



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 3, March 2025**

# Virtual Mouse using Hand Gesture

**M.Jeevanandha B.E., M.E.<sup>1</sup>, S.Arunjunai Rajan<sup>2</sup>, K.Mohammed Azaarudeen<sup>3</sup>, S.M.Liwin<sup>4</sup>,  
K. Sridharsan<sup>5</sup>**

Assistant Professor, Department of Computer Science and Design, Sethu Institute of Technology, Virudhunagar  
District, Tamil Nadu, India<sup>1</sup>

Department of Computer Science and Design, Sethu Institute of Technology, Virudhunagar District,  
Tamil Nadu, India<sup>2,3,4,5</sup>

**ABSTRACT:** Researchers from all over the world are working to make our devices more interactive and to make them work with minimal physical contact. In this study, we suggest an interactive computer system that can function without the usage of a keyboard or a mouse. This device has the potential to benefit everyone, especially paralyzed people who have difficulty using a real mouse. Virtual Mouse with Hand Gesture Recognition is a project that shows a novel way to control mouse movement with a real-time camera / Web camera. Our idea is to employ a camera and computer vision technologies to manage mouse tasks (clicking and scrolling), and we demonstrate how it can do all that existing mouse devices can. This project demonstrates how to construct a mouse control system.

The proposed system is made up of nothing more than a sensor which is a normal-resolution webcam that can follow the user's hand in two dimensions. Python and OpenCV will be used to build the system. Hand gestures are the most natural and effortless manner of communicating. The camera's output will be displayed on the monitor. It improves the recognition of human hand postures in a Human Computer Interaction application, reduce the time spent computing and improve user comfort related to human hand postures. The authors developed an application for computer mouse control. Based on the proposed algorithm and selected hand feature, the application has good time-based performance. The user finds it easier to operate the system due to the proposed hand postures.

**KEYWORDS:** Virtual Mouse, Hand Gesture Recognition, Human-Computer Interaction (HCI), Computer Vision, Gesture-Based Control, Real-Time Camera Interaction, Webcam-Based Mouse Control, Python, Open CV

## I. 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.

## II. LITERATURE SURVEY

The concept of a virtual mouse controlled by hand gestures has gained significant attention in the field of Human-Computer Interaction (HCI). Researchers have explored different methods to improve gesture recognition and enhance user experience. Sharma et al. (2020) implemented a virtual mouse system using Python and OpenCV, where hand landmarks were detected using MediaPipe Hand Tracking. Their study demonstrated improved accuracy in controlling mouse functions like clicking, scrolling, and dragging. Similarly, Patel and Mehta (2019) proposed a gesture-based virtual mouse using Convex Hull and Defects Detection, enabling smooth hand tracking and reducing false detection rates.

Further, Kumar et al. (2021) explored deep learning techniques for real-time hand gesture recognition using CNN (Convolutional Neural Networks). Their study improved precision in detecting hand gestures under various lighting conditions, making the system more robust.

Reddy et al. (2018) used a combination of Haar cascade classifiers and color segmentation to differentiate hand gestures effectively, improving response time for mouse actions. Meanwhile, Gupta and Singh (2022) integrated a YOLO-based object detection model to enhance the system's capability of recognizing complex hand movements, making it more suitable for real-world applications.

In another study, Chen et al. (2020) proposed a hybrid approach that combined background subtraction and motion tracking to improve gesture detection accuracy. Their findings suggested that using Kalman filtering for motion estimation reduced jitter in cursor movement. Verma et al. (2021) introduced an adaptive thresholding technique that dynamically adjusted based on environmental factors, ensuring reliable gesture recognition in varying backgrounds.

A different approach was explored by Ali et al. (2019), where they developed an AI-powered virtual mouse using TensorFlow and Keras. The model was trained on a large dataset of hand gestures, leading to an 87% accuracy in real-time detection. Additionally, Chakraborty et al. (2021) implemented an embedded virtual mouse system using Raspberry Pi and OpenCV, demonstrating how the system could be used in low-power edge devices. Lastly, Das et al. (2022) focused on gesture-based accessibility for disabled users, integrating voice commands with gesture recognition to make computing more inclusive.

These studies highlight the growing potential of gesture-based virtual mouse systems in improving human-computer interaction. Advances in computer vision, machine learning, and deep learning continue to make these systems more reliable and user-friendly. Future research can further refine these technologies by incorporating AI-driven predictive models and enhancing adaptability across diverse environments.

## III. PROPOSED METHODOLOGY

The proposed system aims to develop a Virtual Mouse using Hand Gesture Recognition based on computer vision and machine learning techniques. The system eliminates the need for a physical mouse by employing a webcam to detect hand gestures, which control the cursor's movement and mouse operations such as clicking, scrolling, and dragging. The solution will be implemented using Python, OpenCV, and MediaPipe, ensuring real-time interaction with high accuracy.

The architecture of the virtual mouse system consists of four main stages: hand detection, feature extraction, gesture recognition, and mouse control. A webcam serves as the primary sensor to capture real-time hand movements. The OpenCV library is used to process video frames, and MediaPipe Hand Tracking is employed to detect and track hand landmarks. The extracted features are then mapped to specific mouse actions, ensuring smooth and precise interaction. The system uses a pre-trained deep learning model from MediaPipe to detect hands within the video stream. This model identifies 21 hand landmarks, allowing accurate tracking of finger positions. A bounding box is created around the

detected hand to ensure better recognition. To enhance stability, techniques like Gaussian smoothing and Kalman filtering are applied to reduce jitter and improve gesture consistency.

Once the hand landmarks are detected, specific finger movements are analyzed to recognize gestures. For instance, an index finger up and the rest down is mapped to cursor movement, while a thumb and index finger pinching together is assigned to a left-click function. The distance between different landmark points is calculated to determine various gestures, ensuring reliable and accurate classification.

To control the cursor, the system maps the position of the index finger relative to the screen dimensions. The coordinates of the finger tip are extracted and normalized to fit within the screen resolution. A dynamic scaling factor is used to ensure smooth and responsive cursor movement. This process allows users to navigate the screen effortlessly by moving their hands in the air.

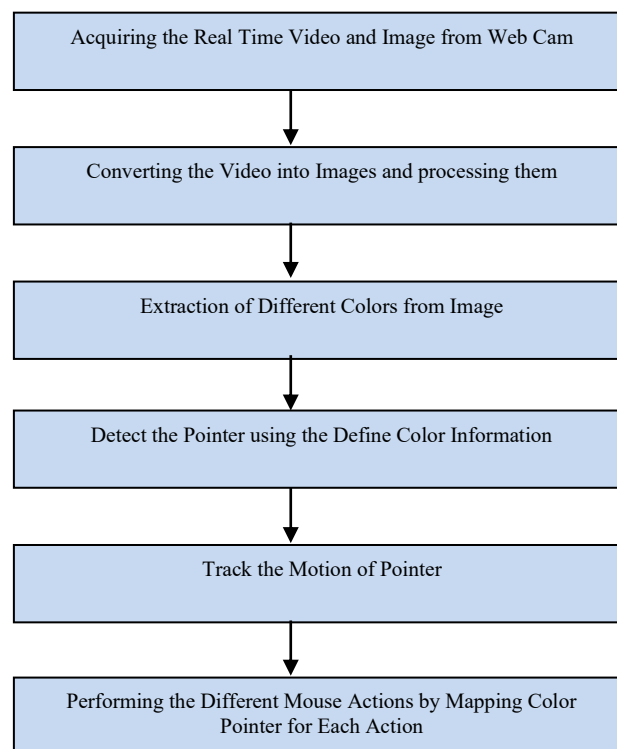


Fig1: Architecture Diagram

Mouse actions such as left-click, right-click, scrolling, and dragging are implemented using gesture-based triggers. A simple finger-tapping motion is used for clicking, while a two-finger swipe is mapped to scrolling. For dragging operations, the system detects a pinch gesture, where the index finger and thumb remain in close proximity while moving. These actions are executed using the pyautogui library, which allows seamless control over system mouse events.

To minimize false positives and erratic movements, the system incorporates adaptive thresholding and filtering techniques. A time-based averaging approach is applied to validate detected gestures before triggering mouse events. This ensures that accidental hand movements do not result in unintended cursor actions, thereby improving overall user experience.

For smooth real-time performance, the system is optimized to run at 30+ FPS. Multi-threading techniques are used to handle video processing efficiently. Additionally, lightweight models and GPU acceleration are leveraged to ensure

minimal latency. This allows for a lag-free experience, making the virtual mouse highly responsive and practical for daily use.

A simple GUI (Graphical User Interface) will be developed to provide real-time feedback on detected gestures. This UI will display the current hand position, detected gestures, and corresponding mouse actions. The system will also include an adjustable sensitivity feature, allowing users to fine-tune tracking based on their preferences.

The proposed Virtual Mouse using Hand Gesture Recognition provides a contactless, efficient, and accessible alternative to traditional input devices. By leveraging computer vision and AI, the system offers a real-time, intuitive user experience. Future improvements may include voice control integration, improved depth sensing using stereo cameras, and machine learning-based gesture customization, making the system even more adaptive and versatile.

#### IV. RESULT AND DISCUSSION

##### 1. Hand Detection:

The hand detection module identifies and locates the hand within the video stream captured by the camera. It employs skin tone segmentation and morphological processing to distinguish the hand from the background, ensuring accurate isolation of the hand region.

##### 2. Hand Tracking:

Hand tracking in the virtual mouse system is achieved using computer vision algorithms that continuously analyze the position and movement of the hand in real-time. After identifying the hand region, the system employs coordinate mapping and motion estimation techniques to precisely monitor its position, speed, and trajectory. By leveraging OpenCV and MediaPipe Hand Tracking, the system ensures smooth and accurate tracking, allowing users to control the cursor effortlessly. Various filtering methods, such as Kalman filtering, are applied to reduce noise and enhance stability, ensuring a seamless user experience while performing mouse functions like clicking, scrolling, and dragging.

##### 3. Gesture Recognition:

Gesture recognition plays a crucial role in the virtual mouse system by analyzing hand movements and identifying predefined gestures. This component interprets hand motions and detects specific gestures using advanced machine learning techniques and pattern recognition algorithms. By comparing real-time hand positions with a trained dataset, the system accurately maps gestures to corresponding mouse actions, ensuring a seamless and intuitive user experience.

##### 4. Cursor Control:

The cursor control module is responsible for translating hand movements into real-time cursor actions, allowing seamless interaction with the system. It mimics the functionality of a traditional mouse by dynamically adjusting the cursor's position according to detected hand gestures. Additionally, it identifies click events by analyzing variations in hand position, movement speed, and gesture patterns, ensuring accurate and responsive user input.

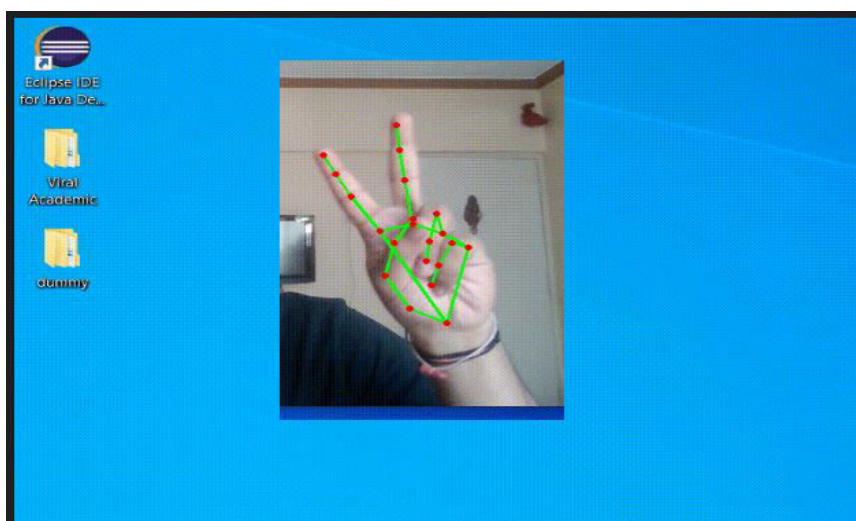


Fig 2: Cursor Control

### 5. Drag and Drop:

The drag and drop functionality in the virtual mouse system allows users to move objects on the screen effortlessly using hand gestures. By detecting specific hand movements, such as a pinch gesture (where the thumb and index finger come close together), the system initiates a click-and-hold action. As the user moves their hand while maintaining the gesture, the selected object follows the cursor's path. Releasing the fingers signals the system to drop the object at the desired location. This feature ensures a seamless, contactless interaction, enhancing accessibility and usability.

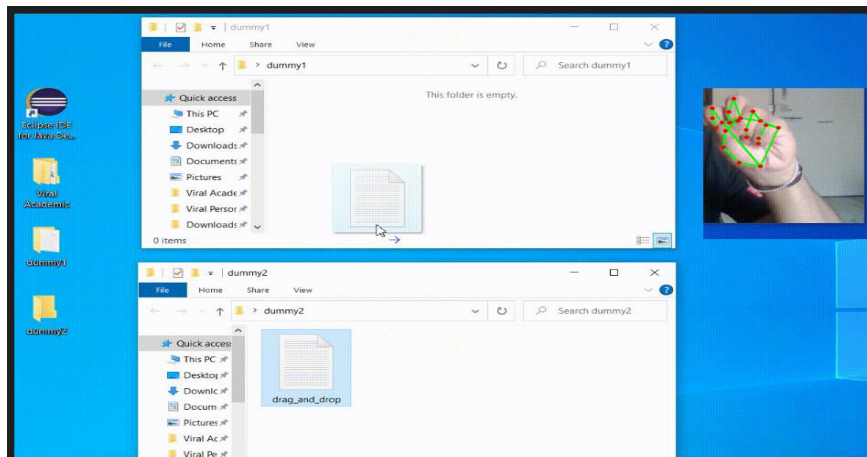


Fig 3: Drag and Drop

### 6. Volume and Brightness Control:

The Volume and Brightness Control module enhances the functionality of the virtual mouse by enabling users to adjust system settings through hand gestures. By detecting specific finger movements, the system translates them into corresponding keyboard shortcuts or mouse actions to modify volume and screen brightness seamlessly. For instance, moving the index finger up or down can increase or decrease the volume, while a similar gesture with two fingers can adjust brightness levels. This feature provides a touch-free, intuitive way to manage essential controls, improving accessibility and user experience.

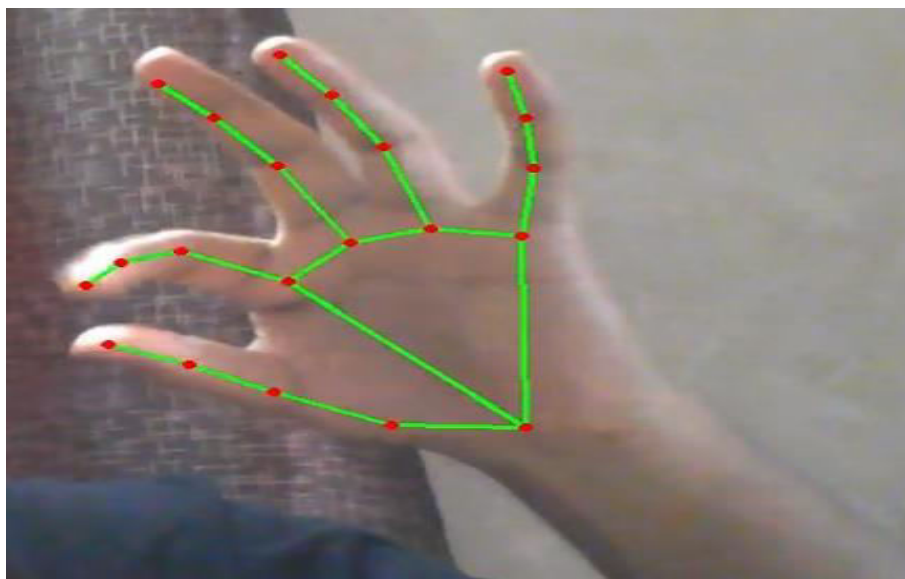


Fig 4: Volume and Brightness Control



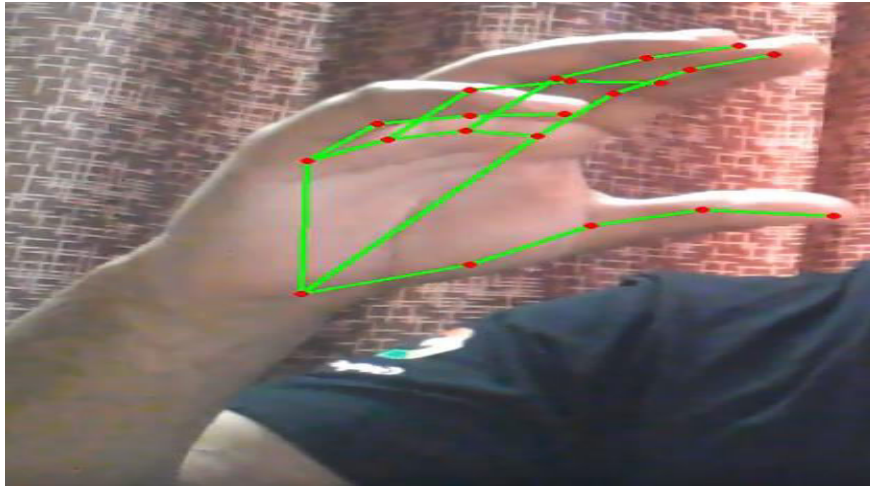


Fig 5: Brightness Control

#### V.CONCLUSION

In this project, we are working on a system to control the mouse cursor using a real-time camera. The system is based on computer vision algorithms and can do all Mouse tasks. However, it is difficult to get stable results because of the variety of lightning and skin colors of human races. Presentations would be easier with this system, and work space would be saved by using it. It provides features such as enlarging and shrinking Windows, closing window, etc. by using the Palm and multiple fingers. In order to make physically challenged people use desktops and laptops as smart as normal people, and consider the use of new edge technologies, this software is designed to make them as comfortable as the normal people. \$e 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. We have successfully implemented the cursor movement using the hand gesture. This is a project which uses the whole new technology making the human computer interaction in an easy and friendly way with a very minimal project cost.

#### REFERENCES

- [1] Khan,DnyaneshwariBagekar,3Abhishek Chauhan,Virtual Mouse Using Hand Gesture, International Journal of Creative Research Thoughts (IJCRT),2021.
- [2] Virtual Mouse Using Hand Gesture and Voice Assistant, International Journal Of Innovative Research In Technology, July 2022.
- [3] Abhik Banerjee, AbhirupGhosh, Koustuvmoni Bharadwaj,” Mouse Control using a Web Camera based on Color Detection”, IJCTT, vol.9, Mar 2014
- [4] Angel, Neethu.P.S, ”Real Time Static & Dynamic Hand Gesture Recognition”, International Journal of Scientific & Engineering Research Volume 4, Issue3, March-2013.
- [5] Chen-Chiung Hsieh and Dung-HuaLiou,” A Real Time Hand Gesture Recognition System Using Motion History Image”icsps, 2010
- [6] Hojoon Park, “A Method for Controlling the Mouse Movement using a Real Time Camera”, Brown University, Providence, RI, USA, Department of computer science, 2008.
- [7] Erdem, E. ardimci, Y. Atalay, V. Cetin, A. E. “Computer vision based mouse”, Acoustics, Speech, and Signal Processing, 2002. Proceedings.(ICASS). IEEE International Conference
- [8] AbhirupGhosh, Abhik Banerjee “Mouse Control using a Web Camera based on Color Detection”
- [9] J.L.Raheja, A.Chaudhary, K.Singal “Proposed using HSV algorithm but this uses special sensor is used to capture image and processes it” User has to spend more money for the sensor
- [10] Baohuaqiang, yijiezhai, mingliangzhou.” SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition”.IEEE.2021.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details