AI Virtual Mouse Controller using Hand Gestures

A Project Report is submitted in partial fulfillment of the requirement of the Mini Project of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

B.LAKSHMI VINAYASRI	(N170156)
G.VARALAKSHMI	(N170243)
G.S.L.PRASANNA	(N170205)
N.PHANISWATHI	(N170210)
V.MOUNIKA	(N170473)

Under the Esteem Guidance of

Mrs.N.Swathi

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies-Nuzivid

Nuzivid, Eluru district, AndhraPradesh - 521202

JUNE2022-AUGUST2022

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

RGUKT-NUZIVID,ELURU Dist - 521202

CERTIFICATE OF COMPLETION

This is to certify that the entitled, **“AI Virtual Mouse Controller using Hand Gestures**” is the bonafied work of **B.Lakshmi Vinayasri (N170156), G.Varalakshmi (N170243), G.S.L.Prasanna (N170205), N. Phani Swathi (N170210), V. Mounika (N170473)** carried out under my guidance and supervision for Mini Project of **Bachelor of Technology** in the department of Computer Science and Engineering under RGUKT IIIT Nuzvid. This work is done during the academic semester June 2022 - August 2022, under our guidance.

------------------------------------	---------------------------------------
Mrs.N.Swathi	**Mr.Sadu Chiranjeevi**
Assistant Professor,	Assistant Professor,
Department of CSE,	Head of the Department,
RGUKT Nuzvid.	Department of CSE,
	RGUKT Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

RGUKT-NUZIVID,ELURU Dist - 521202

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled, **“AI Virtual Mouse Controller using Hand Gestures**” is the bonafied work of **B.Lakshmi Vinayasri (N170156), G.Varalakshmi (N170243), G.S.L.Prasanna (N170205), N. Phani Swathi (N170210), V. Mounika (N170473)** and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in 3rd year of **Bachelor of Technology** for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn , as a recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Mrs.N.Swathi
Assistant Professor,
Department of CSE,
RGUKT Nuzvid.

Mr.Sadu Chiranjeevi
Assistant Professor,
Head of the Department,
Department of CSE,
RGUKT Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

RGUKT-NUZIVID,ELURU Dist - 521202

DECLARATION

We, **B.Lakshmi Vinayasri (N170156), G.Varalakshmi (N170243), G.S.L.Prasanna (N170205), N. Phani Swathi (N170210), V. Mounika (N170473)** hereby declare that the project report entitled **“AI Virtual Mouse Controller using Hand Gestures"**done by us under the guidance of **Mrs.N.Swathi** ,Assistant Professor is submitted for the partial fulfillment for the award of MiniProject in Bachelor of Technology in Computer Science Engineering during the academic semester June 2022-September2022 at RGUKT-Nuzivid.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references.The results embodied in this project report have not been submitted to any other university or institute for the award of any degree.

Date:____________

Place:____________

B.Lakshmi Vinayasri	(N170156)
G. Varalakshmi	(N170243)
G.S.L.Prasanna	(N170205)
N.Phani Swathi	(N170210)
V.Mounika	(N170473)

ACKNOWLEDGEMENT

We would like to express our profound gratitude and deep regards to our guide **Mrs.N.Swathi** for her exemplary guidance, monitoring and constant encouragement to us throughout the B.Tech course. We shall always cherish the time spent with her during the course of this work due to the valuable knowledge gained.We are extremely grateful for the confidence bestowed in us and entrusting our project entitled “**AI MOUSE CONTROLLER USING HAND GESTURES**”. We express gratitude to **Mr. Sadu Chiranjeevi (HOD of CSE)** and other faculty members for being source of inspiration and constant encouragement which helped us in completing the project successfully.Finally, yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

CONTENTS

ABSTRACT

The mouse is one of the wonderful inventions of Human-Computer Interaction (HCI) technology. Currently,wireless mouse or a Bluetooth mouse still uses devices and is not free of devices completely since it uses a battery for power and a dongle to connect it to the PC. In the proposed AI virtual mouse system, this limitation can be overcome by employing webcam or a built-in camera for capturing of hand gestures and hand tip detection using computer vision. The algorithm used in the system makes use of the machine learning algorithm. Based on the hand gestures, the computer can be controlled virtually and can perform left click, right click, scrolling functions, and computer cursor function without the use of the physical mouse. The algorithm is based on deep learning for detecting the hands. Hence, the proposed system will avoid COVID-19 spread by eliminating the human intervention and dependency of devices to control the computer.

1. INTRODUCTION

With the developing technologies within the twenty-one century, the areas of virtual Reality devices that we are using in our daily lifestyle. This paper shows an AI virtual mouse system that take input of hand gestures and detection for fingertip movement to perform mouse operation in computer by using computer vision. The most important objective of the proposed system is to perform mouse operations like click, scroll and cursor movement by employing an in built camera or a web-camera within the computer rather than using a traditional mouse. Hand gesture and fingertip detection by using computer vision is employed as an individual's and computer interaction simply referred to as HCI.

The system makes use of the well-known python libraries like Media Pipe PyAutoGUI for the tracking of the finger traveling in the screen for performing operations like click, drag and scrolling functions.

1.1 Purpose

Encountering the inadequacies and the weaknesses of the traditional Mouse,making use of Virtual Reality for different Mouse Operations like Scrolling up and Down,Cursor Movement and Click. With less effort and without physical contact the system helps to take control over the Mouse.

1.2 Overview

We are developing such application which is combination of AI and Computer Vision. The AI virtual Mouse system that take input of Hand Gestures and Fingertip movement to perform mouse operations like Click,Scroll and Cursor Movement by employing an in-built camera or a traditional mouse.Hand gestures and fingertip detection by using computer vision is employed as an individual's and Computer Interaction, simply referred to as HCI.

1.3 Scope

Virtual Mouse that will soon to be introduced to replace the physical computer mouse to promote convenience while still able to accurately interact and control the computer system.To do that, the software requires to be fast enough to capture and process every image. In order to successfully the user's gesture.

The scope of the project is:

- Real time application.
- User Friendly application.
- Removes the requirement of having a physical mouse.

2. Analysis of Existing and Proposed System

2.1 Problem Statement

Currently, due to COVID and other various diseases, people are afraid of physical touch. So, there is a huge spike in percent of people that are willing to avoid physical touch. To control mouse with any physical touch , there are some inadequacies. For example virtual mouse using colour detection there is a need of colouring or sticking some coloured cloth to fingers. So, to eliminate these inadequacies, our proposed system directly detect fingertips and hand gestures to perform click, scroll and cursor movement functions.Using fingertip detection methods for instant camera access and user-friendly user interface makes it more easily accessible.This system reduces the use of any physical mouse which saves time and also reduces effort.

2.2 Existing System:

■ **Virtual Mouse using Colour Detection:**

Controlling Mouse using a camera based on color detection technique.The user wears colored tapes to provide information to the system. The processing techniques involve an image subtraction algorithm to detect colors.Once the colors are detected the system performs various operations to track the cursor and perform control actions.

2.2.1 Disadvantages

- The presence of other colored objects in the background might cause the system to give an erroneous response.
- The system might run slower on certain computers with low computational capabilities because it involves a lot of complex calculations in very small amount of time.
- Cursor Movement is only in the selected area.

3.Proposed System

Considering the issues and disadvantages in the existing system, we come up with an idea of developing a virtual platform that does not need wear tapes to fingers.Using mediapipe,PyAutoGUI and Opencv our system can directly detect hand landmarks and using fingertips we perform different Mouse functions like click, scroll and cursor Movement.

3.1. Advantages

- High Accuracy Level.
- Cursor Movement in whole Screen.
- No need to wear tapes.
- Works in dark environment.
- Single Hand Detection(Right).
- Less effort and More ease.

4. Requirements

4.1 Hardware Requirements

Operating System	: 64-bit Windows
RAM	: 4GB(Approx)
DISK	: 1 TB hard disk

4.2 Software Requirements

Operating System	: 64-bit Windows
Text Editor	: Python Idle(3.10 version)

4.3 Technologies

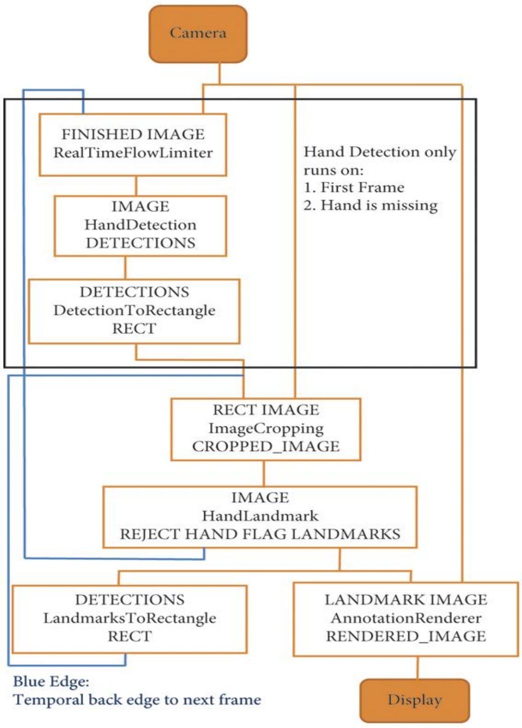
4.3.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

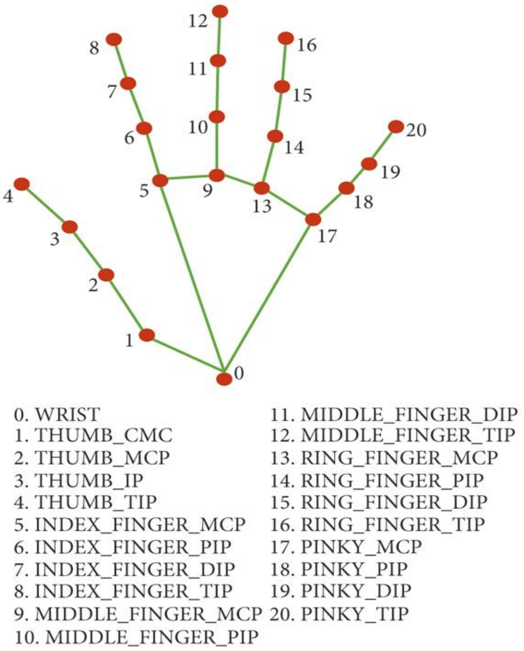
Python programming language is used for developing the AI virtual mouse system, and also, OpenCV which is the library for computer vision is used in the AI virtual mouse system. In the proposed AI virtual mouse system, the model makes use of the MediaPipe package for the tracking of the hands and for tracking of the tip of the hands, and also, Pynput, Autopy, and PyAutoGUI packages were used for moving around the window screen of the computer for performing functions such as left click, right click, and scrolling functions. The results of the proposed model showed very high accuracy level, and the proposed model can work very well in real-world application with the use of a CPU without the use of a GPU.

- **MediaPipe**

  MediaPipe is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google. The MediaPipe framework is useful for cross platform development since the framework is built using the time series data. The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos . The MediaPipe framework is used by the developer for building and analyzing the systems through graphs, and it also been used for developing the systems for the application purpose. The steps involved in the system that uses MediaPipe are carried out in the pipeline configuration. The pipeline created can run in various platforms allowing scalability in mobile and desktops. The MediaPipe framework is based on three fundamental parts; they are performance evaluation, framework for retrieving sensor data, and a collection of components which are called calculators , and they are reusable. A pipeline is a graph which consists of components called calculators, where each calculator is connected by streams in which the packets of data flow through. Developers are able to replace or define custom calculators anywhere in the graph creating their own application. The calculators and streams combined create a data-flow diagram; the graph is created with MediaPipe where each node is a calculator and the nodes are connected by streams .



Single-shot detector model is used for detecting and recognizing a hand or palm in real time. The single-shot detector model is used by the MediaPipe. First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the nonmaximum suppression works significantly better on small objects such as palms or fists. A model of hand landmark consists of locating 21 joint or knuckle co-ordinates in the hand region.



- **OpenCV**

OpenCV is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library. The OpenCV library is used in image and video processing and also analysis such as face detection and object detection.

- **PyAutoGUI**

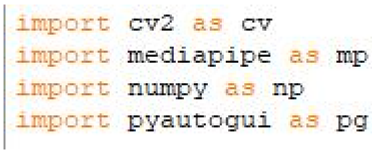
It is cross platform GUI automation module for python. That module keeps tracks of finger in this proposed system.Autopy track the fingertip and tell us which finger is up and which one is down . This process is happening by giving system an input in the form of 0 and 1.From this module the mediapipe module takes output and done the process and give the proper output. From this output opencv visualize everything and create the proper frame from image. It performs both cursor and keyboard functions.

- **NumPy**

NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices.It is an open source project and you can use it freely.NumPy stands for Numerical Python.

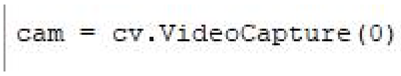
5. Methodologies

5.1 Importing the necessary library



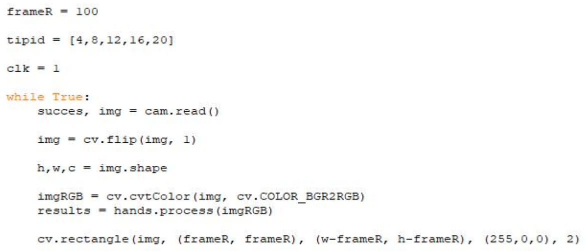
```
import cv2 as cv
import mediapipe as mp
import numpy as np
import pyautogui as pg
```

5.2 Initializing the capturing device



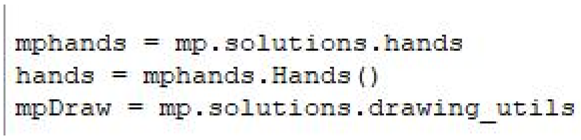
```
cam = cv.VideoCapture(0)
```

5.3 Capturing the frames and processing



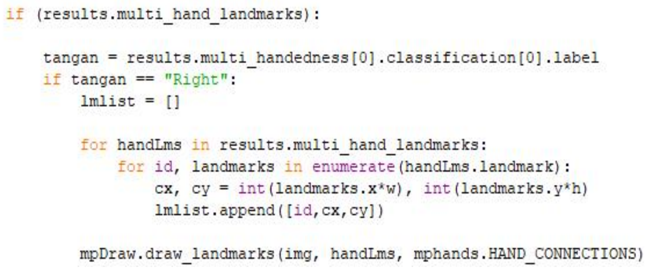
```
frameR = 100

tipid = [4,8,12,16,20]

clk = 1

while True:
    succes, img = cam.read()

    img = cv.flip(img, 1)

    h,w,c = img.shape

    imgRGB = cv.cvtColor(img, cv.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    cv.rectangle(img, (frameR, frameR), (w-frameR, h-frameR), (255,0,0), 2)
```

5.4 Initializing mediapipe



```
mphands = mp.solutions.hands
hands = mphands.Hands()
mpDraw = mp.solutions.drawing_utils
```

5.5 Define hand landmarks



```
if (results.multi_hand_landmarks):

    tangan = results.multi_handedness[0].classification[0].label
    if tangan == "Right":
        lmlist = []

        for handLms in results.multi_hand_landmarks:
            for id, landmarks in enumerate(handLms.landmark):
                cx, cy = int(landmarks.x*w), int(landmarks.y*h)
                lmlist.append([id,cx,cy])

        mpDraw.draw_landmarks(img, handLms, mphands.HAND_CONNECTIONS)
```

6. Main Functionalities

6.1 Cursor Moving

When only the index finger is up and all the fingers down the system allows cursor to move on the entire screen using coordinates of index fingertip.

6.2 Scroll Up

When both the index and middle fingers are up and all other fingers closed the system allows cursor to scroll up if the distance between fingertips of indexfinger and middlefinger is greater than 50.

6.3 Scroll Down

When both the index and middle fingers are up and all other fingers closed the system allows cursor to scroll up if the distance between the fingertips of indexfinger and middlefinger is less than 50.

6.4 Left Click

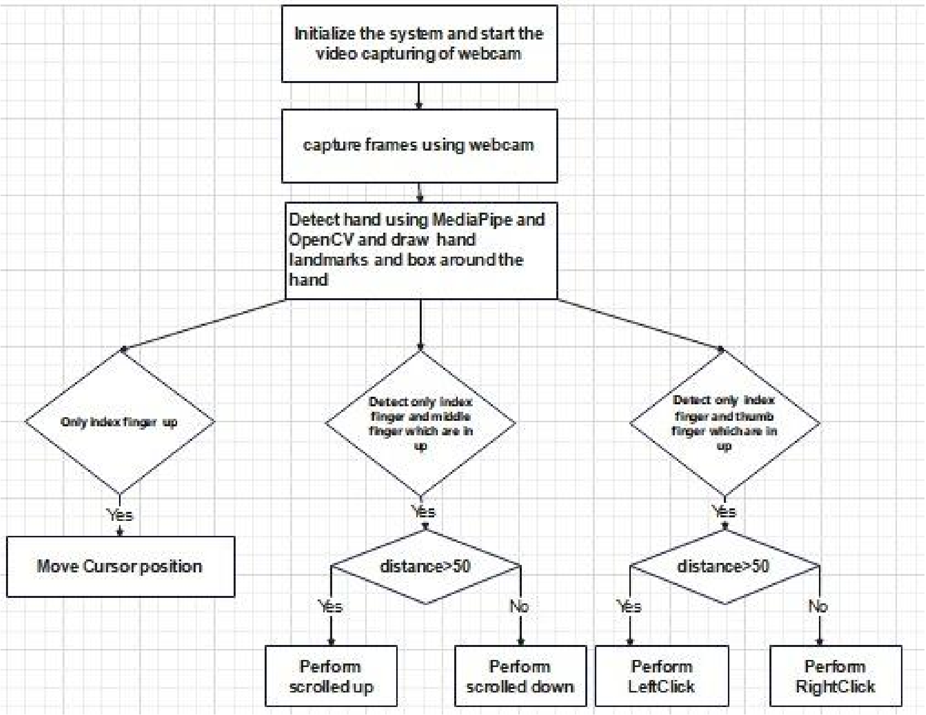
When both the index finger and thumb are up and all other fingers closed the system allows cursor to perform left click if the distance between the fingertips of indexfinger and thumb is greater than 50.

6.5 Right Click

When both the index finger and thumb are up and all other fingers closed the system allows cursor to perform left click if the distance between the fingertips of indexfinger and thumb is greater than 50.

7. Architecture

7.1 Flow Chart


Initialize the system and start the video capturing of webcam
capture frames using webcam
Detect hand using MediaPipe and OpenCV and draw hand landmarks and box around the hand
Only index finger up
Detect only index finger and middle finger which are in up
Detect only index finger and thumb finger which are in up
Yes
Yes
Yes
Move Cursor position
distance>50
distance>50
Yes
No
Yes
No
Perform scrolled up
Perform scrolled down
Perform LeftClick
Perform RightClick

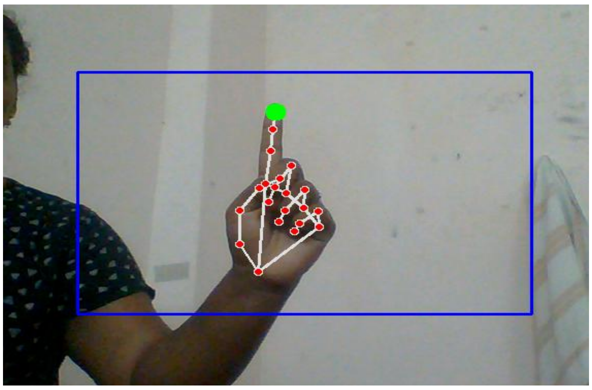
8. Implementation

```
import cv2 as cv
import mediapipe as mp
import numpy as np
import pyautogui as pg

cam = cv.VideoCapture(0)

mphands = mp.solutions.hands
hands = mphands.Hands()
mpDraw = mp.solutions.drawing_utils

screenWidth, screenHeight = pg.size()
print(screenWidth, screenHeight)

frameR = 100

tipid = [4,8,12,16,20]

clk = 1

while True:
   succes, img = cam.read()

   img = cv.flip(img, 1)

   h,w,c = img.shape

   imgRGB = cv.cvtColor(img, cv.COLOR_BGR2RGB)
   results = hands.process(imgRGB)

   cv.rectangle(img, (frameR, frameR), (w-frameR, h-frameR), (255,0,0), 2)

   if (results.multi_hand_landmarks):

      tangan = results.multi_handedness[0].classification[0].label
      if tangan == "Right":
         lmlist = []

         for handLms in results.multi_hand_landmarks:
            for id, landmarks in enumerate(handLms.landmark):
               cx, cy = int(landmarks.x*w), int(landmarks.y*h)
               lmlist.append([id,cx,cy])

         mpDraw.draw_landmarks(img, handLms, mphands.HAND_CONNECTIONS)
         fingers = []
```

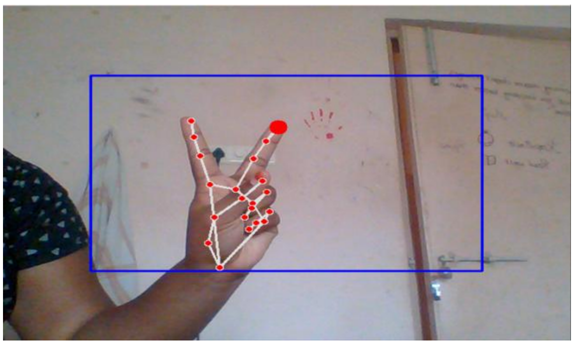
```
            if lmlist[tipid[0]][1] < lmlist[tipid[0]-2][1]:
                fingers.append(1)
            else:
                fingers.append(0)

            for id in range(1,5):
                if lmlist[tipid[id]][2] < lmlist[tipid[id]-3][2]:
                    fingers.append(1)
                else:
                    fingers.append(0)

            if fingers == [0,1,0,0,0]:
                cv.circle(img, (lmlist[8][1], lmlist[8][2]), 10, (0,255,0), cv.FILLED)

                X = np.interp(lmlist[8][1], (frameR, w-frameR), (0, screenWidth))
                Y = np.interp(lmlist[8][2], (frameR, h-frameR), (0, screenHeight))
                pg.moveTo(X, Y, duration = 0.3)
                print("cursor moved")
            if fingers == [0,1,1,0,0]:
                cv.circle(img, (lmlist[12][1], lmlist[12][2]), 10, (0,0,255), cv.FILLED)
                length = abs(lmlist[8][1] - lmlist[12][1])
                if length > 50:
                    pg.scroll(100)
                    print("scrolled up")
                else:
                    pg.scroll(-100)
                    print("scrolled down")
            if fingers==[1,1,0,0,0]:
                cv.circle(img,(lmlist[8][1],lmlist[8][2]),10,(0,0,255),cv.FILLED)
                X = np.interp(lmlist[8][1], (frameR, w-frameR), (0, screenWidth))
                Y = np.interp(lmlist[8][2], (frameR, h-frameR), (0, screenHeight))

                breadth = abs(lmlist[4][1] - lmlist[8][1])
                if breadth>50:
                    pg.click(X,Y,button="left")
                    print("Left Click")
                else:
                    pg.click(X,Y,button="right")
                    print("Right Click")
    cv.imshow("webcam", img)

    if cv.waitKey(20) & 0xFF==ord('q'):
        break

cv.destroyAllWindows()
```

9. Experimental Results

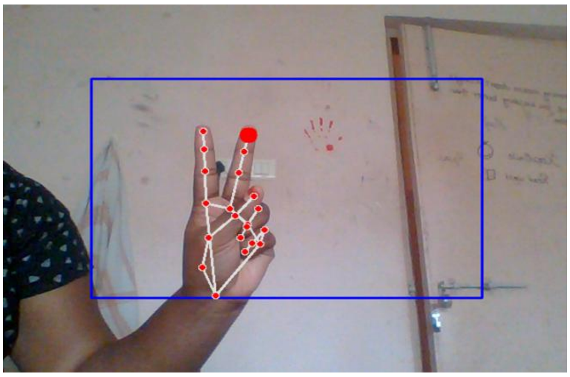
9.1 Cursor Movement



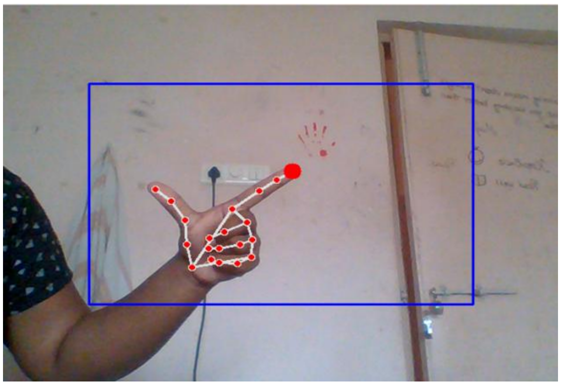
9.2 Scroll Up



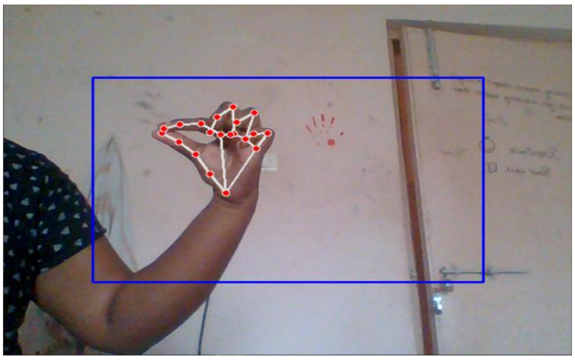
9.3 Scroll Down



9.4 Left Click



9.5 Right Click



10. Conclusion and Future Scope

The System presented AI Virtual Mouse method using OpenCV, PyAutoGUI and Mediapipe by the help of fingertip movement interacted with the computer infront of camera without using any physical device. The approach demonstrated high accuracy and highly accurate gesture eliminates in practical applications.The proposed method overcomes the limitations of already existing virtual mouse systems with advantage of working well in dark and changing light backgrounds.The experimental results shows that the approaching system can perform very well in real-time applications.

11. References

“A Vision Base Application for Virtyal Mouse Interface Using Hand Gesture” IJSRT21NOV703.

https://www.hindawi.com/journals/jhe/2021/8133076/

https://google.github.io/mediapipe/solutions/hands.html