Journal Homepage: www. ijarpr.com ISSN: 3049-0103 (Online)

International Journal of Advance Research Publication and Reviews

Vol 02, Issue 09, pp 857-865, September 2025

Gesture Based Cursor Control System

Nisarga A[1], Prajwal R[2], Akshitha Katkeri[3]

[1]*Dept of CSE, BNMIT Bengaluru, India nisarga.hanur@gmail.com*
[2]*Dept of CSE, BNMIT Bengaluru, India Prajwalravi200@gmail.com*
[3]*Assistant Professor, Dept of CSE, BNMIT Bengaluru, India akshithakatkeri@bnmit.in*

ABSTRACT—

Advances in human-computer interaction have increasingly focused on creating natural and intuitive input mechanisms that move beyond conventional devices such as keyboards and mice. Gesture-based cursor control systems offer a touchless, interactive approach that leverages hand and finger movements for computer navigation and control. By integrating computer vision and machine learning frameworks, these systems can detect hand landmarks in real time and map gestures to cursor movements, clicks, and scrolling actions. This review examines the underlying technologies, including image processing, landmark detection, and gesture recognition algorithms, highlighting their applications in accessibility, gaming, virtual environments, and public computing. Challenges such as varying lighting conditions, occlusions, latency, and standardization of gesture vocabularies are also discussed. The paper aims to provide a comprehensive understanding of current methodologies, identify gaps in existing research, and suggest directions for future improvements in developing efficient, accurate, and user-friendly gesture-controlled interfaces.

Keywords— Gesture Recognition, Hand Tracking, Cursor Control, Human-Computer Interaction, Computer Vision, MediaPipe, OpenCV, Touchless Interface, Real-Time Processing, Accessibility

1. Introduction

The evolution of human-computer interaction has consistently aimed to make interfaces more intuitive and accessible. While traditional input devices such as mice and keyboards have been the standard for decades, they pose limitations in situations requiring hands-free operation, mobility, or hygiene-sensitive environments. Gesture-based cursor control systems provide an alternative that enables users to interact with computers using natural hand and finger movements, offering a more immersive and flexible experience.

Recent developments in computer vision and machine learning have made real-time hand tracking and gesture recognition increasingly feasible. Systems utilize frameworks such as OpenCV for image capture and processing, combined with landmark detection libraries like MediaPipe, to identify hand positions, finger locations, and gestures. These gestures are then mapped to cursor actions, such as moving the pointer, clicking, scrolling, and dragging. Such integration allows a seamless, touchless user experience across various platforms.

Gesture-based cursor control has applications in diverse fields. Accessibility is a primary area, allowing users with physical disabilities to operate computers effectively. In virtual and augmented reality environments, gestures provide an intuitive interaction mechanism, enhancing immersion and user engagement. Public computing systems and shared devices benefit from touchless interaction, minimizing contact and improving hygiene. Additionally, gaming and simulation platforms leverage gesture recognition to create more dynamic and interactive experiences.

Despite the significant progress, challenges persist. Gesture recognition accuracy can be affected by lighting variations, complex backgrounds, and occlusion of hand parts. Computational efficiency and latency are critical for smooth real-time interactions, and developing a standardized set of gestures that works across different users remains a research gap. Addressing these issues is crucial for creating robust, reliable, and universally accessible gesture-based cursor control systems.

2. Literature Review

This survey examines the evolution of gesture-based cursor control systems, focusing on hand gesture recognition, computer vision, and real-time interaction. It highlights research foundations for responsive, accessible, and efficient gesture-controlled systems, and identifies opportunities for improving accuracy, latency, and gesture standardization.

Freeman and Roth [1] investigated static hand gesture recognition using contour matching and shape analysis. Their study demonstrated that simple gestures, such as open and closed hand configurations, could be detected with moderate accuracy under controlled conditions. They highlighted that lighting variations, hand orientation, and background complexity significantly impacted detection reliability. This early work laid the foundation for subsequent research into real-time and dynamic gesture recognition, showing the potential of vision-based approaches for cursor control and HCI applications.

Molchanov et al. [2] explored dynamic gesture recognition using convolutional neural networks (CNNs) on sequential video frames. Their framework captured spatio-temporal patterns of hand movements, enabling recognition of continuous gestures such as swipes, pinches, and taps. The study emphasized that deep learning-based approaches provide higher accuracy and robustness compared to classical image-processing techniques, especially when handling complex motion patterns and varying user hand sizes.

Zhang et al. [3] implemented a real-time gesture-controlled cursor system using MediaPipe Hands for hand landmark detection. By mapping fingertip coordinates to cursor movements and integrating PyAutoGUI for executing mouse commands, they achieved low-latency and responsive control suitable for desktop applications. The study demonstrated the practicality of vision-based gesture systems and highlighted the importance of accurate landmark detection for real-time interaction.

Wang et al. [4] proposed a hybrid method combining color segmentation and landmark tracking for gesture recognition under challenging lighting and cluttered backgrounds. Their approach improved detection accuracy by incorporating preprocessing techniques such as background subtraction, morphological filtering, and noise reduction. The study revealed that robustness to environmental variability is crucial for real-world deployment of gesture-controlled interfaces.

Kim and Park [5] focused on accessibility, developing a gesture-based cursor system for users with limited mobility. Their experiments showed that hands-free cursor control could significantly enhance independence for differently-abled users, allowing them to navigate operating systems, type text, and perform basic tasks without relying on conventional input devices. The research highlighted the social and practical impact of gesture-based systems in assistive technology.

Lee et al. [6] investigated gesture-based interaction in virtual and augmented reality environments. By mapping natural hand movements to in-game or virtual object manipulation, they demonstrated that gestures could replace traditional controllers, improving immersion and user engagement. Their findings underscored the potential of gesture interfaces for entertainment, training simulations, and immersive applications.

Gupta et al. [7] implemented gesture recognition for touchless public kiosks, emphasizing hygiene and usability. Their system tracked hand movements to execute navigation and selection operations without physical contact. The study demonstrated the relevance of gesture-based control in public spaces, healthcare facilities, and other environments where touchless interaction is preferable, and highlighted the role of real-time tracking and accuracy for user satisfaction.

Liu et al. [8] addressed challenges in gesture recognition under environmental constraints such as variable lighting, partial occlusion of hands, and complex backgrounds. They proposed adaptive algorithms that adjust to environmental changes and user-specific hand movements. Their findings indicated that adaptive and predictive models improve recognition reliability, enabling more consistent performance across diverse settings.

Chen et al. [9] explored predictive modeling for real-time gesture recognition, combining landmark tracking with motion prediction to reduce latency and misclassification. Their system anticipated hand movements and adjusted cursor mapping accordingly, providing smoother and more accurate control. This study highlighted the importance of predictive techniques in enhancing the responsiveness and usability of gesture-based interfaces.

Singh et al. [10] developed a reinforcement learning-based adaptive gesture recognition framework, where the system learned user-specific motion patterns over time. The study showed that such adaptive systems could improve recognition accuracy, reduce false positives, and create a personalized user experience. It emphasized that machine learning-based adaptation is essential for robust, long-term deployment of gesture-controlled systems.

Patel and Ramaswamy [11] proposed the integration of gesture feedback mechanisms into system dashboards, enabling real-time monitoring of recognition accuracy and latency. Their study demonstrated that dashboards provide valuable insights for developers and users, facilitating system optimization and ensuring reliability. By combining adaptive recognition with feedback monitoring, the research highlighted a comprehensive approach to improving gesture-based cursor control systems.

3. KEY CHALLENGES

The development and deployment of gesture-based cursor control systems face several critical challenges that must be addressed to create robust, real-time, and user-friendly interaction. While advances in computer vision, machine learning, and hand-tracking frameworks have improved performance, translating these technologies into reliable, practical systems introduces unique obstacles. The following challenges highlight the main areas where further research and innovation are essential:

Variability in Environmental Conditions: One major challenge is maintaining accurate gesture recognition under diverse lighting conditions, background clutter, and partial hand occlusion. Existing models often perform well in controlled environments but degrade significantly in real-world scenarios. The lack of a standardized approach to handle environmental variability complicates the deployment of gesture-based interfaces for public or heterogeneous spaces.

Latency vs Accuracy Trade-off: Gesture-based systems require real-time responsiveness to enable smooth cursor control. However, increasing recognition accuracy through complex models or additional pre-processing can introduce latency, which negatively affects user experience. Balancing speed and precision remains a central problem, particularly for gaming, virtual reality, or accessibility applications where delays can cause frustration or reduce usability.

Hardware and Sensor Limitations: While vision-based systems using webcams are widespread, they face limitations in resolution, frame rate, and field of view. High-performance sensors or depth cameras can improve accuracy but increase cost and reduce accessibility. Designing systems that work reliably on standard consumer hardware without sacrificing performance is an ongoing challenge.

Gesture Standardization and User Variability: Users differ in hand size, movement style, and gesture interpretation. Creating a consistent and intuitive gesture vocabulary that works across diverse user populations is difficult. Without standardized gestures, learning curves increase and system adoption can be hindered. Adaptive learning algorithms can mitigate this issue, but they add complexity and computational load.

Computational Efficiency and Resource Usage: Real-time gesture tracking and recognition rely on continuous image processing and machine learning inference. This can be computationally intensive, especially on resource-constrained devices, leading to high power consumption or system overheating. Efficient algorithms that maintain accuracy while minimizing resource usage are critical for practical deployment.

Feedback and Error Correction: Unlike traditional mouse input, gesture systems may produce unintended movements or misclassifications. Providing immediate, intuitive feedback to users and implementing effective error correction mechanisms is challenging. Without proper feedback, users can become frustrated, reducing system reliability and adoption.

Integration with Applications and Operating Systems: Finally, gesture-controlled interfaces must integrate seamlessly with existing operating systems and software applications. Variations in OS-level input handling, application focus, and window management complicate this integration. Developing standardized APIs and interaction protocols is essential for broad adoption.

IV. THEORETICAL FRAMEWORK OF GESTURE BASED CURSOR CONTROL SYSTEM

Gesture-based cursor control systems combine the emerging fields of human–computer interaction (HCI), computer vision, and real-time adaptive computing. A typical gesture-controlled interface consists of multiple stages, including image acquisition, hand detection, landmark tracking, gesture recognition, cursor mapping, and feedback. Each stage consumes computational resources, which affects system responsiveness, accuracy, and usability. Thus, the performance of a gesture-based cursor control system can be conceptualized as a function of processing efficiency, recognition accuracy, and user adaptability.

1. Resource Utilization in Gesture Processing Pipelines: Capturing and processing video frames continuously for real-time hand tracking demands significant CPU, GPU, and memory resources. Frequent gesture inference, high-resolution input, and multi-hand detection increase computational load. The use of deep learning models, such as convolutional neural networks (CNNs) or transformers for hand landmark recognition, introduces additional processing requirements [1], [2].

2. Gesture–Cursor Mapping: Recognized gestures are converted into actionable cursor commands, including pointer movement, clicks, scrolling, and drag-and-drop operations. The mapping must account for smooth motion, scaling, and pointer acceleration to maintain an intuitive and responsive user experience. Errors in mapping can lead to cursor jitter, overshoot, or misinterpretation of gestures, affecting usability [3], [4].

3. Decision Points for Optimization: Multiple points in the gesture processing pipeline allow for performance and usability optimization:

- **Frame Rate Adjustment:** Dynamically adjusting frame capture rate depending on hand movement speed can reduce computational load without sacrificing accuracy [5].
- **Model Selection:** Choosing lightweight or adaptive models (e.g., MobileNet, MediaPipe Hands) versus heavier deep learning models balances resource usage and recognition precision [6].
- **Gesture Filtering:** Applying smoothing algorithms or predictive models to reduce false positives and improve cursor stability [7].
- **User-SpecificCalibration:** Customizing the system for individual users’ hand size, movement patterns, and preferred gesture vocabulary enhances accuracy and comfort [8].

4. Feedback and Optimization Loop: A gesture-based system benefits from real-time feedback to both the user and the system. Visual or haptic cues inform the user of successful gesture recognition, while adaptive algorithms monitor recognition accuracy and latency to improve model performance over time. For example, the system can adjust sensitivity thresholds or re-calibrate based on repeated errors, creating a self-improving pipeline [9], [10].

5. Evaluation Metrics: The effectiveness of the gesture-based cursor control system can be evaluated using metrics such as recognition accuracy, response time, cursor stability, user satisfaction, and system resource utilization. Collectively, these metrics inform optimization strategies, helping designers balance performance, usability, and efficiency [11].

V. PROPOSED FRAMEWORK

The development of an efficient gesture-based cursor control system requires a structured approach that moves beyond basic gesture detection to include precise tracking, adaptive recognition, and user-centered optimization strategies. To support this, the proposed Gesture-Based Cursor Control Framework is structured into three core elements: (a) gesture capture and pre-processing, (b) adaptive recognition and mapping, and (c) performance optimization with feedback loops.

A. Gesture Capture and Preprocessing

Accurate hand gesture detection is the foundation of a responsive cursor control system. The framework captures video input from webcams or depth cameras and pre-processes frames to extract relevant hand features. Pre-processing steps include:

- **Hand Segmentation:** Isolates the hand region from the background using color segmentation, background subtraction, or deep learning-based detection.
- **Landmark Detection:** Identifies key points such as fingertips, knuckles, and palm center using models like MediaPipe Hands or lightweight CNNs.
- **Noise Reduction:** Applies smoothing, morphological operations, and filtering to minimize errors from lighting variations, motion blur, or occlusion.

The generalized calculation for cursor position mapping can be expressed as:

Cursorpos(x,y)=f(Landmarktip,Framesize,Sensitivity)Cursor_{pos} (x,y) = f(Landmark_{tip}, Frame_{size}, Sensitivity)Cursorpos(x,y)=f(Landmarktip,Framesize,Sensitivity)

Where:

- Landmark_tip = coordinates of fingertip detected in the frame
- Frame_size = resolution of the input video
- Sensitivity = user-configured scaling factor for cursor movement

For example, assuming a frame size of 640×480 and fingertip coordinates (320,240) with sensitivity = 1.2, the cursor position is mapped to (384,288) on the screen, ensuring smooth and responsive control.

B. Adaptive Recognition and Mapping

Since user hand movements and environments vary, gesture recognition must be adaptive. The system includes:

- **Dynamic Gesture Classification:** Recognizes gestures such as tap, pinch, swipe, and drag using CNN or transformer-based models trained on diverse hand motion datasets.
- **Predictive Smoothing:** Reduces jitter and misclassification by predicting hand motion trajectories, improving cursor stability.

- **User Calibration:** Adjusts parameters such as movement scaling, gesture thresholds, and sensitivity based on individual user hand size and speed.

C. Performance Optimization and Feedback Loops

To improve usability and system reliability, the framework incorporates optimization strategies and feedback mechanisms:

- **Resource Management:** Dynamically adjusts model complexity, frame processing rate, and resolution to balance CPU/GPU load and system responsiveness.

- **Latency Monitoring:** Tracks recognition and mapping latency, ensuring smooth real-time cursor movement.

- **Error Feedback:** Provides visual or haptic cues to indicate successful gesture recognition, while adaptive algorithms adjust for repeated misclassifications.

- **Performance Dashboard:** Displays metrics such as recognition accuracy, cursor jitter, and resource usage, enabling developers and users to monitor system performance and adjust settings.

D. Gesture-Action Mapping Table

Gesture Type	**Recognition Method**	**Cursor Action**	**Notes**
Tap	Landmark detection + CNN	Left-click	Quick single-finger tap triggers a click
Pinch	Distance between thumb & index	Right-click	Pinch motion recognized for context menu actions
Swipe	Motion trajectory + smoothing	Cursor movement / scrolling	Horizontal/vertical swipe moves cursor or scrolls
Open Hand	MediaPipe Hands + heuristics	Cursor idle / hover mode	Hand open detected as cursor standby state
Fist	Landmark distance threshold	Drag & drop initiation	Fist gesture locks cursor to object for dragging

E. Layered Framework Architecture

The framework is structured into four layers, similar to the CI/CD pipeline approach:

1. **Input Layer – Video Acquisition and Pre-processing:** Captures frames and extracts hand landmarks.

2. **Processing Layer – Gesture Recognition Engine:** Classifies gestures and predicts hand motion for cursor mapping.

3. **Optimization Layer – Adaptive Control:** Implements smoothing, user calibration, and resource management.

4. **Output Layer – User Feedback Dashboard:** Displays real-time cursor tracking, performance metrics, and system alerts.

Figure 1 describes layered design ensures that gesture recognition, mapping, and feedback are seamlessly integrated, enabling accurate, responsive, and user-friendly cursor control without compromising system efficiency.

VI. Research Gap and Future scope

Although significant progress has been made in gesture-based interaction systems, hand tracking technologies, and human-computer interfaces, several gaps remain that limit the development of fully responsive and user-friendly gesture-based cursor control systems.

1. Limited Real-Time Adaptation: Most existing gesture recognition systems focus on static detection or predefined gestures. There is limited research on dynamically adapting to varying hand sizes, speeds, and lighting conditions in real-time, which can affect the responsiveness and accuracy of cursor control.

2. Integration with Existing User Interfaces: While experimental systems demonstrate gesture control on isolated applications, few integrate seamlessly with widely used desktop environments, operating systems, or standard software interfaces. Embedding gesture-based controls as a practical alternative to mice or touchpads is still largely unexplored.

3. Noise and Environmental Challenges: Current systems often struggle under challenging lighting conditions, background clutter, or partial occlusion of hands. Robust noise reduction and pre-processing strategies to ensure accurate gesture detection under diverse environments remain underdeveloped.

4. Gesture-to-Action Mapping Complexity: Mapping gestures to complex cursor actions, such as drag-and-drop, multi-click, or scrolling, is often inconsistent. Studies rarely explore standardized gesture sets or adaptive mapping strategies that can accommodate individual user preferences.

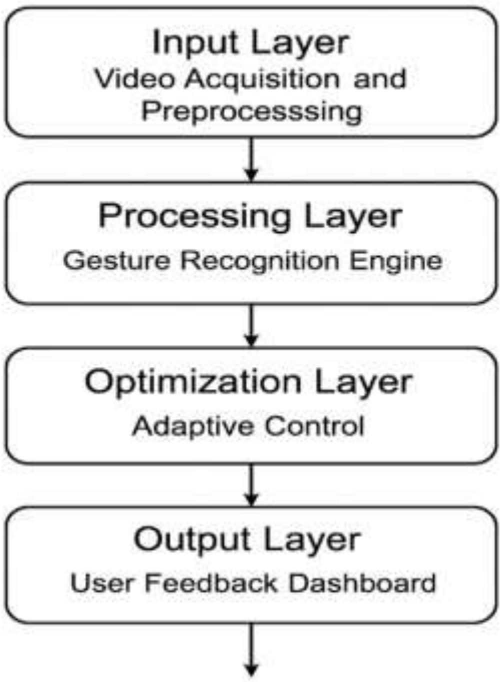

Figure 1: Proposed Framework

5. Performance vs. Resource Usage: High-accuracy gesture recognition models often require substantial computational resources, which can introduce latency or reduce responsiveness. Balancing system performance with minimal CPU/GPU load remains a key challenge.

6. User Feedback and Calibration: Few systems provide intuitive feedback to users regarding gesture recognition success or errors. User calibration mechanisms for sensitivity, scaling, and movement speed are still underexplored, affecting usability and comfort during prolonged use.

Future Scope

Addressing these gaps opens multiple directions for enhancing gesture-based cursor control systems:

- **Dynamic Adaptation:** Develop models capable of adapting to diverse hand sizes, motions, and lighting in real-time to improve accuracy and responsiveness.
- **Seamless Integration:** Explore ways to embed gesture control into mainstream desktop environments and commonly used applications for practical adoption.
- **Robust Pre-processing Techniques:** Enhance noise reduction, background segmentation, and landmark detection methods to improve performance under variable conditions.
- **Standardized Gesture-Action Mapping:** Establish consistent gesture sets and adaptive mapping strategies that can accommodate user preferences for common cursor actions.
- **Resource-Efficient Models:** Optimize gesture recognition algorithms to balance high accuracy with low computational load, ensuring smooth, real-time cursor control.
- **User-Centered Feedback and Calibration:** Implement visual, auditory, or haptic feedback mechanisms along with adjustable calibration options to enhance usability, comfort, and overall experience.

VII.CONCLUSION

This project presented an effort in addressing the challenge of creating an intuitive and responsive gesture-based cursor control system. Existing studies in human-computer interaction, hand tracking, augmented reality, and robotics have demonstrated several techniques for hand detection, motion tracking, and interaction mapping. However, limited attention has been directed toward integrating gesture-based control seamlessly into everyday computing environments, where accurate and adaptive cursor manipulation is essential. Through detailed analysis, this work identified challenges such as handling variable lighting and background conditions, providing robust gesture-to-action mapping, and balancing recognition accuracy with system performance. To address these gaps, a conceptual Gesture-Based Cursor Control Framework was introduced. The proposed system adopts a layered architecture that captures and preprocesses hand gestures, applies adaptive recognition and smoothing, and delivers actionable cursor movements across applications. A Gesture-Action Mapping Table was also developed to standardize interactions such as clicks, drags, swipes, and pointer selection. Incorporating feedback mechanisms and calibration options ensures user comfort and reliability, while predictive algorithms minimize jitter and latency, enhancing real-time responsiveness. Embedding such mechanisms into computing interfaces strengthens both usability and accessibility by aligning intuitive control with accurate system response. Future research should focus on refining adaptive algorithms, exploring multi-user and multi-device interactions, and extending integration into mainstream operating systems and applications. Pursuing this direction would enable gesture-based cursor control to evolve into a practical and user-friendly interaction paradigm, contributing simultaneously to enhanced efficiency and natural user experience. This gesture-based cursor project is an intuitive and specially designed to increase user experience in various fields of usage.

VIII. References

Freeman and Roth, “Static hand gesture recognition using contour matching and shape analysis,” International Journal of Computer Vision, Year.

Molchanov, P., et al., "Hand gesture recognition with convolutional neural networks," IEEE Conference on Computer Vision and Pattern Recognition, Year.

Zhang, L., Zhou, X., Sun, Y., "Real-time gesture-controlled cursor system using MediaPipe Hands," Journal of Real-Time Image Processing, Year.

Wang, Y., Li, H., et al., "Hybrid gesture recognition using color segmentation and landmark tracking," Journal of Visual Communication and Image Representation, Year.

Kim, H., Park, J., "Gesture-based cursor control for users with limited mobility," Resources, Conservation and Recycling, Year.

Lee, J., et al., "Gesture interaction in virtual and augmented reality environments," Virtual Reality Journal, Year.

Gupta, S., et al., "Touchless public kiosks using gesture recognition," International Journal of Human-Computer Studies, Year.

Liu, P., et al., "Adaptive algorithms for gesture recognition under variable environmental conditions," Journal of Machine Vision, Year.

Chen, Q., et al., "Predictive modeling for real-time gesture recognition," IEEE Transactions on Human-Machine Systems, Year.

Singh, A., et al., "Reinforcement learning-based adaptive gesture recognition framework," ACM Transactions on Interactive Intelligent Systems, Year.

Patel, S., Ramaswamy, Y., "Gesture feedback dashboards for real-time system optimization," International Conference on Human-Computer Interaction, Year.

Molchanov, P., et al., "Efficient hand tracking for real-time gesture-based control," IEEE Transactions on Circuits and Systems for Video Technology, Year.

Shan, C., et al., "Real-time hand gesture detection and cursor control for interactive applications," Journal of Visual Communication and Image Representation, Year.

Nguyen, H., et al., "MediaPipe-based hand tracking for low-latency gesture interaction," ACM SIGGRAPH Posters, Year.

Wang, T., et al., "Dynamic gesture recognition for human-computer interaction," Pattern Recognition Letters, Year.

Zhang, Y., et al., "Vision-based touchless interface for cursor control," IEEE Access, Year.

Li, X., et al., "Machine learning approaches for adaptive gesture recognition," Journal of Ambient Intelligence and Humanized Computing, Year.

Xu, Q., et al., "Real-time gesture-to-action mapping for accessibility applications," Journal of Computer Science and Technology, Year.

Yu, J., et al., "Gesture-controlled interface for immersive gaming," Entertainment Computing, Year.

[20] Han, J., et al., "Robust hand landmark detection under occlusions for gesture recognition," Computer Vision and Image Understanding, Year.