Tri-Control

Chetan Kharade[1] Sambhaji Devkate[2] Amruta Golde[3] Kirti Kulkarni[4] Priyanka Kumbhar[5]
[1,2,3,4]Student [5]Guide
[1,2,3,4,5]Department of Information Technology
[1,2,3,4,5]Parvatibai Genba Moze College of Engineering, Pune, Maharashtra, India

Abstract — This paper introduces a comprehensive multimodal system designed for accessible computer control, incorporating eye movement tracking, gesture recognition based on sign language, and voice commands. The system aims to provide a hands-free, user-friendly interface that significantly enhances digital accessibility for users with disabilities, such as motor impairments or vision-related limitations, as well as users seeking efficient multitasking solutions. Implemented using Python and associated libraries (OpenCV, TensorFlow, SpeechRecognition), the platform integrates real-time computer vision, machine learning, and speech recognition technologies. The result is a unified interaction model that empowers users to perform cursor movements, clicks, and various computing tasks with precision and convenience. Extensive testing demonstrated accuracy levels above 90% across different modalities. The system promotes digital inclusion and provides a foundation for scalable assistive technology.

Keywords: Multimodal Human-Computer Interaction, Accessible Computing Interface, Eye Movement Tracking System, Gesture Recognition via Sign Language, Voice-Controlled Computing

I. Introduction

The evolution of human-computer interaction (HCI) has prioritized user convenience, efficiency, and accessibility. Traditional input devices such as the keyboard and mouse, although widely used, are insufficient for individuals with motor impairments or visual disabilities. These devices also present challenges for users engaged in multitasking activities that require minimal manual interaction. The need for alternative interaction methods has led to the development of assistive technologies that rely on eye tracking, gesture control, and voice recognition.

This research presents a multimodal system that combines these three interaction methods, offering a flexible and customizable way to control computer functions. Users can rely on a single input method or combine modalities based on their needs and environmental conditions. This system addresses the limitations of existing assistive devices by offering higher accuracy, adaptability, and user-centered design.

The system is especially beneficial for individuals with physical disabilities, visually impaired users, and users in hands-busy environments such as surgeons or factory workers. The project also aligns with global accessibility standards and the principles of inclusive design. It enables equitable access to digital resources, thereby reducing the digital divide for users with disabilities and contributing to greater social and technological inclusion

II. Related Work

The implementation of assistive input devices has been a focus of great research interest. Eye tracking systems, as reviewed by Chen et al (2020), offer an intuitive method of cursor control through gaze mapping, which translates a person's gaze coordinate into a screen position. Compared to earlier systems that relied heavily on peripherals and were expensive, today's eye-tracking systems based on cameras are both inexpensive and very precise.

Gesture recognition has also made significant strides, thanks to advancements in computer vision and machine learning. Zhang and Wang (2019) noted the applicability of gesture systems in low-resource environments, especially for users with limited limb movement. By analyzing hand shapes, orientations, and motion, these systems can trigger predefined commands or simulate mouse actions.

Voice recognition interfaces, discussed by Lee et al. (2021), are effective in hands-free environments and can serve as primary or supplementary input methods. Recent progress in natural language processing (NLP) and speech-to-text engines has significantly enhanced voice command reliability, even in the presence of accents or ambient noise.

Smith et al. (2022) advocated for multimodal interaction systems, which combine input types to provide a more robust and fault-tolerant interaction environment. They concluded that such systems are more resilient to individual input failure, offer enhanced usability, and provide users with more autonomy.

Privacy remains a concern in tracking systems, as noted by Brown and Thompson (2021). Capturing sensitive biometric data like gaze, gestures, or voice necessitates secure data handling protocols. Their work calls for encrypted processing and anonymization to protect user identity and foster trust in assistive systems.

III. System Design and Architecture

The proposed system follows a modular architecture with three independent yet interoperable modules: Eye Tracking Module Uses facial landmark detection through OpenCV and Dlib. The user's pupil coordinates are calculated in real time and translated into cursor movements. Click simulation is achieved through blink detection or prolonged focus (dwell time).

Module for Gesture Recognition This uses a CNN which has been trained on a gesture labeled dataset. The hand gestures which are captured through the webcam are mapped to cursor actions, such as Scroll, Left-click and Right-click. Voice Command Module Built using Python's SpeechRecognition and Google API, the system recognizes voice input to execute commands like opening apps, clicking,

and scrolling. It supports multiple languages and dialects and adapts to user accent through personalized training.

A central controller arbitrates between these modules, allowing users to operate them concurrently or individually. The user interface provides calibration tools, sensitivity controls, and accessibility presets tailored to various disabilities.

IV. Implementation

We chose Python to implement the system because of its simplicity, strong community support, and powerful libraries for computer vision, speech recognition, and machine learning. It allowed us to prototype quickly and integrate advanced functionalities with minimal overhead.
Libraries used include:

- OpenCV for face and hand tracking
- Dlib for real-time gaze estimation
- TensorFlow for deep learning-based gesture classification
- SpeechRecognition + PyAudio for voice input

Tkinter for GUI development Key features include:

Real-time input feedback and cursor visualization Configurable gesture sets for custom commands Offline and online voice recognition support Custom user profiles for storing accessibility preferences The architecture is device-independent and optimized for low-end hardware to increase accessibility for economically disadvantaged users.

V. Testing and Evaluation

Testing was conducted with 15 participants across various user groups:

- 5 users with motor disabilities
- 5 visually impaired users
- 5 general users (multitasking scenarios)

Key metrics: The eye tracking system performed reliably, achieving an accuracy of around 88% under stable lighting conditions.

Even when lighting varied—such as in dim or uneven environments—it maintained a respectable accuracy of about 81%, demonstrating its robustness in real-world settings.

Gesture Recognition Accuracy: 92% with 5 trained gestures Voice Command Accuracy: 93% (English), 90% (Hindi and Marathi)

Average System Latency: 280ms Divided into 5 categories, the Likert scale evaluated user satisfaction.

Usability achieved an average score of 4.6 and participants cited the following as their strongest points: ease of use, responsiveness, and flexibility.

Multitaskers preferred combining gestures and voice, while users with visual impairments favored voice activated commands

VI. Discussion

Integrating multiple input methods—eye tracking, gesture recognition, and voice commands—adds a valuable layer of flexibility to the system. For example, if a user is in a noisy environment where voice commands may not work well, they can switch to gestures or eye movement. Likewise, someone with limited hand mobility can rely solely on gaze tracking or voice control.

However, some challenges emerged during development and testing: Environmental Sensitivity: Gesture recognition performance is influenced by lighting conditions, while voice commands can be disrupted by background noise.

User Calibration: To achieve optimal accuracy, especially in gaze tracking and voice recognition, a short setup phase is needed to tailor the system to individual users. Latency Variations: When multiple input methods are used at once, slight delays in response time can occur, though the system remains usable.

On the privacy front, user data remains secure. All data is stored locally, with encryption options available. The system is also built to meet Web Content Accessibility Guidelines (WCAG), ensuring that accessibility remains a top priority.

VII. Conclusion

This project successfully demonstrates a practical, inclusive, and customizable solution for hands-free computer interaction. By allowing users to control a cursor using their eyes, hands, or voice, the system makes digital environments more accessible—particularly for individuals with disabilities. Its modular design allows for easy adaptation to different user needs and hardware setups.

High accuracy scores across all input methods, combined with strong user feedback, validate the effectiveness and relevance of the system.

The integration of multiple interaction methods also boosts system reliability, making it more versatile in real-world conditions. In short, the project proves that with thoughtful design and modern technology, we can create meaningful, accessible tools that promote independence and digital inclusion.

VIII. Future Work

Looking ahead, there are several ways the system can be enhanced: Support for Mobile and Wearable Devices: Expanding compatibility to smartphones, tablets, and smart glasses can bring accessibility to more platforms.

AI-Driven Personalization: Incorporating machine learning to adapt the system based on a user's interaction patterns can make the experience more intuitive over time. Emotion Recognition: By detecting emotional cues, the system could adjust its responses or interface to better support user well-being and engagement.

Cloud Synchronization: Allowing users to save their preferences and settings securely across devices will make the system more seamless and user-friendly.

Additionally, collaborations with rehabilitation centers and accessibility organizations are planned to conduct long-term user studies. These efforts will help refine the onboarding process and ensure the system meets real-world needs effectively.

IX. References

[1] Chen, L., Zhao, Y., & Huang, M. (2020).
[2] Eye Gaze Tracking in HCI.

[3] Journal of Eye Tracking Technology, 15(3), 455 470. Zhang, R., & Wang, H. (2019).
[4] Gesture Recognition for Accessible Computing.
[5] Journal of Assistive Computing, 11(2), 98–111. Lee, J., Kim, S., & Cho, H. (2021).
[6] Voice Commands and Assistive Technology.
[7] Assistive Tech Today, 22(4), 211 226. Smith, A., Rogers, T., & Patel, N. (2022).
[8] Multimodal Interaction for Enhanced Accessibility.
[9] Brown, K., & Thompson, R. (2021). Privacy in Tracking Systems.
[10] Privacy and Data Security Journal, 19(2), 105–120