

Virtual Cursor Control Using Eye Tracking and Hand Gestures

SARIYA ANJUM¹, PRITHVIK S GOWDA², SOMASHEKHAR P S³, UDAY GOWDA N S⁴,
YASHAS P GOWDA⁵

¹Asst Professor, Department of Computer Science & Engineering Mysuru Royal Institute of Technology,
Mandya, Karnataka, India.

^{2,3,4,5}Student, Department of Computer Science & Engineering, Mysuru Royal Institute of Technology,
Mandya, Karnataka, India.

Abstract- Human-Computer Interaction (HCI) has evolved significantly with the advancement of Artificial Intelligence and Computer Vision technologies. Traditional input devices such as mouse and touchpads require physical interaction, which may not be suitable for individuals with physical disabilities or in touchless environments. This paper presents a real-time Virtual Cursor Control system using eye tracking and hand gestures. The system uses a standard webcam to detect iris movement for cursor navigation and hand gestures for performing mouse operations such as clicking, scrolling, and dragging. Eye tracking is implemented using MediaPipe FaceMesh, while hand gesture recognition is achieved using MediaPipe Hands. The system eliminates the need for specialized hardware, making it cost-effective and accessible. Experimental results demonstrate high accuracy (up to 90–95% for gesture recognition and ~85–90% for eye tracking) with minimal latency, making it suitable for real-time applications in accessibility systems, healthcare environments, and touchless computing.

Keywords: Human Computer Interaction, Eye Tracking, Hand Gesture Recognition, Virtual Cursor, MediaPipe, Computer Vision, Artificial Intelligence, Touchless Systems

I. INTRODUCTION

Human-Computer Interaction has transitioned from traditional input devices to more natural interaction methods such as gestures and gaze tracking. Conventional devices like mouse and keyboard require physical interaction, which can be inconvenient, unhygienic, or inaccessible for certain users. Recent developments in computer vision and machine learning enable systems to interpret human gestures and eye movements in real time. This paper presents a Virtual Cursor Control System that

combines eye tracking and hand gesture recognition to control a computer without physical contact.

Objectives

1. To develop a real-time touchless cursor control system
2. To implement eye tracking for smooth cursor movement
3. To recognize hand gestures for mouse operations.
4. To integrate both systems into a unified interface.
5. To provide an affordable and accessible solution

II. METHODOLOGY

The proposed system follows a real-time computer vision-based approach for controlling the mouse cursor using eye movements and hand gestures. The system uses a standard webcam as the primary input device and processes the captured video frames continuously to detect facial and hand landmarks. The complete workflow consists of image acquisition, preprocessing, landmark detection, gesture interpretation, and cursor control execution.

Initially, the webcam captures live video frames, which are processed using the OpenCV library for frame resizing, color conversion, and image enhancement. The processed frames are then passed to MediaPipe FaceMesh and MediaPipe Hands models. The FaceMesh module detects facial landmarks, including iris points, which are used to estimate eye movement direction. Simultaneously, the Hands module detects 21 hand landmarks that help identify finger positions and hand gestures.

The eye tracking module maps iris movement to screen coordinates for smooth cursor navigation. Cursor movement is stabilized using smoothing algorithms to reduce jitter and improve user experience. Hand gesture recognition is performed by calculating the distance and orientation between specific finger landmarks. Different gestures are assigned to mouse operations such as left click, right click, scrolling, and dragging.

The interpreted gestures and gaze directions are converted into system-level commands using the PyAutoGUI library. The entire pipeline operates continuously in real time, ensuring low latency and smooth interaction between the user and the computer system. The proposed methodology eliminates the need for specialized hardware and provides an affordable touchless interaction solution suitable for accessibility systems, healthcare applications, and smart environments.

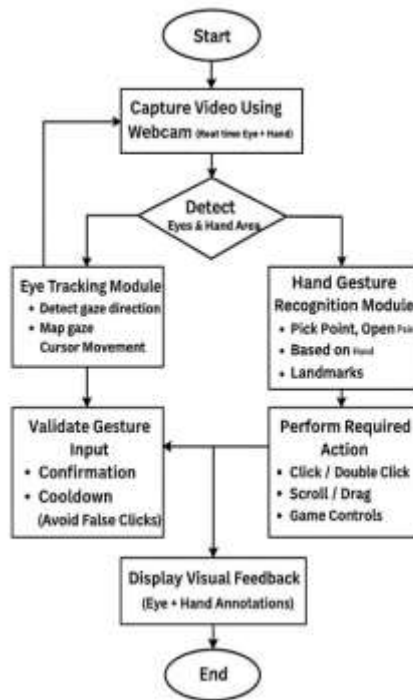


Figure 1: Flowchart

III. MODELING AND ANALYSIS

The proposed virtual cursor control system is designed using a modular architecture consisting of four major layers: Input Layer, Processing Layer, Interpretation Layer, and Execution Layer. The

webcam captures live video frames, which are processed using OpenCV and MediaPipe libraries.

MediaPipe FaceMesh detects facial landmarks and iris positions for eye tracking, while MediaPipe Hands identifies hand landmarks and finger orientations for gesture recognition. The detected eye movements are mapped to screen coordinates for cursor navigation, and hand gestures are interpreted using fingertip distance calculations and threshold-based detection methods to perform mouse operations such as clicking, scrolling, and dragging.

The interpreted actions are converted into system-level commands using the PyAutoGUI library, enabling smooth real-time cursor control. Performance analysis indicates that the system operates at an average speed of 15–25 FPS with low latency ranging from 30–60 milliseconds. Eye tracking achieves approximately 85–90% accuracy, while hand gesture recognition provides 90–95% accuracy under stable lighting conditions. Experimental testing demonstrates stable cursor movement with minimal lag, making the system suitable for efficient and user-friendly touchless human–computer interaction.

Table 1: System Modules and Their Functions

Module	Function
Webcam Input Module	Captures real-time video frames from the webcam
Eye Tracking Module	Detects iris movement and blink actions
Hand Gesture Module	Detects hand landmarks and finger gestures
Gesture Interpretation Module	Interprets gestures into mouse operations
Cursor Control Module	Executes cursor movement and mouse events

Output Display Module	Displays tracking landmarks and gesture feedback
-----------------------	--

The proposed virtual cursor control system is divided into multiple functional modules to ensure efficient and real-time operation. The webcam input module continuously captures live video frames from the user. These frames are processed by the eye tracking and hand gesture modules using MediaPipe and OpenCV libraries. The eye tracking module detects iris landmarks and blink patterns for cursor movement and click operations, while the hand gesture module recognizes finger movements and gestures for scrolling, dragging, and clicking actions.

The gesture interpretation module analyzes the detected landmarks and converts them into specific mouse commands. Finally, the cursor control module executes system-level operations such as cursor movement, clicking, scrolling, and dragging using the PyAutoGUI library. The output display module visually highlights the detected landmarks and gestures, improving usability and interaction feedback.

IV. PROPOSED SYSTEM

The proposed virtual cursor control system is designed to overcome the limitations of existing approaches by integrating eye tracking and hand gesture recognition into a single unified framework. The system utilizes a standard webcam to capture user input and employs MediaPipe-based machine learning models to detect facial and hand landmarks in real time. Eye tracking is used for continuous cursor movement, providing smooth and natural navigation across the screen, while hand gestures are used to perform discrete actions such as clicking, double-clicking, scrolling, and dragging. The combination of these two interaction methods provides greater flexibility and improves overall usability.

The system is implemented using Python and leverages libraries such as OpenCV for image processing, MediaPipe for landmark detection, and PyAutoGUI for controlling the operating system. The

proposed approach eliminates the need for additional hardware, making it affordable and easy to deploy across different platforms. Additionally, the system incorporates smoothing techniques and threshold-based detection to improve stability and reduce unintended actions.

V. EXISTING SYSTEM

Existing virtual cursor control systems primarily rely on either eye tracking or hand gesture recognition individually, resulting in limited functionality and reduced interaction flexibility. Many eye-tracking systems require specialized hardware such as infrared cameras, depth sensors, or wearable devices to achieve accurate gaze estimation. These devices significantly increase system cost and reduce accessibility for general users.

Gesture-based systems using computer vision techniques often depend heavily on lighting conditions and background environments. Variations in illumination, hand orientation, and camera positioning may lead to inaccurate gesture recognition and unstable cursor movement. Some existing systems also support only a limited set of gestures, restricting their usability in practical applications.

Furthermore, several existing approaches are designed for specific domains such as gaming, surveillance, or research purposes rather than general human-computer interaction. The lack of integration between gaze tracking and gesture recognition reduces overall efficiency and user comfort.

The proposed system overcomes these limitations by combining eye tracking and hand gesture recognition within a unified framework using only a standard webcam. The integration of multiple interaction methods improves flexibility, enhances usability, and eliminates the need for expensive external hardware.

Table 2: Comparison Between Existing and Proposed System

Feature	Existing System	Proposed System
Hardware Requirement	Specialized sensors/cameras	Standard webcam
Cost	High	Low
Eye Tracking	Limited	Improved
Gesture Recognition	Basic gestures	Multiple gestures
Real-Time Performance	Moderate	High
Accessibility	Limited	Better accessibility

VI. RESULTS AND DISCUSSION

The proposed system was successfully implemented and tested using a standard laptop webcam and Python-based computer vision libraries. Experimental evaluation demonstrates that the system can effectively perform real-time cursor control using both eye tracking and hand gestures with minimal delay.

The eye tracking module provides smooth cursor movement by accurately estimating iris positions and mapping them to screen coordinates. Cursor stabilization techniques significantly reduce jitter and improve navigation accuracy. The hand gesture recognition module successfully detects gestures such as clicking, scrolling, dragging, and cursor positioning through finger landmark analysis.

The system achieved an average frame processing rate of 15–25 FPS, ensuring smooth real-time interaction without noticeable lag. Experimental observations indicate that eye tracking achieves approximately 85–90% accuracy under stable lighting conditions, while gesture recognition achieves 90–95% accuracy due to reliable hand landmark detection.

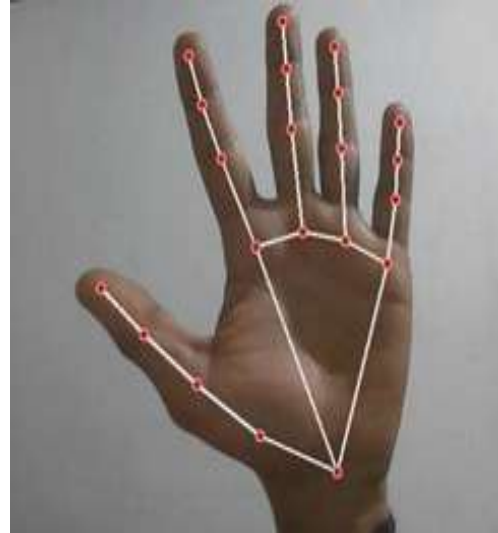


Figure 2: Recognizing hand



Figure 3: Moving the Cursor

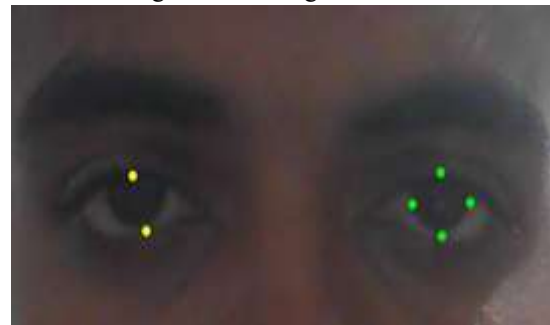


Figure 4: Controlling the Mouse using Eye Gesture

CONCLUSION

The proposed virtual cursor control system demonstrates an effective approach to touchless human-computer interaction by combining eye tracking and hand gesture recognition. The system successfully eliminates the need for physical input devices and provides an accessible, hygienic, and cost-effective solution for modern computing environments. By utilizing computer vision and machine learning techniques, the system achieves real-time performance with high accuracy and responsiveness.

The integration of dual interaction modes enhances flexibility and usability, allowing users to choose the most comfortable method of control. The project highlights the potential of gesture-based interfaces in improving accessibility for individuals with disabilities and enabling innovative applications in areas such as healthcare, gaming, and smart environments. Future enhancements may include the integration of advanced machine learning models for improved gesture recognition, multi-user support, and compatibility with emerging technologies such as augmented and virtual reality.

REFERENCES

- [1] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, vol. 25, no. 11, pp. 120–126, 2000.
- [2] Google MediaPipe, "MediaPipe Hands and FaceMesh Documentation," Google Research, 2024.
- [3] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, vol. 37, no. 3, pp. 311–324, 2007.
- [4] D. Li and D. J. Parkhurst, "Open-Source Software for Real-Time Visible Pupil Tracking," Proceedings of the ACM Symposium on Eye Tracking Research and Applications, pp. 125–128, 2006.
- [5] R. Y. Wang and J. Popović, "Real-Time Hand-Tracking with a Color Glove," ACM SIGGRAPH, pp. 63–72, 2009.
- [6] F. Chollet, *Deep Learning with Python*, Manning Publications, 2018.
- [7] A. Kaushik and R. Jain, "Virtual Mouse Control Using Hand Gesture Recognition," International Journal of Computer Applications, vol. 179, no. 12, pp. 20–25, 2021.
- [8] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518, 2001.