© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

GazeTrack: An AI-Based Multimodal Assistive System

Ms. Nishigandha Somnath Gavali[1], Prof. Ms. Swati S. Hinge[2]

[1]*Student, Artificial Intelligent, Shree Mahavir Education Society's Sanghavi College of Engineering, Nashik - 422 013, Maharashtra, India.*

[2]*Professor, Artificial Intelligent, Shree Mahavir Education Society's Sanghavi College of Engineering, Nashik - 422 013, Maharashtra, India.*

***Abstract*—Human Computer Interaction (HCI) is evolving toward more natural and touchless technologies that improve both usability and accessibility. This paper presents GazeTrack, a personal AI-driven assistive system that allows users to control computers using hand gestures, eye tracking and voice commands, eliminating the need for traditional devices such as keyboards and mice. A standard webcam captures real-time hand movements, which are analyzed using computer vision techniques to recognize gestures for cursor movement, clicking, scrolling, and drag-and-drop actions.**

In addition, voice commands enable users to perform system operations without physical interaction. The system ensures smooth and responsive performance by combining gesture and speech inputs effectively. GazeTrack is particularly beneficial for individuals with physical disabilities, as it reduces effort and supports contactless interaction. Experimental results show high accuracy, quick response, and consistent performance, making it a practical alternative to conventional input methods.

***Index Terms*—Multimodal HCI, Gesture Recognition, Voice Control, Virtual Mouse, AI Interface, Real-Time Interaction**

I. INTRODUCTION

The field of Human Computer Interaction (HCI) has shifted from traditional input devices to more intuitive and contactless interaction methods. With the advancement of Artificial Intelligence and Computer Vision, systems can now understand user intent through gestures and gaze, improving accessibility and user experience. These approaches are especially useful for individuals with physical limitations or in environments where touch-based interaction is not feasible.

However, existing systems often face challenges such as sensitivity to lighting, occlusion, and inconsistent user behavior. To overcome these limitations, this work proposes GazeTrack, a multimodal system that integrates gaze tracking, gesture recognition, and voice commands. The goal is to create a reliable, real-time, and user-friendly interface that enhances interaction efficiency while reducing dependency on physical devices.

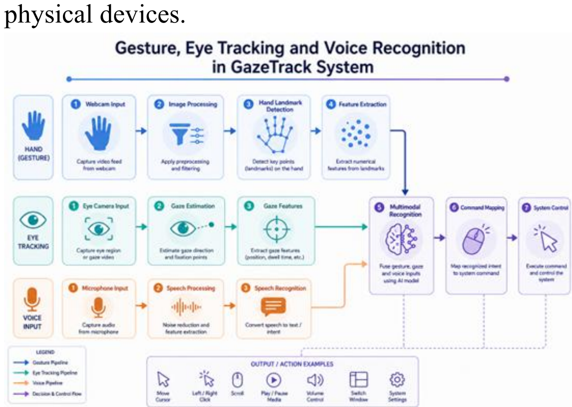

Figure 1. GazeTrack- Multimodal human-computer interaction platform

Figure 1. shows a multimodal GazeTrack system that combines gesture, eye tracking, and voice inputs. Hand gestures are captured through a webcam, processed, and recognized using an AI model. Eye tracking detects gaze direction to control cursor movement and screen interaction. Voice input is processed to convert speech into commands or actions. All inputs are fused together to execute system controls like clicking, scrolling, and navigation efficiently.

II. LITERATURE REVIEW

Recent research in HCI highlights the importance of multimodal interaction, where multiple inputs such as gesture, gaze, and speech are combined to improve system performance. Studies show that integrating these modalities enhances accuracy and reduces ambiguity compared to single-input systems [1]. Deep learning techniques, particularly CNNs, have significantly improved gaze tracking and gesture recognition capabilities [2], [3].

Gesture recognition also plays a vital role in natural interaction, with advancements in computer vision and deep learning enabling reliable detection of hand movements [3], [7]. However, issues like lighting variation and occlusion still affect performance [8]. To address these limitations, recent research emphasizes multimodal frameworks that combine gaze, gesture, and voice inputs, resulting in improved accuracy, robustness, and user experience [5], [12]. Such developments support practical applications like virtual mouse systems and form the basis for advanced solutions like GazeTrack [6], [9].

Table 1. An overview of recent surveys in the field

Author / Year	Methodology	Limitations
O. A. K. et al. (2024) [1]	Survey on multimodal HCI systems	No practical implementation
J. Chen et al. (2025) [2]	CNN-based gaze estimation	Lacks multimodal integration
S. Rautaray & A. Agrawal (2024) [3]	Vision-based gesture recognition survey	Sensitive to lighting and occlusion
M. T. Rahman et al. (2026) [4]	Multimodal HCI study	No gesture support
K. Nguyen & H. Lee (2025) [5]	Deep learning multimodal framework	High computational cost
R. Sharma & P. Gupta (2025) [6]	AI virtual mouse system	No gaze tracking
M. Linardakis et al. (2025) [7]	Survey on gesture recognition	No real-time implementation
W. Zhang et al. (2021) [8]	Dynamic gesture recognition	Limited to gesture modality
P. U. Vignesh et al. (2023) [9]	Virtual mouse using gestures	Limited functionality
T. S. C. Reddy et al. (2023) [12]	Multimodal gesture review	No implementation
Z. Zhang (2012) [13]	Kinect-based motion tracking	Requires special hardware
S. Mitra & T. Acharya (2007) [14]	Foundational gesture survey	Outdated approaches
Proposed System: GazeTrack (2026)	AI-based multimodal assistive system	Depends on lighting and webcam quality (typical constraint)

III. METHODOLOGY

The proposed system integrates gesture recognition, gaze tracking, and voice commands into a unified framework for hands-free interaction. The process begins with capturing real-time video and audio inputs using a webcam and microphone. These inputs are preprocessed to enhance quality through normalization, feature extraction, and noise reduction. Gesture detection is performed using MediaPipe and OpenCV, while gaze tracking maps eye movements to screen coordinates. Voice commands are processed using speech recognition techniques. All inputs are combined using a multimodal fusion approach to improve accuracy and responsiveness. The system then executes actions such as cursor movement, clicking, and scrolling in real time. Finally, performance is evaluated based on accuracy, response time, and user experience.

This project focuses on building a smart system called the AI Gaze Gesture Navigator, which combines gesture recognition, eye gaze tracking, and voice commands into one interactive platform. The goal is to create a natural, hands-free way to control digital devices, making technology easier and more accessible for everyone.

First, the system gathers real-time video from a webcam to detect hand gestures and eye movements while capturing audio through a microphone for voice commands. The captured data is then cleaned and processed images are resized and backgrounds removed to improve clarity, and audio noise is filtered out for clear speech recognition.

Next, gestures are identified using MediaPipe and OpenCV, along with convolutional neural networks that can recognize actions like clicking, dragging, and scrolling. These gestures are then linked to actual computer functions through PyAutoGUI. For eye tracking, tools like Dlib or OpenFace locate eye landmarks and calculate the pupil's position to control the cursor based on where the user looks.

Voice inputs are processed through the Google Speech API or advanced AI models that interpret spoken commands like "open file" or "scroll down." All three

inputs gesture, gaze, and voice are combined in a single control system, with specific rules deciding which input takes priority depending on the situation.
The fused controls translate directly into cursor movement and system actions, making the interaction smooth and immediate. The complete system is then rigorously tested for speed, precision, and comfort across different users and environments.

IV. PROPOSED WORK

This work introduces GazeTrack as a contactless interaction system that enables users to control computers using natural inputs. The system leverages AI and computer vision to interpret gestures, gaze, and voice commands in real time. These inputs are integrated to provide precise and efficient control of system functions.
The proposed approach focuses on improving accessibility, usability, and interaction speed. It is designed for applications in assistive technology, education, and smart environments, offering a flexible and user-friendly alternative to traditional input devices.

V. SYSTEM ARCHITECTURE

The system consists of multiple layers. The input layer captures data from the webcam and microphone. The processing layer performs feature extraction and recognition using AI techniques. The fusion layer integrates all inputs to ensure smooth interaction. The control layer converts inputs into system actions, and the output layer delivers the final user experience.
The workflow includes data acquisition, preprocessing, feature extraction, model evaluation, and real-time deployment, ensuring efficient and reliable system performance.

- Input Layer: This layer collects real-time data through a webcam for gesture and gaze detection, and a microphone for capturing voice commands.
- Processing Layer: The captured data is processed using image enhancement, hand and eye landmark detection, gaze estimation, and speech-to-text conversion with the help of AI and computer vision techniques.
- Fusion Layer: Inputs from gesture, gaze, and voice are combined and coordinated to resolve conflicts and ensure smooth and accurate interaction.
- Control Layer: The processed inputs are translated into system actions such as cursor movement, clicking, scrolling, and executing application commands, enabling complete hands-free control.
- Output Layer: The system provides a seamless and intelligent user experience, improving accessibility and comfort, especially for users with physical limitations, and supporting applications in areas like healthcare and education.

VI. WORKFLOW DESCRIPTION:

The process starts with data collection, followed by preprocessing to clean and standardize inputs. Advanced models then extract relevant features and are evaluated for performance and reliability. Finally, the optimized system is deployed for real-time use, ensuring efficient and user-friendly multimodal interaction.

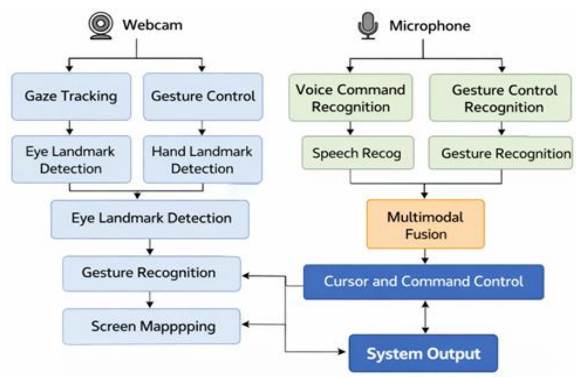

Figure 2. Workflow Architecture of GazeTrack System

VII. RESULTS AND DISCUSSION

The GazeTrack system was tested to evaluate its real-time performance. The results show that gesture recognition and gaze tracking provide accurate cursor control, while voice commands are executed with minimal delay. The system demonstrates stable performance with low latency under normal conditions. The findings indicate that combining multiple input methods improves accuracy and usability compared to single-input systems. Although factors like lighting and noise can affect performance, the system remains reliable in most scenarios. The results highlight the effectiveness of multimodal interaction in creating user-friendly and accessible systems.

Figure 3 illustrates a comparison between traditional input devices and the proposed GazeTrack system across key performance metrics. Although conventional methods achieve slightly higher accuracy, GazeTrack greatly reduces user effort and enhances accessibility. These findings indicate that the proposed system provides an effective and well-balanced alternative, particularly in improving usability and inclusiveness.

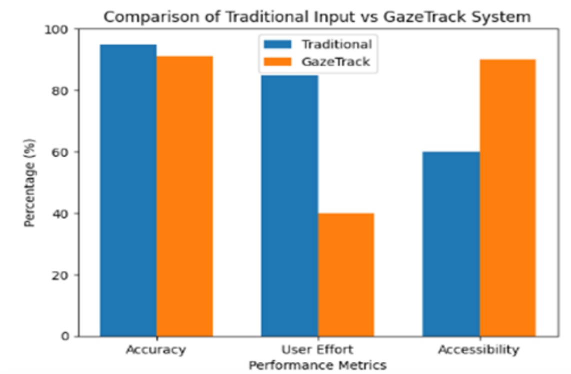

Figure 3. Comparison of Traditional Input vs GazeTrack System

VIII. EXPERIMENTAL SETUP AND OUTPUT SCREENS

A. System Configuration

The GazeTrack system is built using Python and uses different libraries to enable real-time interaction. It uses OpenCV to capture and process video from the camera, and MediaPipe to detect hand movements and facial features accurately. Voice commands are handled using SpeechRecognition, while actions like clicking and scrolling are performed using automation tools such as PyAutoGUI and Autopy.

The system combines eye tracking, hand gestures, and voice commands into one platform, allowing the user to control the computer in different ways. It also includes a smart feature that automatically switches between eye and hand control based on what the user is doing, making the system easier and more comfortable to use.

B. Hardware Setup

The experimental setup was conducted on a standard computing system with the following specifications:

- Processor: Intel i5 or equivalent
- RAM: 8 GB
- Camera: Integrated webcam (720p)
- Operating System: Windows 10 or 11

The webcam was positioned at eye level to ensure optimal detection of facial landmarks and hand gestures. Proper illumination was maintained to minimize detection errors and improve system accuracy.

C. Software Environment

The system was implemented in a real-time processing environment with the following components:

- Python 3.x
- OpenCV for video capture and processing
- MediaPipe for hand and face mesh detection
- SpeechRecognition for voice input processing
- AI-based NLP module for natural language command interpretation

The integration of these components enables simultaneous processing of visual and auditory inputs.

D. Experimental Procedure

The system was evaluated under multiple operational modes to analyze its performance:

1. Eye Tracking Mode: Cursor movement controlled using gaze direction and blink detection for click operations.
2. Hand Gesture Mode: Interaction through finger gestures including cursor movement, clicking, dragging, scrolling, and zooming.
3. Voice Command Mode: Execution of system-level commands using natural language input.

Figure 4, 5 and 6 shows the output screens of GazeTrack. Each mode was tested independently and in combination to evaluate system responsiveness, usability, and accuracy.

E. Output Screens

1) Eye Tracking

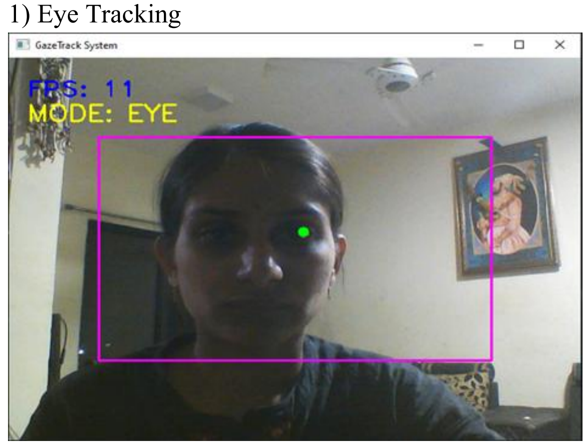

Figure 4. Eye Tracking

2) Hand Gesture

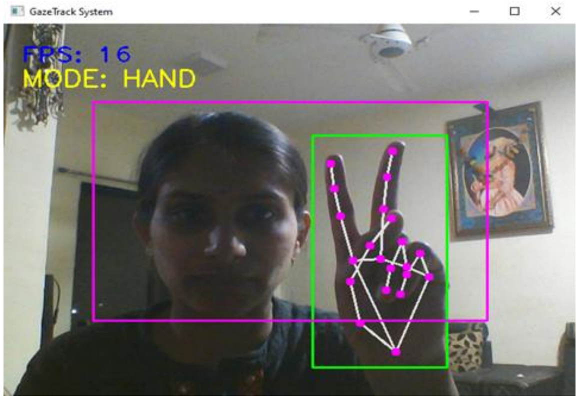

Figure 5. Hand Gesture

3) Voice Command

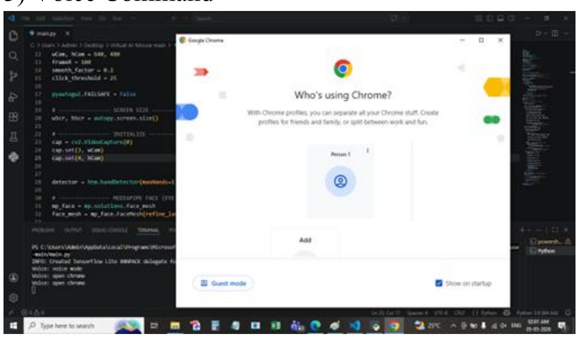

Figure 6. Voice Mode

The experimental results demonstrate that the proposed GazeTrack system effectively integrates multiple input modalities into a unified interface. The system provides a flexible and intuitive interaction mechanism, particularly beneficial for assistive applications. The combination of computer vision and AI-based voice processing enables a robust and user-friendly human-computer interaction system.

IX. CHALLENGES

GazeTrack faces several practical challenges that can affect its performance and usability. Since the system relies heavily on vision-based tracking, its accuracy is sensitive to lighting conditions; poor illumination, shadows, or excessive brightness can interfere with reliable detection of eye and hand features. Additionally, voice recognition may struggle in noisy environments, where background sounds can reduce the accuracy of speech-to-text conversion. The system also involves significant computational complexity, as it must process multiple input streams gaze, gesture, and voice in real time, requiring efficient algorithms and adequate hardware resources. Furthermore, user-specific calibration is often necessary to ensure precise gaze tracking and gesture recognition, as variations in individual features and behaviors can impact the system's overall accuracy.

X. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

This paper presented GazeTrack, a multimodal HCI system that enables touch-free interaction using gesture, gaze, and voice inputs. The system improves usability and accessibility by providing an intuitive interface and reducing reliance on traditional devices. However, the system is affected by environmental factors such as lighting and background noise. Future work will focus on improving robustness, supporting advanced gestures, and enhancing performance using advanced AI models. The system can be further extended to applications in virtual reality, healthcare, and smart environments.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my guide, Prof. Ms. Swati S. Hinge, for her invaluable guidance, continuous support, and encouragement throughout the course of this project. His/her expertise and insightful suggestions were instrumental in shaping the direction and quality of this work.

I am also thankful to the Head of Department and the Principal of Sanghavi College of Engineering for providing the necessary resources and support that enabled the successful completion of this project.

I extend my appreciation to all the teaching and non-teaching staff of the Department of Artificial Intelligence for their cooperation and assistance whenever required.

Finally, I would like to convey my heartfelt thanks to my family and friends for their constant motivation, understanding, and encouragement throughout this journey.

REFERENCES

[1] O. A. K. et al., "Multimodal human–computer interaction: A review," IEEE Access, vol. 12, pp. 10234–10250, 2024.

[2] J. Chen, L. Wang, and Y. Zhao, "Appearance-based gaze estimation using deep learning: A review," Pattern Recognition, vol. 140, pp. 109–123, 2025.
[3] S. Rautaray and A. Agrawal, "Vision-based hand gesture recognition: A survey," Artificial Intelligence Review, vol. 57, pp. 1–30, 2024.
[4] M. T. Rahman et al., "Gaze and speech in multimodal human-computer interaction: A scoping review," 2026.
[5] K. Nguyen and H. Lee, "Multimodal deep learning for human–computer interaction," IEEE Transactions on Multimedia, vol. 27, pp. 455–470, 2025.
[6] R. Sharma and P. Gupta, "AI-based virtual mouse using hand gesture and voice recognition," International Journal of Emerging Technology and Research, vol. 13, no. 7, pp. 45–52, 2025.
[7] M. Linardakis, I. Varlamis, and G. Th. Papadopoulos, "Survey on hand gesture recognition from visual input," IEEE, 2025.
[8] W. Zhang et al., "Dynamic hand gesture recognition based on short-term sampling," IEEE/CAA Journal of Automatica Sinica, 2021.
[9] P. U. Vignesh et al., "AI virtual mouse using hand gesture recognition," IRJAEH, 2023.
[10] "Survey on hand gesture recognition from visual input," arXiv, 2025; PeerJ Review, 2025.
[11] M. Thaqib-ul-Rahman et al., "Deep learning approaches for gesture recognition," IEEE Access, 2024.
[12] T. S. C. Reddy et al., "Multimodal gesture recognition: A review," Sensors, 2023.
[13] Z. Zhang, "Microsoft Kinect sensor and its effect," IEEE MultiMedia, vol. 19, no. 2, pp. 4–10, Apr.–Jun. 2012.
[14] S. Mitra and T. Acharya, "Gesture recognition: A survey," IEEE Transactions on Systems, Man, and Cybernetics Part C, vol. 37, no. 3, pp. 311–324, May 2007.