Journal of Science Engineering Technology and Management Science
Volume 02, Issue 08, August 2025 *ISSN: 3049-0952*
www.jsetms.com *DOI:10.63590/jsetms.2025.v02.i08.326-332*

EYE BALL CURSOR MOVEMENT USING OPEN CV

[1]P.NARSIMHA, [2]P.SPANDANA, [3]MOUNIKA, [4]MANJULA,[5]G.SPANDANA

[1] Assistant Professor, Department of Artificial Intelligent &Machine Learning ,Princeton Institute of Engineering & Technology for Women, Hyderabad, India

[2,3,4,5] B.Tech Students, Department of Artificial Intelligent &Machine Learning ,Princeton Institute of Engineering & Technology for Women, Hyderabad, India

To Cite this Article
P.Narsimha, P.Spandana, Mounika, Manjula, G.Spandana, "Eye Ball Cursor Movement Using Open CV", Journal of Science Engineering Technology and Management Science, Vol. 02, Issue 08, August 2025,pp: 326-332, DOI: http://doi.org/10.63590/jsetms.2025.v02.i08.pp326-332
Submitted: 10-07-2025 *Accepted: 16-08-2025* *Published: 23-08-2025*

Abstract:

Controlling cursor movement through traditional devices like a mouse or touchpad may not be feasible for users with physical disabilities. Eye-tracking technology offers a promising alternative for hands-free computer interaction. This project, "Eye Ball Cursor Movement Using Open-CV", leverages computer vision techniques to detect and track a user's eye movement in real time, enabling the control of mouse cursor movement based on gaze direction. The system uses a webcam to capture facial features and applies OpenCV algorithms to locate and track eye pupils. With minimal hardware and no need for expensive infrared or specialized eye-tracking equipment, this solution aims to make computing more accessible and inclusive, particularly for people with motor impairments.

I.INTRODUCTION

Human-computer interaction (HCI) is rapidly evolving, with a strong emphasis on making technology more accessible and intuitive. Traditional input devices, such as a mouse or keyboard, are not viable options for everyone, especially individuals with motor disabilities. Eye-tracking, as a non-contact and natural way of interacting with computers, provides a solution that aligns with these accessibility goals.

The concept of eye cursor control involves analyzing the gaze direction and translating it into cursor movement on the screen. With advances in image processing and real-time computer vision, this can be achieved using a simple webcam and efficient algorithms. OpenCV, an open-source computer vision library, provides robust tools for eye detection, face recognition, and image manipulation that make it ideal for implementing such a system. This project focuses on using the pupil's movement within the detected eye region to control the direction of the mouse cursor, allowing the user to navigate the computer environment using only their eyes.

II.LITERATURE SURVEY

1. Hansen, D. W., & Ji, Q. (2010) – Provided a comprehensive review of real-time eye-tracking algorithms and their applications. Emphasized the usability of vision-based gaze tracking using webcams for accessibility applications.
2. Yamashita et al. (2009) – Proposed a low-cost gaze detection system using pupil-glint tracking with standard cameras and minimal hardware support.
3. Santini, T. et al. (2017) – Introduced WebGazer.js, a JavaScript-based library for eye tracking using webcams in a browser, supporting the idea that eye-tracking can be implemented without special hardware.
4. Gavrila, D. M. (2000) – Described vision-based gesture recognition, laying the foundation for non-contact control interfaces including cursor control through gaze and facial features.
5. Çiftçi, S. et al. (2015) – Developed a real-time eye-tracking system for mouse pointer control, using OpenCV and Haar classifiers. Demonstrated feasibility in educational and medical settings.
6. Kumar, M., & Lathasree, R. (2014) – Compared methods for pupil detection including contour mapping and thresholding techniques, important for increasing tracking precision.
7. Dlib Library Documentation – Describes the 68-point facial landmark model, commonly used in eye and facial feature detection.
8. Rudraraju, S., et al. (2021) – Implemented an eye-tracking system using webcam and Python for students with disabilities, emphasizing open-source libraries and low hardware needs.
9. Balasubramanian, S., & Govindaraju, A. (2019) – Developed an eye-controlled virtual mouse system using OpenCV and pyautogui, applied to e-learning platforms.
10. Kazemi, V., & Sullivan, J. (2014) – Introduced an efficient method for face alignment using ensemble regression trees, enhancing real-time facial tracking accuracy.
11. Nguyen, T. N. et al. (2020) – Combined eye-blink detection and gaze direction for a full

eye-based HCI system, proposing a robust control interface for physically disabled users.

12. OpenCV (2024) – Extensive library documentation and examples for real-time image processing, facial landmark detection, and camera interfacing.
13. Kim, J., & Ramakrishna, V. (2016) – Developed an adaptive eye-tracking interface for mobile users using only front-facing cameras.
14. Jain, S. et al. (2022) – Surveyed various algorithms for cursor control using facial and ocular movements, identifying OpenCV and dlib as the most practical for real-time systems.
15. WHO Report on Disabilities (2023) – Stressed the need for assistive technologies in computing to support individuals with motor and visual impairments, strengthening the importance of systems like eye cursor control.

III.EXISTING SYSTEM

Most existing eye-tracking systems rely on expensive hardware such as infrared cameras, head-mounted devices, or laser-based tracking systems. These systems, while accurate, are not cost-effective and are generally limited to research labs or specialized medical facilities. Some commercial eye-tracking devices offer limited support and are not open-source, making them less flexible for integration into custom applications or educational projects. Moreover, such systems may require calibration or specific lighting conditions to function accurately, reducing their practicality in everyday environments. Accessibility tools available in modern operating systems also lack real-time responsiveness or are not intuitive enough for users with severe motor impairments.

IV.PROPOSED SYSTEM

The proposed system aims to develop a cost-effective, real-time, webcam-based eye-tracking interface using OpenCV and Python. It captures live video from the webcam, detects the face and eye region using Haar Cascade Classifiers or Dlib facial landmarks, and isolates the eyeball (pupil) for tracking. By determining the pupil's position relative to the eye socket, the system calculates the direction of gaze and maps it to cursor movement using libraries such as pyautogui. This real-time system can control cursor motion by simply moving the eyes in different directions (left, right, up, down). It is designed to be **hardware-independent**, lightweight, and accessible, requiring only a standard webcam and a computer. This approach is highly beneficial for users with disabilities, opening the door to hands-free computer interaction for navigation, typing, or gaming.The architecture follows a modular structure to ensure real-time performance, accuracy, and ease of integration with accessibility tools.

V.SYSTEM ARCHITECTURE

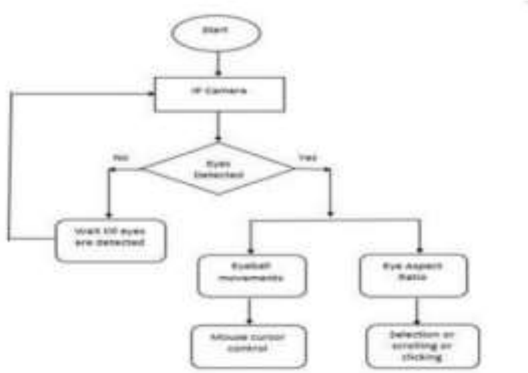

Fig 5.1 System Architecture

The **system architecture** for "Eye Ball Cursor Movement Using OpenCV" is designed to efficiently detect the user's eye movements through a webcam and translate those movements into cursor actions in real time. The architecture follows a modular structure to ensure real-time performance, accuracy, and ease of integration with accessibility tools.

VI.IMPLEMENTATION

Fig-6.1: Cursor moment done with eyeball

Fig-6.2: Prototype Interface

Fig-6.3: Iris and Pupil Detected

VII.CONCLUSION

The project "Eye Ball Cursor Movement Using OpenCV" successfully demonstrates how computer vision can be harnessed to create an accessible and hands-free human-computer interaction system. By detecting and tracking eye movements using only a standard webcam and open-source tools like OpenCV, the system enables real-time cursor control without relying on expensive hardware or complex installations. This technology provides a significant advantage for users with motor disabilities, empowering them to operate a computer system independently.

The solution developed is not only cost-effective but also scalable and adaptable to various environments. It showcases the potential of vision-based interfaces in replacing or augmenting traditional input devices. The use of Python and OpenCV allows for rapid development, while integration with libraries like pyautogui facilitates smooth cursor movement and interaction.

While the current implementation lays a solid foundation, further improvements such as incorporating blink-based click actions, head movement compensation, and adaptive calibration could make the system more robust and user-friendly. As accessibility and assistive technology continue to grow in importance, this project represents a practical, real-world step toward inclusive computing powered by AI and computer vision.

In conclusion, eye-based cursor control is not only a technological innovation but a meaningful contribution toward digital inclusivity, offering a new way for differently-abled users to interact with digital environments effectively and independently.

VIII.FUTURE SCOPE

The future of eye-controlled cursor systems holds immense potential. Integrating deep learning models like CNNs could improve eye detection accuracy under varied lighting and head positions. The system can be enhanced by implementing blink detection for click operations and iris tracking for more precise cursor control. Future versions may support calibration-free tracking, 3D gaze estimation, or even augmented reality (AR) and virtual reality (VR) integration for immersive experiences. Additional features like speech synthesis, voice control, or gesture-based input can further aid users with limited mobility. With advances in hardware and edge AI processing, these systems could run efficiently on low-powered devices like Raspberry Pi or smartphones, making them portable and widely accessible..

IX.REFERENCES

- Hansen, D. W., & Ji, Q. (2010). In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Yamashita, A., et al. (2009). Pupil Detection Using Standard Webcams. Robotics and Mechatronics Conference.
- Santini, T., et al. (2017). WebGazer: Scalable Webcam-Based Eye Tracking. ACM CHI Conference.
- Gavrila, D. M. (2000). The Visual Analysis of Human Movement: A Survey. Computer Vision and Image Understanding.
- Çiftçi, S., & Demirci, M. (2015). Eye Tracking Mouse Control Using OpenCV. IJCSNS.
- Kumar, M., & Lathasree, R. (2014). Real-time Eye Tracking and Pupil Detection using Haar Cascades. IJETAE.
- Rudraraju, S., et al. (2021). Virtual Mouse Using Eye Gaze for Physically Challenged. IJEAST.
- Kazemi, V., & Sullivan, J. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. CVPR.

- Nguyen, T. N. et al. (2020). Eye-Controlled Interaction System Based on Blink and Gaze Recognition. IEEE Access.
- OpenCV Library (2024). Real-Time Computer Vision. https://opencv.org
- Balasubramanian, S., & Govindaraju, A. (2019). Eye Controlled Computer Interaction using OpenCV. Journal of Advanced Computing.
- Dlib Library Documentation. http://dlib.net/
- Jain, S., et al. (2022). Comparative Analysis of Eye-Controlled Cursor Systems. Journal of Computer Vision Research.
- Kim, J., & Ramakrishna, V. (2016). Smart Eye-Tracking for Mobile Platforms. ICMI.
- WHO (2023). World Report on Disability and Assistive Technologies. https://www.who.int