimization Hand Gesture Recognition." throughout this work, the limitation is keep frames square measure needed to be processed for hand segmentation and skin component detection. Vinay Kr. Pasi, Saurabh Singh, and Pooja Kumari in 2016 projected "Cursor management victimization Hand Gestures" among the IJCA Journal. The system proposes completely totally different|the various bands to perform different functions of the mouse. The limitation is it depends on various colors to perform mouse functions. Chaitanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 projected "Virtual Mouse victimization Hand Gesture" where the model detection relies on colors. But, alone few mouse functions square measure performed

2.3 Proposed System

The main objective of the planned AI virtual mouse system is to develop alternate to the regular and ancient mouse system to perform and management the mouse functions, and this could be achieved with the help of an interior net camera that captures the hand gestures and hand tip then processes these frames to perform the particular mouse performs like left click, right click, and scrolling perform. Hence, the projected system will avoid COVID-19 unfold by eliminating the human intervention and dependency of devices to control the pc, this vision at terribly use full throughout covid for maintain social distance & untouchability.

Virtual Mouse exploitation Hand gesture recognition permits users to manage mouse with the assistance of hand gestures. System's digital web camera is employed for following hand gestures. pc vision techniques were used for gesture recognition. OpenCV python package consists as video capture that is employed to capture information from a live video, main issue we want to spot the applications the model goes to develop so the development of the mouse movement without touching or using of the mouse.

For the aim of detection of hand gestures and hand gesture, the Media Pipe framework is employed, and OpenCV library is employed for computer machine vision the rule makes use of the machine learning ideas to trace and acknowledge the hand gestures and hand tip.

CHAPTER-3

SYSTEM ANALYSIS AND DESIGN

3.1 Requirement Specification

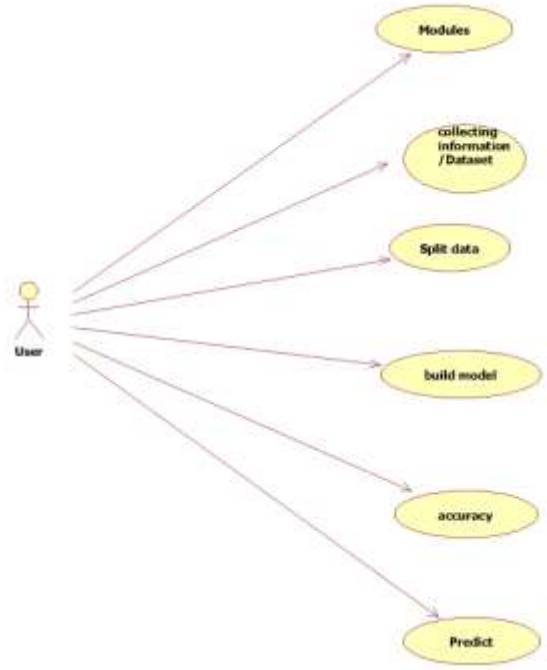
3.1.1 Functional Requirement

- **Softwares**
  - Windows 7-10
  - Visual Studio or Pycharm.
  - Python 3.8.10
- **Virtual Mouse**
  - OpenCV python package
  - Mediapipe python package
  - Autopy python package

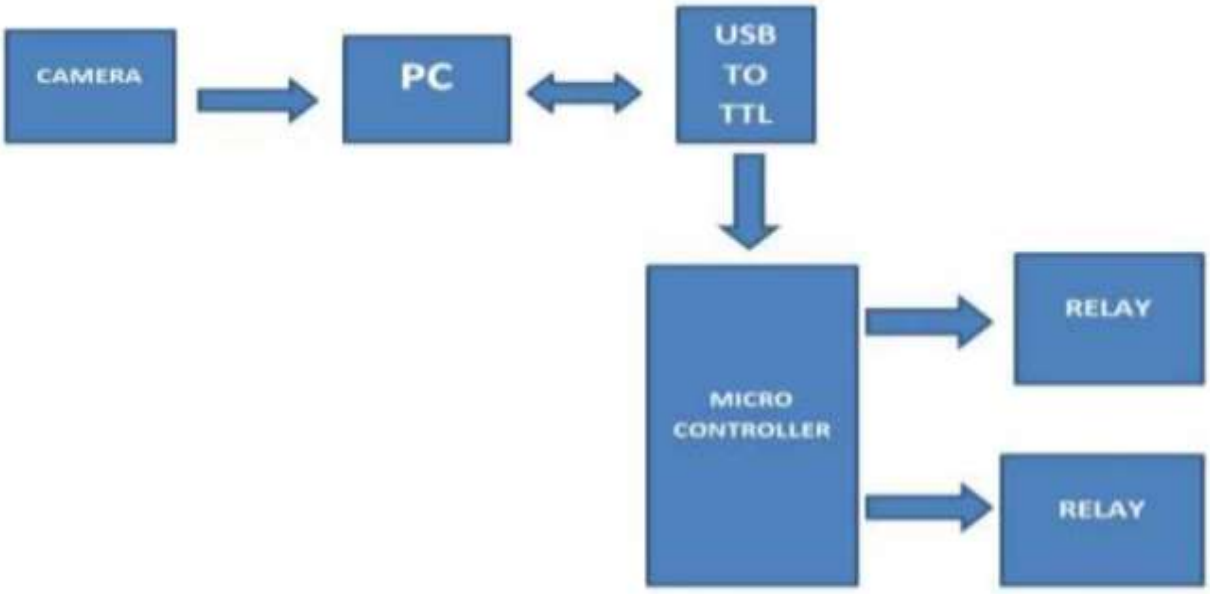
3.1.2 Non-Functional Requirement

- **Security Requirement**
  - This app is safe to use.
  - All functions are fully tested & save to use
  - Don't impact on personal data
- **Safety Requirement**
  - This app don't requires an internet connection after installation.
  - This app need some space to download python package.

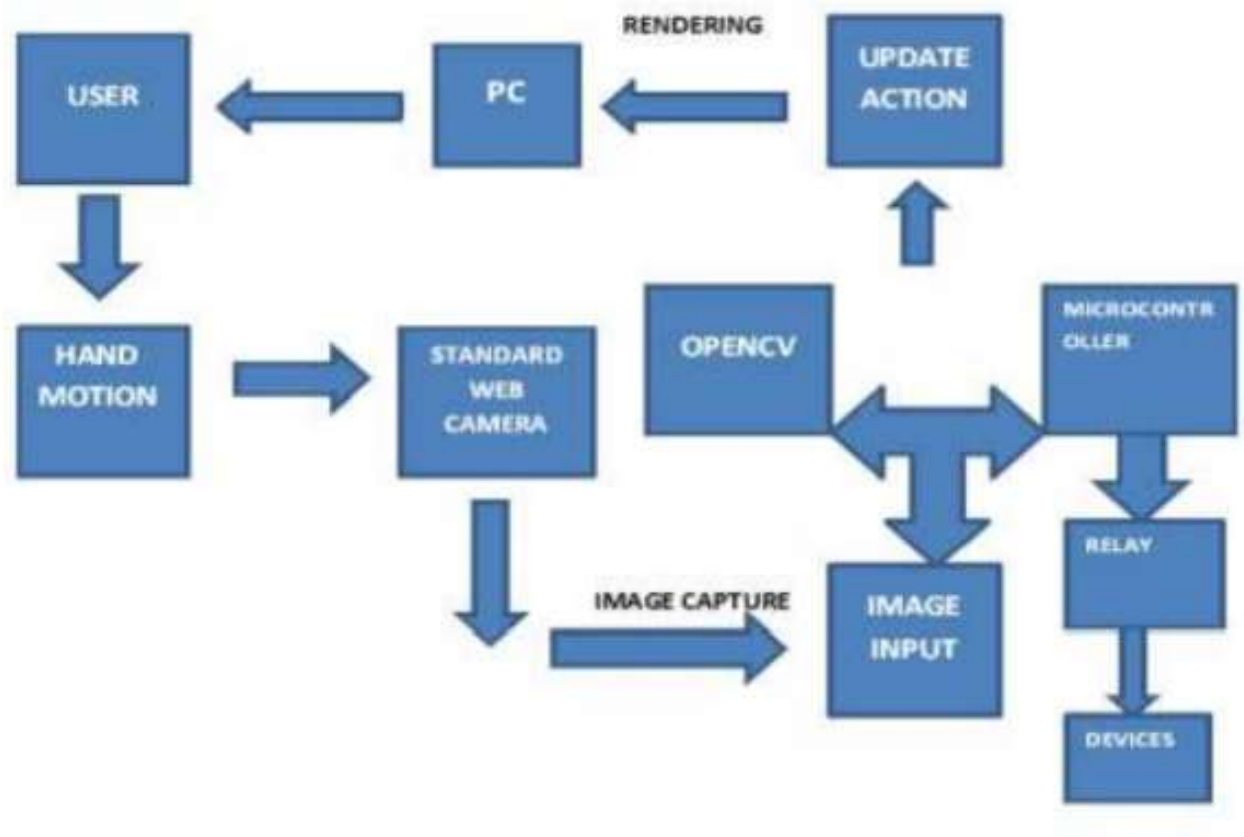
3.2 Use Case Diagram



3.3 Block Diagram

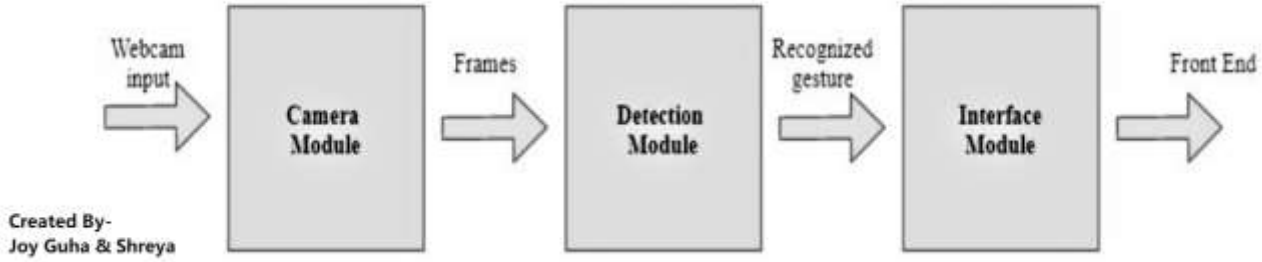


3.4 Sequence Diagram

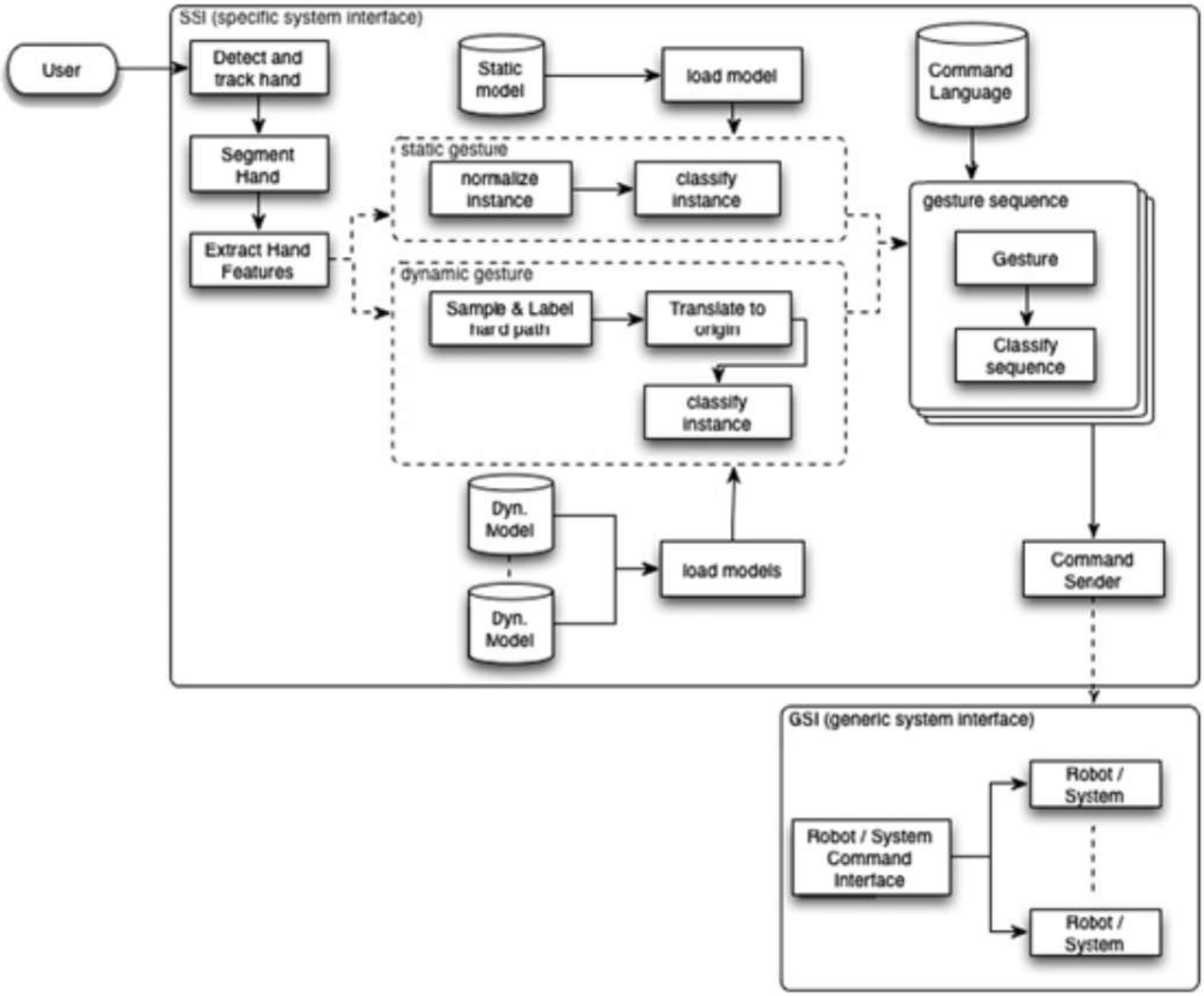


3.5 Use-Case Diagram

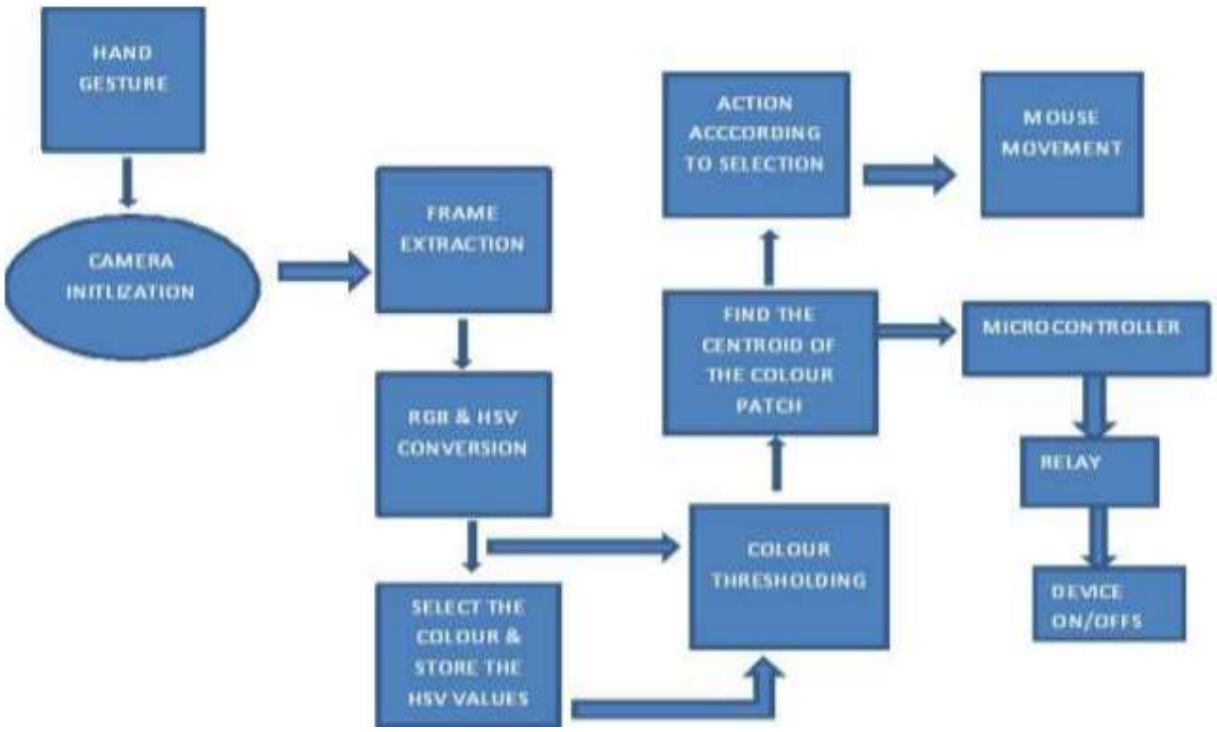
Level 0 DFD



Level 1 DFD



3.6 System Architectural Diagram



What is Media Pipe?

Media Pipe is a framework that is employed for applying in a vvarious machine learning pipeline, associate with an open source framework of Google. The Media Pipe framework is helpful for cross platform development since the framework is made mistreatment the statistic information. The Media Pipe framework is multi-modal, wherever this framework is often applied to varied audios and videos .The Media Pipe framework is employed by the developer for building and analyzing the systems through graphs, and it conjointly been used for developing the systems for the appliance purpose. The steps concerned within the system that uses Media Pipe square measure administrated within the pipeline configuration.

The pipeline created will run in numerous platforms permitting measurability in mobile and desktops. The Media Pipe framework is predicated on 3 elementary parts; they're performance analysis, framework for retrieving detector information, and a group of parts that square measure known as calculators, and those they square measure reusable. A pipeline may be a graph that consists of parts known as calculators, wherever every calculator is connected by streams during which the packets of knowledge flow through. Developers square measure able to replace or outline custom calculators anyplace within the graph making their own application. The calculators and streams combined produce a data-flow diagram; the graph (Figure 1) is made with Media Pipe wherever every node may be a calculator and therefore the nodes square measure connected by streams.

Here we have attached the media pipe figure for the reference that will give you a knowledge about the media pipe how it work and how it collect the gesture of the hand . A pipeline may be a graph that consists of parts known as calculators, wherever every calculator is connected by streams during which the packets of knowledge flow through. Developers square measure able to replace or outline custom calculators anyplace within the graph making their own application. The Media Pipe framework is employed by the developer for building and analyzing the systems through graphs, and it conjointly been used for developing the systems for the appliance purpose. The steps concerned within the system that uses Media Pipe square measure administrated within the pipeline configuration.

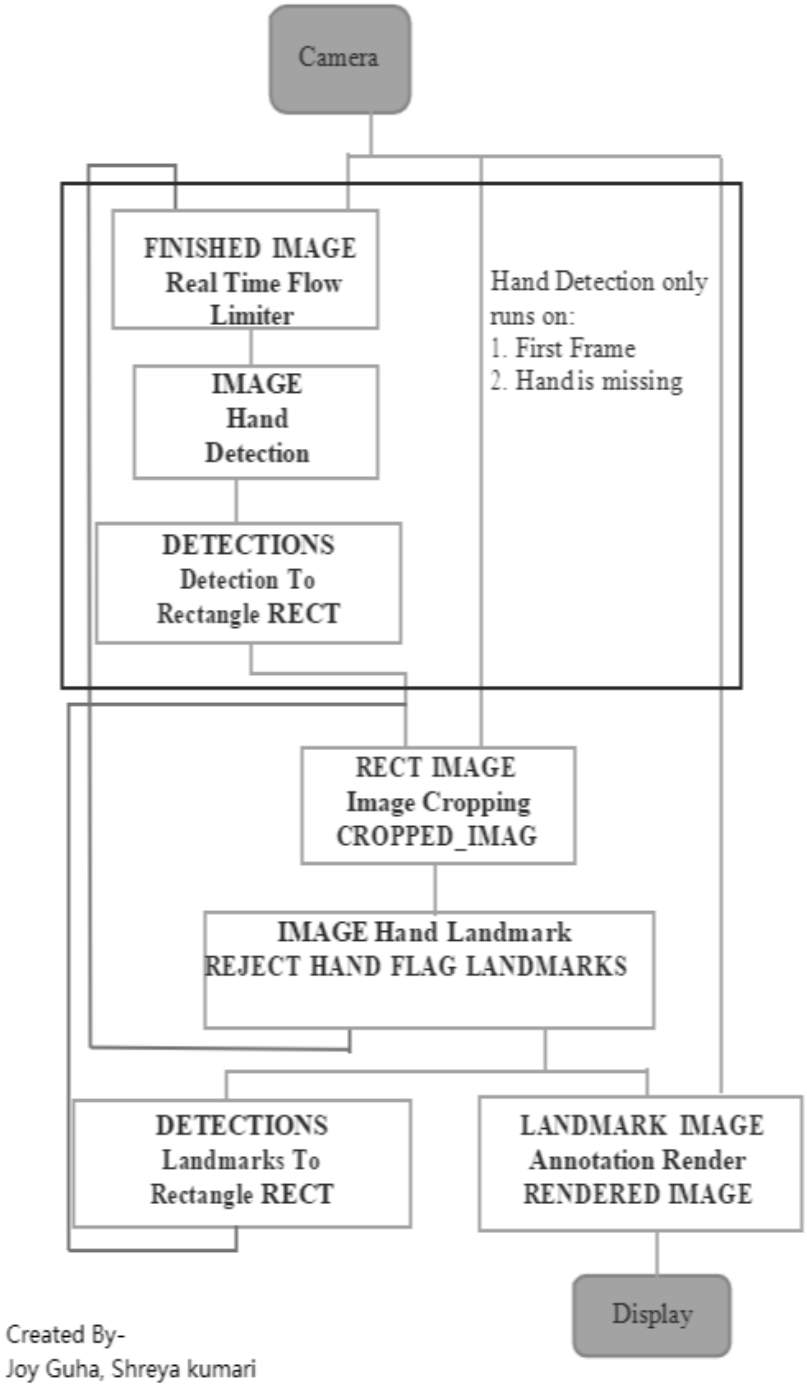


Figure of Media Pipe hand tip recognition layout.

Single-shot is employed for detection & recognizing a finger and palm in real time victimization notebook computer net cam. Detector system is employed by the Media Pipe, in the hand detection module of python, its style for a finger and hand detection model as a result of it's simple to coach hand. The planed model of hand purpose mark consists of twenty one joint purpose and co-ordinates within the hand

1. WRIST
2. THUMB CMC
3. THUMB MCP
4. THUMB IP
5. THUMB TIP
6. INDEX FINGER MCP
7. INDEX FINGER PIP
8. INDEX FINGER DIP
9. INDEX FINGER TIP
10. MIDDLE FINGER MCP
11. MIDDLE FINGER PIP
12. MIDDLE FINGER DIP
13. MIDDLE FINGER TIP
14. RING FINGER MCP
15. RING FINGER PIP
16. RING FINGER DIP
17. RING FINGER TIP
18. PINKY MCP
19. PINKY PIP
20. PINKY DIP
21. PINKY TIP

3.2. What is OpenCV?

OpenCV is laptop vision library that contains Associate in Nursing image-processing algorithms for Associate in Nursing object detection. OpenCV could also be a library of python programing language, and amount of your time laptops vision applications ar usually developed by victimization the non-public computer vision library. The OpenCV library is utilized AN exceedingly in a very image and video method Associate in Nursing conjointly analysis sort of a face detection and an object detection.

3.3. Methodology

Nursing image-processing algorithms for Associate in Nursing object detection. OpenCV may be a library of python programming language, and period of time pc vision applications are often developed by victimization the pc vision library. The OpenCV library is employed in a picture and (also the) video process Associate in Nursing Associate in Nursing also analysis like a face detection and an object detection. In fig 4 the methodology is explained.

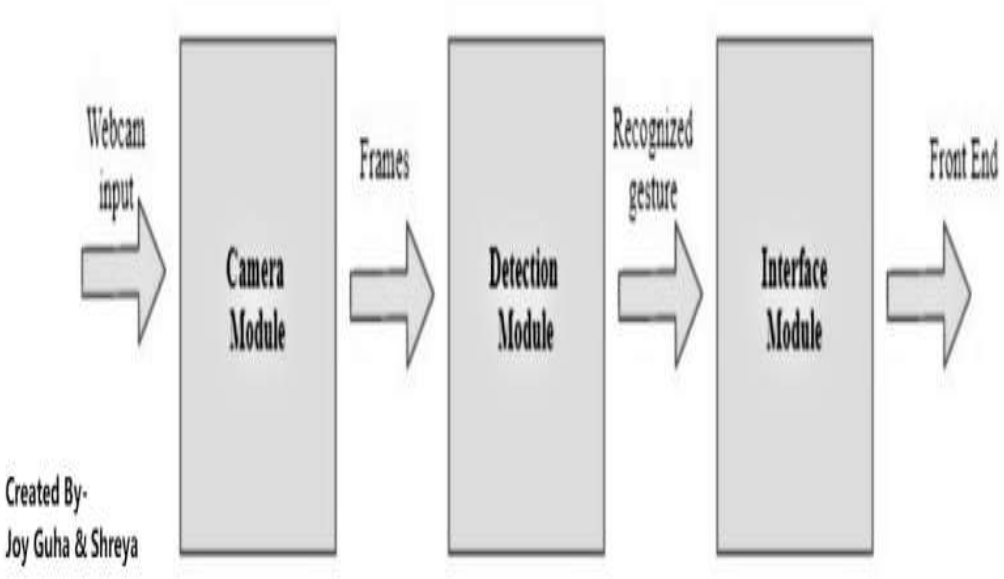
3.4 Related Work

Here unit some connected works assigned on virtual mouse victimization hand gesture detection by sporting a glove among the hand and in addition victimization color tips among the hands for gesture recognition, but they are now further correct in mouse functions. The recognition is not thus correct as result of sporting gloves; in addition, the gloves don't seem to be fitted to some users, and in some cases, the recognition is not thus correct as results of the failure of detection of color tips. Some efforts square measure created for camera-based detection of the hand gesture interface. In 1990, Quam introduced associate early hardware-based system; throughout this technique, the user got to wear a knowledge Glove. The projected system by Quam tho' provides results of higher accuracy; but it's powerful to perform variety of the gesture controls victimization the system. Dung-Hua Liou, ChenChiung Hsieh, and David Lee in 2010 projected a study on "A period of time Hand Gesture Recognition System victimization Motion History Image." the foremost limitation of this model is further subtle hand gestures. Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013 projected a study on "Cursor system victimization Hand Gesture Recognition." throughout this work, the limitation is keep frames square measure needed to be processed for hand segmentation and skin component detection. Vinay Kr. Pasi, Saurabh Singh, and Pooja Kumari in 2016 projected "Cursor management victimization Hand Gestures" among the IJCA Journal.

The system proposes completely totally different|the various bands to perform different functions of the mouse. The limitation is it depends on various colors to perform mouse functions. Chaitanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 projected "Virtual Mouse victimization Hand Gesture" where the model detection relies on colors. But, alone few mouse functions square measure performd.

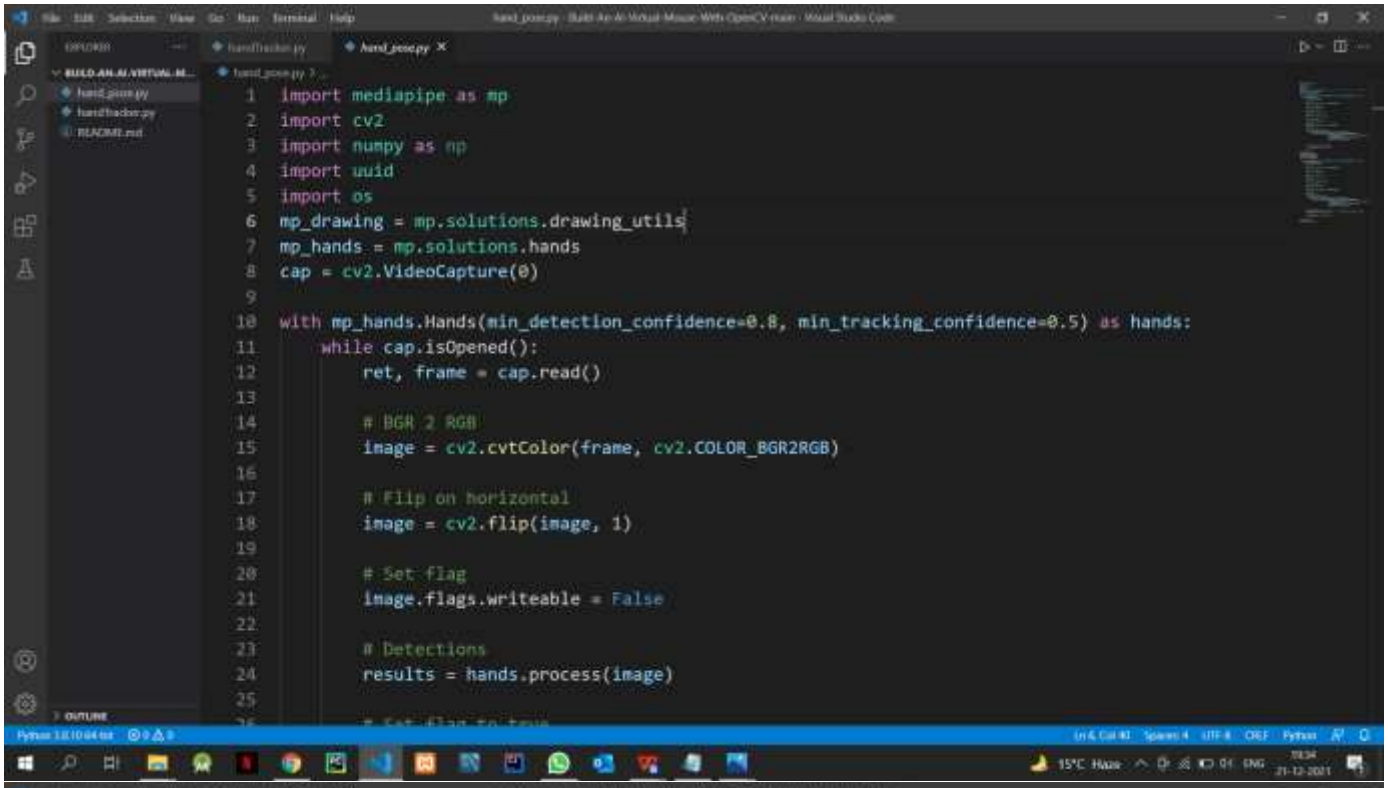
3.5 Algorithm Used for AI Virtual Mouse for Hand Gesture Tracking

For the aim of detection of hand gestures and hand gesture, the Media Pipe framework is employed, and OpenCV library is employed for computer machine vision the rule makes use of the machine learning ideas to trace and acknowledge the hand gestures and hand tip.

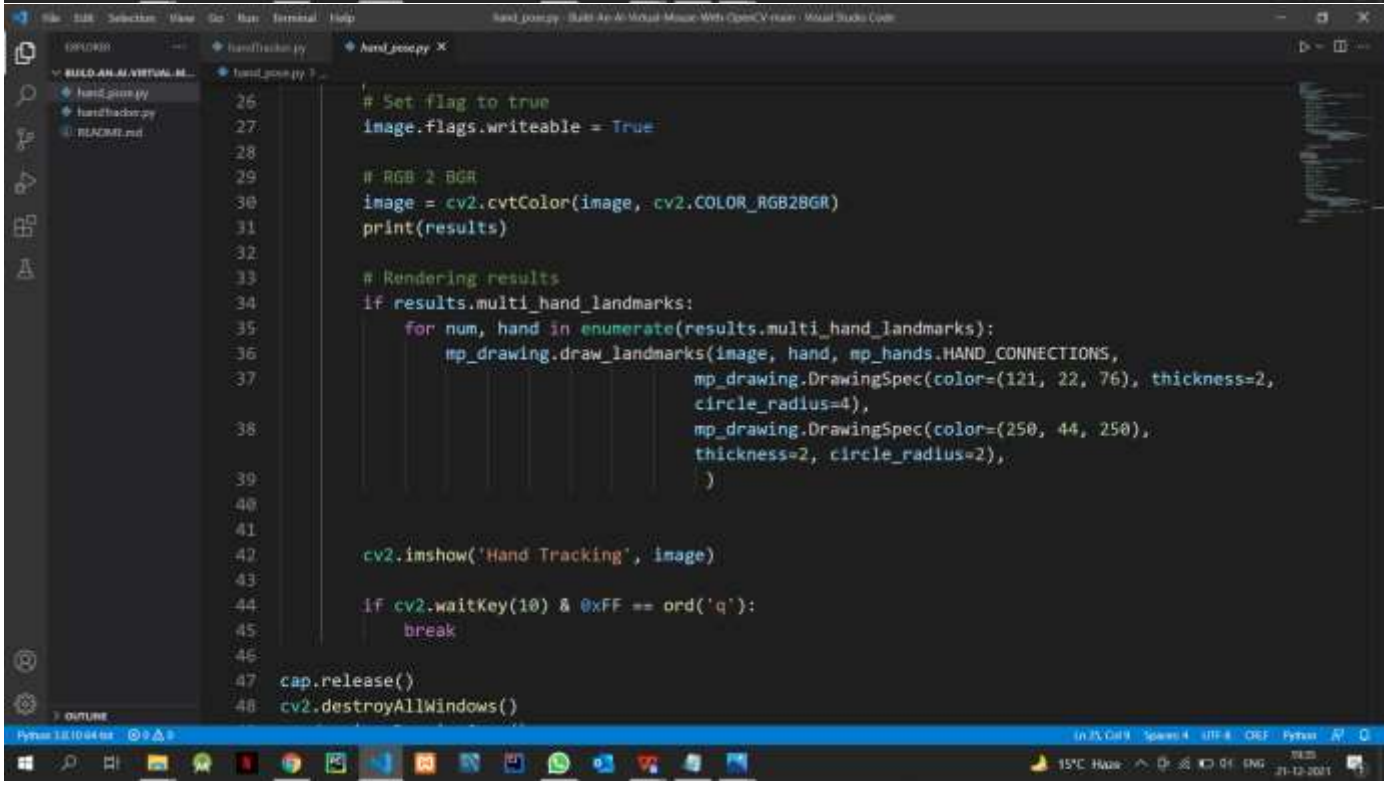


3.6 Algorithm and Pseudo Code

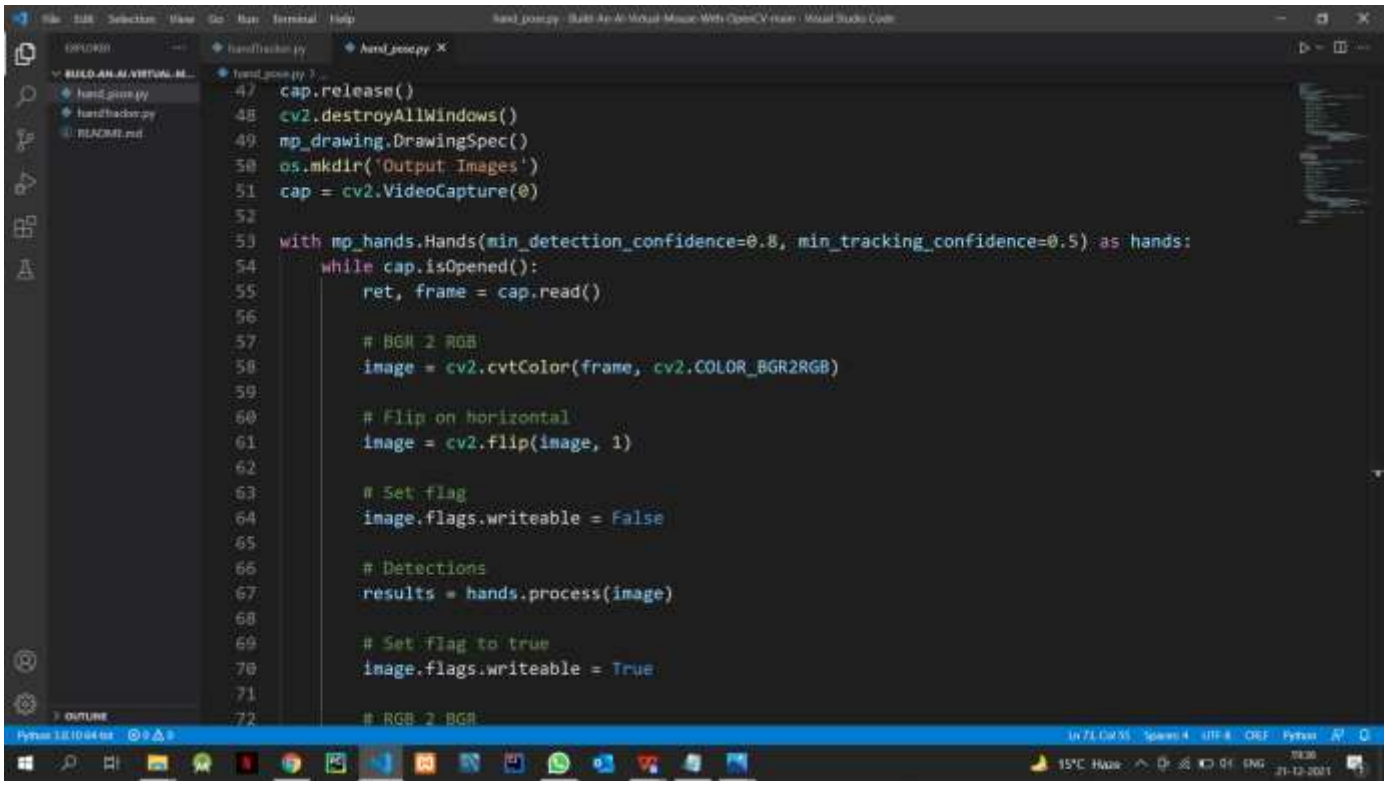
- **Hand Pose module**

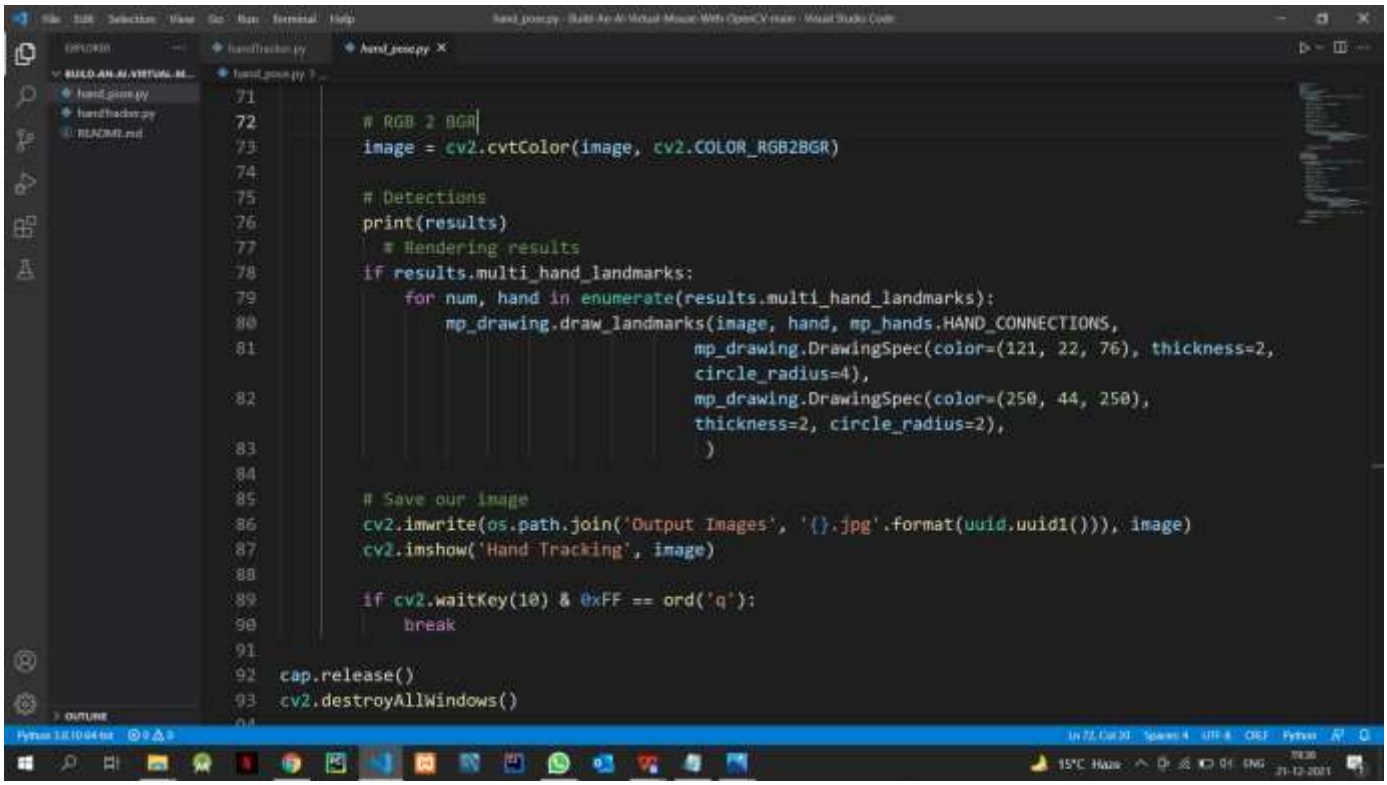


```python
import mediapipe as mp
import cv2
import numpy as np
import uuid
import os
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands
cap = cv2.VideoCapture(0)

with mp_hands.Hands(min_detection_confidence=0.8, min_tracking_confidence=0.5) as hands:
    while cap.isOpened():
        ret, frame = cap.read()

        # BGR 2 RGB
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        # Flip on horizontal
        image = cv2.flip(image, 1)

        # Set flag
        image.flags.writeable = False

        # Detections
        results = hands.process(image)
```

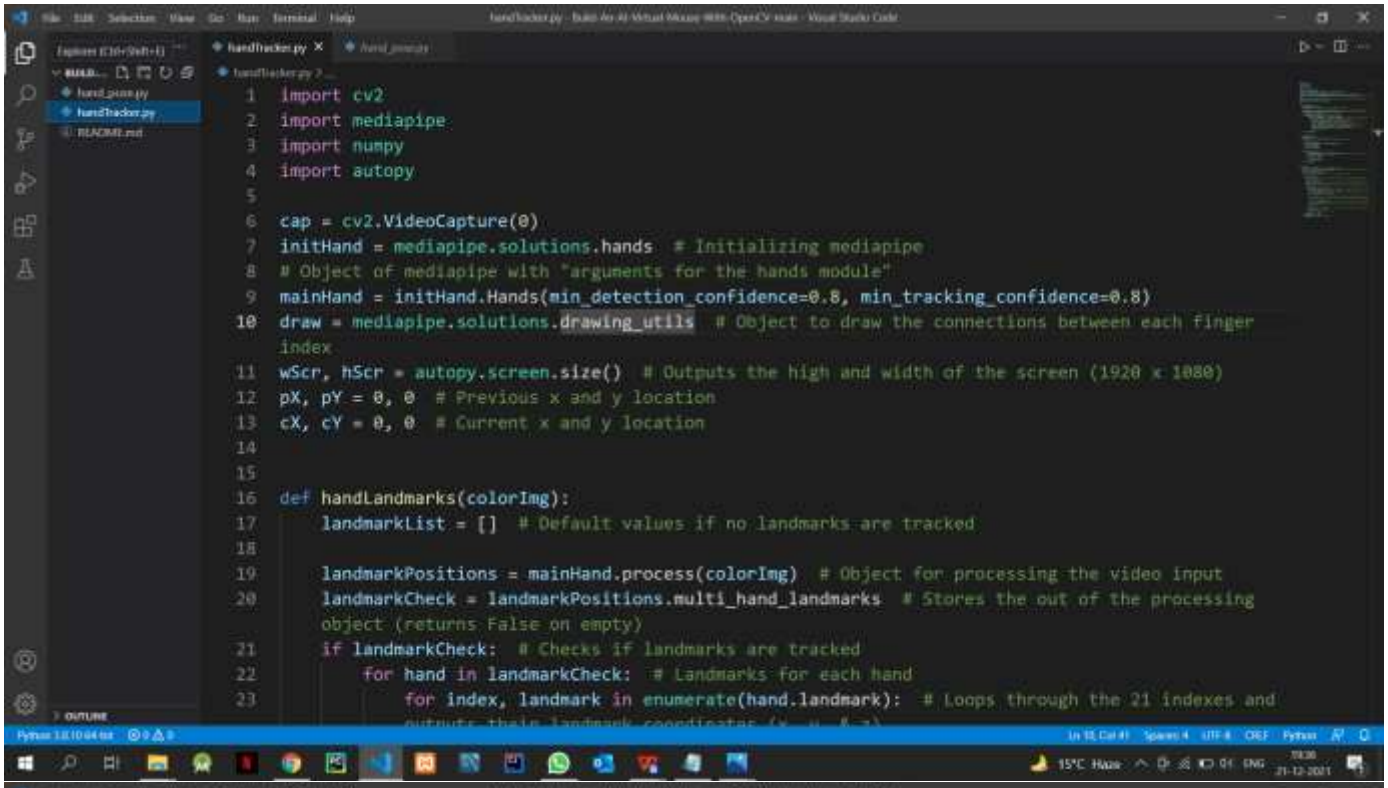


```python
        # Set flag to true
        image.flags.writeable = True

        # RGB 2 BGR
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
        print(results)

        # Rendering results
        if results.multi_hand_landmarks:
            for num, hand in enumerate(results.multi_hand_landmarks):
                mp_drawing.draw_landmarks(image, hand, mp_hands.HAND_CONNECTIONS,
                                          mp_drawing.DrawingSpec(color=(121, 22, 76), thickness=2,
                                          circle_radius=4),
                                          mp_drawing.DrawingSpec(color=(250, 44, 250),
                                          thickness=2, circle_radius=2),
                                           )


        cv2.imshow('Hand Tracking', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

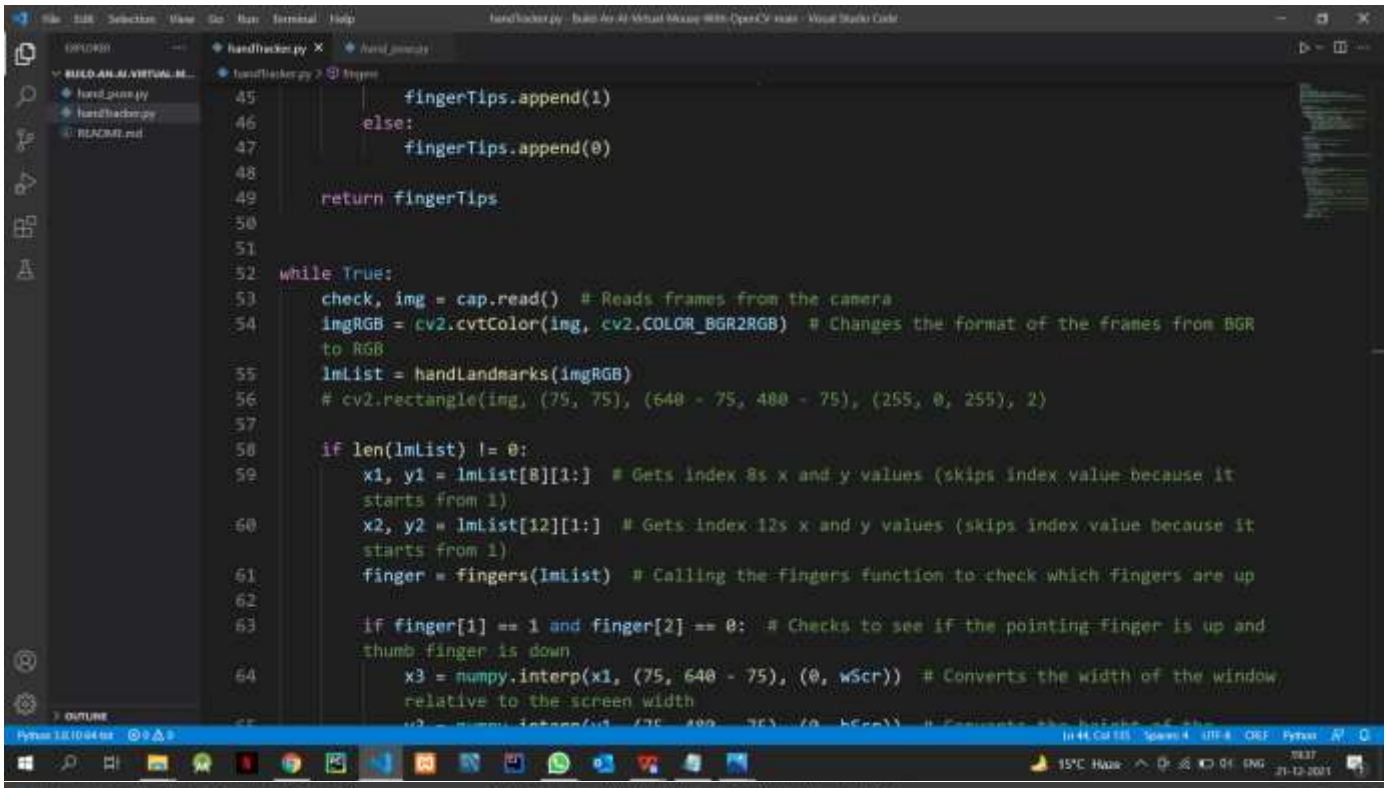

cap.release()
cv2.destroyAllWindows()
mp_drawing.DrawingSpec()
os.mkdir('Output Images')
cap = cv2.VideoCapture(0)
with mp_hands.Hands(min_detection_confidence=0.8, min_tracking_confidence=0.5) as hands:
while cap.isOpened():
ret, frame = cap.read()
# BGR 2 RGB
image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
# Flip on horizontal
image = cv2.flip(image, 1)
# Set flag
image.flags.writeable = False
# Detections
results = hands.process(image)
# Set flag to true
image.flags.writeable = True
# RGB 2 BGR


hand_pose.py
handTracker.py
# RGB 2 BGR
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
# Detections
print(results)
# Rendering results
if results.multi_hand_landmarks:
for num, hand in enumerate(results.multi_hand_landmarks):
mp_drawing.draw_landmarks(image, hand, mp_hands.HAND_CONNECTIONS,
mp_drawing.DrawingSpec(color=(121, 22, 76), thickness=2,
circle_radius=4),
mp_drawing.DrawingSpec(color=(250, 44, 250),
thickness=2, circle_radius=2),
)
# Save our image
cv2.imwrite(os.path.join('Output Images', '{}.jpg'.format(uuid.uuid1())), image)
cv2.imshow('Hand Tracking', image)
if cv2.waitKey(10) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()

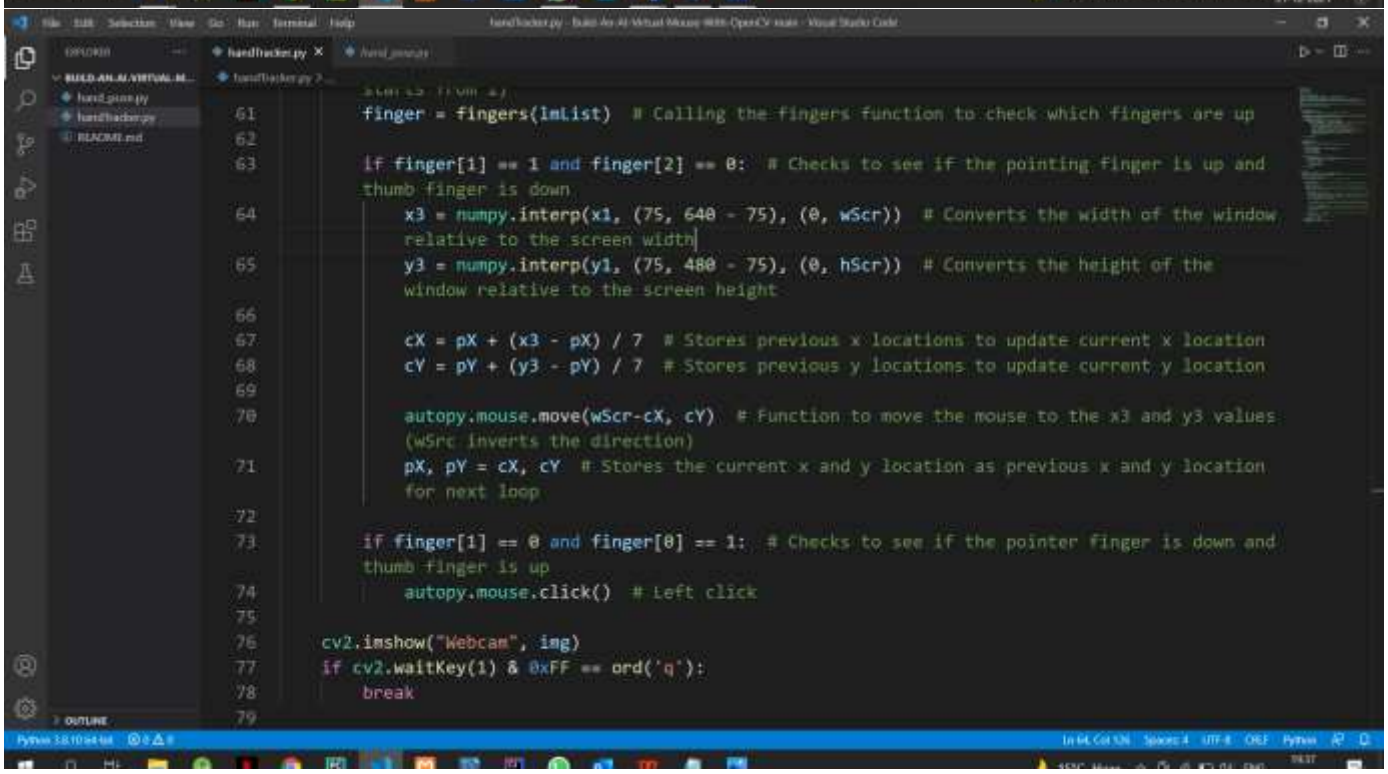
- **Hand tracing module**



```python
import cv2
import mediapipe
import numpy
import autopy

cap = cv2.VideoCapture(0)
initHand = mediapipe.solutions.hands  # Initializing mediapipe
# Object of mediapipe with "arguments for the hands module"
mainHand = initHand.Hands(min_detection_confidence=0.8, min_tracking_confidence=0.8)
draw = mediapipe.solutions.drawing_utils  # Object to draw the connections between each finger index
wScr, hScr = autopy.screen.size()  # Outputs the high and width of the screen (1920 x 1080)
pX, pY = 0, 0  # Previous x and y location
cX, cY = 0, 0  # Current x and y location


def handLandmarks(colorImg):
    landmarkList = []  # Default values if no landmarks are tracked

    landmarkPositions = mainHand.process(colorImg)  # Object for processing the video input
    landmarkCheck = landmarkPositions.multi_hand_landmarks  # Stores the out of the processing object (returns False on empty)
    if landmarkCheck:  # Checks if landmarks are tracked
        for hand in landmarkCheck:  # Landmarks for each hand
            for index, landmark in enumerate(hand.landmark):  # Loops through the 21 indexes and outputs their landmark coordinates (x, y, & z)
                draw.draw_landmarks(img, hand, initHand.HAND_CONNECTIONS)  # Draws each individual index on the hand with connections
                h, w, c = img.shape  # Height, width and channel on the image
                centerX, centerY = int(landmark.x * w), int(landmark.y * h)  # Converts the decimal coordinates relative to the image for each index
                landmarkList.append([index, centerX, centerY])  # Adding index and its coordinates to a list

    return landmarkList


def fingers(landmarks):
    fingerTips = []  # To store 4 sets of 1s or 0s
    tipIds = [4, 8, 12, 16, 20]  # Indexes for the tips of each finger

    # Check if thumb is up
    if landmarks[tipIds[0]][1] > lmList[tipIds[0] - 1][1]:
        fingerTips.append(1)
    else:
        fingerTips.append(0)

    # Check if fingers are up except the thumb
    for id in range(1, 5):
        if landmarks[tipIds[id]][2] < landmarks[tipIds[id] - 3][2]:  # Checks to see if the tip of the finger is higher than the joint
```



```
            fingerTips.append(1)
        else:
            fingerTips.append(0)

    return fingerTips


while True:
    check, img = cap.read()  # Reads frames from the camera
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  # Changes the format of the frames from BGR
    to RGB
    lmList = handLandmarks(imgRGB)
    # cv2.rectangle(img, (75, 75), (640 - 75, 480 - 75), (255, 0, 255), 2)

    if len(lmList) != 0:
        x1, y1 = lmList[8][1:]  # Gets index 8s x and y values (skips index value because it
        starts from 1)
        x2, y2 = lmList[12][1:]  # Gets index 12s x and y values (skips index value because it
        starts from 1)
        finger = fingers(lmList)  # Calling the fingers function to check which fingers are up

        if finger[1] == 1 and finger[2] == 0:  # Checks to see if the pointing finger is up and
        thumb finger is down
            x3 = numpy.interp(x1, (75, 640 - 75), (0, wScr))  # Converts the width of the window
            relative to the screen width
```



```
        finger = fingers(lmList)  # Calling the fingers function to check which fingers are up

        if finger[1] == 1 and finger[2] == 0:  # Checks to see if the pointing finger is up and
        thumb finger is down
            x3 = numpy.interp(x1, (75, 640 - 75), (0, wScr))  # Converts the width of the window
            relative to the screen width
            y3 = numpy.interp(y1, (75, 480 - 75), (0, hScr))  # Converts the height of the
            window relative to the screen height

            cX = pX + (x3 - pX) / 7  # Stores previous x locations to update current x location
            cY = pY + (y3 - pY) / 7  # Stores previous y locations to update current y location

            autopy.mouse.move(wScr-cX, cY)  # Function to move the mouse to the x3 and y3 values
            (wScr inverts the direction)
            pX, pY = cX, cY  # Stores the current x and y location as previous x and y location
            for next loop

        if finger[1] == 0 and finger[0] == 1:  # Checks to see if the pointer finger is down and
        thumb finger is up
            autopy.mouse.click()  # Left click

    cv2.imshow("Webcam", img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

CHAPTER-4

RESULTS AND DISCUSSION

4.1 User Interface

Screen Name	Description
Camera window	This window is for user to see theri hand image and give them area to move their hand.
Cursor Point	This is the point of mouse cursor in which user can understand where they move the mouse.

Table-1.1

Working output

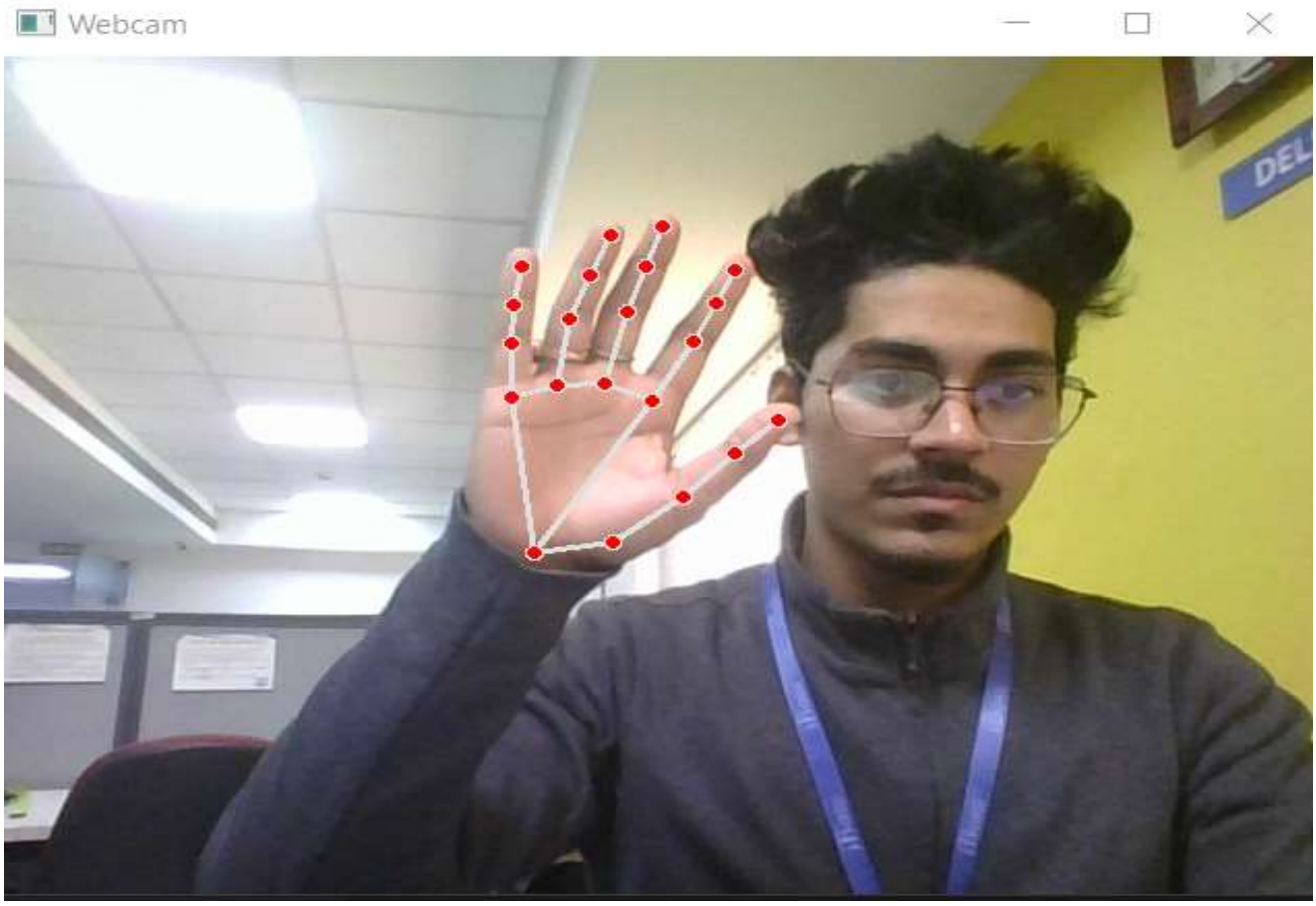


Fig 1.1 capturing the hand

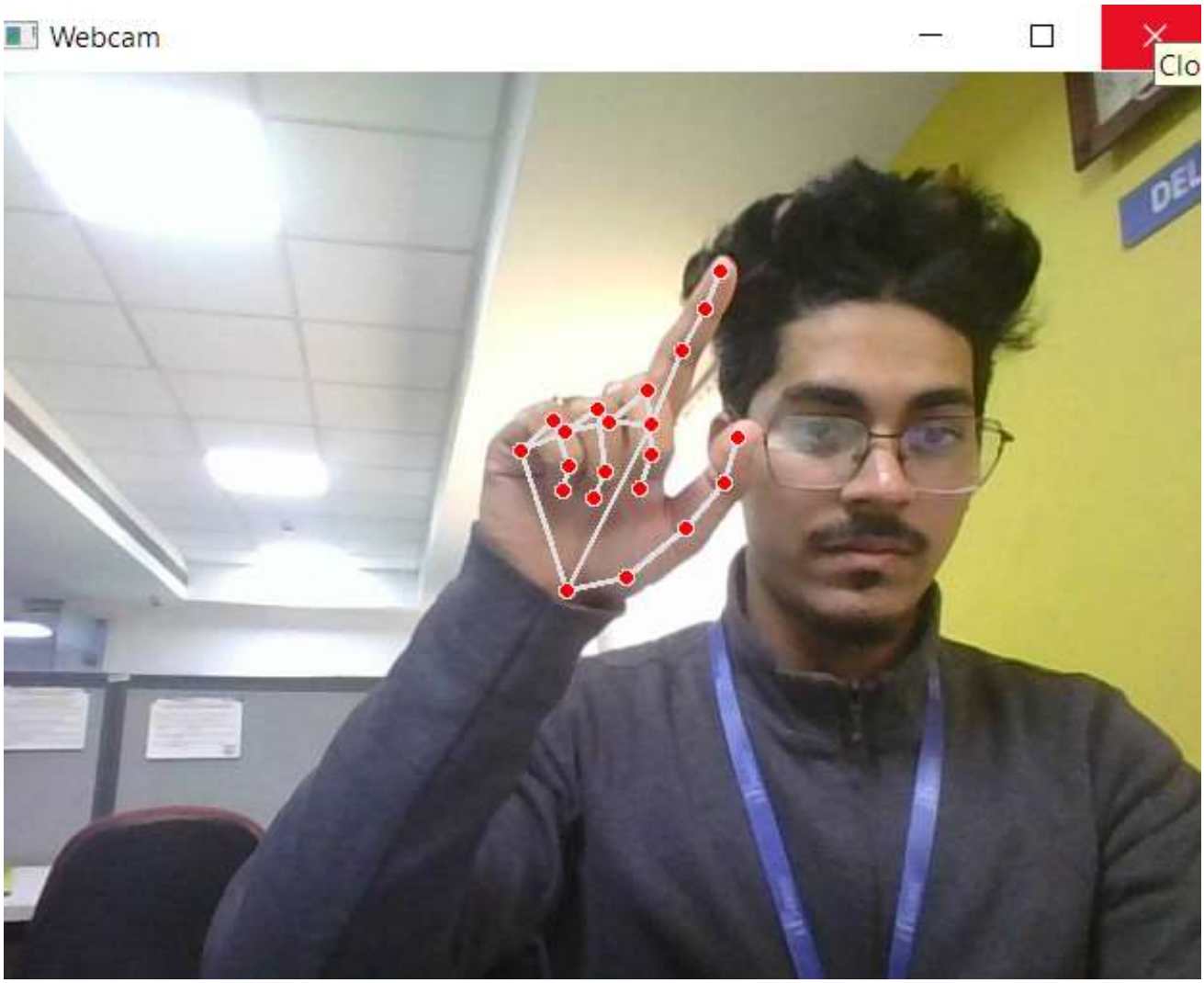


Fig 1.2 moving the mouse cursor

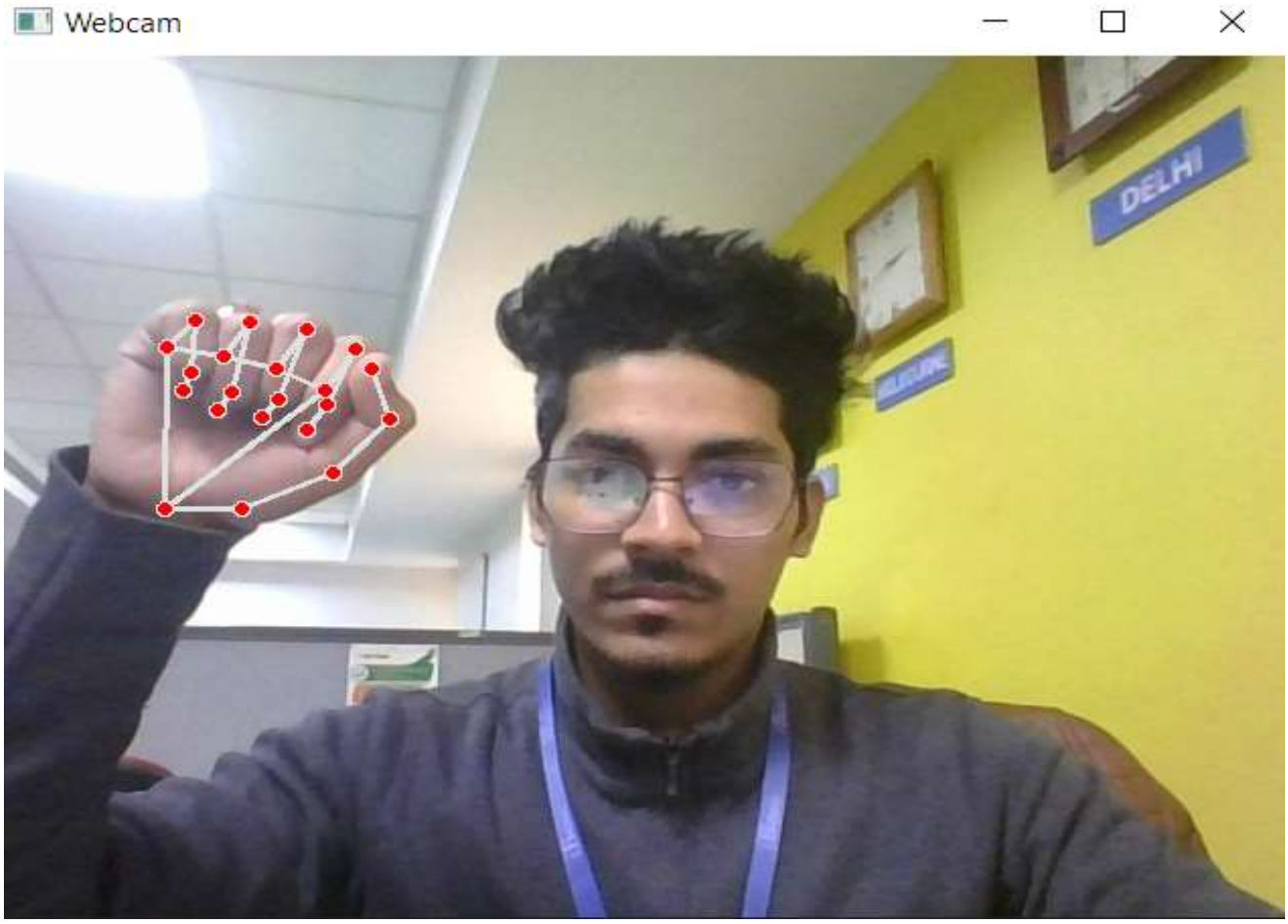


Fig 1.3 clicking of mouse

Result Analysis

- **Cursor Move**

With the help of OpenCV module we have a tendency to tend to were able to sight the inexperienced color connected to our fingers. As presently as a result of the inexperienced Color is detected two rectangles square measure designed around the finger denoting the inexperienced color. The OpenCV module uses web camera place in in our computer and camera consequently can sight the inexperienced color. The gap between the two Rectangles designed around the inexperienced color is calculated and thus the purpose of line is delineate by the dot and according to the movement of the dot the pointer moves around the screen. The pointer can hover around all the directions as we have a tendency to tend to move our fingers. Presently the mouse is completely controlled by hand gestures.

- **Click**

After activity the upper than operations. The gap between a pair of fingers has been calculated and additionally the middle is controlling the mouse movement presently as we've a bent to bring our a pair of fingers shut getting ready to the middle the gap Between the two fingers is shrunken and additionally the tetragon detected in each finger is removed and a circle is obligatory around every the fingers that performs the press operation. And as shortly as we've a bent to extend the gap the clicking operation stops and alone the indicator moves around. The clicking operation is performed by increasing and decreasing the distance between a pair of rectangles intentional around the fingers, presently if we've a bent to stay out a pair of fingers shut multiple clicking operation is performed.

Data Availability

The hand pursuit data used to support the findings of this study unit of measurement boxed within the article. $e study uses Google’s framework; thence, no new data unit of measurement needed to educate the model.

CHAPTER-

5.1 CONCLUSION

The main objective of the AI virtual mouse system is to control the mouse pointer functions by victimization the hand gestures instead of using a physical mouse. The projected system could also be achieved by using a photographic camera or a integral camera that detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions.

From the results of the model, we have a tendency to square measure able to return to a conclusion that the projected AI virtual mouse system has performed alright and includes an even bigger accuracy compared to the prevailing models and in addition the model overcomes most of the constraints of the prevailing systems. Since the projected model has larger accuracy, the AI virtual mouse could also be used for real-world applications, and also, it should be used to reduce the unfold of COVID-19, since the projected mouse system
Can be used with regards to exploitation hand gestures whereas not exploitation the traditional physical mouse. The model has some limitations like small decrease in accuracy in right click mouse perform and some difficulties in clicking and dragging to select out the text. Hence, we'll work next to beat these limitations by up the fingertip detection rule to supply extra correct results

5.2 FUTURE SCOPE

The planned AI virtual mouse has some limitations like little decrease in accuracy of the proper click mouse perform and collectively the model has some difficulties in execution clicking and dragging to choose the text. These unit variety of the constraints of the planned AI virtual mouse system, and these limitations square measure overcome in our future work. Moreover, the planned methodology is also developed to handle the keyboard functionalities along with the mouse functionalities regarding that's another future scope of Human-Computer Interaction (HCI).

VR and AR have most scope within the future. What we're witnessing these days is merely the start. Where they need increased play experiences, they need conjointly seeped into alternative industries in addition. Producing corporations, as an example, is mistreatment VR for simulating their processes till the stage of getting the end-product. This provides

Them an enormous wide-awake before the begin producing. Likewise, it's conjointly wont to train staff remotely, giving them a practical-like expertise of operating. In another case, AR/VR is employed in preparation that is proving to be exceptionally helpful for the aim. So, if you wish to figure with VR, be assured that the technology are relevant for the years to come back.

5.3 APPLICATIONS

The AI virtual mouse system is useful for many applications it are typically accustomed crop the realm for victimization the physical mouse, and it are typically used in things where we've got an inclination to cannot use the physical mouse. Ssystem eliminates the usage of devices, and it improves the human-computer interaction,The projected model options an even bigger accuracy of 99% which is far larger than that of different projected models for virtual mouse, and its many applications.

Application are following:-

1. In the COVID-19 state of affairs, it isn't safe to the devices by touching them as a results of it's about to result in a possible state of affairs of unfold of the virus by touching.
2. AI virtual mouse is accustomed play virtual reality and raised reality-based games whereas not the wireless or wired mouse devices.
3. Persons with problems in their hands can use this system to control the mouse functions at the computer.
4. At intervals the sphere of computing, the projected system like HCI is employed for dominant robots in coming up with and style, the projected system is employed for coming up with regarding for prototyping.
5. 2D and 3D painting or image editing is drawn by the AI virtual mouse system just by the hand gestures.

CHAPTER-6

6.1 REFERENCE

[1]Shshank Prasaad, Shubhr Sinha " Real-time Object Detection and Tracking in Unknown Environment" IEEE

[2]Artificial Intelligence concept [Online].
Available:https://en.wikipedia.org/wiki/Artificial_intelligene.

[3]Machine Learning concept [Online].
Available:https://en.wikipedia.org/wiki/Machine_learning

[4]Open CV python package [Online].
Available: https://opencv.org/

[5] PyautoGUI python package[Online].
Available:https://pyautogui.readthedocs.io/en/latest/

[6] L. $omas, "Virtual mouse using hand gesture," International Research Journal of Engineering and Technology (IRJET, vol. 5, no. 4, 201

[7] P. Nandhini, J. Jaya, and J. George, "Computer vision system " in Proceedings of the 2013 International Conference on Current Trends in Engineering and Technology (ICCTET), pp. 85–87, Coimbatore, India, July 2013.

[8] J. Jaya and K. $anushkodi, "Implementation of certain system for medical image diagnosis," European Journal of Scientifific Research, vol. 53, no. 4, pp. 561–567, 2011.

[9] P. Nandhini and J. Jaya, "Image segmentation for food quality evaluation using computer vision system,"

International Journal of Engineering Research and Applications, vol. 4, no.2, pp. 1–3, 2014.

[10] J. Jaya and K. $anushkodi, "Implementation of classification system for medical images," European Journal of Scientific Research, vol. 53, no. 4, pp. 561–569, 2011.