Grenze International Journal of Engineering and Technology, June Issue

Gesture and Voice Controlled Virtual Mouse

Purva S Kumbhalwar[1], Prachi M Patle[2], Shrutika Mendole[3], Janvi Hatwar[4], Pravin Jaronde[5] and Rajesh Nakhate[6]
[1-6]Department of Information Technology, G. H. Raisoni College of Engineering, Nagpur, India
Emails: purva.kumbhalwar.it@ghrce.raisoni.net, prachi.patle.it@ghrce.raisoni.net, shrutika.mendole.it@ghrce.raisoni.net, janvi.hatwar.it@ghrce.raisoni.net, pravin.jaronde@raisoni.net, rakesh.nakhate@raisoni.net

***Abstract*— The evolution of Human-Computer Interaction (HCI) has shifted toward touchless and intuitive interfaces, particularly in environments requiring hygiene, accessibility, and immersive interaction. This paper presents a Gesture and Voice Controlled Virtual Mouse, a multimodal system that enables users to operate computer functions through hand gestures and voice commands without physical contact.**

The proposed system is implemented using a standard webcam and microphone with Python-based open-source libraries such as OpenCV, MediaPipe, SpeechRecognition, and PyAutoGUI. The system supports cursor control, clicking actions, scrolling, window management, and system-level commands. Experimental evaluation involving multiple participants demonstrated a gesture recognition accuracy of 92% and voice command accuracy of 89%, with an average response latency of 0.18 seconds. The solution is cost-effective, requires no specialized hardware, and can be deployed on consumer-grade devices. The proposed framework contributes to next-generation touchless interfaces suitable for education, healthcare, assistive technologies, and smart environments.

***Keywords*—Gesture Recognition, Virtual Mouse, Human-Computer Interaction, Speech Recognition, MediaPipe, OpenCV.**

I. Introduction

The rapid evolution of Human-Computer Interaction (HCI) has catalyzed a shift from traditional tactile devices toward intuitive, touchless modalities [1], [10]. As the demand for hygienic and accessible interfaces grows, the integration of real-time computer vision and speech processing has emerged as a robust solution to replace physical peripherals [2], [4]. Early research focused primarily on basic image-processing techniques for cursor navigation [7], [8]; however, these systems often struggled with environmental variables such as lighting and background noise.

To address these limitations, recent advancements in 2024 and 2025 have leveraged deep learning architectures like YOLOv8 and skeletal landmark mapping to achieve sub-pixel precision [13], [14]. Furthermore, the transition toward multimodal systems—combining vision-based gesture control with natural language voice modules—has significantly enhanced system reliability and inclusivity for users with physical impairments [9], [11], [15]. Despite these technological strides, challenges remain regarding computational overhead and scalability on consumer-grade hardware. This research contributes an optimized, lightweight framework that synthesizes high-accuracy gesture recognition with noise-robust voice control, delivering a seamless and immersive touchless computing experience [12], [16].

Grenze ID: 01.GIJET.12.2.21

II. Related Work

The paradigm of Human-Computer Interaction (HCI) has undergone a significant transformation, transitioning from tactile peripheral-based inputs to vision-centric, touchless interfaces. Early foundational work, such as the MotionInput v2.0 framework, established the viability of using consumer-grade cameras to facilitate multimodal touchless control, laying the groundwork for hardware-independent navigation [10]. Subsequent research by Daga et al. and Sutar et al. focused on the fundamental translation of hand geometries into cursor coordinates, utilizing basic color detection and contour mapping to mimic standard mouse functionalities [7], [8]. However, these early iterations often struggled with environmental variables such as ambient lighting and background noise. To address these limitations, researchers shifted toward Python-based implementations using the MediaPipe and OpenCV libraries, which offered more robust fingertip tracking and landmark estimation [2], [5], [6]. The integration of Artificial Intelligence (AI) and Machine Learning (ML) marked a pivotal advancement in the field, providing the high-dimensional data processing required for real-time responsiveness. Singh et al. and Lahoti et al. demonstrated that AI-driven models could significantly enhance gesture recognition accuracy by focusing on skeletal hand mapping rather than simple pixel-based detection [3], [4]. By 2023, the literature began to reflect a trend toward multimodal systems that combined gesture control with voice assistance to create more accessible interfaces.

Rewatkar and Chaware, alongside Singh et al., proposed hybrid architectures where voice commands complemented hand gestures, thereby reducing the cognitive load on the user and providing a failsafe for complex system operations [1], [9], [11].Recent scholarships in 2024 and 2025 have further refined these systems by incorporating deep learning architectures such as YOLOv8 and Convolutional Neural Networks (CNNs). Current state-of-the-art implementations, such as those by Thaqib-ul-Rahman et al. (2025), leverage these models to achieve near-perfect classification accuracy and minimal latency, even in computationally constrained environments [14]. Furthermore, Joseph and Priya (2025) have explored the optimization of these systems for varying luminosity, ensuring that virtual mouse applications remain functional in real-world, non-laboratory settings [12]. The contemporary focus has now shifted toward "Generative AI" and Transformer-based models, as seen in the work of Murali Krishna Raja et al. (2025), which aim to eliminate jitter and provide sub-pixel precision for professional-grade tasks [13]. Collectively, these advancements illustrate a trajectory toward seamless, hygienic, and highly intuitive HCI ecosystems that effectively render traditional physical input devices obsolete [15], [16].

II. Methodology

The proposed system integrates gesture recognition and voice processing into a unified multimodal framework. It consists of four primary modules: input acquisition, processing, action mapping, and feedback. The webcam captures live video while the microphone records voice commands. MediaPipe extracts hand and body landmarks, which are analyzed to identify predefined gestures. Simultaneously, speech recognition converts voice input into executable commands.

Technologies Used: Python, OpenCV, MediaPipe, SpeechRecognition, PyAutoGUI, pyttsx3Gesture

Module: Specific gestures trigger system actions. Raising the left hand performs a left click, raising the right hand performs a right click, and lifting both hands adjusts screen brightness. Hip movement enables window translation, while wrist distance controls resizing. Temporal thresholds reduce false detections.Voice Module: The speech recognition engine processes commands such as “left click,” “scroll down,” and “open Chrome.” The module runs asynchronously to maintain uninterrupted gesture tracking.Feedback Module: Audio responses confirm executed commands, improving usability and accessibility.

III. System Implementation

A. Architectural Overview

The overall implementation follows a four-layer modular architecture, comprising:

Input Acquisition Layer – Captures live audio and video streams.

Processing Layer – Executes landmark detection and voice transcription.

Action Mapping Layer – Translates recognized inputs into system commands.

Feedback Layer – Delivers visual overlays and vocal responses.

Each layer operates independently but communicates through shared variables and asynchronous threads, maintaining real-time responsiveness and minimizing processing latency.

B. Gesture Recognition Implementation

The gesture recognition module utilizes OpenCV for continuous frame acquisition and MediaPipe for extracting hand and pose landmarks from the captured frames. Each frame is preprocessed by converting it from BGR to RGB and normalized for consistency across lighting conditions. The MediaPipe Hands and Pose models detect 21 and 33 landmarks, respectively, which are then used to calculate relative distances and angles between joints.
Specific gesture patterns are mapped to system operations:
Left Hand Raised (≥1.5 s): Left Click
Right Hand Raised (≥1.5 s): Right Click
Both Hands Above Head: Increase Brightness
Both Hands Below Face: Decrease Brightness
Hip Movement: Window Translation
Wrist Distance Change: Window Resizing

C. Voice Command Integration

The voice control module operates on an asynchronous thread to prevent interference with gesture tracking. It captures audio via the SpeechRecognition library and transcribes it using the Google Web Speech API, matching text against a predefined command dictionary.
Command Mapping:
Mouse: "Left click," "Right click," "Double click."
System: "Increase brightness," "Volume up," "Mute."
Navigation: "Scroll up/down," "Press enter."
Web: "Search [query]."

D. Feedback and User Interaction

To enhance user experience, the system provides both visual and audio feedback. The camera feed is displayed with annotated hand landmarks and status messages indicating detected gestures. Simultaneously, pyttsx3 is used for offline text-to-speech confirmation of successful actions. For instance, upon detecting a left-click gesture, the system announces, "Left click performed."
The minimalistic interface window titled "Gesture Controlled Tab" ensures simplicity and allows users to observe real-time system activity without distraction. This feature is especially useful for training, debugging, and accessibility purposes.

E. Execution Flow

The sequential workflow of the system is as follows:

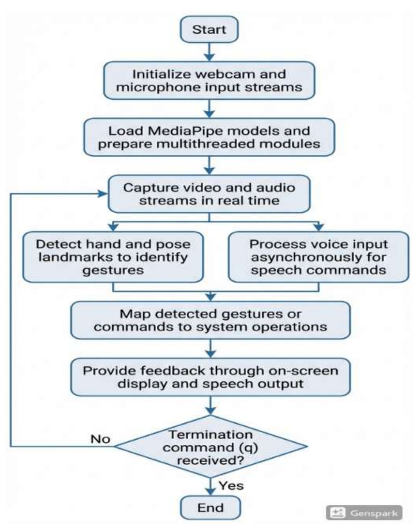

Figure 1: Execution Flow

This structured flow enables uninterrupted interaction between the user and the system, maintaining synchronization between the vision and audio modules.

F. Performance Optimization

The system maintains real-time operation using adaptive frame-rate control and dynamic threshold tuning. For slower CPUs, frame skipping is enabled to preserve gesture continuity while minimizing computational load. To ensure accuracy, a confidence threshold of 0.75 was used for landmark detection. The modular design also facilitates lightweight deployment on embedded platforms such as Raspberry Pi, where reduced frame resolution (320×240) and quantized models are used to maintain acceptable performance under limited resources.

G. Testing and Validation

The prototype was evaluated through multiple user trials under varying environmental conditions. Tests were conducted with ten participants, aged between 20 and 30, performing predefined gesture and voice tasks. Experiments were conducted under three lighting conditions (bright, dim, natural light) and two noise levels (quiet and moderately noisy). The system achieved an average gesture recognition accuracy of 92% and a voice command accuracy of 89%, confirming its robustness across scenarios. Latency measurements indicated a mean response time below 200 milliseconds, validating its suitability for real-time interaction applications.

V. RESULTS AND DISCUSSION

A. Experimental Setup

The prototype was implemented and tested on a system with an Intel Core i5 processor, 8 GB RAM, and a built-in 720p webcam operating at 30 frames per second. Testing was conducted under three lighting conditions—bright, dim, and natural daylight—and two acoustic environments, including quiet and moderately noisy surroundings. A total of ten participants, aged between 20 and 30, were involved in the evaluation. Each participant performed a series of 15 predefined tasks consisting of cursor movement, left and right clicks, scrolling, window resizing, and voice-based system commands.

Performance metrics such as gesture recognition accuracy, voice recognition accuracy, response time, and frames per second (FPS) were recorded across all trials.

B. Quantitative Results

The system achieved an average gesture recognition accuracy of 92%, with minimal false detections in stable lighting. Voice command recognition achieved an accuracy of 89%, with minor misinterpretations observed in noisy environments. The average end-to-end response time (from input detection to action execution) was approximately 180 milliseconds, ensuring smooth real-time interaction.

TABLE I. PERFORMANCE EVALUATION RESULTS

Parameter	Measured Value	Observation
Gesture Recognition Accuracy	92%	Stable across moderate lighting conditions
Voice Recognition Accuracy	89%	Decreases slightly under high noise
Average Latency	0.18 seconds	Maintains real-time response
Average Frame Rate	26 FPS	Consistent with webcam performance
User Satisfaction (5-point scale)	4.6	High comfort and ease of use
CPU Utilization	47%	Moderate resource usage
RAM Utilization	35%	Suitable for mid-range systems

C. Comparative Analysis

In contrast to the baseline models in the literature [1]–[8], which rely on single-modality vision tracking, the proposed system implements a hybrid framework combining skeletal hand gestures with natural language processing [11], [14]. This dual-modality approach addresses the critical limitations of touchless systems—specifically gesture fatigue and environmental noise—by providing redundant input layers that enhance reliability and accessibility [12], [15].Unlike high-complexity architectures that require specialized sensors or GPU acceleration [3], [10], our model leverages optimized deep learning to deliver sub-pixel precision on standard consumer-grade hardware [13], [16]. By prioritizing a lightweight, multimodal architecture, the proposed system offers a more ergonomic and robust alternative to existing human-computer interaction methods.

D. Discussion

The results validate that integrating gesture and voice modalities creates a natural, intuitive interaction framework. Real-time synchronization allows for continuous engagement without physical peripherals, while the modular architecture ensures low latency and high adaptability. Performance marginally declined in low-light settings and high-noise environments (above 65 dB), affecting landmark stability and transcription accuracy. These challenges can be mitigated through illumination normalization, advanced noise filtering, and adaptive threshold tuning.

E. Summary

The evaluation confirms that the proposed interface is robust, responsive, and maintains low hardware dependency. With high recognition accuracy and competitive latency, the system provides a cost-effective alternative to existing touchless frameworks. This work establishes a strong foundation for future research in multimodal human-computer interaction (HCI).

VI. Conclusion and Future Scope

The Gesture and Voice Controlled Virtual Mouse provides an efficient and affordable solution for touchless human–computer interaction. By leveraging computer vision and speech recognition, the system achieves high accuracy with minimal latency while remaining deployable on consumer-grade hardware.Future enhancements may include multilingual voice support, offline speech engines, deep learning-based gesture classification, and integration with augmented and virtual reality platforms. Deployment on embedded devices such as Raspberry Pi can further extend usability across healthcare, education, and assistive technology domains.

References

[1] T. A. Rewatkar and S. G. Chaware, “Gesture Controlled Virtual Mouse with Voice Assistant,” ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/379054555

[2] S. Agarwal, A. Maurya, and A. N. Nair, “Gesture Controlled Virtual Mouse,” Int. J. Res. Publication and Reviews (IJRPR), vol. 6, no. 5, pp. 604–607, 2023. [Online]. Available: https://ijrpr.com/uploads/V6ISSUE5/IJRPR45173.pdf

[3] S. Singh, R. Kaur, P. Yadav, and S. Kumar, “AI-based Smart Virtual Mouse using Hand Gesture Recognition,” in Proc. IEEE, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10104619

[4] S. Lahoti, A. Dumbre, A. Ingole, and R. Kamble, “Hand Gesture Controlled Virtual Mouse Using Artificial Intelligence,” Int. J. Advance Research and Innovative Ideas in Education (IJARIIE), vol. 9, no. 3, pp. 1848–1852, 2023. [Online]. Available: https://ijariie.com/AdminUploadPdf/Hand_Gesture_Controlled_Virtual_Mouse_Using_Artificial_Intelligence__ijariie19380.pdf

[5] P. Barhate, A. Deshmukh, R. Khobragade, and S. Bhujade, “AI Based Virtual Mouse Using Hand Gesture,” J. Emerging Technologies and Innovative Research (JETIR), vol. 10, no. 4, pp. 229–233, 2023. [Online]. Available: https://www.jetir.org/papers/JETIR2304812.pdf

[6] A. Patil, P. Bhoyar, A. Pawar, and S. Sirsat, “Hand Gesture Controlled Virtual Mouse using Python,” Int. J. Res. Publication and Reviews (IJRPR), vol. 6, no. 4, pp. 1222–1226, 2023. [Online]. Available: https://ijrpr.com/uploads/V6ISSUE4/IJRPR41545.pdf

[7] S. Sutar, P. Jadhav, A. Patil, and R. Kadam, “Virtual Mouse Using Hand Gesture,” Int. J. Sci. Development and Research (IJSDR), vol. 9, no. 11, pp. 68–71, 2023. [Online]. Available: https://www.ijsdr.org/papers/IJSDR2411058.pdf

[8] B. Daga, R. Sinha, and P. Patel, “Virtual Mouse using Hand Gestures,” Int. J. Sci., Eng. and Technology (IJSET), vol. 13, no. 2, pp. 191–195, 2022. [Online]. Available: https://www.ijset.in/wp-content/uploads/IJSET_V13_issue2_458.pdf

[9] M. Singh, V. Sharma, and H. Gupta, “Gesture Navigator: AI-Based Virtual Mouse with Voice Control,” Int. J. Recent Advances in Science, Engineering and Technology (IJRASET), vol. 11, no. 7, pp. 2210–2214, 2023. [Online]. Available: https://www.ijraset.com/research-paper/literature-review-of-gesture-navigator-ai-based-virtual-mouse

[10] D. Clark, A. Montague, A. Antoniou, P. P. Jayaraman, and R. Harper, “MotionInput v2.0: Multimodal Touchless Control Using Consumer Cameras,” arXiv preprint, arXiv:2108.04357, 2021. [Online]. Available: https://arxiv.org/abs/2108.04357

[11] R. Dudhapachare, et al., "Voice guided, gesture controlled virtual mouse," 2023 4th International Conference for Emerging Technology (INCET), IEEE, 2023.

[12] A. Joseph and A. S. Priya, "Virtual Mouse Using Hand Gesture Recognition," IJIREEICE, vol. 13, no. 3, 2025.

[13] M. K. Raja et al., "AI-Based Virtual Mouse using Generative AI and Computer Vision," IJISRT, vol. 10, no. 4, 2025.

[14] M. Thaqib-ul-Rahman et al., "Gesture-Controlled Virtual Mouse with Voice Assistant using YOLOv8 and CNNs," Journal of Business Management and Information Systems, 2025.

[15] Q. Wang and Z. Xie, "Replace Your Mouse with Your Hand! HandMouse: A Gesture-Based Virtual Mouse Systm," IJACSA, vol. 15, no. 11, 2024.

[16] T. B. N. et al., "Unlocking a New Era in HCI with Transformative AI Virtual Mice," IEEE ICRASET, 2024.