International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 201-209
Published Online October 2022 in IJEAST (http://www.ijeast.com)

AI AND ML BASED GESTURE CONTROLLED VIRTUAL MOUSEACTIONS SYSTEM - A NOVEL APPROACH

Pruthvi Kumar P
Dept. of Computer Science and Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India – 577204

Dr. Nirmala Shivanand
Professor, Dept. of Computer Science and Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India - 577204

***Abstract*— In the current system, a wired/wireless mouse device is used. This needs a battery, a USB cable, and a dongle to connect mouse to a PC. In this work, AI and ML based gesture controlled virtual mouse actions system is proposed and developed. This is a novel approach. In this approach, human and PC interaction is achieved using hand gestures. To detect hand gestures, to control the position of the virtual mouse with the fingertip rather than a physical mouse device, a media pipe model is used. This proposed system is implemented in Python and it will reduce the likelihood of COVID-19 spreading by eliminating human intervention and the dependency on devices to control the PC directly.**

***Keywords*— AIML, Python, Virtual Mouse, Computer Vision, Media pipe**

I. INTRODUCTION

Nowadays, numerous mouse options are available in the market today, such as Wireless Mouse, Wired Mouse, Optical Mouse, Laser Mouse, Gaming Mouse, Stylus Mouse, Presentation Mouse, and Vertical Mouse. Regardless, users don't know how much the physical mouse accuracy improves. The mouse will always have some limitations since it is a hardware input device integrated with the computer. Like any other physical object, the mouse will have a limited lifetime within which it is functional, and after its lifetime, it is needed to change the physical mouse. As technology evolves, everything becomes virtualized.

The AI [12] and ML based gesture controlled virtual mouse actions system utilizes human hand gesture using hand and fingers tips detection to perform mouse operations on the PC window screen using computer vision techniques. The main goal is to perform mouse operations in real-time. These functions include cursor movement, left/right-click, scroll up/down, drag/drop objects, and volume/brightness actions performed using the system's built-in camera rather than traditional mouse devices. Also, the fingertip of a hand gesture could be monitored using webcam.

Gesture recognition virtual mouse is a detection technique that employs the mathematical interpretation to detect, track and recognize human hand gestures as given instructions. These may be in any form, such as a hand picture or pixel image. that requires more minor computing difficulties for the hand recognition to work using a webcam while using a wireless or Bluetooth mouse and the dongle connected to the PC. It needs more significant computational problems, and a battery is required to power the mouse. In this proposed system, the webcam captures frames and then processes the captured video frames, recognizing various hand gestures before performing the specific mouse function.

The proposed approach works on an inexpensive central processing unit (CPU) processor without graphics processing unit (GPU). The system webcam captures hand tips and fingertips images at a rate of 30fps (frames per second), and it provides long-distance simultaneous fingertip tracking to control the mouse cursor with very high precision. The model can perform exceptionally well in real-world and real-time applications using just a CPU and no GPU.

II. LITERATURE REVIEW

Many prior experiments on the virtual mouse used hand gesture detection by wearing a glove in hand and colored tips in the hands for gesture recognition. Further, these methods were not much accurate in mouse operations.

Paper [1] Based on the hand's shape coordinates, the Kcosine algorithm method is utilized to locate the fingertip position. The system monitors fingers in real-time, 30 frames per second on a computer with single CPU and

Kinect V2. This fingertip-gesture-based interface enables people to interact with computers by hand. The hand area of interest and palm centre are initially retrieved by utilising in-depthskeleton-joint information images from a Microsoft Kinect Sensor. The shapes of the hand and fingers are then retrieved and characterized using a border-tracing technique. Finally, the fingertip position is mapped to RGB images to operate the mouse cursor on a virtual screen. However, the detection depth of the Kcosine algorithm is restricted and unsuitable for outdoor applications.

Paper [2] Developed Hand gesture tracking generates images from camera frames by detecting the shape and forming aconvex hull around it. This research employs two approaches for monitoring fingers: one uses coloured caps and the other employs hand motion recognition. This method has been subjected to extensive testing in real-world circumstances. To execute mouse activities, hand features are extracted using the area ratio of shapes as well as the area of features per square centimeter.

Paper [3] Designed hand gestures is utilized to create an optical mouse and keyboard. The to computer's camera will interpret the images of various movements done by a person's hand. The mouse will move according to the movement of the gestures, even performing right and left clicks using the defect calculations of the convex hull algorithm, which generates the mouse and keyboard functions with the defects. It can detect and identify hand gestures in order to control mouse and keyboard operations and generate a real-world user interface. This project has the potential to be extremely large in medical science where computing is necessary but limitations are However, if some external noise or fault is found in the working region of the camera, the Convex Hull method may create problems.

Paper [4] Based on the object tracking system to control mouse cursor actions system that uses hand gestures captured frames from a webcam using an HSV color detection technique. This system enables the user to control the computer cursor with their hand bearing colour caps or tapes that the PC webcam tracks, as well as perform mouse operations such as left click, right click, and double click using various hand gestures. The author implemented this project using the Python programming language. The drawback is due to noise in the captured live webcam frames and it cannot work in low light conditions.

Paper [5] Designed a virtual mouse system based on HCI that employs computer vision and hand motions. The system uses awebcam or built-in camera to take frames and analyses them using color segmentation and detection techniques in order to make the frames trackable. After that, it recognizes various user gestures and executes the mouse operations. So, a user can primarily do left clicks, right clicks, and double clicks, as well as scroll up and down operations with their hand in various gestures. The limitation is that noise is present in the gathered live webcam frames, it cannot function in low light.

Paper [6] Based on the openCV, wx, and numpy libraries were used to create the virtual mouse model. The model will launch a webcam to collect the user's fingertip colour caps frames. The mouse movement is dependent on the highlighted colour that the user specifies for mouse movement. This project may be beneficial for presentations as well as eliminate the weight of additional hardware devices. However, it does not operate inlow-light conditions since noise appears in collected frames.

Paper [7], Developed work focuses on improving human computer interaction systems [11] using hand gestures in 3-D space by employing two USB cameras that are orthogonal to each other to acquire top and side views of different hand movements in position. The hand's pointing motion is estimated and mapped to the coordinate system of the screen. It also employed additional hand motions to complete the actions of virtual mouse movements and hand pointing to point at the screen, as well as other operations like folder/object selection. However, the inventor of this solution uses two cameras to detect the mouse, which is very expensive. Click operations don’t work properly. system camera windowlagging issues.

Paper [8] Designed a system can recognise particular human gestures use to communicate information for device control is one of the main objectives of gesture recognition. By implementing real-time gesture recognition, a user can control the volume of a desktop by making a particular hand gesture infront of a video camera connected to a computer. With the aid of the OpenCV module, create a hand gesture volume control system. In this project no need for a keyboard or mouse, the system may be controlled here via hand gestures. This is straightforward software control with hand gestures. Using simple hand gestures and a desktop-based volume control system, a user may adjust the volume in real-time. Additionally, we suggest using mouse cursor motion control. however, it only has the ability to increase and decrease volume but don’t perform direct volume mute operation.

Paper [9] Based on hand gesture recognition is an intelligent, intuitive, and practical method of interacting between humans and robots (HRI). In this study, the author gathered data for gesture recognition using a novel data glove named YoBu. The extreme learning machines (ELM) and SVM models gathered information on static gestures, created a gesture dataset, and examined which variables are crucial for classification. This gesture identification is based on a 54-dimension data glove called the ELM Kernel, which can perform better. Future robotic teleoperation based on gesture recognition may benefit from this study. nonetheless, it has a drawback since the user must wear gloves.

Paper [10] Designed a real-time tracking based virtual mouse application and virtual painting using finger colour caps based on gesture patterns and executed using a camera with the aid of artificial digital vision that integrates image processing and gesture recognition. Because the whole system is wireless and based on hand motion tracking and gesture recognition of coloured objects, it performs air gestures, which will be captured and created on the laptop screen and projected onto a wall for the benefit of the class attendees. This application is particularly for professors or lecturers who are colour blind, have poor vision, or have vertebral deformities, damaged legs, or spinal abnormalities. however, the user's usage of coloured caps to operate the application is a constraint.

From the detailed literature survey, it is evident that controller hand and finger tips are not very accurate and do not provide much hassle-free control throughout all mouse activities.

III. PROPOSED SYSTEM

The main aim of the proposed work is to perform mouse operations using human hand and fingers tips in virtual. The system collects real-time datasets by capturing streaming frames of data from a live webcam with the help of the OpenCV python library. The Media pipe models are used to track, detect, and recognize the hand gestures from each individual frame that is captured.

The proposed system module is divided into 4 parts: the hand tracking module, mouse module, volume module, and brightness module. The hand tracking module provides all of the essential libraries, and the hand detection class contains data about hand tracking, hand detection, hand location, hand landmarks, which fingertip is up, distance between two finger tips, and capture frame time period. To perform mouse operations, the following Mouse class module provides four mouse action methods: cursor movement, left/right click, scroll up/down, and drag/drop objects. Following that, the Volume and Brightness class module methods contain system data to perform increase/decrease/mute volume and increase/decrease control brightness.

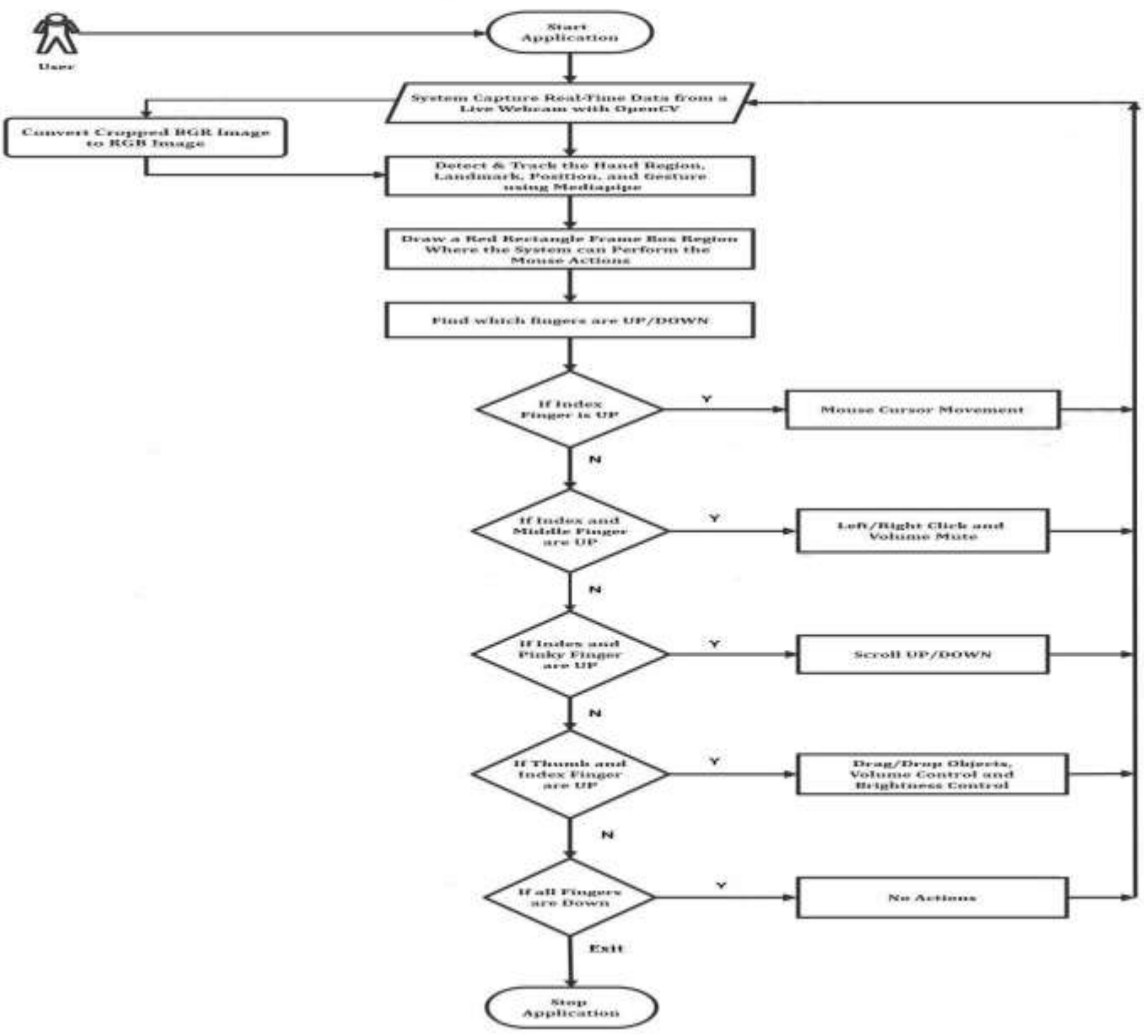

Fig. 1. Flowchart of Working Real-Time Virtual Mouse Model

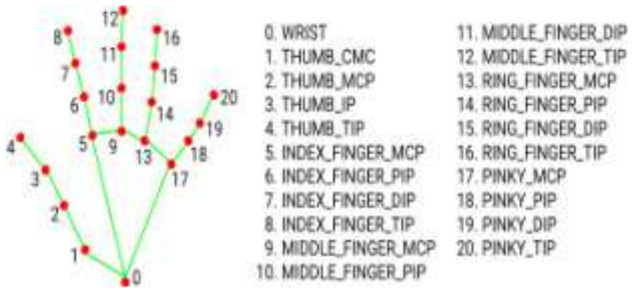

Fig. 2. 21 Hand Landmarks

A. Capturing the System Webcam video frames and processing the Hand image

The proposed gesture controlled virtual mouse system is built on a media pipe model and OpenCV libraries that capture real-time data frames from webcams and utilize them as input for further computation.

Here, the system's primary camera captures user hand frames as shown in Figure 1.

cap = cv2.VideoCapture(0)

The webcam, where each frame is collected until the application is terminated. The video frames captured by webcam are saved in BGR color format. The BGR color format must be converted to RGB with OpenCV can process the frames. Then, OpenCV reads frame by frame using the media pipe model to recognize hand landmarks and position to detect the hand tips and fingertip's actions in the video frames.

B. Initializing Media pipe Model and Draw Rectangular Region for Mouse Moving through the Window

Initializing the media pipe model is responsible for monitoring hand gestures and other events. Next, set the minimum confidence for hand detection and hand tracking by

self.hands=self.mpHands.Hands(self.mode=False,self.maxHands=1,self.modelCon=1,self.detectionCon=0.85, self.trackCon=0.15)

i.e., this threshold is fixed to attain good accuracy for hand gesture detection. Then, draw an axis line across the fingertips and hand tips using the media pipe. This keeps track of finger movement and establishes connections with neighbouring fingers.

The preconfigured media pipe model is being used to capture the 21 hand landmarks and monitor them as shown in Figure 2 If no landmarks are identified, the default value is zero. This indicates that no action is required. Otherwise, it returns the coordinate value of the X, Y, and Z axes, indicatingthat action is needed.

Following the capture of all hand landmarks from the webcam, a red rectangle box appears around the cv2 Window. This converts fingertip coordinates from the webcam screen to the PC window full screen for controlling mouse operation. When the hands are detected, as well as find which finger is up for performing the specific mouse actions, a rectangular region box is created in cv2 window screen. where the mouse moves the cursor across the PC window full screen and performs the mouse operation which is shown in Figure 1.

C. List of Virtual Mouse Operations Developed in Proposed System Model

The proposed system developed with OpenCV and Media pipe Model detects the hand tip and fingertips and draws a red rectangular window box around the cv2 window using a media pipe model that calculates the coordinates of the fingertips of the hand from the screen capture cv2 window to be able in the system. Therefore, the virtual mouse should perform some actions on the PC window screen as the user moves his/her fingertip. The following functions of the virtual mouse are implemented.

i. Movement of Mouse Cursor around the PC Window

The Hand Gestures the index finger is up with fingerUp ID = 1. That, i.e., the virtual mouse is in moving mode. When the users move their index finger around on the rectangle region, the cv2 window screen correspondingly mouse cursor also move in the same direction on the PC window screen. The speed at which the mouse cursor moves is proportional to the speed of the hand pointing gesture movement as shown in Figure 3.

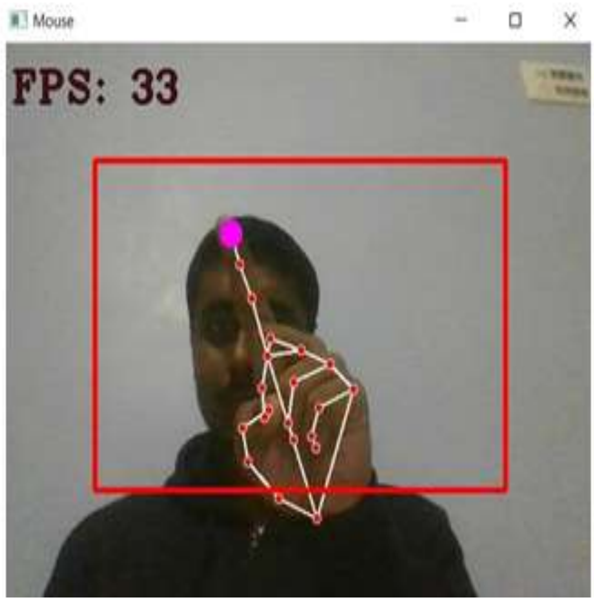

Fig. 3. Virtual Mouse Movement Action

ii. Left Click

The Hand Gestures of the index and middle fingers are up with fingerUp ID = 1 and fingerUp ID = 2. A mouse performs the left-click action on the PC window screen

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 201-209
Published Online October 2022 in IJEAST (http://www.ijeast.com)

when the distance between the index and middle fingertips is less than 40 cm or when they are brought closer together. The speed of the mouse's left-click action is proportional to the distancevictory gesture from the start point as shown in Figure 4.

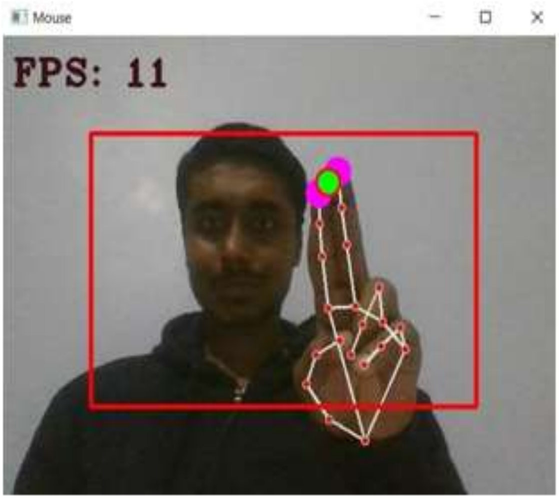

Fig. 4. Virtual Mouse Left Click Action

iii. Right Click

The Hand Gestures of the index and middle fingers are up with fingerUp ID = 1 and fingerUp ID = 2. A mouse performs the right-click action on the PC window screen when the index and middle fingertips are wider than one another, or when the gap between adjacent fingers range between less than 95cm and more than 110cm. The speed of the mouse right-clickaction is proportional to the distance victory gesture from the start point as shown in Figure 5.

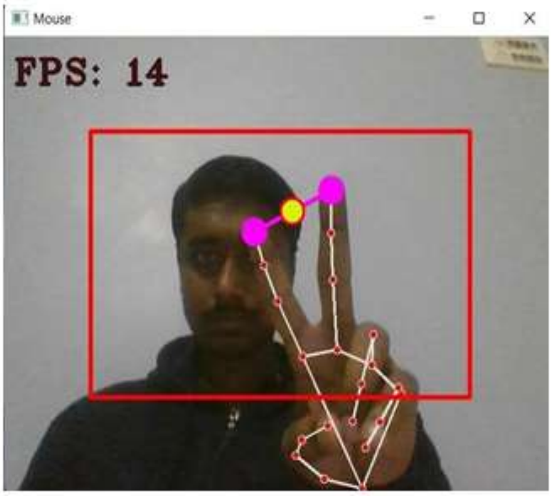

Fig. 5. Virtual Mouse Right Click Action

iv. Scroll UP

The Hand Gestures of the index and pinky fingers are up with fingerUp ID = 1 and fingerUp ID = 4. A mouse performs the scroll-up action on the PC window screen when the distance between the index and pinky fingertips is less than 90 cm or when they are brought closer together. The speed of the mouse scroll-up action is proportional to the distance horns gesture from the start point, and it is controlled vertical/horizontal by horns gesture actions as shown in Figure 6

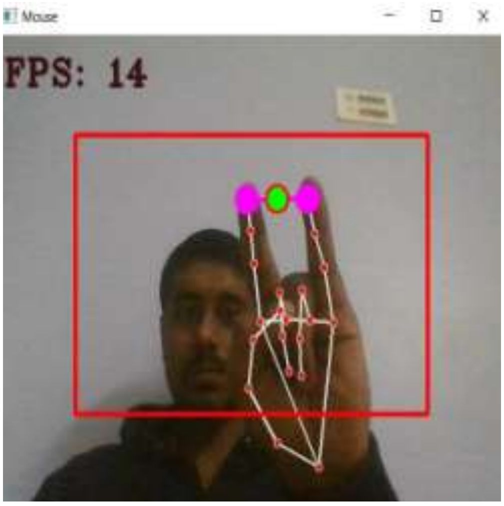

Fig. 6. Virtual Mouse Scroll UP Action

v. Scroll Down

The Hand Gestures of the index and pinky fingers are up with fingerUp ID = 1 and fingerUp ID = 4. A mouse performs the scroll-down action on the PC window screen when the distance between the index and pinky fingertips is greater than 90 cm or when they are wider than each other. The speed of the mouse scroll-down action is proportional to the distance horns gesture from the start point, and it is controlled vertical by horns gesture actions as shown in Figure 7.

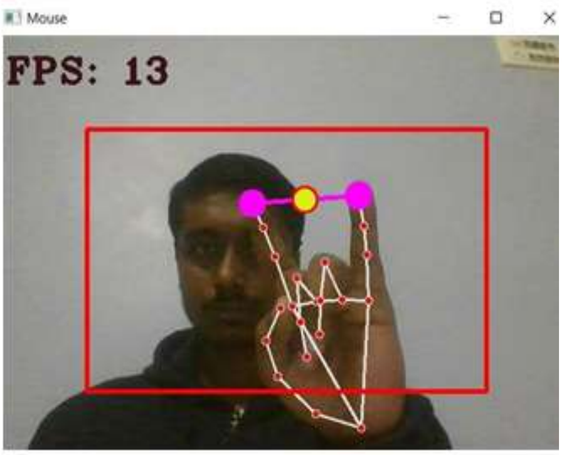

Fig. 7. Virtual Mouse Scroll Down Action

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 6, ISSN No. 2455-2143, Pages 201-209
Published Online October 2022 in IJEAST (http://www.ijeast.com)

vi. Drag Object

The Hand Gestures of the thumb and index fingers are up with fingerUp ID = 0 and fingerUp ID = 1. A mouse performs the drag object action on the PC window screen when the distance between the thumb and index fingertips is less than 50 cm or when they are brought closer together. It can move objects(files/folders) from one directory to another. The speed of the mouse drag action is proportional to the distance pinch gesture movement, and it is controlled drag actions by pinch gesture movement as shown in Figure 8.

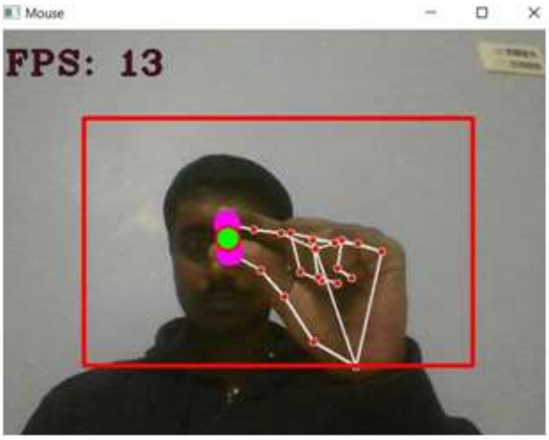

Fig. 8. Virtual Mouse Drag Object Action

vii. Drop Object

The Hand Gestures of the thumb and index fingers are up with fingerUp ID = 0 and fingerUp ID = 1. A mouse performs the drop object action on the PC window screen when the index and middle fingertips are wider than one another, or when the gap between adjacent fingers range between 60cm and 65cm. It can drop objects(files/folders) from one directory to another. It is controlled drop objects by pinch gesture action as shown in Figure 9.

Fig. 9. Virtual Mouse Drop Object Action

viii. Volume Controller

The Hand Gestures of the thumb and index fingers are up with fingerUp ID = 0 and fingerUp ID = 1. A mouse controls the volume action on the PC window screen when the thumb and index fingertips are closer together to reduce the volume to 0% or wider than one another to increase the volume to 100% depending on the length of the adjacent fingers, which range should be between 50 cm and 150 cm. The increase/decrease of volume control rate is proportional to the distance pinch gesture movement from the start point as shown in Figure 10.

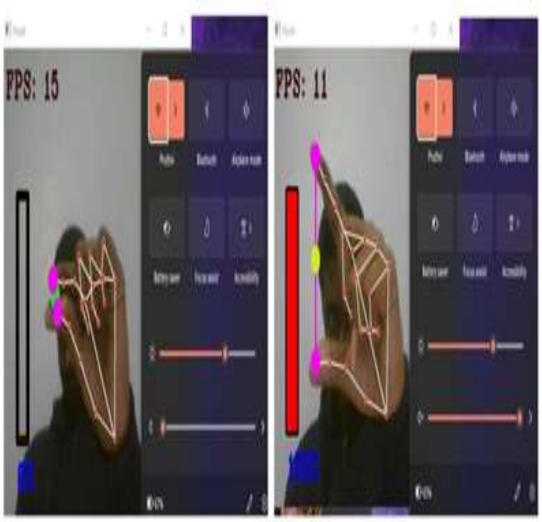

Fig. 10. Virtual Mouse Volume Control Action

ix. Volume Mute

The Hand Gestures of the index and middle fingers are up with fingerUp ID = 1 and fingerUp ID = 2. A mouse performs the volume mute action on the PC window screen when the distance between the index and middle fingertips is less than 40 cm or when they are brought closer together. The speed of the mouse volume mute action is proportional to the distance victory gesture from the start point as shown in Figure 11.

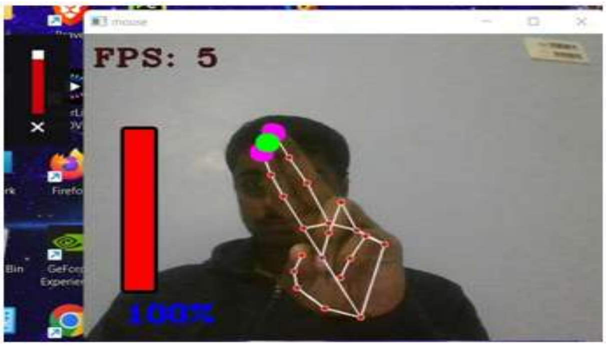

Fig. 11. Virtual Mouse Volume Mute Action

x. Brightness Controller

The Hand Gestures of the thumb and index fingers are up with fingerUp ID = 0 and fingerUp ID = 1. A mouse controls the brightness action on the PC window screen when the thumb and index fingertips are closer together to reduce the volume to 0% or wider than one another to increase the volume to 100% depending on the length of the adjacent fingers, which range should be between 50 cm and 150 cm.The increase/decrease of brightness control rate is proportional to the distance pinch gesture movement from the start point as shown in Figure 12.

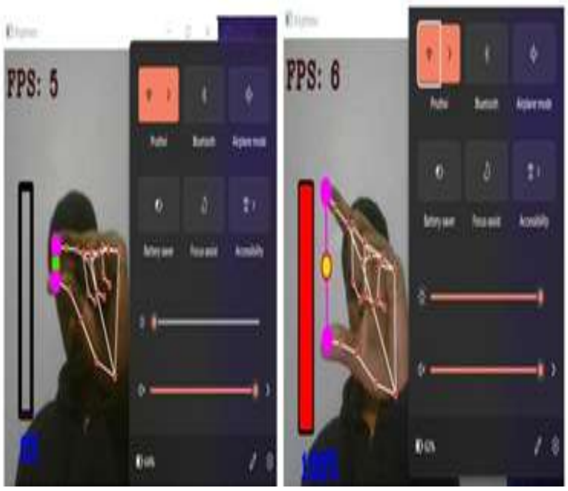

Fig. 12. Virtual Mouse Brightness Controller Action

xi. No Actions

The Hand Gestures of the finger's tips are down the PC doesn't perform any mouse actions in the window screen as shown in Figure 13.

Fig. 13. Virtual Mouse No Actions

IV. EXPERIMENTAL RESULTS

The proposed system is built on the principle of simulating mouse actions on a PC screen window by using modern AI and ML technologies, such as computer vision technology, to improve human and PC interaction. During the analysis and prediction of the proposed model dataset, real-time data was captured, such as hand tracking, fingertips detection, and hand gesture recognition from the webcam with OpenCV. The results of the prediction model recognizing hand gestures were achieved under various lighting conditions and varying distance between hand and webcam. Testing this approach is carried out in various lighting conditions and at varying distances from the webcam. The proposed approach is tested by each user 16 times by four different peoples: professionals, ordinary peoples, students, and expert peoples. The users performed mouse operations eight times in standard, low, high, and dim lighting conditions, and eight times at close, long, and normal hand distances from the webcam, The results of mouse actions are recorded in table 1.

Table -1: Users Tested Experimental Result

Mouse Operations	Expt.1	Expt. 2	Expt.3	Expt. 4	Accuracy
Mouse Cursor Movement	100	100	100	100	100/100
Left Click	100	100	100	99	100/100
Right Click	99	100	99	98	99/100
Scroll UP	99	100	99	99	99/100
Scroll Down	100	100	100	99	100/100
Drag Object	98	99	98	97	98/100
Drop Object	98	99	98	97	98/100
Volume Control	100	100	100	100	100/100
Brightness Control	100	100	100	100	100/100
No Actions	100	100	100	100	100/100
Total Result	994	998	994	989	994/1000

Table 1 shows the users manually tested results of a virtual mouse actions with professional peoples, ordinary peoples, students, and expert peoples. The results indicate no statistically significant difference between real and virtual mouse operations. Users can perform dynamic virtual mouse operations with 99% accuracy. After analyzing the mouse operations perform admirably in a variety of illumination situations, noisy backgrounds, distance tracking. The prediction model obtained astounding accuracy after examining the outcomes of the suggested model. As shows in figure 14 bar graph chart that demonstrates the percentage level of mouse operation accuracy that was attained after each user's virtual mouse activity.

Fig. 14. Level of Accuracy of virtual mouse actions

In existing works, analysis of results were not done. In our work, the analysis of results are conducted by giving our software application to different types of users (professional peoples, ordinary peoples, students, and expert peoples). Based on this accuracy is computed. Table 1 show the accuracy of various mouse actions tested by different types of peoples.

IV. CONCLUSION

The proposed system uses a PC webcam to capture 30fps real-time frames with OpenCV. The mediapipe model recognises those frames to process hand gestures and perform mouse operations like left/right-click, scroll up/down, drag/drop objects, and volume/brightness control without using a physical mouse devices. It is evident from test results, that the proposed system model performs well even in dim lighting conditions. The mouse operations work very well in real-world and real-time applications using just a CPU and no GPU. The proposed model's accuracy is 99%.

V. REFERENCE

[1] Tran, D.S., Ho, N.H., Yang, H.J., Kim, S.H. and Lee, G.S. (2021). Real-time Virtual Mouse System using RGB-D Images and Fingertip Detection. In Proceeding of International Conference on Multimedia Tools and Applications, pp.10473-10490.

[2] Reddy, Vantukala VishnuTeja, Thumma Dhyanchand, Galla Vamsi Krishna, and Satish Maheshwaram. (2020). Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition. In Proceeding of International Conference in Hyderabad Section, IEEE, pp.1-5.

[3] Chowdhury, S.R., Pathak, S., Praveena, M.A. (2020). Gesture Recognition Based Virtual Mouse and Keyboard. In Proceeding of 4th International Conference on Trends in Electronics and Informatics, IEEE, pp.585-589.

[4] Shetty, Monali, Christina A. Daniel, Manthan K. Bhatkar, and Ofrin P. Lopes. (2020). Virtual Mouse Using Object Tracking. In Proceeding of 5th International Conference on Communication and Electronics Systems, IEEE, pp.548-553.

[5] Shibly, Kabid Hassan, Samrat Kumar Dey, Md Aminul Islam, Shahriar Iftekhar Showrav. (2019). Design and development of hand gesture based virtual mouse. In Proceeding of 1st International Conference on Advancesin Science, Engineering and Robotics Technology, IEEE, pp.1-5.

[6] Varun, K.S., Puneeth, I. and Jacob, T.P. (2019). Virtual mouse implementation using OpenCV. In Proceeding of 3rd International Conference on Trends in Electronics and Informatics, IEEE, pp.435-438.

[7] Shajideen, S. M. S., Preetha, V. H. (2018). Hand Gestures- Virtual Mouse for Human Computer Interaction. InProceeding of International Conference on Smart Systems and Inventive Technology, IEEE, pp.543-546.

[8] SK. Abdul Soniya, R.V. Harshitha, Y. Veera Reddy, Dr. Ratna Raju Mukiri. (2018). A Novel Approach: Hand Gesture Volume Control System. In Proceeding of International Journal of Research, volume-7, issue-7, pp.200-206.

[9] Lu, D., Yu, Y. and Liu, H. (2016). Gesture recognition using data glove: An extreme learning machine method. InProceeding of IEEE international conference on robotics and biomimetics, IEEE, pp.1349-1354.

[10] Bhattacharjee, A., Jana, I., Das, A., Kundu, D., Ghosh, S. and Gupta, S.D. (2015). A novel probabilistic approach of colored object detection and design of a gesture based real-time mouse tracking along with virtual teaching intended for color-blind people. In Proceeding of 2nd International

Conference on Signal Processing and Integrated Networks, IEEE, pp.512-519.

[11] Anusha S, N Vignesh Karthik and Sampada K S. (2018). Comparative Study on Voice Based Chat Bots. International Journal of Computer Sciences and Engineering, Vol. 6, No. 12, pp. 172-175.

[12] Sampad Mondal, Subham Paul, Somjeet Ganguly and Ratul Dey. (2019). Virtual Personal Assistant with Face Recognition Login System Using Machine Learning and Artificial Intelligence. International Journal of Engineering Science and Computing, Vol. 9, No. 4, pp. 21603-21605.