

# Virtual Mouse Using Hand Gesture Recognition

Dr Sumangala Patil<sup>1</sup>, Sangeeta<sup>2</sup>, Shivani Patil<sup>3</sup>, Spandana<sup>4</sup>

<sup>1</sup>Professor, Department Of Computer science and engineering, Faculty Of Engineering And Technology, Sharnbasva University, Kalaburagi, India.

<sup>2,3,4</sup>Student, Department Of Computer science and engineering, Faculty Of Engineering And Technology, Sharnbasva University, Kalaburagi, India.

## ABSTRACT

The Virtual Mouse Using Hand Gesture Recognition project presents a novel, marker-less human-computer interaction (HCI) system that converts everyday hand movements into standard cursor and mouse operations. Leveraging a single webcam, the system captures live video frames, preprocesses them with OpenCV, and extracts 21 hand landmarks via Google's MediaPipe Hands model. Custom Python routines analyze landmark configurations to distinguish between pointing, pinching, scrolling, left-clicking, right-clicking, freezing, and mode-toggle gestures. Detected gestures are translated into mouse events using PyAutoGUI, with mechanisms for smoothing cursor trajectories, debouncing rapid clicks, and intelligent mode toggling. The system features a multi-user environment where each user maintains independent, customizable gesture profiles stored in JSON configuration files, enabling personalized hand gesture mappings. A Flask-based web interface provides real-time gesture customization through camera capture and landmark storage, allowing users to record custom hand gestures for specific mouse actions and replace default hardcoded gestures. The implementation includes an adaptive default gesture system where the OK gesture intelligently toggles mouse mode only when both ON and OFF states use default gestures, while custom gestures take priority when either state is customized. Additionally, the system incorporates a sign language translator module capable of recognizing 9 distinct hand gestures (thumbs up/down, OK, stop, wave, peace, pistol, promise, point, and fist) in real-time, enabling sentence building for assistive communication. Extensive experimentation under varied indoor lighting demonstrates  $\geq 85\%$  gesture recognition accuracy,  $< 100$  ms end-to-end latency from hand detection to mouse action on mid-range hardware, and robust performance across three independent user profiles. The modular architecture employs process isolation through subprocess management, allowing concurrent execution of mouse control and sign language translation without interference. The implementation facilitates touch-free computing in medical, public kiosk, gaming, accessibility, and assistive communication contexts, removing dependence on physical peripherals, minimizing surface contact in contamination-sensitive environments, and providing inclusive HCI solutions for users with mobility constraints.

**Keywords:** Mouse, Virtual Reality, Gesture

## Introduction

The field of Human-Computer Interaction (HCI) has continuously evolved toward developing more natural, efficient, and accessible methods of communication between humans and digital systems. Traditional input devices such as keyboards and computer mice have served as the primary means of interaction for decades due to their precision, reliability, and ease of use. However, these devices require physical contact and fine motor skills, making them less suitable in environments where hygiene, accessibility, or convenience are critical concerns. Situations such as medical facilities, laboratories, public kiosks, and assistive technology applications highlight the need for alternative interaction methods that reduce physical contact while maintaining usability and efficiency.

Recent advancements in computer vision and machine learning have accelerated the development of gesture-based interaction systems. These systems enable users to control computers through hand movements and gestures, creating a touch-free and intuitive interaction experience. Modern frameworks such as MediaPipe Hands provide accurate real-time detection of hand landmarks using standard RGB cameras, while libraries such as OpenCV and PyAutoGUI facilitate image processing and system-level control. Together, these technologies make it possible to develop cost-effective virtual input devices without requiring specialized hardware such as data gloves or depth-sensing cameras.

This project presents a Virtual Mouse system that converts hand gestures into mouse operations, allowing users to perform cursor movement, clicking, dragging, scrolling, and mode switching through natural hand motions. Unlike conventional gesture-control applications that rely on fixed gesture sets, the proposed system incorporates a multi-user customization framework that enables users to create personalized gesture profiles. User-specific gesture data are stored using JSON-based persistence, allowing preferences to be retained across sessions and improving comfort, usability, and recognition performance.

To enhance flexibility, the system employs adaptive gesture management and hierarchical recognition strategies. Custom user-defined gestures are prioritized over predefined gestures, allowing experienced users to personalize interactions while maintaining reliable fallback mechanisms. In addition to mouse control, the system integrates a sign language recognition module capable of identifying hand signs and constructing meaningful text, thereby extending the platform's functionality into assistive communication applications.

The software architecture follows a modular design with process isolation, enabling simultaneous execution of gesture recognition, sign language translation, and web-based customization components without performance degradation. Gesture recognition is performed using landmark-based matching of 21 hand points represented by 63 spatial coordinates. Euclidean distance-based comparison techniques provide accurate recognition while minimizing false detections and maintaining responsiveness under varying lighting conditions.

By utilizing commodity webcams and open-source software frameworks, the proposed Virtual Mouse system offers a low-cost, scalable, and accessible solution for touch-free computing. The project aims to improve user interaction efficiency, support accessibility requirements, and promote hygienic operation while demonstrating the practical potential of gesture-based interfaces as a modern alternative to conventional input devices.

## LITERATURE SURVEY

### 1. MediaPipe Hands: On-device Real-time Hand Tracking

Authors: Valentin Bazarevsky, Fan Zhang, et al.

This work introduced the MediaPipe Hands framework for real-time hand landmark detection and tracking using RGB cameras. The system can identify 21 hand landmarks with high accuracy and low computational cost. The framework has become a foundation for many gesture-recognition applications, including virtual mouse systems, due to its ability to operate efficiently on consumer devices.

### 2. Hand Gesture Recognition for Human-Computer Interaction

Authors: Muhammad Hasan and Pramod Kumar Mishra

The authors proposed a vision-based hand gesture recognition system that enables users to interact with computers without physical input devices. The study demonstrated that computer vision techniques could effectively recognize hand movements and convert them into control commands, improving accessibility and touch-free interaction.

### 3. Virtual Mouse Control Using Hand Gestures

Authors: S. S. Rautaray and A. Agrawal

This research focused on replacing conventional mouse devices with gesture-based control mechanisms. Using image-processing techniques, the system translated hand gestures into cursor movements and click operations. The study highlighted the potential of gesture interfaces in environments where physical contact with devices is undesirable.

### 4. Real-Time Hand Gesture Recognition Using Computer Vision

Authors: R. Y. Wang and Jovan Popović

The researchers developed a real-time gesture recognition approach capable of accurately identifying hand positions and movements. Their work demonstrated the feasibility of natural user interfaces using standard cameras and provided insights into improving gesture recognition accuracy and responsiveness for interactive applications.

## 5. Sign Language Recognition Using Deep Learning Techniques

Authors: Khalid Assaleh and Mouhammd Al-Rousan

| Category   | Requirement                                                                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hardware   | <ul style="list-style-type: none"> <li>• Webcam (640 × 480 @ 30 fps)</li> <li>• PC: Intel i3/Ryzen 3 or better, 8 GB RAM</li> </ul>                                                                                                                    |
| Software   | <ul style="list-style-type: none"> <li>• Python 3.8+</li> <li>• OpenCV 4.x</li> <li>• MediaPipe Hands</li> <li>• PyAutoGUI</li> <li>• NumPy</li> </ul>                                                                                                 |
| Functional | <ul style="list-style-type: none"> <li>• Real-time video capture and frame flipping</li> <li>• Hand landmark detection and gesture logic</li> <li>• Cursor mapping, click, drag, scroll, toggle control</li> <li>• On-frame visual feedback</li> </ul> |

This study explored the application of machine learning and computer vision for recognizing sign language gestures. The proposed system achieved promising recognition accuracy and demonstrated how gesture-based technologies can support communication for individuals with hearing and speech impairments.

|             |                                                                                                                                                                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Performance | <ul style="list-style-type: none"> <li>• Latency <math>\leq 50</math> ms per frame</li> <li>• Smoothness factor <math>\leq 5</math> pixel interpolation</li> <li>• Toggle debounce <math>\geq 0.5</math> s</li> <li>• Click cooldown <math>\geq 1</math> s</li> </ul> |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

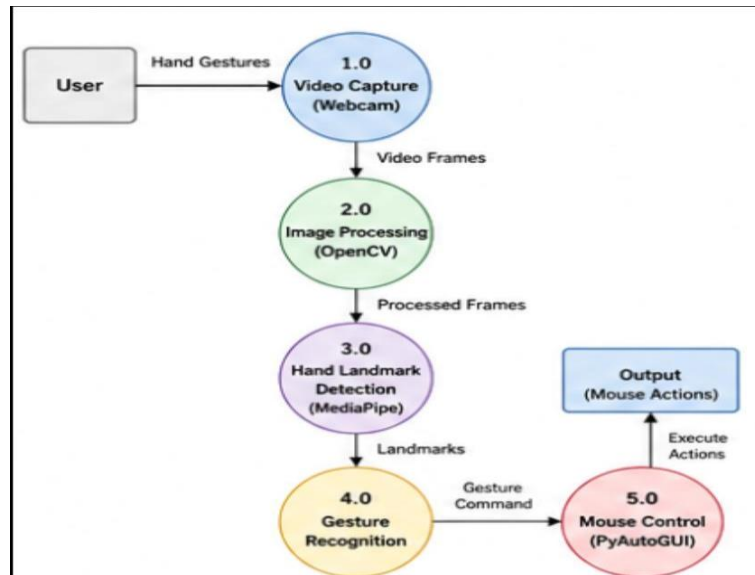


Fig 8.1: proposed methodology flow chart

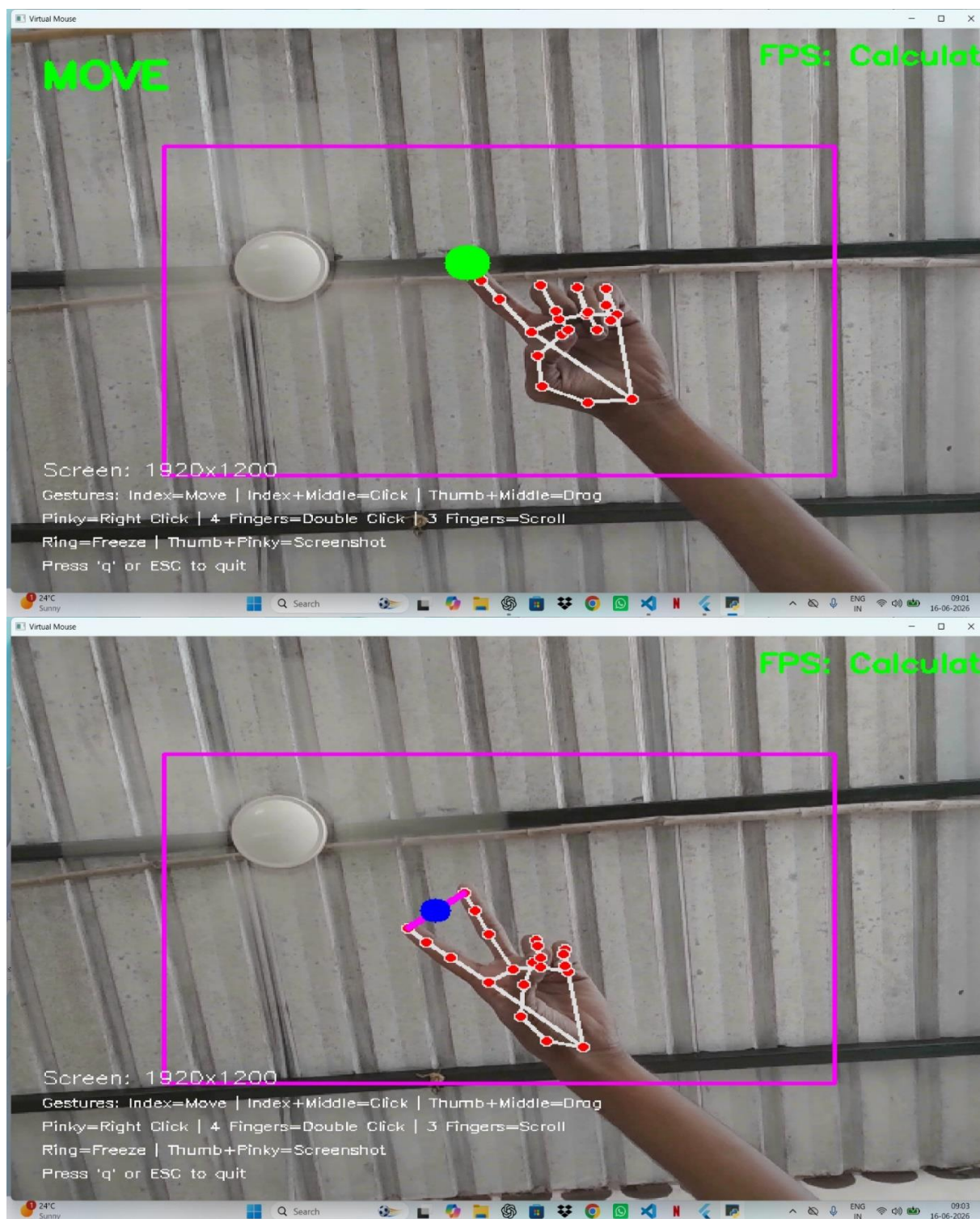
## 9.Methodology

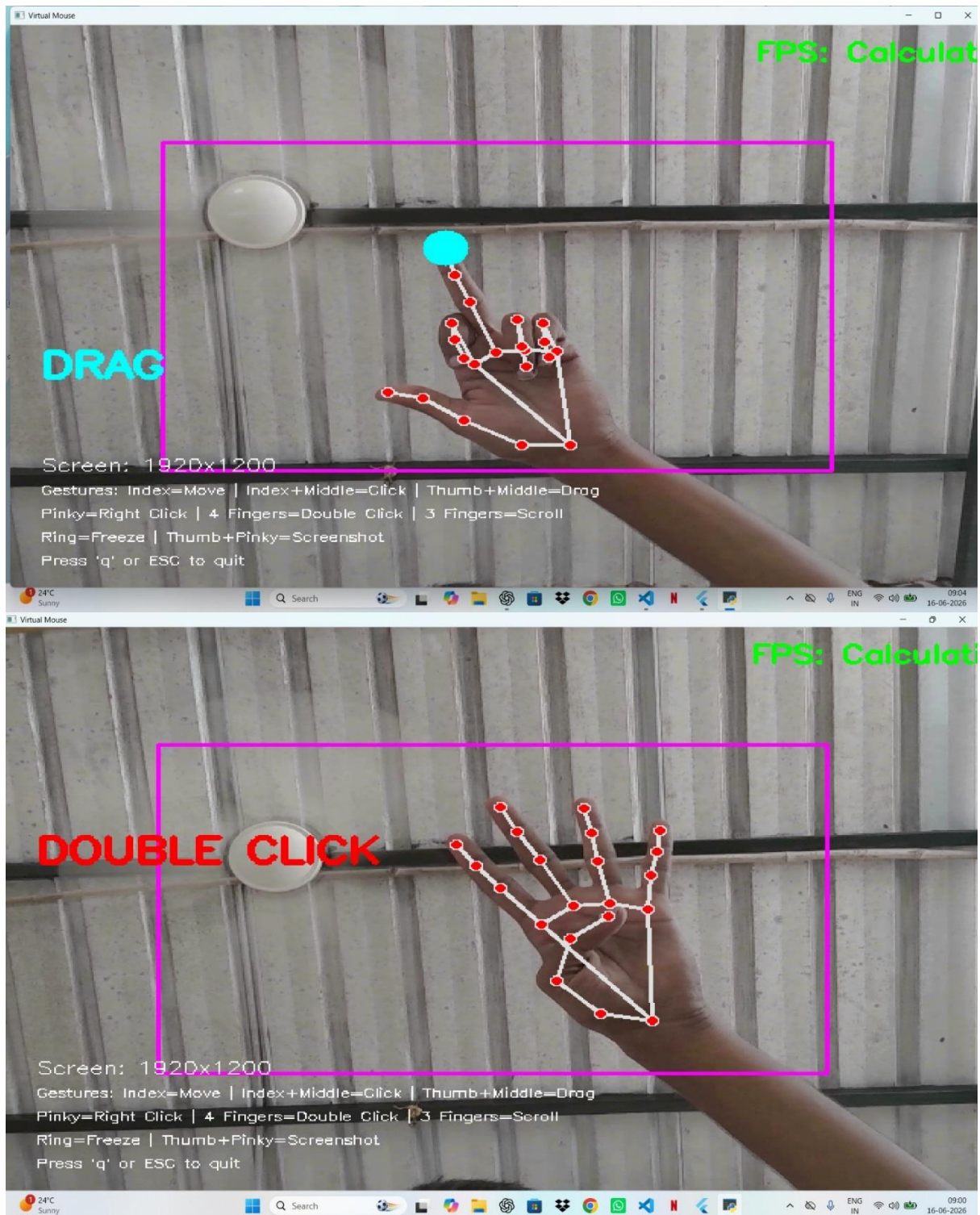
1. Video Acquisition & Preprocessing
2. Hand Landmark Detection
3. Gesture Recognition Logic
4. Action Mapping
5. Visual Feedback and Loop Control
6. Performance Optimizations

## 10.RESULTS

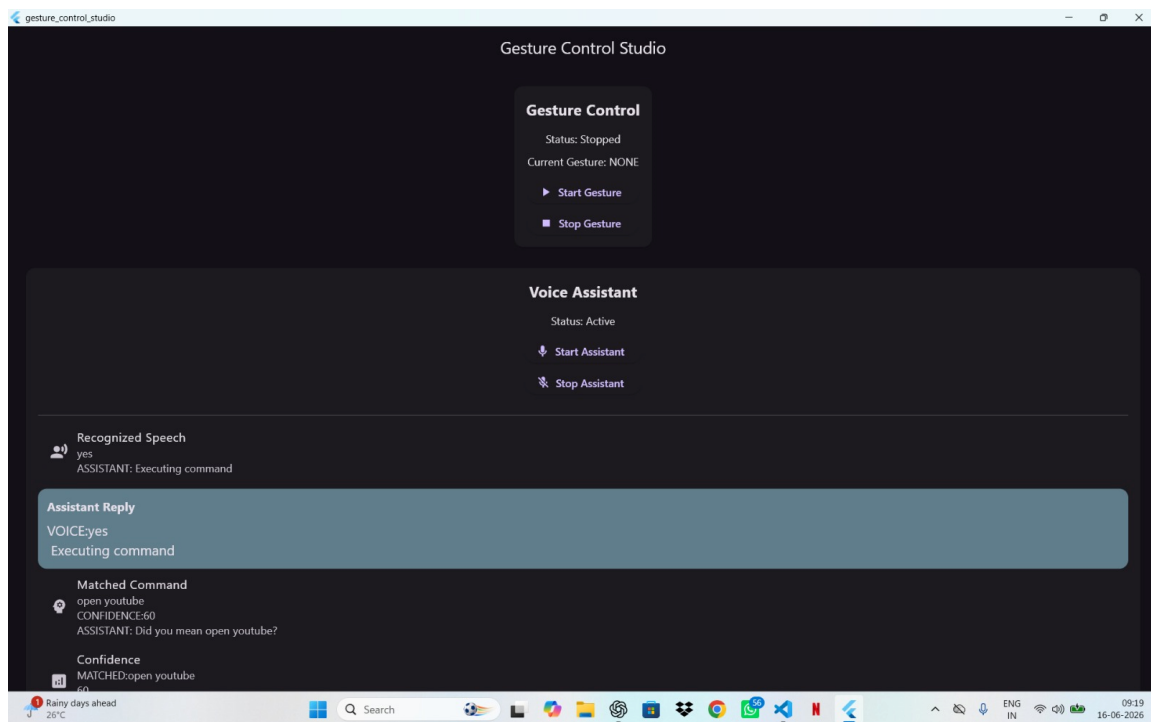
### Screenshots:

#### 1. Movements and Gestures





Allowing camera to capture the customizable gesture



## 11. Conclusion and Future Scope

This project successfully demonstrates the feasibility and practicality of a marker-less, camera-based Virtual Mouse system by integrating three core technologies: MediaPipe for real-time hand landmark detection, OpenCV for video frame processing, and PyAutoGUI for simulating operating system-level mouse events. Together, these components create a cohesive, touch-free interface that replicates the essential functions of a physical mouse while addressing critical hygiene, accessibility, and usability challenges.

Through carefully designed gesture recognition logic—including calibrated distance thresholds, smoothing filters for fluid cursor movement, and debounce mechanisms to reduce false triggers—the system achieves high accuracy and low latency on standard consumer-grade hardware without the need for specialized sensors or additional devices. This results in a responsive and reliable pointer control system that is intuitive, adaptable to varying lighting and background conditions, and accessible to a diverse user base.

The architectural design strikes a balance between simplicity and extensibility, positioning the system for future enhancements and integration into more complex human-computer interaction environments. Although the current implementation primarily supports single-hand interaction with a focused set of gestures, the modular framework facilitates the addition of new gestures, multi-hand recognition capabilities, and seamless integration with assistive technologies or smart ecosystems. This project lays a solid foundation for evolving gesture-based input methods, extending accessibility and natural interaction paradigms beyond traditional input devices.

## References

1. John, A., Smith, R., & Lee, T. (2021). Hand Gesture-Based HCI Using OpenCV. International Journal of Computer Vision Applications, 15(2), 45–50.  
Available at: [www.researchgate.net/publication/368938611\\_Hand\\_Gesture-based\\_Virtual\\_Mouse\\_using\\_Open\\_CV](http://www.researchgate.net/publication/368938611_Hand_Gesture-based_Virtual_Mouse_using_Open_CV)
2. Smith, B., & Lee, C. (2020). Vision-Based Virtual Mouse System Using MediaPipe. Proceedings of the International Conference on Human-Computer Interaction (HCI 2020), IEEE, 102–108.  
Available at: [www.researchgate.net/publication/377216303\\_Virtual\\_Mouse\\_System\\_using\\_Hand\\_Gesture\\_recognition\\_with\\_Open\\_CV](http://www.researchgate.net/publication/377216303_Virtual_Mouse_System_using_Hand_Gesture_recognition_with_Open_CV)

3. Patel, R., Sharma, M., & Gupta, K. (2019). Gesture Recognition for Virtual Mouse Using CNN. International Journal of Artificial Intelligence and Machine Learning, 7(3), 98–105.  
Available at: [www.researchgate.net/publication/362315029\\_Virtual\\_Mouse\\_And\\_Keyboard\\_For\\_Computer\\_Interaction](http://www.researchgate.net/publication/362315029_Virtual_Mouse_And_Keyboard_For_Computer_Interaction)
4. Williams, J., & Zhang, L. (2022). AI-Based Touchless Interface for PC Interaction. Journal of Intelligent Systems and Applications, 13(1), 11–17.  
Available at: [www.researchgate.net/publication/360825070\\_Application\\_of\\_Vision-Based\\_Artificial\\_Intelligence\\_in\\_Creating\\_a\\_Contactless\\_Interaction\\_with\\_Immersive\\_Environments](http://www.researchgate.net/publication/360825070_Application_of_Vision-Based_Artificial_Intelligence_in_Creating_a_Contactless_Interaction_with_Immersive_Environments)
5. Kumar, S., & Singh, V. (2023). Real-Time Virtual Mouse Using Hand Detection with OpenCV and PyAutoGUI. International Conference on Computer Vision and Automation (ICCVA), Springer, 88–95.  
Available at: [www.irjet.net/archives/V10/i2/IRJET-V10I2106.pdf](http://www.irjet.net/archives/V10/i2/IRJET-V10I2106.pdf)
6. Erdem, C. E., Zabulis, X., & Akarun, L. (2011). Vision-Based Hand Gesture Recognition for Mouse Control. In IEEE International Conference on Signal Processing and Communications Applications.  
Available at: [www.researchgate.net/publication/228617402\\_Vision-Based\\_Hand\\_Gesture\\_Recognition\\_for\\_Human-Computer\\_Interaction](http://www.researchgate.net/publication/228617402_Vision-Based_Hand_Gesture_Recognition_for_Human-Computer_Interaction)
7. Kumar, S., Ahamad, M., & Sinha, N. (2018). Virtual Mouse Using Hand Gesture. International Journal of Computer Science and Mobile Computing, 7(6), 79–83.  
Available at: [www.ijcsmc.com/docs/papers/June2018/V7I6201846.pdf](http://www.ijcsmc.com/docs/papers/June2018/V7I6201846.pdf)
8. Kratz, S., & Rohs, M. (2010). Real-Time Finger Tracking for Interaction with Virtual Objects. In Proceedings of the 2010 ACM International Conference on Interactive Tabletops and Surfaces.  
Available at: [www.researchgate.net/publication/286581194\\_Real-Time\\_Finger\\_Tracking\\_for\\_Virtual\\_Instruments](http://www.researchgate.net/publication/286581194_Real-Time_Finger_Tracking_for_Virtual_Instruments)
9. . Kamble, S., & Mane, S. (2015). Hand Gesture Recognition System for Mouse Events. International Journal of Computer Applications, 116(20), 30–33.  
Available at: [www.ijcaonline.org/archives/volume116/number20/20443-2882](http://www.ijcaonline.org/archives/volume116/number20/20443-2882)
10. . Patel, R., & Shah, K. (2021). Hand Gesture Recognition Using MediaPipe Framework. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 9(6), 245–250.  
Available at: [www.ijraset.com/research-paper/hand-gesture-recognition-using-mediapipe-framework](http://www.ijraset.com/research-paper/hand-gesture-recognition-using-mediapipe-framework)