International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

Next-gen interaction experience using virtual mouse system

Naga Sesha Lakshmi Pokanati [1], Satvik Reddy Alla [2, *], Sai Raja Pradeep Pampana [2], Ashan Mohammad [2], Swamy Sakala [2] and Veera Venkata Naga Surya Devi Sai Chintam [2]

[1] Assistant Professor, Department of Computer Science and Engineering (CSE), Aditya College of Engineering and Technology, Surampalem - Pin 533437, Andhra Pradesh, India.

[2] UG Student, Department of Computer Science and Engineering (CSE), Aditya College of Engineering and Technology, Surampalem - Pin 533437, Andhra Pradesh, India.

International Journal of Science and Research Archive, 2025, 15(01), 348-354

Publication history: Received on 19 February 2025; revised on 29 March 2025; accepted on 01 April 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.15.1.0899

Abstract

The Virtual Mouse System is an innovative, touch-free control mechanism that replaces traditional computer mice using computer vision and artificial intelligence (AI). With a standard camera or webcam, it interprets hand, head, and eye gestures to perform actions like cursor movement, clicking, and scrolling. Designed for both general users and individuals with physical disabilities, it provides an intuitive, accessible, and futuristic interaction method. Key technologies like OpenCV and AI enable real- time gesture recognition, making it suitable for applications in accessibility, gaming, and education. This system eliminates physical devices, offering convenience and a modern dimension to human-computer interaction. This project focuses on developing a Virtual Mouse System using Python, MediaPipe, and PyAutoGUI, enabling users to control a computer cursor with simple hand gestures. The system leverages MediaPipe's Hand Tracking API to detect hand landmarks and recognize specific gestures, which are then mapped to mouse actions such as cursor movement, clicking, dragging, and scrolling.

Keywords: Virtual Mouse; Gesture Recognition; Computer Vision; MediaPipe; Human-Computer Interaction; PyAutoGUI; Accessibility

1. Introduction

AI Virtual Mouse is a software that allows users to give inputs of a mouse to the system without using the actual mouse. To the extreme, it can also be called hardware as it uses an ordinary camera [1]. A virtual muse can usually be operated with multiple input devices, which may include an actual mouse or computer keyboard. The virtual mouse uses a web camera with the help of different image processing techniques [2]. Using figures detection methods for instant Camera access and a user-friendly interface makes it more easily accessible. The system is used to implement a motion- tracking mouse, a physical mouse that saves time and also reduces effort. The hand movements of a user are mapped into mouse inputs. A web camera is set to take images continuously [3].

The requirement for simple and effective input methods grows as computer devices importance in our everyday lives continues to grow. Input technologies like touchscreens and voice recognition have become increasingly sophisticated and practical in recent years, replacing the conventional mouse and keyboard arrangement, which has been the norm for decades. But even these cutting-edge input techniques have their limitations, especially when it comes to circumstances in which making physical touch with a device is either impractical or undesirable.

Gestures such as finger pinching or tapping are detected and converted into mouse clicks, while palm movement controls cursor navigation. The creation of a virtual mouse that enables users to operate an on-screen pointer using just

* Corresponding author: A Satvik Reddy

their hands and eyes is examined in this research study. The virtual mouse is designed to be precise and dependable, giving users an easy and practical input technique that can be applied in several situations. The algorithms utilized to track hand and eye motions, the user interface, and the feedback mechanisms for the virtual mouse will all be covered in length in this article. We will also go through the outcomes of our testing, which show how useful the virtual mouse is in several situations.

Strategies for Implementing the Next-Gen Virtual Mouse System

- Implementing a Virtual Mouse in Python involves using MediaPipe for hand tracking, OpenCV for image processing, and PyAutoGUI for controlling the cursor. The system captures real-time hand movements via a webcam, detects gestures, and translates them into mouse actions like clicking, scrolling, and dragging, enabling touchless interaction.
- The Next-Gen Virtual Mouse System requires tools like Python for development, OpenCV for image processing, and MediaPipe for hand tracking. PyAutoGUI enables cursor control, while NumPy and Matplotlib assist in data handling and visualization. A webcam is essential for capturing gestures, and an IDE like PyCharm or VS Code facilitates coding and debugging.
- The next step is system design and architecture development. This involves defining the system flow, designing the modules, and determining the interaction between components. The technology stack is finalized, and a prototyping phase ensures feasibility.
- Next, the coding and implementation phase begins, where core functionalities like gesture recognition, cursor control, and UI design are developed. The system undergoes unit testing to verify individual modules, followed by integration testing to ensure seamless operation. Finally, user testing and performance evaluation refine the system, preparing it for deployment and future enhancements based on feedback.

2. Literature review

Several efforts have been made to develop systems capable of recognizing gestures for controlling virtual devices. Zhang et al. [1] introduced a background image differencing technique to segment hand gestures under complex backgrounds. Shetty et al. [2] designed a virtual mouse system using object tracking through HSV color filtering to detect color-marked fingers. Vasanthagokul et al. [3] proposed a virtual mouse designed for accessibility enhancement by integrating gesture recognition for mouse operations.

Ashish Mhetar et al. [4] developed a simple virtual mouse application using color caps and webcam-based tracking for cursor control and clicking. Devanshu Singh et al. [5] utilized OpenCV to design a hand gesture-based virtual mouse that could identify finger positions to simulate mouse actions. Shibly et al. [6] combined hand segmentation and contour analysis to implement gesture-based pointer movement.

In recent developments, Jiang et al. [7] leveraged depth information to build a robust real-time hand gesture recognition system. Lee and Han [8] utilized convolutional neural networks to identify key poses in gesture sequences. Park et al. [9] proposed a human-machine interaction system with gesture recognition focused on high responsiveness and low latency. Chang et al. [10] presented a fingertip detection algorithm for precise real-time interaction.

Gesture-based human-computer interaction has been extensively researched in recent years, particularly with advancements in computer vision and machine learning. Earlier systems often relied on external hardware such as infrared sensors, accelerometers, or colored gloves to track hand motion. For example, Garg et al. proposed a glove-based system using accelerometers to detect finger orientation for mouse control [1]. However, such systems limited user flexibility and introduced hardware dependency. With the emergence of deep learning and real-time hand-tracking frameworks, non-intrusive methods have gained prominence. MediaPipe by Google offers a robust framework for 3D hand tracking using only a standard webcam, significantly reducing system complexity and cost [7].

Researchers such as Mitra and Acharya emphasized the importance of intuitive gesture control in virtual environments, proposing static and dynamic gesture recognition algorithms using contour-based methods [1]. More recent work by Zhang et al. applied convolutional neural networks (CNNs) to improve gesture classification accuracy in real time [2]. Head and eye tracking techniques have also evolved, with Dlib and OpenCV enabling precise facial landmark detection without additional sensors [3][4]. These technologies have been leveraged in accessibility applications, gaming, and assistive devices, showing potential in creating fully touchless computer interfaces [5].

3. Methodology

The methodology provides a detailed explanation of the techniques used for each system component. It outlines the methods and function calls necessary for implementation as proposed in the module.

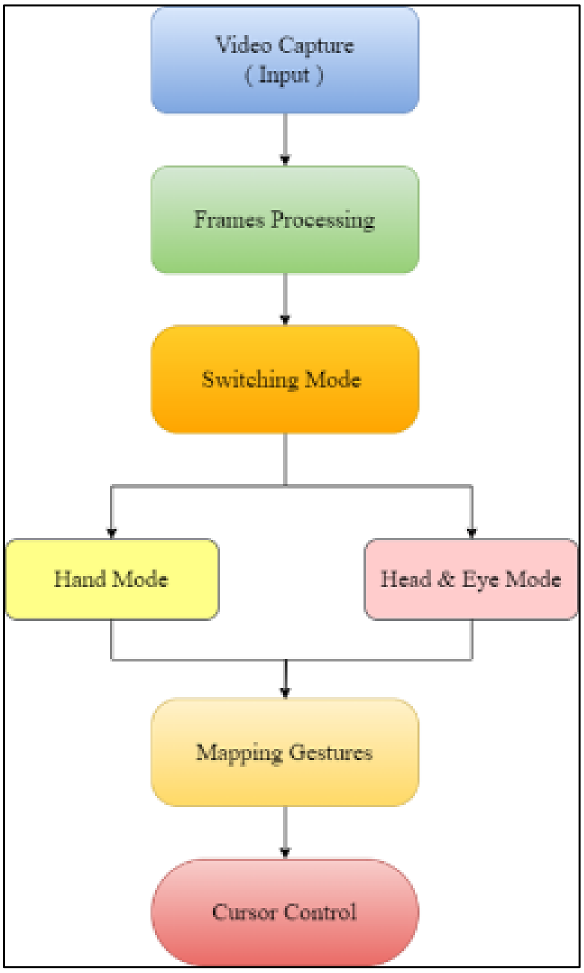

Figure 1 System architecture

3.1. Capturing Eye Movements

A webcam or infrared sensor captures real-time images of the user's eyes. The system continuously monitors pupil position, blinking, and gaze direction.

3.2. Eye Tracking and Detection

- OpenCV and MediaPipe are used to detect facial landmarks, focusing on the eyes and pupils.
- Haar Cascade Classifier or Deep Learning models help recognize eye positions with high accuracy.

3.3. Mapping Nose & Eye Gestures to Mouse Actions

- Looking Left/Right → Moves the cursor horizontally.
- Looking Up/Down → Moves the cursor vertically.
- Long Blink → Simulates a mouse click.
- Squinting → Triggers custom actions like scrolling.

3.4. Processing and Implementing Cursor Control

- PyAutoGUI is used to convert eye gestures into real-time mouse movements.
- The system ensures minimal delay for seamless interaction.

3.5. Testing and Optimization

- The system is tested under various lighting conditions, eye shapes, and user behaviors to enhance accuracy.
- Calibration settings allow users to adjust sensitivity and response speed for better comfort.

3.6. Switching Logic

- The system intelligently switches between eye tracking and hand tracking modes based on user input.
- A predefined gesture (e.g., a specific hand sign or eye movement) toggles between control modes.

- The switching mechanism ensures seamless interaction, allowing users to shift between different control methods without manual intervention.

3.7. Capturing Hand Movements

- A webcam continuously captures real-time video of the user's hand.

- The system processes these frames to track finger positions and hand gestures.

3.8. Hand Detection and Tracking

- MediaPipe Hand Tracking and OpenCV are used to detect hand landmarks (finger joints and palm position).
- The AI model identifies key points on the hand, allowing precise tracking of finger movements.

3.9. Mapping Hand Gestures to Mouse Actions

- Moving index finger while bending thumb → Controls cursor movement on the screen.
- Index Finger bent → Mouse left-click.
- Middle Finger bent → Mouse right-click.

3.10. Processing and Cursor Control Implementation

- Python libraries like PyAutoGUI process the detected gestures and convert them into real-time mouse commands.
- Ensures smooth operation with minimal latency.

3.11. Testing and Performance Optimization

- The system is tested under varied lighting conditions and different hand positions to improve accuracy.
- Users can adjust sensitivity and calibrate the system for better control.

4. Result and Outputs

Figure 2 Hand Mode Recognition

Figure 3 Cursor Movement

Figure 4 Right Click Action

Figure 5 Left Click Action

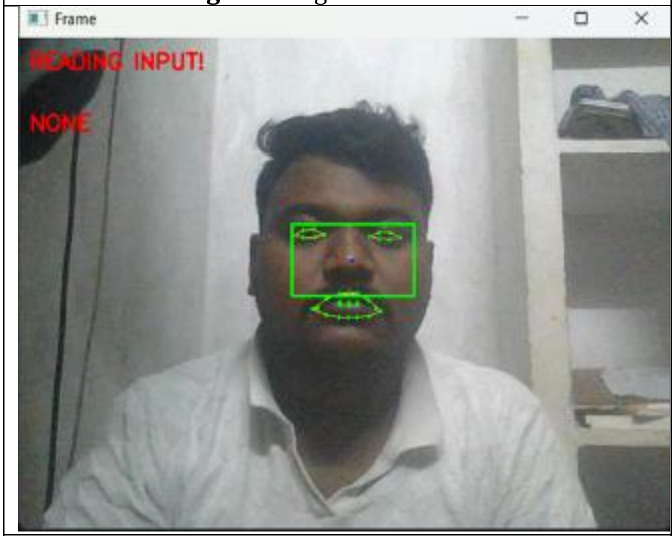

Figure 6 Head & Eye Gesture Recognition

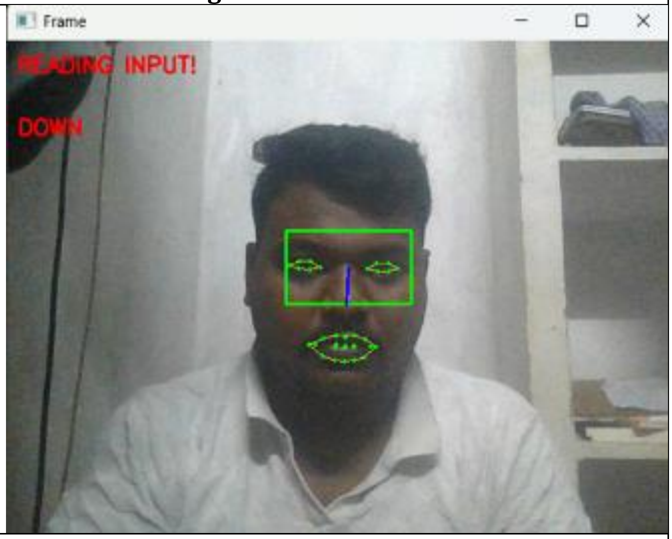

Figure 7 Cursor Moving Downwards

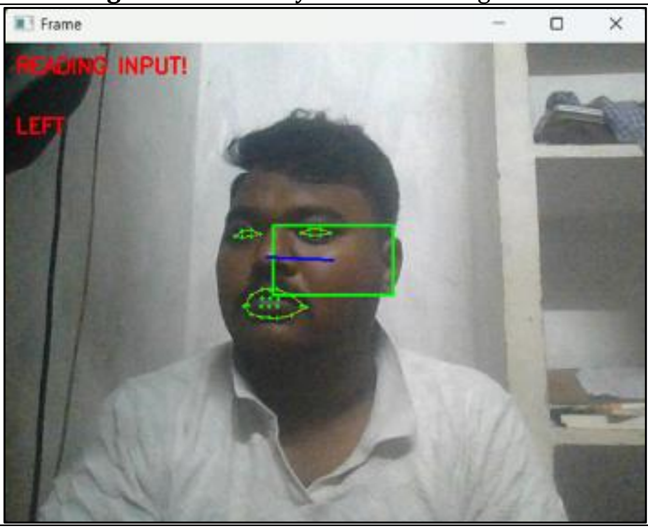

Figure 8 Cursor Moving Leftwards

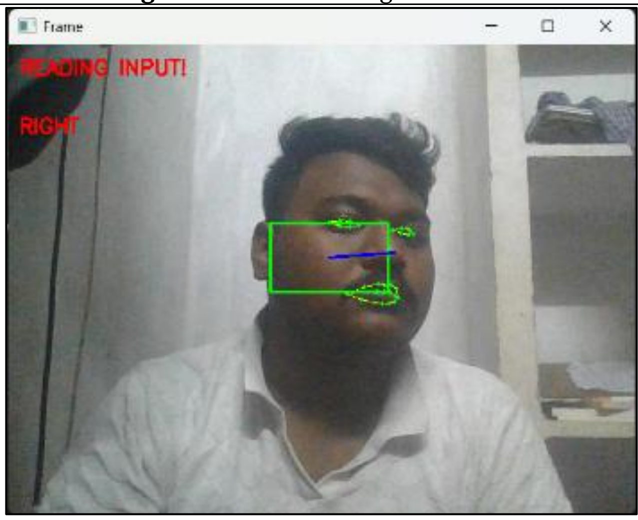

Figure 9 Cursor Moving Rightwards

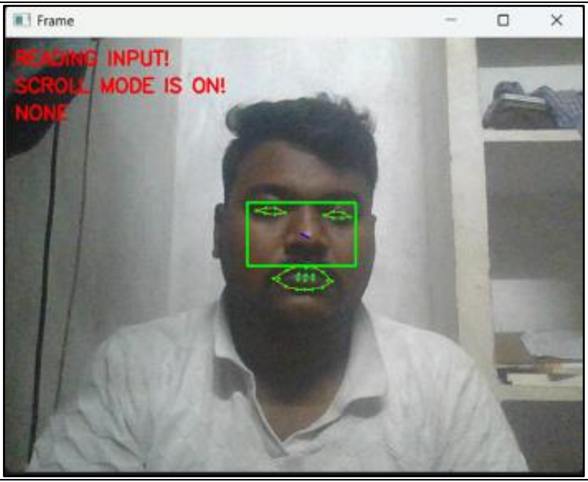

Figure 10 Scroll Mode ON

5. Conclusion and future scope

The Next-Gen Virtual Mouse System using hand gestures provides a touch-free, intuitive, and futuristic way of interacting with computers. By utilizing computer vision and AI-based tracking, users can perform mouse actions such as moving the cursor, clicking, scrolling, and dragging without a physical device. This enhances accessibility for individuals with disabilities and improves user convenience in various fields, including gaming, presentations, and hands-free computing.

In the future, this system can be further enhanced by incorporating eye tracking, voice commands, and AI-driven gesture recognition for greater accuracy and efficiency. Additionally, integrating this technology into AR/VR environments, smart TVs, and IoT devices can expand its usability. Future developments can also focus on reducing processing latency, improving gesture recognition in low- light conditions, and enhancing multi-hand interactions.

Overall, the Virtual Mouse System is a step towards the future of human-computer interaction, making technology more accessible, immersive, and user-friendly.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Q. Y. Zhang, F. Chen, and X. W. Liu, "Hand Gesture Tracking and Fragmentation Based on Difference Background Image with Intricate Background," Deliberations of the 2008 International Conference on Embedded Technology and Systems, Sichuan, 29- 31 July 2008, pp. 338 343

[2] M. Shetty, C. A. Daniel, M. K. Bhatkar and O. P. Lopes, "Virtual Mouse Using Object Tracking," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 548-553, doi: 10.1109/ICCES48766.2020.9137854.

[3] S. Vasanthagokul, K. Vijaya Guru Kamakshi, G. Mudbhari and T. Chithrakumar, "Virtual Mouse to Enhance User Experience and Increase Accessibility," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, 10.1109/ICIRCA54612.2022.9985625

[4] Ashish Mhetar, B K Sriroop, Kavya AGS, Ramanath Nayak, Ravikumar Javali, Suma K V. "Virtual Mouse", Proceedings of IEEE International Conference on Circuits, Communication, Control and Computing (I4C 2014), pp. 69-72, 2014.

[5] Devanshu Singh , Mr. Ayush Singh , Mr. Aniket Kumar Singh Shailesh Chaudhary , Asst. Prof. JayvirSinh Kher, “Virtual Mouse using OpenCV”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol.9 Issue XII pp. 1055-1058, 2021.

[6] Shibly, K. H., Dey, S. K., Islam, M. A., & Showrav, S. I. (2019, May). Design and development of hand gesture based virtual mouse. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5). IEEE

[7] Jiang, H., Zhou, C., Wang, Z., Zhang, Y., & Huang, X. (2020). A Real Time Hand Gesture Recognition System Based on Depth Information. Journal of Sensors,2020,8858632 https://doi.org/10.1155/2020/8858632

[8] Lee, B. K., & Han, S. S. (2017). Real-time hand gesture recognition based on key poses using convolutional neural networks. Journal of Real-Time Image Processing, 14(2), 363-374 doi.org

[9] Park, S., Lim, S., & Kim, S. (2019). A Novel Real- Time Hand Gesture Recognition System for Human– Machine Interface. Sensors, 19(9), 1953. doi.org

[10] Chang, C.-T., Wu, C.-H., & Lin, T.-Y. (2018). A Real-Time Fingertip Detection Method for Human- Computer Interaction.IEEE Access, 6, 60742–60749 doi.org.