EYECON: A MULTIMODAL HUMAN-COMPUTER INTERFACE INTEGRATING GAZE TRACKING AND VOICE COMMANDS

Suyash Subhash Narawade
Dept. of Computer Engineering
Rajiv Gandhi College of Engineering (RGCOE)
Karjule Harya, India
Email: suyashnarawade440@gmail.com

Shailesh Vilas Kadam
Dept. of Computer Engineering
Rajiv Gandhi College of Engineering (RGCOE)
Karjule Harya, India
Email: shaileshkadam209@gmail.com

Sadik Samir Pathan
Dept. of Computer Engineering
Rajiv Gandhi College of Engineering (RGCOE)
Karjule Harya, India
Email: sadikpathan044@gmail.com

Dattatray Mininath Shirke
Dept. of Computer Engineering
Rajiv Gandhi College of Engineering (RGCOE)
Karjule Harya, India
Email: dattatrayshirke02@gmail.com

***Abstract*—Human-Computer Interaction (HCI) continues to evolve towards more intuitive and accessible control paradigms, particularly for users with motor impairments. This paper presents EyeCon, a novel multimodal interface that synergizes real-time eye tracking with robust voice recognition to provide a hands-free method for system navigation and control. The system leverages a standard webcam and the Mediapipe framework for high-fidelity facial landmark and iris tracking, translating gaze direction into cursor movements on the screen. Concurrently, an offline voice recognition module processes vocal commands such as "click," "scroll," and "open," which are mapped to system-level actions. An integration layer serves as the core of EyeCon, intelligently fusing the continuous stream of gaze data with discrete voice commands to execute user intentions accurately. This dual-modal approach significantly reduces the ambiguity and cognitive load associated with purely gaze-based or voice-based systems, specifically addressing the "Midas touch" problem. We detail the system architecture, implementation of core algorithms for calibration, gaze mapping, and command fusion, and present a comprehensive performance evaluation. The results demonstrate the system's potential as an effective, low-cost, and accessible assistive technology that requires no specialized hardware.**

***Index Terms*—Human-computer interaction, eye-gaze tracking, multimodal interface, assistive technology, Mediapipe, voice recognition.**

I. INTRODUCTION

The evolution of Human-Computer Interaction (HCI) has consistently aimed to create more natural and seamless communication between users and digital devices. While the keyboard and mouse remain the dominant input methods, they present significant accessibility challenges for individuals with severe motor disabilities, such as those with amyotrophic lateral sclerosis (ALS), spinal cord injuries, or cerebral palsy. This has spurred research into alternative interaction modalities designed to bridge this accessibility gap, including voice control, brain-computer interfaces (BCIs), and eye-gaze tracking [1].

However, single-modal interfaces often have inherent limitations. Voice-only systems can be cumbersome and inefficient for spatial tasks like cursor control. BCIs, while powerful, are often invasive or require extensive training and expensive hardware. Gaze-only systems, which map eye movements directly to cursor control, frequently suffer from the "Midas touch" problem, where unintended actions are triggered by a natural, prolonged stare [2]. This issue can make interfaces frustrating and error-prone.

The synergy of multiple input streams, as explored in multimodal systems, has been shown to create more robust, flexible, and less ambiguous user interfaces [3]. By combining modalities, the strengths of one can compensate for the weaknesses of another. This paper explores such a multimodal approach, aiming to create a system that is not only functional but also accessible and low-cost.

We introduce EyeCon, a system that combines the spatial accuracy of eye tracking for pointing with the explicit command execution of voice recognition for activation. Our primary contributions are:

1) The design and implementation of a low-cost, multimodal HCI system that functions effectively using only a standard webcam and microphone.
2) A robust integration logic that uses gaze stability analysis in conjunction with discrete voice commands to effectively mitigate the "Midas touch" problem.

3) An empirical evaluation of the system's performance, quantifying its accuracy and responsiveness for common desktop control tasks, and establishing its viability as a practical assistive tool.

This paper is structured as follows: Section II reviews related work in the field. Section III provides an in-depth look at the system architecture and methodology. Section IV presents our experimental setup and performance results. Section V discusses the implications and limitations of our findings. Finally, Section VI concludes the paper and outlines future research directions.

II. RELATED WORK

The pursuit of hands-free computing has a rich history, with significant advancements in both gaze and voice technologies.

A. Gaze Tracking Systems

Early eye-tracking systems often required specialized, expensive infrared hardware that illuminated the eye and tracked corneal reflections (known as Pupil Center Corneal Reflection, or PCCR). While highly accurate, their cost and setup complexity limited their widespread adoption. In recent years, the focus has shifted towards appearance-based methods that use standard webcams and computer vision techniques.

The "Eye Tracking for Everyone" project was a landmark study that demonstrated a scalable approach using Convolutional Neural Networks (CNNs) on a large, crowdsourced dataset to achieve high accuracy with standard webcams [1]. This proved the feasibility of low-cost, software-driven gaze tracking. The use of deep learning for facial landmark detection has become the state-of-the-art, with frameworks like Mediapipe providing highly optimized models that deliver real-time performance on commodity hardware [4]. These models can predict a dense mesh of facial landmarks, which is crucial for establishing a stable head-pose and accurately locating the pupils and irises.

B. Voice Control Interfaces

Voice recognition has become ubiquitous, integrated into smartphones, smart speakers, and desktop operating systems. However, many popular systems (e.g., Siri, Google Assistant, Alexa) rely on cloud-based processing. This approach raises privacy concerns, requires a constant internet connection, and can introduce network latency. For applications in assistive technology, where reliability and privacy are paramount, offline systems are often preferred.

Open-source toolkits like Kaldi have enabled the development of high-performance, offline speech recognition models [5]. These toolkits provide the foundation for engines like Vosk, which we utilize in our system [6]. Vosk offers lightweight models that can run entirely on a local machine, ensuring privacy and responsiveness without depending on an internet connection.

C. Multimodal HCI

The fusion of gaze and voice is a powerful and well-established paradigm in HCI. Research has shown that combining eye-gaze for pointing with speech for issuing commands can enable complex tasks, like controlling a robotic arm, with greater efficiency and naturalness than either modality alone [7]. Studies have also explored the use of dwell time as an activation method in gaze-based systems, where a user stares at an element for a fixed duration to trigger a click. However, this can lead to false positives and a less fluid user experience, as it interrupts the natural flow of visual scanning [8].

Our work directly addresses the shortcomings of dwell-time activation by replacing it with explicit voice commands. This approach builds on the established principles of multimodal interaction to create a practical, low-ambiguity solution for general desktop navigation.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The EyeCon system is composed of three primary software modules: the Eye Tracking Module, the Voice Recognition Module, and the Integration Layer. These modules work in concert to translate user intentions into system actions. The overall architecture is depicted in Fig. 1.

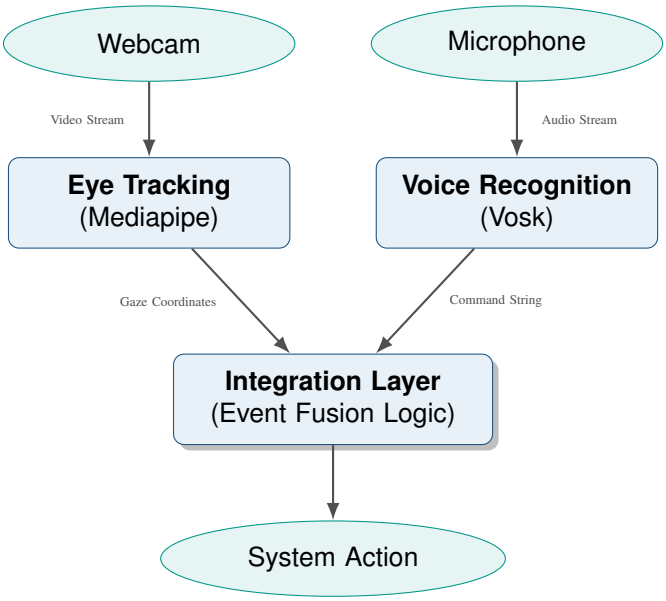

Fig. 1: System Architecture of EyeCon, showing data flow from input sensors through processing modules to the final system action.

A. Eye Tracking Module

This module is the core of the spatial navigation component. We use Google's Mediapipe framework for its high-performance Face Mesh model, which predicts 468 3D facial landmarks in real-time [4].

1) Calibration: To map the user's eye movements to screen coordinates, a one-time, 5-point calibration is performed. The user is asked to look sequentially at five known points (four corners and the center) on the screen. For each point, the system records the corresponding 3D iris landmark coordinates from Mediapipe. This process builds a user-specific dataset that correlates the raw gaze vectors with absolute screen positions (Fig. 2).

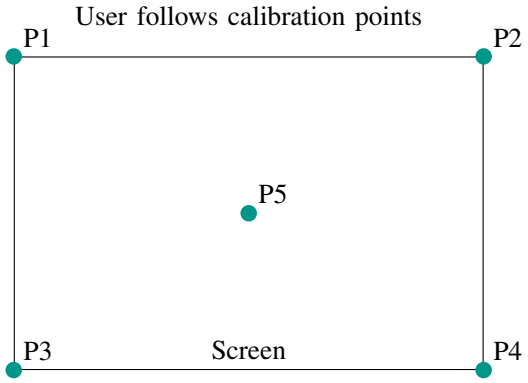

Fig. 2: The 5-point calibration process on the user's screen.

2) Gaze Vector and Screen Mapping: For each frame from the webcam, we first establish a 3D head coordinate system using stable facial landmarks (e.g., nose tip, chin, and forehead points). This allows us to compensate for small head movements. The gaze vector $\vec{g}$ is calculated as the vector originating from the estimated 3D center of the eyeball to the 3D center of the pupil, as provided by Mediapipe's iris model (Fig. 3).

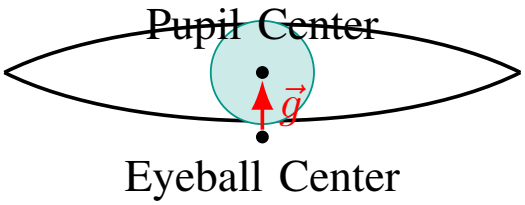

Fig. 3: Conceptualization of the gaze vector $\vec{g}$ originating from the eyeball center to the pupil center.

This vector is then transformed into the 2D screen coordinate space. We use a second-order polynomial regression model, trained on the calibration data, to learn the mapping function f:

$$(S_x, S_y) = f(g_x, g_y, g_z) \tag{1}$$

where (S_x, S_y) are the screen coordinates and (g_x, g_y, g_z) are the components of the gaze vector. A moving average filter over the last 5 frames is applied to the final coordinates to reduce jitter and create a smoother cursor motion.

B. Voice Recognition Module

This module interprets voice commands using the Vosk Speech Recognition Toolkit [6]. We use a small, pre-trained English model (approx. 50 MB) which is loaded into memory at startup. A predefined command grammar is specified:

```
[ "click", "double click", "scroll up",
  "scroll down", "open file", "close window" ]
```

This limited vocabulary significantly improves recognition accuracy and speed. The system continuously listens to the microphone audio stream, processes it in chunks, and when a complete utterance matching the grammar is detected, the corresponding command string is sent to the Integration Layer.

C. Integration Layer

This is the core logic that fuses the two modalities. It operates as a state machine. When a voice command is received, it checks if the user's gaze has been stable for a minimum duration (a 300ms dwell-time filter). Gaze stability is determined by calculating the variance of the last N gaze coordinates stored in a buffer. If the variance is below a predefined threshold, the gaze is considered stable. If the gaze is stable, the voice command is executed at the average coordinate of the gaze buffer. This prevents accidental actions while the user is scanning the screen. Algorithm 1 shows the pseudocode.

Algorithm 1 Integration Layer Logic

```
Initialize: ‘gaze_history‘ ← empty queue of size N
while running do
    ‘current_gaze‘ ← Get smoothed gaze coordinates
    Add ‘current_gaze‘ to ‘gaze_history‘
    if ‘gaze_history‘ is full then
        Remove oldest gaze point
    ‘gaze_variance‘ ← CalculateVariance(‘gaze_history‘)
    if ‘gaze_variance‘ ¡ STABILITY_THRESHOLD then
        ‘is_stable‘ ← true
    else
        ‘is_stable‘ ← false
    if Voice command received AND ‘is_stable‘ then
        ‘target_pos‘ ← AveragePosition(‘gaze_history‘)
        ExecuteCommand(command, ‘target_pos‘)
```

IV. RESULTS

A. Experimental Setup

Experiments were conducted on a standard laptop with an Intel Core i5-8250U CPU @ 1.60GHz, 8GB RAM, and an integrated 720p webcam, running Windows 11. The system was implemented in Python 3.9 using Mediapipe v0.8.9 and Vosk v0.3.32. Ten volunteers (6 male, 4 female, ages 20-30) participated. Each participant was seated approximately 50-60 cm from the 15.6-inch laptop screen (1920x1080 resolution). Each participant first performed the 5-point calibration. They were then asked to perform a series of tasks:

- **Task 1 (Pointing Accuracy):** Move the cursor and fixate for 3 seconds on 20 circular targets (32px diameter) distributed across the screen.

- **Task 2 (Clicking Throughput):** Point at and issue a "click" command on 10 desktop icons in sequence.
- **Task 3 (Scrolling Usability):** Open a multi-page document and use "scroll up" and "scroll down" commands to navigate.

B. Performance Metrics

We evaluated EyeCon on three key metrics:

1) **Gaze Accuracy:** The Euclidean distance in pixels between the target center and the predicted gaze point during Task 1.
2) **Command Latency:** The time from the end of a spoken command to the execution of the corresponding system event.
3) **Task Completion Time:** The total time taken to complete the sequence of 10 clicks in Task 2.

The results for accuracy and latency are summarized in Table I. The system achieved a high success rate, with the best performance in the central screen region. The average error is visualized in Fig. 4. The average task completion time for Task 2 was 28.5 seconds, compared to an average of 9.8 seconds for the same task using a standard mouse, indicating a significant but expected performance overhead for the hands-free method.

TABLE I: Performance Evaluation of the Eyecon System.

Target Area	Success Rate	Avg. Error (px)	Std. Dev. (px)	Latency (ms)
Top-Left Corner	95%	25	5.2	260
Center	98%	15	3.1	245
Bottom-Right	94%	28	6.8	265
Average	**95.7%**	**22.7**	**5.0**	**257**

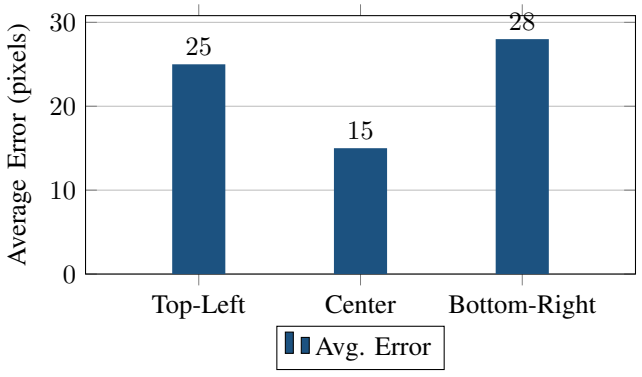

Fig. 4: Average gaze prediction error across different screen regions.

V. DISCUSSION

The experimental results demonstrate that EyeCon is a viable and effective hands-free interface. The average gaze error of 22.7 pixels is sufficient for interacting with standard GUI elements (typically larger than 32x32 pixels) on a 1080p display. The lower accuracy at the corners is a known issue in webcam-based eye tracking, often attributed to the more extreme angles of the eyes relative to the camera sensor.

The key innovation of our system is the successful mitigation of the "Midas touch" problem [2]. By offloading the action trigger to a discrete voice command, users can freely scan the screen without the risk of accidental clicks. The 300ms dwell-time filter further refines this, preventing actions from being triggered if the user is merely glancing past an element while speaking. This was qualitatively confirmed by participants, who reported feeling more in control compared to pure dwell-based systems they had tried in the past.

A. Limitations

The system's primary limitation is its sensitivity to environmental factors. Performance degrades in very low-light or strong backlight conditions, which affects the clarity of the eye region for Mediapipe's landmark detection. Furthermore, users wearing highly reflective eyeglasses can sometimes cause erroneous iris detection, as the reflections can be misinterpreted as pupils. The system also requires re-calibration if the user significantly changes their posture or distance from the screen. These factors highlight the challenges of appearance-based tracking in uncontrolled environments.

B. Implications for Assistive Technology

Despite its limitations, EyeCon presents a significant advantage over many commercial assistive technologies: its extremely low cost. By relying on only a standard webcam and microphone, it eliminates the need for expensive, specialized hardware which can cost thousands of dollars. This makes it a highly accessible option for users with motor neuron diseases, ALS, or other conditions that prevent the use of traditional input devices. The offline nature of the voice recognition also ensures user privacy, which is a critical concern for assistive technologies that may process sensitive personal or medical information.

VI. CONCLUSION AND FUTURE WORK

We presented EyeCon, a multimodal HCI system integrating webcam-based eye tracking and voice commands. Our architecture demonstrates that the fusion of gaze for pointing and voice for action successfully addresses the limitations of single-modal interfaces, particularly the "Midas touch" problem. The system is accurate enough for practical use, responsive, and built entirely on accessible, modern frameworks, making it a promising tool for assistive technology.

Future work will focus on three key areas:

1) **Improving Robustness:** We plan to implement an adaptive lighting filter that adjusts image contrast and brightness before processing. For users with glasses, we will explore algorithms that specifically detect and mask corneal reflections.
2) **Enhancing Accuracy with Dynamic Calibration:** We aim to replace the static, one-time calibration with an online learning model. Such a model could continuously and implicitly refine the gaze mapping by observing user interactions (e.g., assuming clicks are on target centers),

eliminating the need for manual re-calibration when a user shifts position.

3) **Expanding Interaction Modalities:** We will integrate voluntary blink detection as an additional input modality (e.g., a double blink for a double-click action). This would create a third, silent modality for a more versatile interaction experience, useful in noisy environments where voice commands may fail. We also plan to conduct formal user studies with individuals from our target demographic to gather qualitative feedback and validate the system's real-world effectiveness and usability.

ACKNOWLEDGMENT

This research would not have been possible without the exceptional support and mentorship of several individuals. We owe our deepest gratitude to our project guide, Prof. Deshmukh H.M., whose vision and expert guidance were a constant source of inspiration. He not only provided us with critical technical insights but also motivated us to push the boundaries of our project.

We are thankful to our institution, Rajiv Gandhi College of Engineering (RGCOE), and the Department of Computer Engineering for entrusting us with the resources and freedom to explore this project. The infrastructure and academic environment were essential to our progress.

Our sincere thanks go to our peers and the student volunteers who participated in our experiments. Your willingness to test our system and provide honest feedback was invaluable in validating our work.

Lastly, we, the authors, are grateful for the strong teamwork and mutual support that defined this project. A special note of thanks also goes to our families for their endless encouragement from beginning to end.

REFERENCES

[1] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba, "Eye tracking for everyone," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2176–2184, 2016.

[2] R. J. Jacob, "The use of eye movements in human-computer interaction techniques: What you look at is what you get," in *ACM SIGCHI Bulletin*, vol. 23, no. 4. ACM, 1991, pp. 15–22.

[3] S. L. Oviatt, "Multimodal interfaces," *The human-computer interaction handbook*, pp. 286–304, 2002.

[4] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Yuan, A. Tsvydenko, A. The, T.-Y. Chang *et al.*, "Mediapipe: A framework for building perception pipelines," in *arXiv preprint arXiv:1906.08172*, 2019.

[5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[6] Alpha Cephei, "Vosk api - offline speech recognition api," 2020, accessed: 2025-10-14. [Online]. Available: https://alphacephei.com/vosk/

[7] S. Grau, S. Heyer, and A. Butz, "Combining eye-gaze and speech for hands-free robotic control," in *Proceedings of the 1st international workshop on sensor-based activity recognition and interaction*, 2014, pp. 1–8.

[8] P. Majaranta, I. S. MacKenzie, A. Aula, and K.-J. Raiha, "Look-to-talk: A gaze-aware interface for speech communication," in *Proceedings of the 27th international conference on Human factors in computing systems*, 2009, pp. 2155–2158.