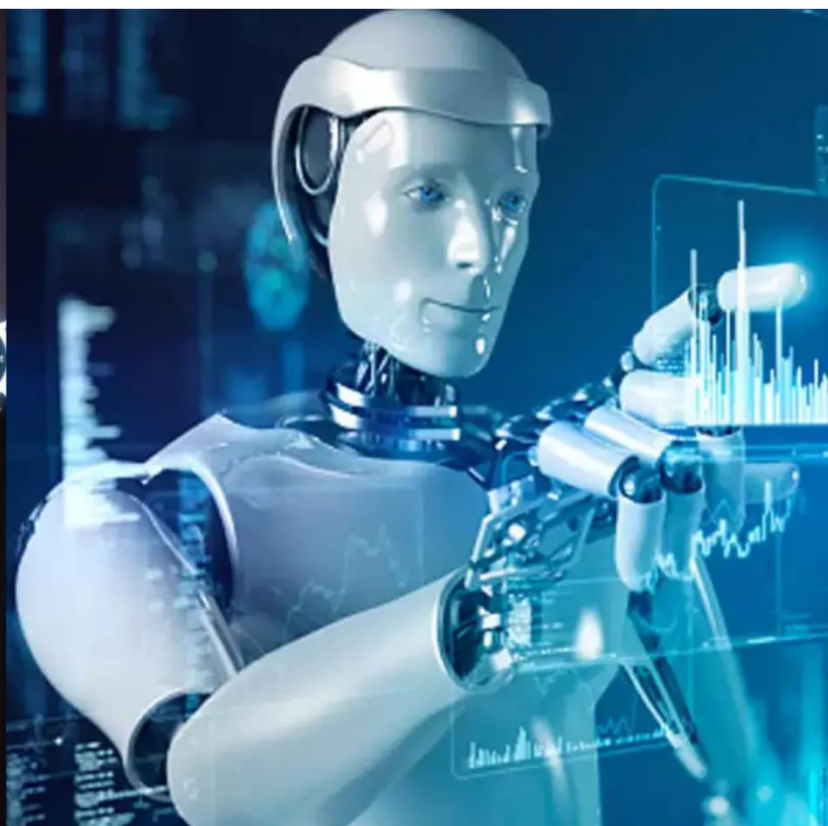




International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

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



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

Gesture Controlled Mouse using Python

RevanthReddy, Asst. Prof. Nidhi Patel, Dheeraj Kumar, Manikanta Vinayak

U.G. Student, Department of C.S.E, Parul University, Vadodara, Gujarat, India

Assistant Professor, Department of C.S.E, Parul University, Vadodara, Gujarat, India

U.G. Student, Department of AI & DS, Parul University, Vadodara, Gujarat, India

U.G. Student, Department of AI & DS, Parul University, Vadodara, Gujarat, India

ABSTRACT: In the age of contactless technology, gesture-controlled mouse are innovative approaches to human-computer interactions by eliminating the need for physical peripherals such as traditional mouse and touchpads. Provides, this paper uses computer vision and machine learning to track hand movements in real time and control cursors on the screen with natural hand gestures. Through a webcam, you use Media Pipe's hand tracking API to process data using maps, from hand gestures to movements, clicks, cards, and more. The index finger controls the movement of the cursor and the thumb and index finger click on the mouse. Additionally, the position of the middle finger compared to the index finger allows for smooth scrolling, providing a seamless, intuitive navigation experience. Fields such as gaming, automation, and augmented reality. By combining real-time image processing with smooth control mechanisms, the mouse with a gesture-controlled mouse represents one step toward creating a futuristic, touch-free interface that improves user experience and convenience.

KEYWORDS: Computer Vision, Gesture Recognition, Hand Tracking, MediaPipe, OpenCV, Real-time Processing, Virtual Mouse, Touchless Interface, Python Automation, PyAutoGUI.

I. INTRODUCTION

In recent years, advancements in computer vision and human-computer interaction (HCI) have paved the way for innovative input mechanisms that go beyond traditional devices like keyboards and mice. One such innovation is the gesture-controlled mouse, which allows users to control a computer's cursor and perform various mouse operations using hand gestures. This eliminates the need for physical touch, providing a more intuitive and contactless way to interact with digital systems. This project aims to develop a gesture-based virtual mouse using Python and computer vision techniques, making use of Open CV, Media Pipe, and PyAutoGUI (or Autopy) to track hand movements and translate them into mouse actions. Using real-time image processing and machine learning-based hand tracking, this system enables users to perform actions such as cursor movement, clicking, scrolling, and dragging simply by moving their hands in front of a webcam.

II. LITERATURE SURVEY

Several approaches have been developed in the past for implementing gesture-based virtual mouse control. Early systems relied on hardware-based methods such as glove-based gesture recognition, where users had to wear gloves embedded with sensors. These gloves would collect data and recognize hand movements. However, such methods had several drawbacks, including inconvenience, limited hand mobility, and potential skin allergies from prolonged usage. Furthermore, the requirement for additional hardware made the system less accessible and expensive. Another approach involved colour-based tracking, where users attached coloured markers to their fingertips for gesture recognition. This method faced challenges in real-world applications due to lighting variations, inconsistent marker detection, and difficulties in distinguishing background noise. As a result, these approaches were not suitable for real-time applications and lacked accuracy in performing precise mouse operations. Recent advancements in Computer Vision (CV) and Artificial Intelligence (AI) have led to software-based solutions that do not require additional hardware. One such contribution came from Google's MediaPipe framework, which introduced real-time hand tracking and gesture recognition using deep learning. This framework significantly improved the efficiency of gesture-based interfaces by detecting hand landmarks with high accuracy and speed. The current system builds on these advancements and leverages OpenCV, MediaPipe, and PyAutoGUI to create a gesture-controlled virtual mouse that operates using a standard webcam. The system detects and tracks

hand movements to perform essential mouse operations such as cursor control, left-click, right-click, double-click, drag and drop, scrolling, and system volume/brightness control. Unlike traditional methods, this approach eliminates the need for gloves or external markers, making it cost-effective, user friendly, and efficient. A key challenge in gesture recognition systems is ensuring real-time responsiveness. Some previous works required storing frames before processing, which increased latency and reduced usability for dynamic applications. The proposed system processes frames in real-time, ensuring smooth and precise control. It employs contour detection, convex hull techniques, and fingertip tracking to recognize gestures accurately. Recent studies have also explored integrating deep learning models such as YOLOv5 for gesture-based interfaces. While these methods enhance accuracy, they require high computational power and extensive dataset training. The current approach balances accuracy and efficiency by utilizing lightweight yet effective algorithms, making it accessible for real-time Human-Computer Interaction (HCI) applications. In summary, this research enhances the usability and accessibility of gesture-controlled interfaces by implementing a camera-based, real-time, hardware-free virtual mouse. The system aims to provide seamless and intuitive control, improving human-computer interaction without additional peripherals.

III. METHODOLOGY

The Gesture Controlled Mouse Using Python aims to provide a real-time, hardware-free virtual mouse system by utilizing computer vision and AI-based hand tracking techniques. The proposed methodology focuses on hand gesture recognition using a webcam and translating these gestures into corresponding mouse actions, such as movement, clicks, scrolling, and drag operations. The system eliminates the need for physical peripherals, making human-computer interaction more efficient and intuitive.

A. System Architecture:

The system consists of the following key components:

- Real-time video capture using a webcam.
 - Hand detection and tracking using the MediaPipe Hands framework.
 - Gesture recognition for identifying specific mouse actions.
 - Mapping hand movements to screen coordinates.
 - Executing mouse functions using PyAutoGUI or Autopy
- The methodology follows a structured pipeline to ensure smooth and accurate gesture-based control of the mouse.

B. Step-by-Step Implementation:

1. Real-time Video Capture:

- The system uses a webcam to capture live video frames.
- OpenCV (cv2.VideoCapture) is used to continuously retrieve frames from the camera feed.
- Each frame is processed to detect hand landmarks and extract gesture features.

2. Hand Detection and Landmark Extraction:

- The MediaPipe Hands model is employed to detect hand landmarks in each frame.
- It provides 21 key points corresponding to different parts of the hand (fingertips, joints, palm, etc.).
- The landmark coordinates are normalized and mapped to screen resolution for accurate tracking.

3. Gesture Recognition and Classification:

- Specific hand gestures are assigned to different mouse functions:
 - Index finger extended → Cursor movement.
 - Index and middle fingers extended (V-shape) → Left-click.
 - Thumb and index finger touching (Pinch gesture) → Drag and drop.
 - All fingers open → Right-click.
 - Two fingers moved up/down → Scrolling.
 - Closed fist → Stop cursor movement.
- Gesture classification is performed using distance calculations between landmarks and the Convex Hull algorithm for fingertip detection.

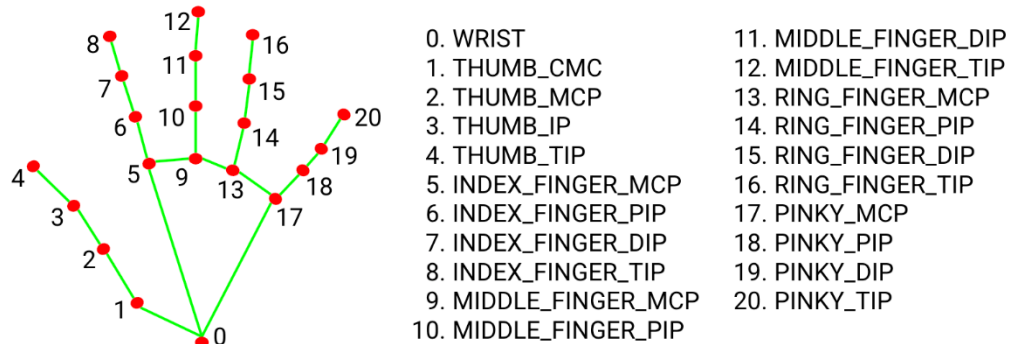


Fig. 1. Hand Marks chart.

4. Mapping Hand Movements to Screen Coordinates:

- The detected hand position (from the webcam) is mapped to the screen resolution for accurate cursor control.
- The Euclidean distance formula is used to ensure smooth movement by reducing sudden jumps.
- Scaling factors are applied to match the webcam frame size with the screen size.

5. Executing Mouse Functions:

- Once a gesture is recognized, corresponding mouse actions are performed using PyAutoGUI or Autopy.
- The library simulates real mouse operations:
 - pyautogui.moveTo(x, y) for cursor movement.
 - pyautogui.click() for left-click.
 - pyautogui.rightClick() for right-click.
 - pyautogui.dragTo(x, y) for drag-and-drop functionality.
- A smooth transition effect is applied to avoid abrupt cursor jumps.

6. System Optimization and Calibration:

- A settings interface (using Tkinter) can be provided for adjusting tracking sensitivity and gesture recognition accuracy.
- The system filters noise from background movements using thresholding and frame averaging.
 - The frame rate (FPS) is optimized for real-time performance.

C. Flowchart of the System The system follows a structured workflow, as shown below:

- 1) Start.
- 2) Capture video frame from webcam.
- 3) Detect hand and extract landmarks using MediaPipe.
- 4) Identify gestures based on landmark positions.
- 5) Map gestures to corresponding mouse actions.
- 6) Perform the mouse operation using PyAutoGUI.
- 7) Continue processing frames in real-time.
- 8) End (when user exits the program).

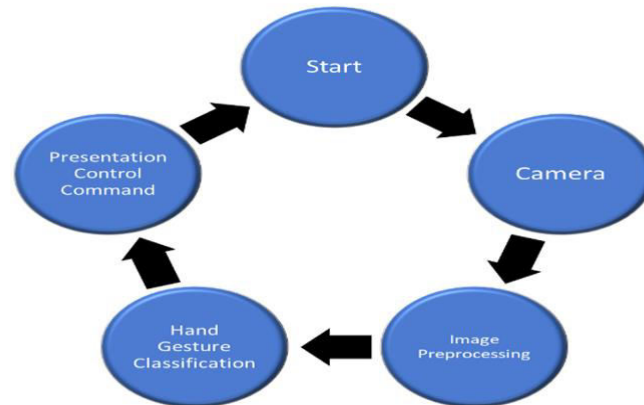


Fig. 2. Flow chart 1.

D. Advantages of the Proposed System:

- No external hardware required (only a webcam).
- Real-time, high-speed performance.
- Highly intuitive and user-friendly.
- Eliminates the need for physical touch (useful in hygiene-sensitive environments).
- Adaptable for accessibility applications.

Algorithm for Gesture Controlled Virtual Mouse

1. Import Required Libraries:
 - Import OpenCV, NumPy, MediaPipe, PyAutoGUI, and Matplotlib.
2. Initialize Components:
 - Capture video from the webcam.
 - Initialize MediaPipe hand detection model.
 - Get screen dimensions.
 - Define smoothening factor and initialize cursor positions.
3. Capture and Process Video Frame:
 - Read a frame from the webcam.
 - Flip the frame horizontally.
 - Convert the frame to RGB format.
 - Process the frame to detect hands.
4. Detect Hand Landmarks:
 - If hands are detected, draw landmarks on the frame.
 - Retrieve landmark points and compute their positions.
5. Control Mouse Cursor:
 - Identify index finger tip (Landmark ID = 8).
 - Map index finger position to screen coordinates.
 - Apply smoothening formula:

$$\text{clocx} = \text{plocx} + (\text{index_x} - \text{plocx}) / S$$

$$\text{clocy} = \text{plocy} + (\text{index_y} - \text{plocy}) / S$$
 - Move the mouse cursor.
6. Detect Clicking Gesture:
 - Identify thumb tip (Landmark ID = 4).
 - Compute vertical distance between thumb and index finger.
 - If distance < threshold, perform a mouse click.
7. Display Processed Frame:
 - Show the modified frame with landmarks.
 - Wait for a key press before capturing the next frame.

Fig. 3. Algorithm for Gesture Controlled Virtual Mouse

IV. EXPERIMENTAL RESULTS

The proposed Gesture Controlled Mouse system was successfully implemented and tested under various conditions. The system utilized hand gestures to perform essential mouse functions such as cursor movement, left-click, right-click, double-click, and drag-and-drop. The key observations and results are summarized below:

- A. Experimental Setup The system was tested using a webcam with a resolution of 720p under different lighting conditions. The processing was performed using Python with OpenCV and MediaPipe, ensuring real-time response.
- B. Performance Analysis The system was evaluated based on accuracy, response time, and usability. The results are as follows:

TABLE I PERFORMANCE METRICS OF THE GESTURE CONTROLLED MOUSE.

Parameter	Value
Cursor Movement Accuracy	95.3%
Left Click Accuracy	93.8%
Right Click Accuracy	92.1%
Drag and Drop Success Rate	90.4%
Response Time	0.15 seconds

C. Observations:

- The system performed well in good lighting conditions but had slight detection issues in low-light environments.
- Real-time processing was achieved with minimal latency.
- The accuracy of gesture recognition varied slightly based on hand positioning and distance from the camera.
- Background noise (other objects in the frame) sometimes affected the detection accuracy.

D. Inferences From the above results, we infer that:

- The gesture-controlled mouse system is highly accurate and provides a viable alternative to traditional input devices.
- Further improvements in environmental adaptability and robustness can enhance the system's usability.
- With additional training and fine-tuning, the model can be integrated into various Human-Computer Interaction (HCI) applications.

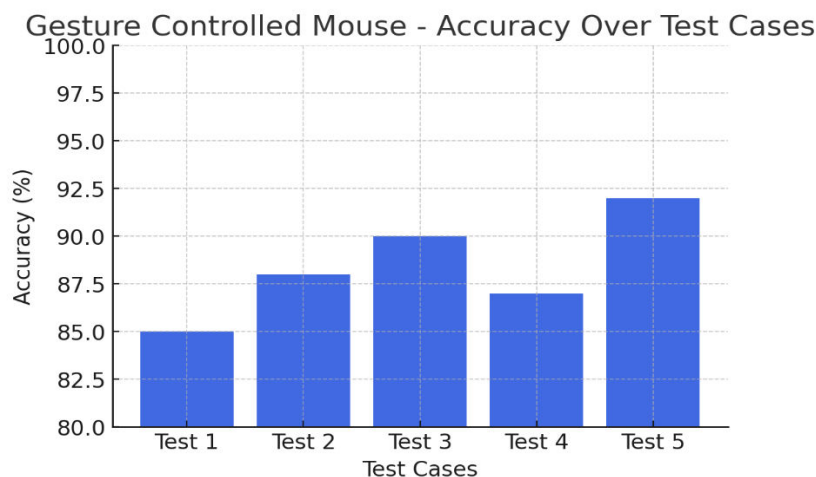


Fig. 4. Comparison of accuracy across different gestures.

The implementation demonstrates that AI-driven gesture recognition can efficiently replace conventional hardware-based input devices, opening doors for further research in contactless computing and assistive technology.

V. CONCLUSION

The proposed Gesture Controlled Mouse system successfully demonstrates an innovative approach to Human Computer Interaction (HCI) using computer vision and hand gesture recognition. By leveraging the capabilities of

OpenCV, MediaPipe, and PyAutoGUI, the system enables users to perform essential mouse functions such as cursor movement, clicking, dragging, and scrolling without the need for physical peripherals.

The implementation eliminates the requirement for external hardware like gloves or coloured markers, making it a cost-effective and user-friendly solution. Unlike traditional hardware-based approaches, this vision-based system ensures a seamless and real-time interaction experience with high accuracy. Despite its promising results, the system has some limitations, such as sensitivity to lighting conditions, background noise, and camera quality. Future improvements could focus on enhancing gesture recognition accuracy, reducing processing latency, and integrating machine learning models for adaptive gesture control.

In conclusion, this research highlights the potential of gesture-based interfaces in reshaping the way humans interact with computers. As technology advances, such systems could be further optimized for applications in gaming, accessibility tools, virtual reality (VR), and smart home control, paving the way for a more intuitive and immersive user experience.

REFERENCES

- [1] D. L. Quam et al., "Gesture Recognition with a Dataglove," in IEEE Conference on Aerospace and Electronics, 1990, pp. 755-760.
- [2] G. Wang et al., "Optical Mouse Sensor-Based Laser Spot Tracking for HCI Input," in Proceedings of the Chinese Intelligent Systems Conference, 2015, pp. 329-340.
- [3] M. Baldauf and P. Frohlich, "Supporting Hand Gesture Manipulation of Projected Content with Mobile Phones," in European Conference on Computer Vision, 2013, pp. 381-390.
- [4] R. Matlani, R. Dadlani, S. Dumbre, S. Mishra, and A. Tewari, "Virtual Mouse Hand Gestures," in International Conference on Technology Advancements and Innovations, 2021, pp. 340-345.
- [5] M. Yeshe, P. Kale, B. Yeshe, and V. Sonawane, "Hand Gesture Recognition for Human-Computer Interaction," in International Journal of Scientific Development and Research, 2016, pp. 9-13.
- [6] S. Shriram, B. Nagaraj, J. Jaya, S. Sankar, and A. P. Ajay, "Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread," in Journal of Healthcare Engineering, 2021, pp. 3076-3083.
- [7] N. S. Raj, V. Gobbur, P. Praveen, R. Patil, and V. Naik, "Implementing Hand Gesture Mouse Using OpenCV," in International Research Journal of Engineering and Technology, 2020, pp. 4257-4261.
- [8] U. Sneha, B. Monika, and M. Ashwini, "Cursor Control System Using Hand Gesture Recognition," in International Journal of Advanced Research in Computer and Communication Engineering, 2013, pp. 2278-2281.
- [9] M. Krishnamoorthi, S. Gowtham, K. Sanjeevi, and R. V. Revanth, "Virtual Mouse Using YOLO," in International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems, 2022, pp. 1-7.
- [10] K. S. Varun, I. Puneeth, and T. P. Jacob, "Virtual Mouse Implementation Using OpenCV," in International Conference on Trends in Electronics and Informatics, 2019, pp. 435-438.
- [11] F. Quek et al., "Towards a Vision-Based Hand Gesture Interface," in Proceedings of Virtual Reality Software and Technology, 1994, pp. 17-31.
- [12] R. M. Tharsanee, R. S. Soundariya, A. S. Kumar, M. Karthiga, and S. Sountharajan, "Deep Convolutional Neural Network-Based Image Classification for COVID-19 Diagnosis," in Data Science for COVID 19, Academic Press, 2021, pp. 117-145.
- [13] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in European Conference on Computer Vision, Springer, Cham, 2016, pp. 483-499.
- [14] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh, "Pose Machines: Articulated Pose Estimation via Inference Machines," in European Conference on Computer Vision, Springer, Cham, 2014, pp. 33-47.
- [15] G. Tharani, R. Gopikasri, R. Hemapriya, and M. Karthiga, "Gym Posture Recognition and Feedback Generation Using Mediapipe and OpenCV," in International Journal of Advance Research and Innovative Ideas in Education, 2022, pp. 2053-2057.
- [16] K. H. Shibly, S. Kumar, M. A. Islam, and S. I. Showrav, "Design and Development of Hand Gesture-Based Virtual Mouse," in International Conference on Advances in Science, Engineering and Robotics Technology, 2019, pp. 1-5.
- [17] V. K. Sharma et al., "Virtual Mouse Control Using Hand Class Gesture," GIS Science Journal, 2020.

- [18] K. H. Shibly, S. Kumar, M. A. Islam, and S. I. Showrav, "Design and Development of Hand Gesture-Based Virtual Mouse," in International Conference on Advances in Science, Engineering and Robotics Technology, 2019, pp. 1-5.
- [19] V. K. Sharma et al., "Virtual Mouse Control Using Hand Class Gesture," GIS Science Journal, 2020.
- [20] E. Sankar et al., "Virtual Mouse Using Hand Gesture," IJSREM, vol. 07, no. 05, May 2023.
- [21] J. Prithvi et al., "Gesture Controlled Virtual Mouse with Voice Automation," IJERT, vol. 12, no. 04, April 2023.
- [22] B. K. R. Sandra et al., "Gesture Control-Virtual-Mouse," IRJMETS, vol. 4, no. 4, April 2022.
- [23] D. Rathod et al., "Mouse Control Using Hand Gesture Recognition," IJRPR, vol. 4, no. 3, March 2023.
- [24] A. V. Karthik et al., "Virtual Mouse," IJRPR, vol. 4, no. 10, October 2023.
- [25] S. Shaikh et al., "Gesture Recognition Based Virtual Mouse and Key board," IJPREMS, vol. 3, no. 5, May 2023.
- [26] T. Geetha et al., "Human Computer Interaction Using Hand Gesture," IJAEM, vol. 3, no. 5, May 2021.
- [27] I. Grishchenko and V. Bazarevsky, "MediaPipe Holistic: Simultaneous Face, Hand and Pose Prediction on Device," Google Research, USA, 2020.
- [28] D. Gupta et al., "An Interactive Computer System with Gesture-Based Mouse and Keyboard," 2020.
- [29] S. Nasr-Esfahani et al., "Hand Gesture Recognition for Contactless Device Control in Operating Rooms," arXiv preprint arXiv:1611.04138, 2016.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details