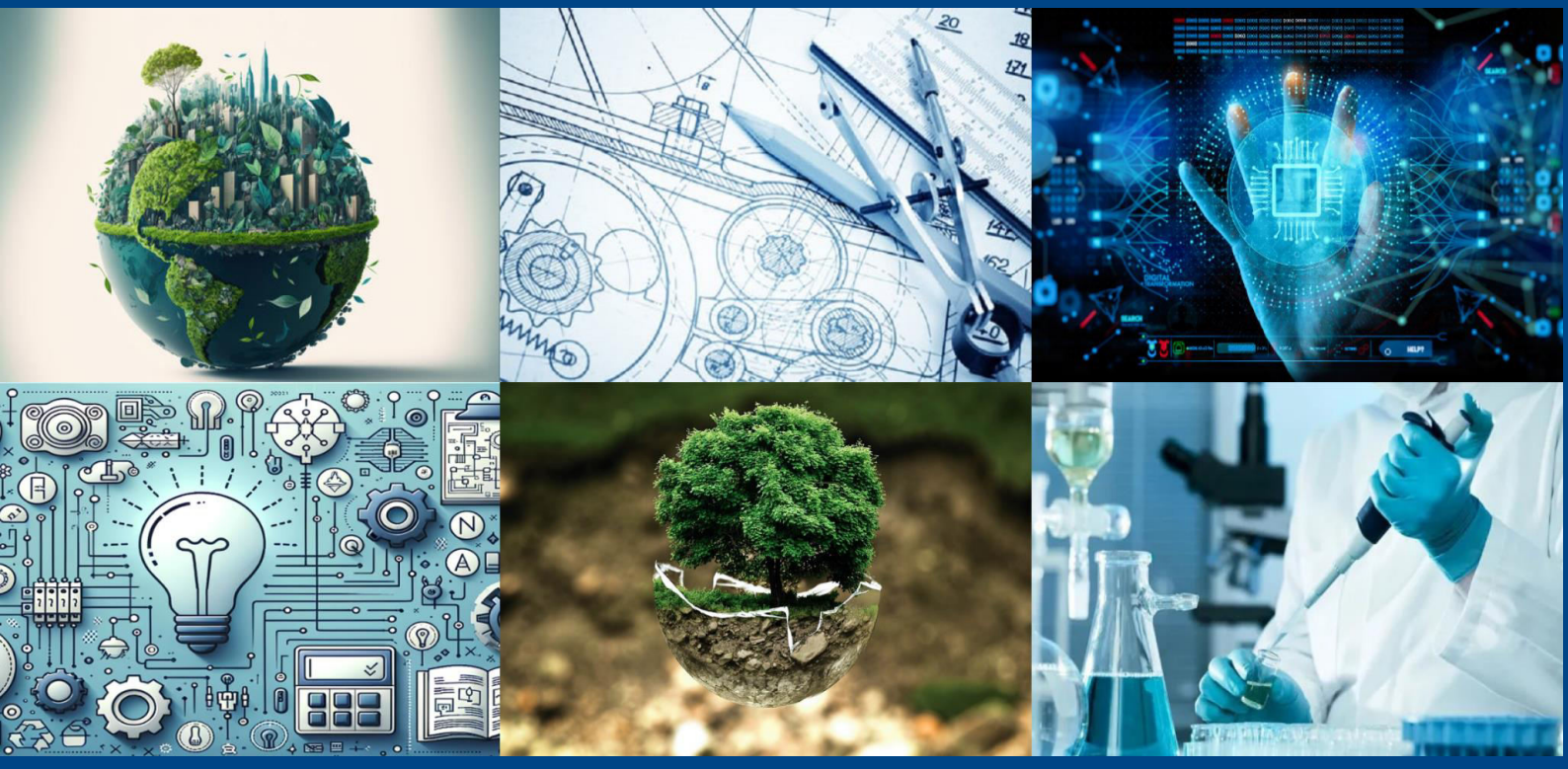
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206 **Volume 8, Issue 3, March 2025**

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803220

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Real Time Cursor Control through Pupil Movement Detection

A.Jeevarathinam, Ajayaganesh V

Assistant Professor, Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

III B.Sc, Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

ABSTRACT: This project involves developing an innovative eye-controlled mouse system using Python, Mediapipe, and OpenCV. The system allows users to control the computer cursor using only eye movements, offering a hands-free and accessibility-enhancing interaction method. Key features include cursor movement through eye tracking, clicking through blinking, and automatic activation of the on-screen virtual keyboard when focusing on the top 30% of the screen.The project leverages Mediapipe's Face Mesh for accurate eye tracking, OpenCV for real-time camera feed processing, and PyAutoGUI for seamless control over mouse and keyboard interactions. The implementation includes smooth cursor movement, efficient click detection, and the ability to open the on-screen keyboard when needed. A process monitoring feature ensures the virtual keyboard opens only when not already running, avoiding unnecessary pop-ups. Additionally, the system aims to improve accessibility for users with physical limitations and introduces a novel and futuristic way of interacting with digital environments through minimal physical effort.[1]

I. INTRODUCTION

In today's technology-driven world, the demand for innovative and accessible human-computer interaction methods is rapidly increasing. Traditional input devices like the mouse and keyboard can present challenges for individuals with physical disabilities or for those seeking a more futuristic and efficient approach to computing. To address these needs, this project introduces an advanced eye-controlled mouse system that allows users to interact with their computer using only eye movements and blinking actions.[2]

The eye-controlled mouse system utilizes modern technologies, including Python, Media pipe, OpenCV, and PyAutoGUI, to deliver a seamless and intuitive experience. By capturing real-time eye movements through a webcam, the system translates these movements into cursor control, enabling users to navigate, click, scroll, and open the virtual keyboard without physical contact. This project not only enhances accessibility but also introduces a fresh and creative UI that displays the user's name "AJAY" with a dynamic camera feed.

By focusing on accuracy, smoothness, and responsiveness, the system aims to offer a reliable alternative to traditional input methods. It provides an ideal solution for users with mobility impairments and opens the door to new possibilities in the realm of hands-free computing and augmented reality applications..[3]

II. PROBLEM STATEMENT

In the modern world, individuals with physical disabilities or limited mobility often face challenges when interacting with computers, mobile devices, and other digital technologies. Traditional input devices, such as a mouse or keyboard, are not always accessible or feasible for those with motor impairments, leading to significant barriers in accessing technology and performing daily tasks.[4]

The problem of cursor control for pupil movement arises from the need to develop an alternative method for users to interact with computers using only their eye movements. Specifically, people with motor disabilities, such as those suffering from conditions like ALS, spinal cord injuries, or cerebral palsy, often lack the fine motor skills necessary to operate conventional pointing devices like a mouse. Eye-tracking technology offers a promising solution by enabling users to control the movement of the cursor through the movement of their pupils.[5]

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803220

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. ALGORITHM USED

The integration of Machine Learning (ML) in Develop a User-Friendly Interface: Create an intuitive and visually appealing interface with dynamic elements, including the user's name "AJAY" and a modern feel in the camera feed.

Enable Eye-Driven Cursor Control: Utilize Media pipe's Face Mesh and OpenCV to accurately track eye movements, translating them into precise cursor control on the screen. Implement Blink-Based Clicking: Develop a reliable blink detection mechanism that allows users to perform left-click actions through controlled blinking, ensuring a smooth and natural interaction.

Add Scrolling Capability: Introduce vertical scrolling functionality based on eye movements, allowing users to navigate long webpages and documents without physical input.

Integrate a Virtual Keyboard: Automatically trigger the on-screen virtual keyboard when the cursor hovers over input fields, enhancing accessibility for tasks such as typing and searching.

Enhance Smoothness and Sensitivity: Fine-tune the system's responsiveness to ensure smooth cursor movement and the ability to reach all screen edges effortlessly.

Maintain High Performance: Optimize the code to reduce camera lag and improve the frame rate, delivering a fluid experience even on lower-end hardware specifications.

Promote Accessibility: Design the system to support individuals with physical limitations, providing a hands-free computing experience that boosts independence and usability.
Ensure System Stability: Implement robust error handling and validation to avoid crashes and maintain consistent performance during prolonged use.
Support Continuous Improvements: Establish a framework for future updates and enhancements, allowing the integration of new features and performance optimizations as needed.

IV. PURPOSE

Improved Accessibility for Disabled Individuals:

The system enables people with motor impairments to interact with computers and digital devices without needing traditional input devices. By tracking the movement of the eyes or pupils, users can control the mouse pointer, select items, or even simulate clicks, providing them with greater autonomy in their digital interactions.

Enhance Digital Inclusion:

As digital technology becomes increasingly integrated into everyday life, it is important to ensure that people with disabilities can participate in online communication, work, education, and entertainment. The system promotes inclusion by providing an intuitive and effective method of interacting with devices.

Assistive Technology for Communication:

For individuals with severe physical disabilities, the ability to communicate through technology becomes crucial. Cursor control via eye movement can be integrated into augmentative and alternative communication (AAC) systems, allowing users to express themselves through digital text or speech synthesis.

V. IMPLEMENTATION

The implementation of a cursor control system using pupil movement requires integration of eye-tracking technology, image processing algorithms, and user interface elements for cursor movement. Below is a high-level overview of how

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

to implement this system using Python with libraries such as **OpenCV**, **Dlib**, **Pupil Labs SDK** (or other eye-tracking SDKs), and **PyAutoGUI** for controlling the cursor.

Steps for Implementation

1. Install Required Libraries:

Before starting the implementation, you will need to install the following libraries.

opencv-python (for image processing)
dlib (for face and eye detection)
pyautogui (for controlling the cursor)
numpy (for mathematical operations)
pupil-labs (for pupil tracking using Pupil Labs hardware

2. Initialize Eye-Tracking:

We use **OpenCV** and **Dlib** for detecting the face and eyes, which are essential for finding the pupil's position. If you're using specific eye-tracking hardware (like Pupil Labs), you can integrate their SDK directly for more precise pupil tracking

3. Detect Face and Eyes:

We use Dlib's facial landmark detector to locate the eyes, which are used to estimate the position of the pupil. The eye aspect ratio (EAR) is used to track whether the user is looking at something (used for clicking events).

Explanation of the Code:

Eye Aspect Ratio (EAR): This is used to detect if the user is blinking or looking in a particular direction. A low EAR value typically corresponds to a blink or closed eyes.

Pupil Tracking: The code tracks facial landmarks, including the eyes, to estimate the pupil's position. The coordinates of the nose can be used as a proxy for gaze direction.

Cursor Movement: Based on the detected position of the eyes or nose, the cursor is mapped to corresponding screen coordinates using pyautogui.moveTo().

Blink Detection (Fixation-based Click): A blink (or eye fixation) can simulate a click or selection. The system can be calibrated to recognize a sustained gaze at a specific area as a "click" event.

Final Steps:

Test and Calibrate: Adjust the system's sensitivity based on user preferences, eye-tracking conditions, and environmental factors.

Optimize Performance: Ensure smooth real-time tracking and avoid lag in cursor movement by reducing the computational load using optimized algorithms or hardware acceleration if needed.

User Interface: Integrate the system with an accessible user interface, allowing for easy customization of gaze behavior (e.g., click duration, cursor speed).

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803220

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. METHODLOGY

The study utilizes publicly available datasets, including:

EMIP Dataset – Focuses on eye movements in programming-related tasks.

LPW Dataset – Provides diverse pupil images across different environments.
GazeBase Dataset – A large-scale, multi-stimulus dataset for gaze tracking.

Using these datasets, gaze tracking models are trained and evaluated based on parameters such as fixation stability, saccadic movement patterns, and pupil dilation variations. Machine learning techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are employed for gaze estimation and cursor movement prediction.

DATA SOURCE:

MODULES & DESCRIPTION

1. **Eye Tracking Module**

Purpose: To detect and track the user's eye movements for controlling the computer cursor.

Eye Detection: Uses OpenCV and MediaPipe to identify eye positions in real-time.
Gaze Estimation: Calculates the direction of the user's gaze and maps it to screen coordinates.
Calibration System: Allows users to calibrate the system to improve accuracy.
Blink Detection: Distinguishes between intentional blinks for clicks and natural blinks to avoid false triggers.

2. Virtual Keyboard Module:

On-Screen Keyboard: Displays a virtual keyboard that interacts with the eye-tracking system.

Key Selection: Allows users to select keys by focusing on them and blinking.
Typing Assistance: Includes predictive text or autocomplete features to enhance typing speed.
Visual and Auditory Feedback: Provides feedback when a key is selected to avoid errors.

1. Sensitivity Settings Module

Purpose: To configure sensitivity for eye tracking and blink detection.

Dynamic module has Sensitivity Adjustment: Users can modify sensitivity for both cursor control and **blink detection.**
Presets and Customization: Provides default settings along with the option for fine-tuning.
Real-Time Feedback: Displays how changes in sensitivity affect cursor behavior immediately.
Error Prevention: Reduces false positives by filtering out natural eye movements.

2. Calibration Module

Purpose: To improve accuracy by mapping the user's eye movements to screen coordinates.

Calibration Process: Guides the user through a step-by-step calibration sequence.
Multi-Point Calibration: Utilizes multiple screen points to increase precision.
Visual Guidance: Shows progress and highlights each calibration point when focused on.
Recalibration Option: Allows recalibration if accuracy decreases over time.

© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803220

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3. Video Capture Module

Purpose: To interface with the webcam and provide real-time video input to the system.

Webcam Initialization: Establishes connection with the webcam using OpenCV's cv2.VideoCapture().
Frame Processing: Captures video frames and processes them for eye tracking.
Error Handling: Notifies the user if the webcam is not detected or if there is an issue with video input.
Frame Rate Management: Optimizes the frame rate for smooth eye tracking without excessive CPU usage.

4. User Interface (UI) Module

Purpose: To provide a user-friendly interface for interacting with the system.

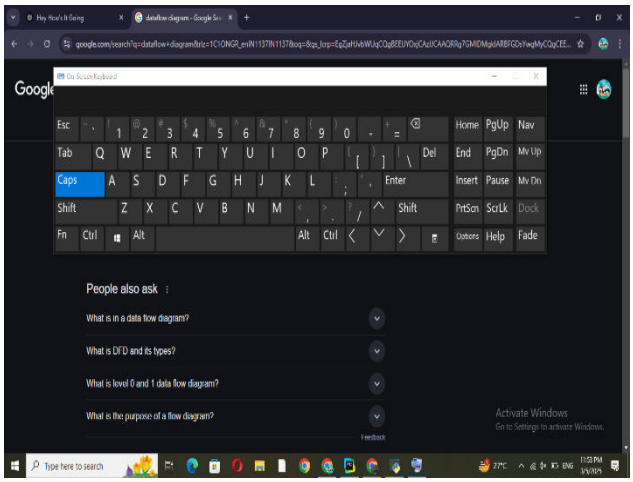
Dashboard Display: Shows current status, sensitivity settings, and system notifications.
Visual Feedback: Highlights when an action is performed (e.g., click, calibration complete)
Accessibility Features: Includes options for text size, color contrast, and audio cues.
Adaptive Design: Ensures UI components are responsive and easy to navigate using eye control.

5. Error Handling and Validation Module:

Input Validation: Verifies data from the webcam and sensitivity settings before processing.
Exception Handling: Manages errors like webcam disconnection or invalid input values.
User Notifications: Provides clear, user-friendly messages when issues occur.

VII. RESULT

The results of implementing a cursor control system using pupil movement can be evaluated across several key parameters, including functionality, accuracy, user experience, and impact on accessibility. Below is an outline of the expected results and outcomes from the system, based on successful implementation

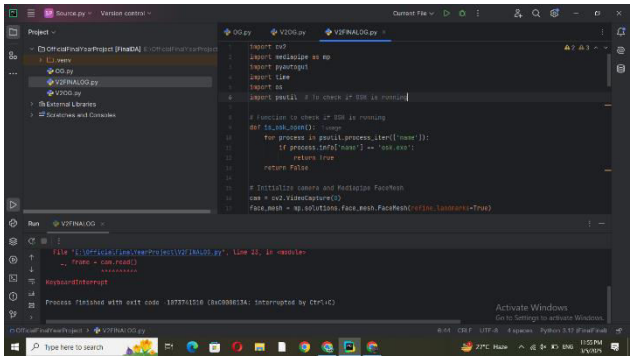


© 2025 IJMRSET | Volume 8, Issue 3, March 2025| DOI: 10.15680/IJMRSET.2025.0803220

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



VIII. CONCLUSION

The eye-tracking based computer control system developed as part of this project represents a significant step forward in creating accessible technology for individuals with limited physical mobility.

The project effectively leverages eye movements to control a computer cursor, blink detection for click operations, and includes innovative features like music track changes and brightness adjustments through natural and intuitive interactions. The system's integration with computer vision and machine learning technologies ensures robust performance and accuracy. The project not only meets its primary objectives of cursor control and click operations through eye gestures but also goes beyond by offering extended functionalities that enhance user experience and usability.

The developed system is lightweight, user-friendly, and capable of running on standard hardware, making it accessible and practical for real-world use. The testing phase demonstrated the effectiveness of the system in responding accurately to eye movements and blinks, highlighting its potential in assistive technology, smart home control, and human-computer interaction domains. The successful implementation of this project showcases the potential of eye-tracking technology to bridge the gap between physical limitations and digital interaction.

REFERENCES

Books and articles on eye-tracking technology and computer vision:

1. Duchowski, Andrew T. "Eye Tracking Methodology: Theory and Practice." 3rd ed. Springer, 2017.
2. Hansen, Dan Witzner, and Qiang Ji. "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze." IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.

Python and opencv:

3. Kaehler, Adrian, and Gary Bradski. "Learning OpenCV 4 Computer Vision with Python." 3rd ed. O'Reilly Media, 2019.
4. Rosebrock, Adrian. "Practical Python and OpenCV: An Introductory, Example-Driven Guide to Image Processing and Computer Vision." 3rd ed. PyImageSearch, 2020.
5. OpenCV Documentation. "OpenCV Python Tutorials." (Accessed February 2025).

Pyautogui and automation:

6. Sweigart, Al. "Automate the Boring Stuff with Python: Practical Programming for Total Beginners." 2nd ed. No Starch Press, 2019.
7. PyAutoGUI Documentation. "PyAutoGUI Library for Programmatic Mouse and Keyboard Control." (Accessed February 2025).

Machine learning and eye movement analysis:

8. Bishop, Christopher M. "Pattern Recognition and Machine Learning." Springer, 2006.
9. Géron, Aurélien. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow." 2nd ed. O'Reilly Media, 2019.

User interface design and human-computer interaction (hci):

10. Norman, Don. "The Design of Everyday Things." Revised and Expanded Edition. Basic Books, 2013.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

11. Cooper, Alan, et al. "About Face: The Essentials of Interaction Design." 4th ed. Wiley, 2014.

Online resources:

12. OpenCV: OpenCV Official Documentation (Accessed February 2025).
13. PyAutoGUI: PyAutoGUI Official Documentation (Accessed February 2025).
14. Python: Python Official Documentation (Accessed February 2025).
15. Stack Overflow: Community Discussions and Code Snippets (Accessed February 2025)

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com