DEEP LEARNING-BASED REAL-TIME AI VIRTUAL MOUSE SYSTEM USING COMPUTER VISION TO AVOID COVID-19 SPREAD

SEMINAR REPORT

SUBMITTED BY

JOMIN K MATHEW

TKM20MCA2021

TO

THE APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIRMENTS FOR THE AWARD OF DEGREE

IN

MASTER OF COMPUTER APPLICATION

TKM COLLEGE OF ENGINEERING KOLLAM

KERALA

JULY 2022

DEPARTMENT OF MCA

TKM COLLEGE OF ENGINEERING KOLLAM

2020-22

CERTIFICATE

This is to certify that the report entitled **DEEP LEARNING-BASED REAL-TIME AI VIRTUAL MOUSE SYSTEM USING COMPUTER VISION TO AVOID COVID-19 SPREAD** submitted by **JOMIN K MATHEW (TKM20MCA-2021)**, to the APJ Abdul Kalam Technological University in partial fulfillment of the M.C.A degree in Master of Computer Application is a bonafide record of the seminar work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Dr. FOUSIA M SHAMSUDEEN
(Seminar Coordinator)
Professor and Head
Department of MCA
TKM College of Engineering
Kollam

DECLARATION

I **JOMIN K MATHEW** hereby declare that the seminar report **DEEP LEARNING-BASED REAL-TIME AI VIRTUAL MOUSE SYSTEM USING COMPUTER VISION TO AVOID COVID-19 SPREAD**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Application of the **APJ Abdul Kalam Technological University, Kerala** is a bonafide work done by me under supervision **of Prof. FOUSIA M SHAMSUDEEN.**

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam **JOMIN K MATHEW**

22-07-2022

ACKNOWLEDGEMENT

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Prof. Dr. FOUSIA M SHAMSUDEEN**, Head of Department, **MCA, TKM COLLEGE OF ENGINEERING** Kollam for providing me with all the necessary facilities and support.

I profusely thank all other faculty members in the department and all other members of department of **MCA, TKM COLLEGE OF ENGINEERING** for the guidance and mentorship throughout the course.

I owe thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this seminar.

JOMIN K MATHEW

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.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

With the development technologies in the areas of augmented reality and devices that we use in our daily life, these devices are becoming compact in the form of Bluetooth or wireless technologies. This paper proposes an AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision. The main objective of the proposed system is to perform computer mouse cursor functions and scroll function using a web camera or a built-in camera in the computer instead of using a traditional mouse device. Hand gesture and hand tip detection by using computer vision is used as a HCI [1] with the computer. With the use of the AI virtual mouse system, we can track the fingertip of the hand gesture by using a built-in camera or web camera and perform the mouse cursor operations and scrolling function and also move the cursor with it. While using a wireless or a Bluetooth mouse, some devices such as the mouse, the dongle to connect to the PC, and also, a battery to power the mouse to operate are used, but in this paper, the user uses his/her built-in camera or a webcam and uses his/her hand gestures to control the computer mouse operations. In the proposed system, the web camera captures and then processes the frames that have been captured and then recognizes the various hand gestures and hand tip gestures and then performs the particular mouse function. 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.

1.1. Problem Description and Overview. The proposed AI virtual mouse system can be used to overcome problems in the real world such as situations where there is no space to use a physical mouse and also for the persons who have problems in their hands and are not able to control a physical mouse. Also, amidst of 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 overcome these problems since hand gesture and hand Tip detection is used to control the PC mouse functions by using a webcam or a built-in camera.

1.2. Objective. 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.

CHAPTER 2

LITERATURE SURVEY

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. An electronically instrumented glove which provides information about finger and hand position. A total of 22 gestures in three classes were investigated. The first class contained gestures which only involved finger flexure. The second class contained gestures which required both finger flexure and hand orientation. The third class of gestures required finger motion in addition to flexure and orientation. Only four sensors were necessary to positively identify specific gestures from groups of up to 15 gestures. The results show the specific number of sensors required to positively identify a gesture from a group. This depends on the number of gestures in a group, as well as the class of gestures.

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. A face based adaptive skin color model and a motion history image based hand moving direction detection method were proposed. There are four dynamic hand gestures hand moving up, moving down, moving left, and moving right and two static hand gestures fist and waving hand defined in this paper. These hand gestures are natural and simple. Harr-like features were designed to detect the four directional dynamic hand gestures. Static hand gestures were extracted by the face based adaptive skin color model and detected by checking a face based ROI.

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. The A new technique has been proposed to increase the adaptability of the system. We have developed a system to control the mouse cursor and implement its function using a real time camera. The goal of this project is to create a system that will recognize the hand gestures and control the computer/laptop according to those gestures. The project will also benefit the mobile systems where using pointing devices like mouse is difficult. Implementation of all the mouse tasks such as left and right clicking, double clicking and starting the applications using the gestures like notepad, paint, command prompt etc. Before actual implementing gesture comparison algorithms, skin pixel detection and hand segmentation from stored frames need to be done. The project is also developed in such a way that the user, new to the system will just have to install the set up and not run the whole project.

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. A new technique has been proposed to increase the adaptability and response time of the system. We have developed a system to control the mouse cursor and implement its function using a real time camera. Implementation of all the mouse tasks such as left and right clicking, double clicking and scrolling up & down, starting the applications using the gestures like notepad, paint, command prompt etc. This system is developed in such a way that the user, new to the system will just have to install the set up and not run the whole project. In this system, an object tracking based virtual mouse application has been developed and implemented using a webcam. The proposed system has been implemented in MATLAB environment using MATLAB Image Processing Toolbox, OpenCV library.

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. Gesture recognition gives the best interaction between human and machine. Gesture recognition is also important for developing alternative human computer interaction modalities. It enables human to interface with machine in a more natural way. Gesture recognition can be used for many applications like sign language recognition for deaf and dumb people, robot control etc. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Digital Canvas is an extension of our system which is gaining popularity among artists, by which the artist could create 2D or 3D images using the Virtual Mouse technology using the hand as brush and a Virtual Reality kit or a monitor as display set. This technology can be used to help patients who don’t have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person's motions are tracked and interpreted as commands. The major extension to this work can be done to make system able to work at much complex background and compatible with different light conditions. It can be made as an effective user interface and which can include all mouse functionalities. And also, it would be ideal to research into advanced mathematical materials for image processing and investigate on different hardware solutions that would result in more accurate hand detections. Not only did this project show the different gesture operations that could be done by the users but it also demonstrated the potential in simplifying user interactions with personal computers and hardware systems.

CHAPTER 3
METHODOLOGY

3.1 PROPOSED SYSTEM

The proposed AI virtual mouse has performed very well in terms of accuracy when compared to the other virtual mouse models. The novelty of the proposed model is that it can perform most of the mouse functions such as left click, right click, scroll up, scroll down, and mouse cursor movement using fingertip detection, and also, the model is helpful in controlling the PC like a physical mouse but in the virtual mode.

3.2 TECHNOLOGIES USED

- Python programming language is used for developing the AI virtual mouse system.
- OpenCV which is the library for computer vision is used in the AI virtual mouse system.
- The MediaPipe Framework for tracking of the hands and for tracking the tip of the hands.
- PyAutoGUI package were used for moving around the window screen of the computer for performing functions such as left click, right click and scrolling functions.

3.2.1 Computer Vision: Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

OpenCV: OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Application of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below:

- Face recognition.
- Automated inspection and surveillance.
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds.
- Interactive art installations.
- Anamoly (defect) detection in the manufacturing process (the odd defective products).
- Street view image stitching.
- Video/image search and retrieval.
- Robot and driver-less car navigation and control.
- Object recognition.
- Medical image analysis.
- Movies – 3D structure from motion.
- TV Channels advertisement recognition.

OpenCV Functionality:

- Image/video I/O, processing, display (core, imgproc, highgui).
- Object/feature detection (objdetect, features2d, nonfree).
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab).
- Computational photography (photo, video, superres).
- Machine learning & clustering (ml, flann).
- CUDA acceleration (gpu).

Image-Processing: Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it. If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

3.2.2 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 calculator, and they are reusable. A pipeline is a graph which consists of components called calculators, where each calculators 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 (Figure 1) 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.

Possibilities with MediaPipe:

- Human Pose Detection and Tracking High-fidelity human body pose tracking, inferring a minimum of 25 2D upper-body landmarks from RGB video frames.
- Face Mesh 468 face landmarks in 3D with multi-face support.
- Hand Tracking 21 landmarks in 3D with multi-hand support, based on high-performance palm detection and hand landmark model
- Holistic Tracking Simultaneous and semantically consistent tracking of 33 pose, 21 per-hand, and 468 facial landmarks.
- Hair Segmentation Super realistic real-time hair recoloring.
- Object Detection and Tracking Detection and tracking of objects in the video in a single pipeline.

- Face Detection Ultra-lightweight face detector with 6 landmarks and multi-face support.
- Iris Tracking and Depth Estimation Accurate human iris tracking and metric depth estimation without specialized hardware. Tracks iris, pupil, and eye contour landmarks..
- 3D Object Detection Detection and 3D pose estimation of everyday objects like shoes and chairs.

MediaPipe Holistic: Mediapipe Holistic is one of the pipelines which contains optimized face, hands, and pose components which allows for holistic tracking, thus enabling the model to simultaneously detect hand and body poses along with face landmarks. one of the main usages of MediaPipe holistic is to detect face and hands and extract key points to pass on to a computer vision model.

Landmarks that can be detected using Mediapipe Holistic:

1. Pose Landmarks

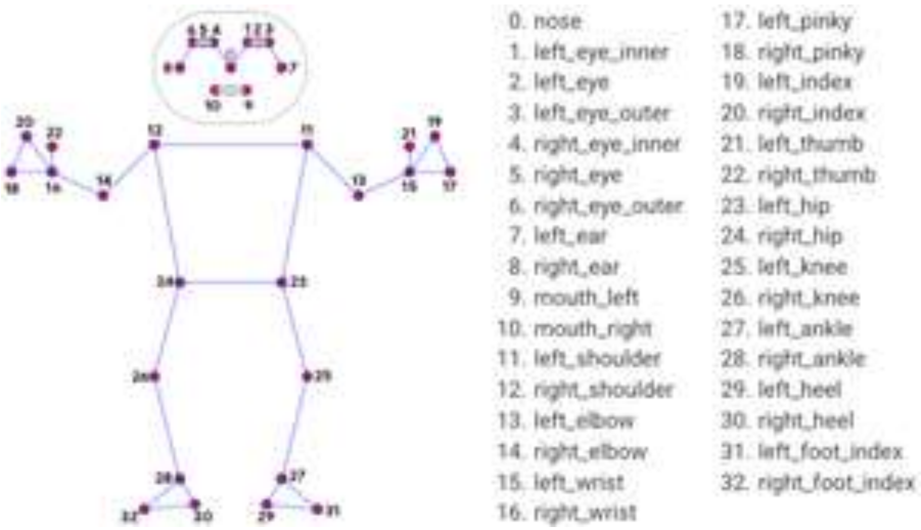


Fig 1: Pose Landmark

2. Hand Landmarks

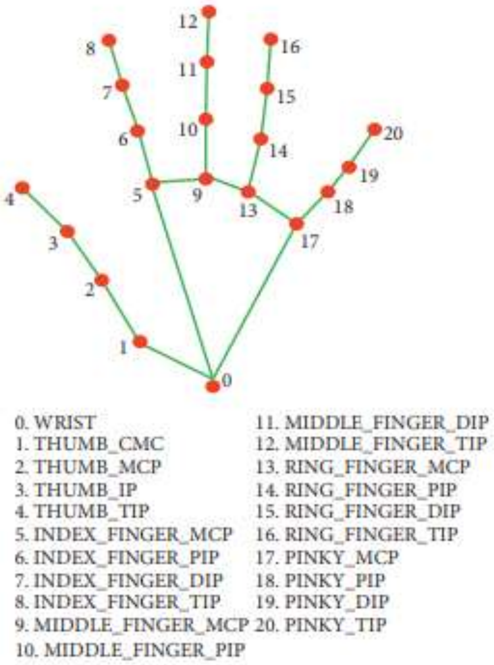


Fig 2: Hand Landmark.

3. Face Landmarks

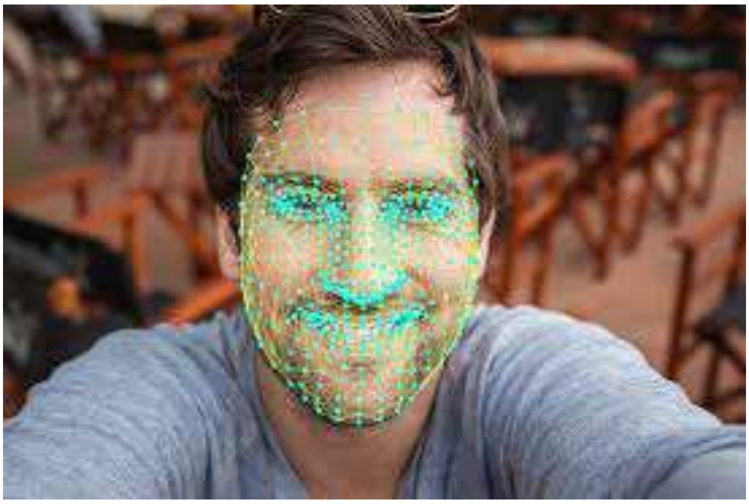


Fig 3: Face Landmark.

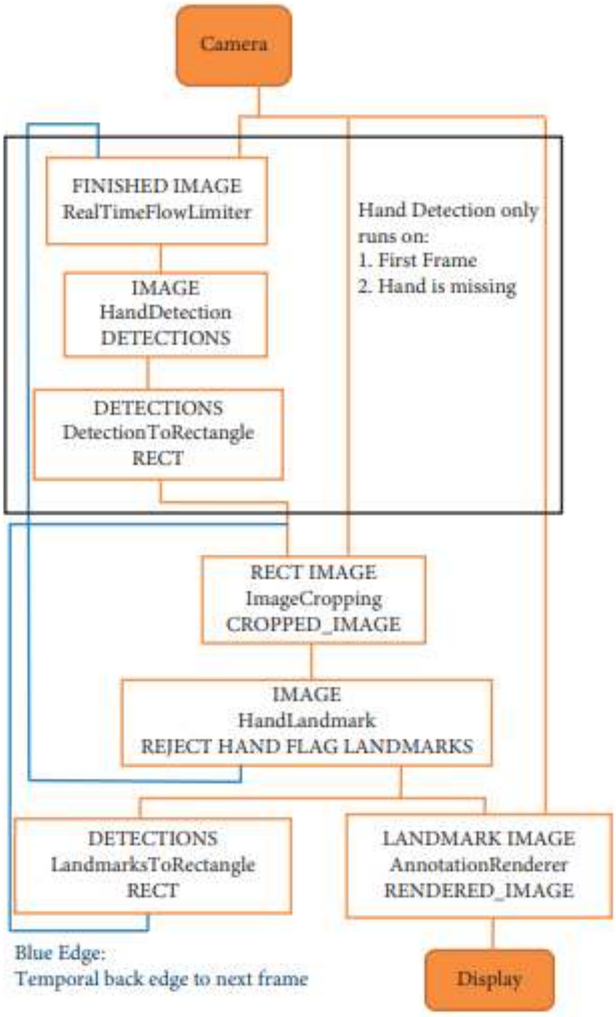


Fig 4: MediaPipe Hand Recognition graph.

3.2.3 PyAutoGUI: PyAutoGUI is essentially a Python package that works across Windows, MacOS X and Linux which provides the ability to simulate mouse cursor moves and clicks as well as keyboard button presses.

Controlling mouse movements using pyautogui module:

Python tracks and controls mouse using coordinate system of screen. Suppose the resolution of your screen is 1920X1080, then your screen’s coordinate system looks like:

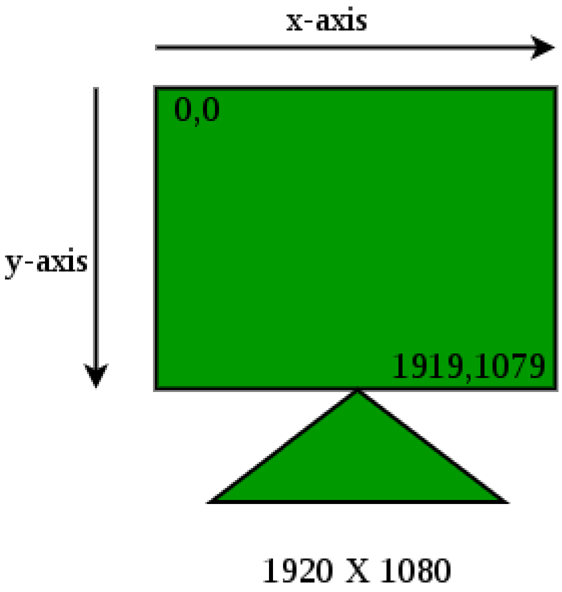


Fig 5: System of screen.

- **size():** This function is used to get Screen resolution.
- **moveTo():** use this function to move the mouse in pyautogui module.

  pyautogui.moveTo(100, 100, duration = 1)

  This code uses moveTo() function, which takes x and y coordinates, and an optional duration argument. This function moves your mouse pointer from it's current location to x, y coordinate, and takes time as specified by duration argument to do so. Save and run this python script to see your mouse pointer magically moving from its current location to coordinates (100, 100), taking 1 second in this process.

- **moveRel()** function: moves the mouse pointer relative to its previous position.

  pyautogui.moveRel(0, 50, duration = 1)

  This code will move mouse pointer at (0, 50) relative to its original position. For example, if mouse position before running the code was (1000, 1000), then this code will move the pointer to coordinates (1000, 1050) in duration 1 second.

- **position():** function to get current position of the mouse pointer.

  print(pyautogui.position())

  coordinates where your mouse was residing at the time of executing the program.

- **click():** Function used for clicking and dragging the mouse.

  ```
  pyautogui.click(100, 100)
  ```

  This code performs a typical mouse click at the location (100, 100). We have two functions associated with the drag operation of the mouse, dragTo and dragRel. They perform similar to moveTo and moveRel functions, except they hold the left mouse button while moving, thus initiating a drag.

- **scroll():** scroll function takes no. of pixels as an argument, and scrolls the screen up to a given number of pixels.

  ```
  pyautogui.scroll(200)
  ```

  This code scrolls the active screen up to 200 pixels.

- **typewrite():** You can automate typing of the string by using typewrite() function. just pass the string which you want to type as an argument of this function.

  ```
  pyautogui.click(100, 100)
  pyautogui.typewrite("hello Geeks !")
  ```

  Suppose a text field was present at coordinates 100, 100 on-screen, then this code will click the text field to make it active and type “hello Geeks!” in it.

- **Passing key names**: You can pass key names separately through typewrite() function.

  ```
  pyautogui.typewrite(["a", "left", "ctrlleft"])
  ```

  This code is the automatic equivalent of typing “a”, pressing the left arrow key, and pressing the left control key.

- **Pressing hotkey combinations**: Use hotkey() function to press the combination of keys like ctrl-c, ctrl-a, etc.

  ```
  pyautogui.hotkey("ctrlleft", "a")
  ```

  This code is the automatic equivalent of pressing left ctrl and “a” simultaneously. Thus in windows, this will result in the selection of all text present on the screen.

3.3 WORKING OF MODULES

- **The Camera Used in the AI Virtual Mouse System.**

  The AI virtual mouse system is based on the frames that have been captured by the webcam in a laptop or PC. By using the Python computer vision library OpenCV, the video capture object is created and the web camera will start capturing video. The web camera captures and passes the frames to the AI virtual system.

- **Capturing the Video and Processing**

  The AI virtual mouse system uses the webcam where each frame is captured till the termination of the program. The video frames are processed from BGR to RGB color space to find the hands in the video frame.

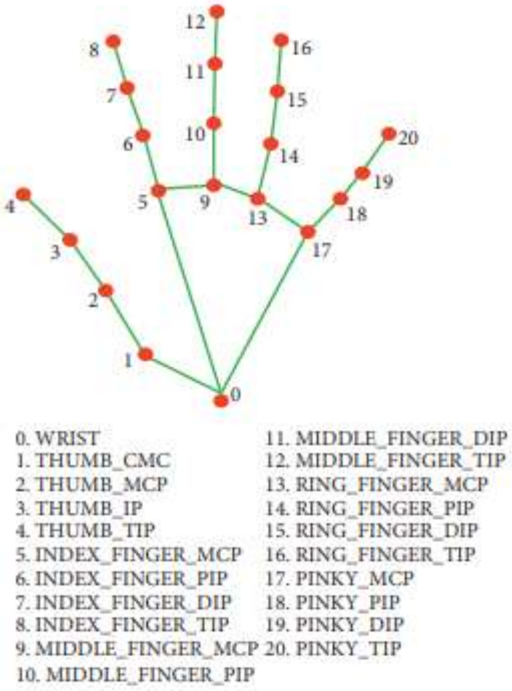


Fig 6: Co-ordinates or land mark in the hand.

- **Rectangular Region for Moving through the Window.**

  The AI virtual mouse system converts the coordinates of fingertip from the webcam screen to the computer window full screen for controlling the mouse.

- **Detecting Which Finger Is Up and Performing the Particular Mouse Function.**

  In this stage, we are detecting which finger is up using the tip Id of the respective finger that we found using the MediaPipe and the respective co-ordinates of the fingers that are up and according to that, the particular mouse function is performed.

- **For the Mouse Cursor Moving around the Computer Window.**

  If the index finger is up with tip Id = 1 or both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up, the mouse cursor is made to move around the window of the computer using the AutoPy package of Python.

- **For the Mouse to Perform Left Button Click**

  If both the index finger with tip Id = 1 and the thumb finger with tip Id= 0 are up and the distance between the two fingers is lesser than 30px, the computer is made to perform the left mouse button click using the pynput Python package.

- **For the Mouse to Perform Right Button Click.**

  If both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is lesser than 40 px, the computer is made to perform the right mouse button click using the pynput Python package.

- **For the Mouse to Perform Scroll up Function.**

  If both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is greater than 40 px and if the two fingers are moved up the page, the computer is made to perform the scroll up mouse function using the PyAutoGUI Python package.

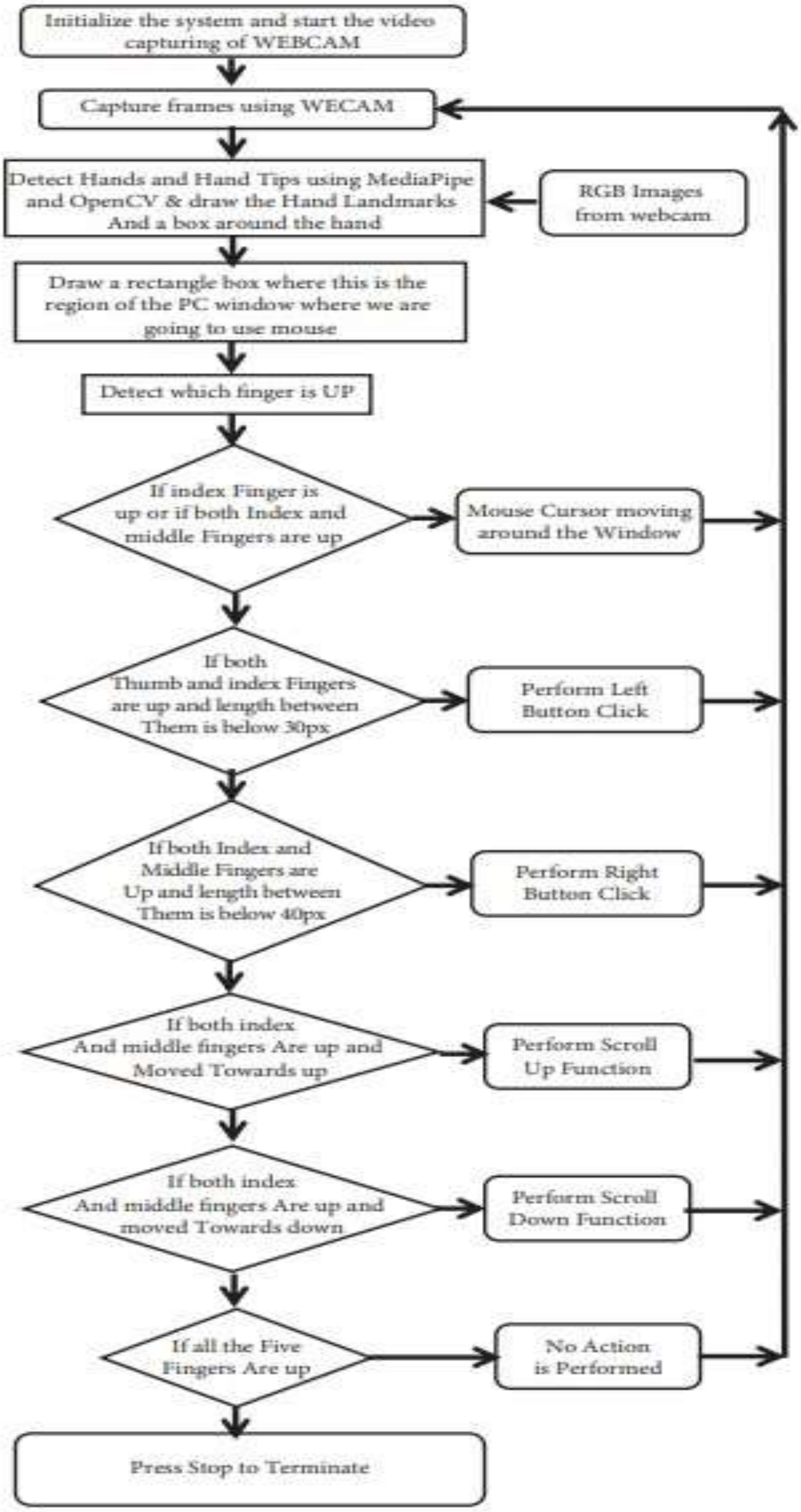
- **For the Mouse to Perform Scroll down Function.**

  If both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is greater than 40px and if the two fingers are moved down the page, the computer is made to perform the scroll down mouse function using the PyAutoGUI Python package.

- **For No Action to be Performed on the Screen.**

  If all the fingers are up with tip Id = 0, 1, 2, 3, and 4, the computer is made to not perform any mouse events in the screen.

FLOWCHART


Initialize the system and start the video capturing of WEBCAM
Capture frames using WECAM
Detect Hands and Hand Tips using MediaPipe and OpenCV & draw the Hand Landmarks And a box around the hand
RGB Images from webcam
Draw a rectangle box where this is the region of the PC window where we are going to use mouse
Detect which finger is UP
If index Finger is up or if both Index and middle Fingers are up
Mouse Cursor moving around the Window
If both Thumb and index Fingers are up and length between Them is below 30px
Perform Left Button Click
If both Index and Middle Fingers are Up and length between Them is below 40px
Perform Right Button Click
If both index And middle fingers Are up and Moved Towards up
Perform Scroll Up Function
If both index And middle fingers Are up and moved Towards down
Perform Scroll Down Function
If all the Five Fingers Are up
No Action is Performed
Press Stop to Terminate

CHAPTER 4
RESULT AND DISCUSSION

In the proposed AI virtual mouse system, the concept of advancing the human-computer interaction using computer vision is given. Cross comparison of the testing of the AI virtual mouse system is difficult because only limited numbers of datasets are available. The hand gestures and fingertip detection have been tested in various illumination conditions and also been tested with different distances from the webcam for tracking of the hand gesture and hand tip detection. This test has been made in different light conditions and at different distances from the screen, and each person tested the AI virtual mouse system 10 times in normal light conditions, 5 times in faint light conditions, 5 times in close distance from the webcam, and 5 times in long distance from the webcam.

It can be seen that the proposed AI virtual mouse system had achieved an accuracy of about 99%. From this 99% accuracy of the proposed AI virtual mouse system, we come to know that the system has performed well. The accuracy is low for “Right Click” as this is the hardest gesture for the computer to understand. The accuracy for right click is low because the gesture used for performing the particular mouse function is harder. Also, the accuracy is very good and high for all the other gestures. Compared to previous approaches for virtual mouse, our model worked very well with 99% accuracy.

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.

Major applications:

- The proposed model has a greater accuracy of 99% which is far greater than the that of other proposed models for virtual mouse, and it has many application

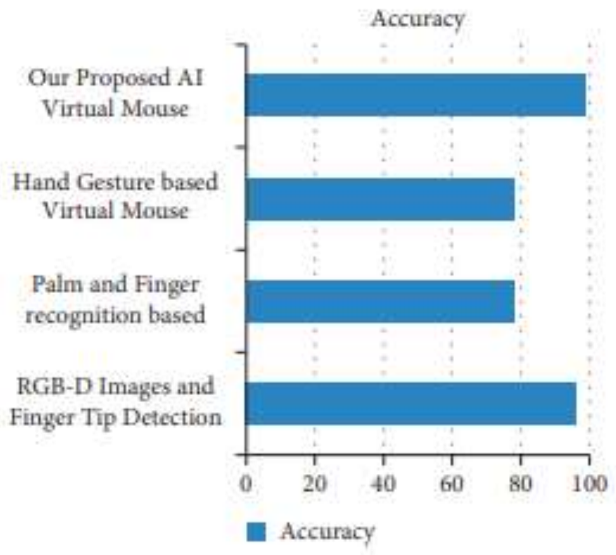


Fig 7: Graph for comparison between the models.

- 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

- In the field of robotics, the proposed system like HCI can be used for controlling robotsIn designing and architecture, the proposed system can be used for designing virtually for prototyping.

4.1 OUTPUTS

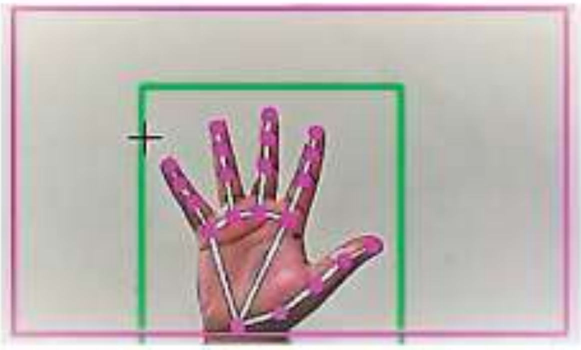


Fig 8: Rectangular box for the area of the computer screen where we can move the cursor.

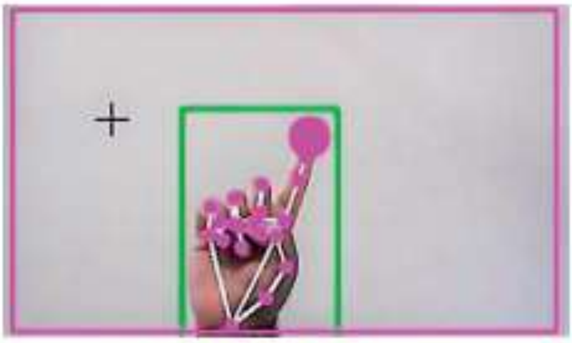


Fig 9: Mouse cursor moving around the computer window.

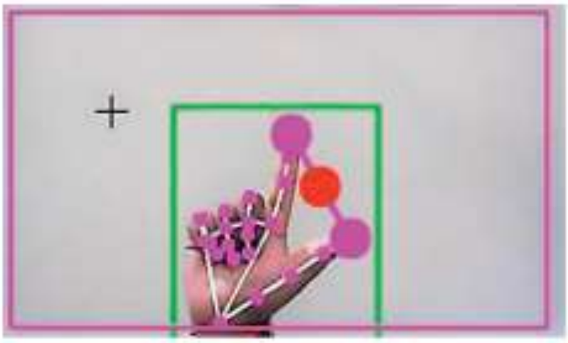


Fig 10: Gesture for the computer to perform left button click.

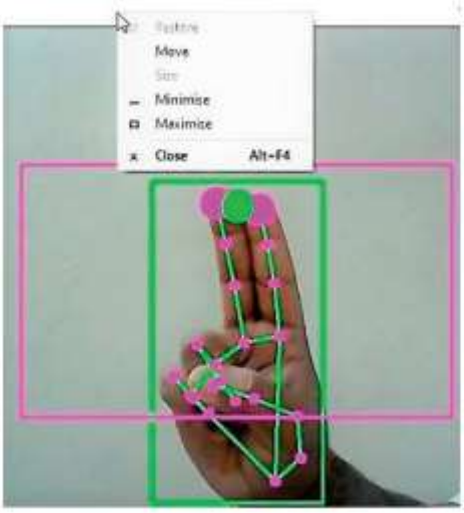


Fig 11: Gesture for the computer to perform right button click.

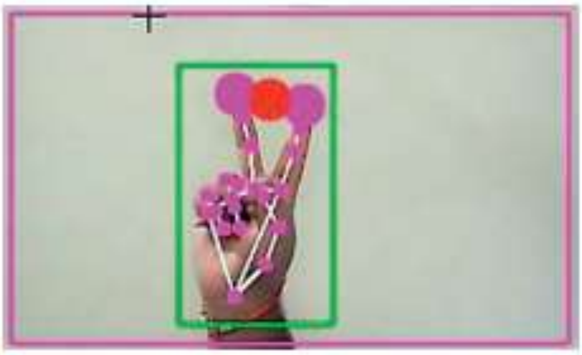


Fig 12: Gesture for the computer to perform scroll up & down function.

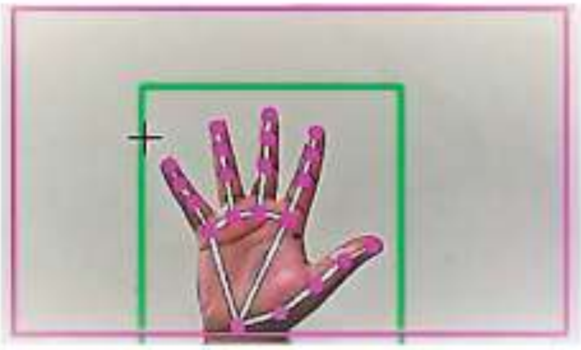


Fig 13: Gesture for the computer to perform no action

CHAPTER 5
CONCLUSION

The main objective of the AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions. From the results of the model, we can come to a conclusion that the proposed AI virtual mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, the AI virtual mouse can be used for real-world applications, and also, it can be used to reduce the spread of COVID-19, since the proposed mouse system can be used virtually using hand gestures without using the traditional physical mouse. The model has some limitations such as small decrease in accuracy in right click mouse function and some difficulties in clicking and dragging to select the text. Hence, we will work next to overcome these limitations by improving the fingertip detection algorithm to produce more accurate results.

5.1 FUTURE ENHANCEMENT

The proposed AI virtual mouse has some limitations such as small decrease in accuracy of the right click mouse function and also the model has some difficulties in executing clicking and dragging to select the text. These are some of the limitations of the proposed AI virtual mouse system, and these limitations will be overcome in our future work. Furthermore, the proposed method can be developed to handle the keyboard functionalities along with the mouse functionalities virtually which is another future scope of Human-Computer Interaction (HCI).

REFRENCES

[1] J. Katona, “A review of human–computer interaction and virtual reality research fields in cognitive Info Communications,” Applied Sciences, vol. 11, no. 6, p. 2646, 2021.

[2] D. L. Quam, “Gesture recognition with a DataGlove,” IEEE Conference on Aerospace and Electronics, vol. 2, pp. 755–760, 1990.

[3] 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, July 2010.

[4] S. U. Dudhane, “Cursor control system using hand gesture recognition,” IJARCCE, vol. 2, no. 5, 2013.

[5] K. P. Vinay, “Cursor control using hand gestures,” International Journal of Critical Accounting, vol. 0975–8887, 2016.

[6] L. Thomas, “Virtual mouse using hand gesture,” International Research Journal of Engineering and Technology (IRJET, vol. 5, no. 4, 2018.

[7] P. Nandhini, J. Jaya, and J. George, “Computer vision system for food quality evaluation—a review,” 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. Thanushkodi, “Implementation of certain system for medical image diagnosis,” European Journal of Scientific 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. Thanushkodi, “Implementation of classification system for medical images,” European Journal of Scientific Research, vol. 53, no. 4, pp. 561–569, 2011.