# Cursor Pointer Control Using Eye and Hand Gestures

Bharath S
*Department of Computer Applications*
*PES University, Bengaluru, India.*
bharath7sms@gmail.com

***Abstract*— This project explores an innovative approach to human-computer interaction by combining eye and hand gestures to control the cursor pointer on a computer screen. The traditional input devices such as a mouse or touchpad are augmented with a system that utilizes both eye-tracking technology and hand gestures for a more intuitive and immersive user experience. The UI presents users to switch between two distinct interaction mode. The eye gestures enable precise cursor control based on the user's gaze, allowing users to perform cursor movements and click operations. The hand-gestures enable users to perform click operations, cursor movement, drag and drop, volume control and scrolling. The integration of these two input modalities creates a multifaceted and adaptable system that caters to variety of user preferences and accessibility needs.**

**Keywords—eye, hand, gestures, python, cursor**

## I. INTRODUCTION

In today's world, where computers have become an indispensable part of our daily lives, the way we interact with them is increasingly important. Traditionally, we have relied on familiar tools like the mouse and touchpad to navigate our screens. However, this project aims to revolutionize this interaction by introducing a new, seamless, and intuitive method.

By combining cutting-edge eye-tracking technology with intuitive hand gestures, we are introducing a system that responds not only to where you look but also to how you move your hands. This innovative approach allows users to control their computers in a more natural and effortless manner.

Imagine being able to move the cursor around the screen simply by looking at it, or performing actions like clicking and dragging with simple hand and eye gestures. This level of integration between human movement and computer response is groundbreaking and has the potential to transform the way we use technology on a daily basis.

One of the primary goals of this project is to make computer use more accessible to everyone, including those who may find traditional input devices challenging. By eliminating the need for complex peripherals and relying instead on natural human gestures, we are breaking down barriers to entry and opening up new possibilities for users of all abilities.

But our vision extends beyond mere functionality. We are not only focused on making the technology work; we are equally dedicated to making it easy and enjoyable for people to use. By letting your eyes and hands take the lead, we are striving to create a computer experience that feels friendly, intuitive, and effortless.

Through extensive research, development, and testing, we are working tirelessly to refine our system and ensure that it meets the needs and expectations of users from all walks of life. From enhancing productivity in the workplace to providing new avenues for entertainment and creativity, the potential applications of this technology are virtually limitless.

Ultimately, our aim is to empower users to interact with their computers in a way that feels natural and intuitive, enhancing their overall experience and enabling them to accomplish more with less effort. With our eyes and hands as our guides, we are charting a course towards a future where technology seamlessly integrates into our lives, enriching our experiences and expanding our possibilities.

### *A. Problem Scenario*

Interacting with a computer cursor through the standard mouse or touchpad can often prove to be a challenging and uncomfortable experience for many users. Regardless of age or ability, individuals frequently encounter difficulties when attempting to navigate the computer interface using these traditional input devices. This widespread issue is further exacerbated by the potential for discomfort and the risk of repetitive strain injuries associated with prolonged use of the conventional mouse. The limitations of conventional input devices extend beyond mere physical discomfort. They may also fail to provide the ease and comfort necessary for an enjoyable computing experience. Recognizing these challenges, our project endeavors to tackle the issues associated with traditional cursor control methods, aiming to introduce a more accessible and user-friendly alternative.

*B. Proposed Solution*

This project focuses on revolutionizing the way we interact with the computer cursor by eliminating the need for a conventional mouse or touchpad. Instead, we introduce a novel approach that allows users to control the cursor through intuitive hand and eye gestures. Picture being able to effortlessly guide the cursor across the screen by simply moving your hand or looking at where you want it to go. It's akin to playing with your computer using natural, instinctive movements, making the entire interaction process both enjoyable and accessible to everyone. This approach not only simplifies cursor control but also opens up new possibilities for a more intuitive and engaging computing experience, catering to a broad spectrum of users.

## II. LITERATURE REVIEW

The literature review plays a critical role in laying the foundation for the current research by examining existing literature related to the problem domain. The primary aim is to investigate the research already conducted on similar problems, thereby gaining insights into the available data, materials, and findings.

*A. Existing Systems*

In the paper titled “Eye Tracking Mouse for Human-Computer Interaction,” presented at the 2013 E-Health and Bioengineering Conference (EHB) by authors Robert Gabriel Lupu, Florina Ungureanu, and Valentin Siriteanu, they introduced an innovative system utilizing eye-tracking techniques to facilitate human-computer interaction. The system, designed for mouse control based on eye movements, showcased advancements in reliable and hands-free computer interaction. However, a significant limitation was identified—specifically, the system ceased to function when any liquid, such as eyeliner or mascara, was present in the eyes. This constraint, particularly pertinent for female users who frequently use eye cosmetics, highlights a scenario in which the system's effectiveness is compromised. The research underscores the importance of addressing such limitations to enhance the robustness and adaptability of eye-tracking technology for diverse user scenarios in real-world applications. [1]

In their 2014 paper titled "Face and Eye Tracking for Controlling Computer Functions," presented at the 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON), authors Chairat Kraichan and Suree Pumrin introduced a face and eye-controlled system developed using MATLAB. The system employed a webcam to enable mouse control through facial and eye movements, showcasing an innovative approach to hands-free computer interaction. However, a notable limitation was identified—the system's effective functionality was confined to a few centimeters, potentially restricting its practical usability. This work contributes to the field of human-computer interaction and highlights the importance of addressing operational constraints to enhance the system's adaptability for broader application. [2]

The paper titled "Pupil Centre Coordinates Detection Using the Circular Hough Transform Technique," published in the 2015 38th International Spring Seminar on Electronics Technology (ISSE) by Radu Gabriel Bozomitu, Alexandru Pasarica, Vlad Cehan, Cristian Rotariu, and Constantin Barabasa, introduces a system employing Hough Transform Techniques through a webcam for detecting pupil center coordinates. While demonstrating an effective method for pupil detection, the system faces a significant limitation—it operates at a non-real-time pace. The sequential process of capturing the body, face, eyes, and then the pupil results in substantial time delays, hindering the system's responsiveness. This drawback highlights the need for optimizations to transform the system into a real-time solution and enhance its practical applicability in various contexts. [3]

The In 2020 Gubbala Durga Prasanth and P. Srinivas Reddy proposed a paper on “Virtual mouse implementation using OpenCV”. They used OpenCV, Deep learning, NumPy and anaconda technology. Its major drawback is that it is very complex to use. [4]

In 1990, Quam introduced an early hardware-based system; in this system, the user should wear a Data Glove. The proposed system by Quam although gives results of higher accuracy, but it is difficult to perform some of the gesture controls using the system. Data Gloves typically had sensors embedded in the glove to detect hand and finger movements. While they could provide a more accurate representation of hand gestures compared to some other methods, they did have limitations. [5]

## III. HARDWARE AND SOFTWARE REQUIREMENTS

*A. Hardware*

1) *Laptop or PC with Web Camera*
2) *8GB RAM*
3) *256GB or more SSD/HDD*
4) *Intel i3 or AMD Ryzen 5 Processor*

*B. Software*

1) *Windows Operating System*
2) *Python*

## IV. SYSTEM DESIGN

### A. Architecture Diagram

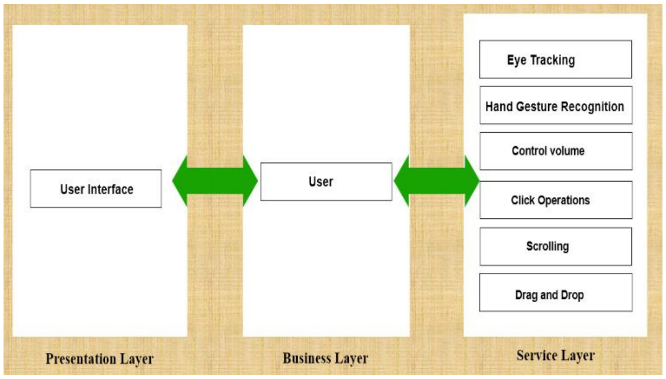


Figure 1 Architecture Diagram

In this system, users seamlessly provide input through their eyes and hands. The system captures and interprets the user's natural eye and hand gestures, translating them into precise on-screen cursor motions and facilitating mouse-free interactions.

The Presentation Layer is the user-facing component of the cursor control using eye and hand gesture system. It encompasses the graphical user interface (GUI) elements, providing users with an intuitive and visually effective platform to interact with the system. This layer includes on-screen cursor representation focusing on delivering a user-friendly experience.

The business layer also known as the logic layer, is the layer responsible for handling the core business logic and functionality of the system. In this layer, the system's fundamental operations, algorithms, and data processing occur.

The users can control the cursor by making hand gestures in the air. The system captures the movements and translates them into on-screen actions. Hand gestures can be used to simulate mouse clicks and perform selection.

Users can control the cursor on the screen by simply moving their eyes. Eye-tracking technology detects the direction of gaze and translates it into cursor movement.

The Click functionality in cursor pointer control using eye and hand gestures refers to the action of simulating a mouse click without physically pressing a button. This is done by both hand gesture or by just blinking the eye.

The Scrolling functionality in cursor pointer control using hand gestures allows users to scrolling through content, such as web pages or documents, without physically interacting with a traditional scrolling device like a mouse wheel or touchpad. The users just have to use their hand and specified gesture to scroll.

The Drag and drop functionality allows users to interact with digital content by selecting, moving, and dropping items on a screen. When using hand gestures for drag and drop, the goal is to simulate these actions without the need for physical devices like a mouse.

Volume control using hand gestures involves manipulating audio levels without the need for physical buttons or sliders.

### B. Block Diagram

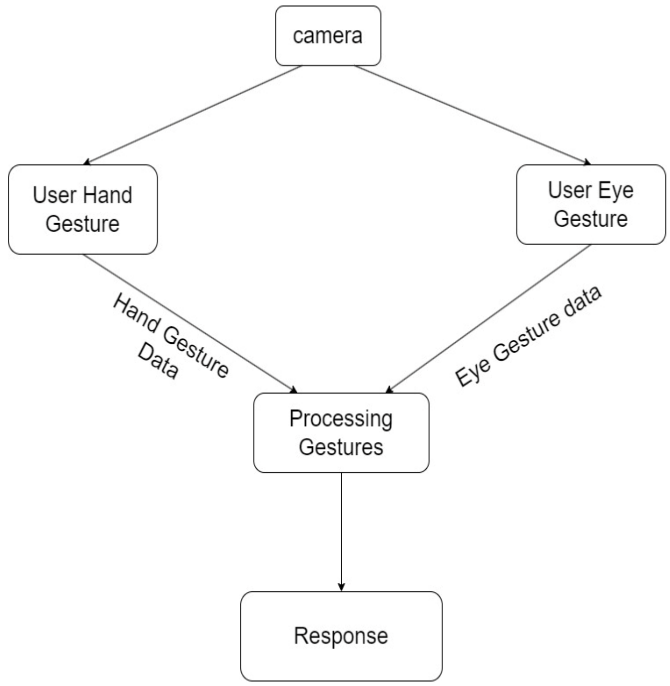


Figure 2 Block diagram

The diagram shows a system for controlling a computer cursor using both hand gestures and eye tracking.

The hand and eye gestures will be captured by a camera. The hand gesture data and eye tracking data are the inputs to the system. The hand gesture data is captured by the camera or sensor and processed to determine the specific gesture being made. The eye tracking data is used to determine where the user is looking on the screen.

The system processes the hand gesture data to determine the specific gesture being made. This may involve pattern recognition algorithms or other techniques to identify the gesture.

Once the gesture has been identified, control signals are generated to move the cursor on the screen. The control signals may be sent directly to the computer or may be processed further to refine the cursor movement.

The computer receives the control signals and moves the cursor accordingly.

The final component of the system is the cursor itself, which is moved on the screen in response to the hand gestures and eye tracking data.

## C. Process Flow

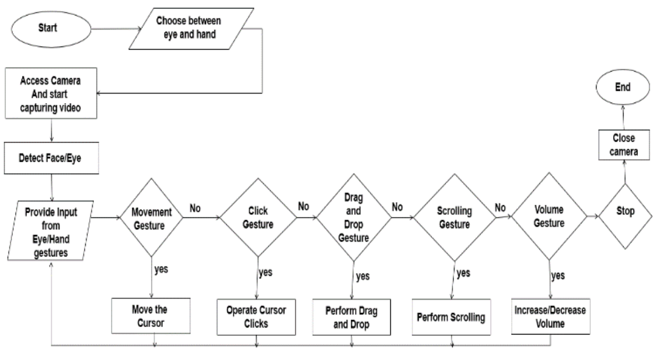


Figure 3 Process flow diagram

The process begins with starting the system and accessing the camera to capture video. The system then detects the user's hand and eyes using computer vision techniques. The system tracks the user's eye and hand movements to provide input for controlling the cursor and performing various actions. If the user moves their eyes or hands in a certain way, the system moves the cursor to the corresponding location on the screen. If the user performs a click gesture (e.g., by blinking their eyes or making a hand gesture), the system performs a click action on the current cursor location. If the user performs a drag-and-drop gesture through hands the system performs a drag-and-drop action on the corresponding screen elements. If the user performs a scrolling gesture, the system performs a scrolling action on the current screen. If the user performs a volume gesture, the system increases or decreases the volume accordingly. The process ends with closing the camera and stopping the system.

## D. Use Case Diagram

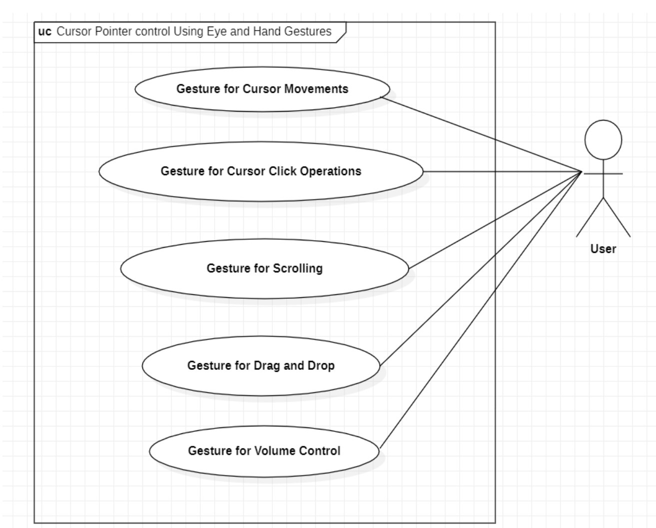


Figure 4 Use Case Diagram

The use case diagram for the cursor pointer control using eye and hand gestures system visually represents the interaction scenarios between users and the system. The primary actor is the user.

The primary actor represents the user interacting with the system. The person who will be using the system to control the cursor pointer.

1) *Gesture for Cursor Movements: A use case where the user can move the cursor pointer by performing a specific gesture with their eyes or hands.*

2) *Gesture for Cursor Click Operations*: A use case where the user can perform click operations (e.g. left-click, right-click) by performing a specific gesture with their eyes or hands.
3) *Gesture for Scrolling:* A use case where the user can scroll through a document or webpage by performing a specific gesture with their hands.
4) *Gesture for Drag and Drop:* A use case where the user can drag and drop items by performing a specific gesture with their hands.
5) *Gesture for Volume Control:* A use case where the user can control the volume of the system by performing a specific gesture with their hands.

## E. Activity Diagram

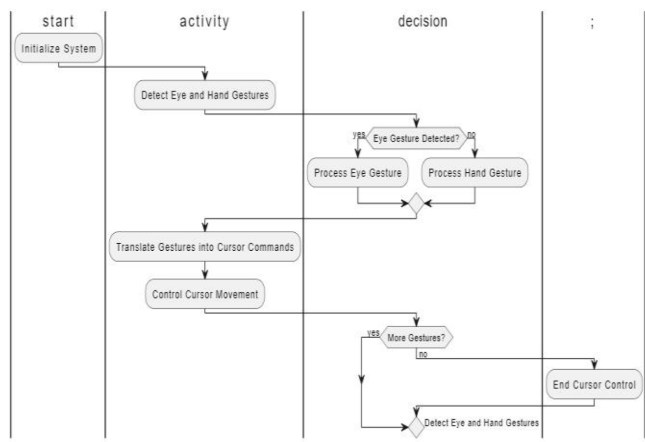


Figure 5 Activity Diagram

1) *Initialize System:* The process begins with initializing the system.
2) *Detect Eye and Hand Gestures:* The system continuously monitors and detects both eye and hand gestures.
3) *Decision Point:* The diagram branches based on whether an eye gesture is detected. If an eye gesture is detected, the system proceeds to Process Eye Gesture. If no eye gesture is detected, the system proceeds to Process Hand Gesture.

4) *Process Eye Gesture:* The system processes the detected eye gesture.

5) *Process Hand Gesture:* The system processes the detected hand gesture.

6) *Translate Gestures into Cursor Commands*: Regardless of the type of gesture (eye or hand), the system translates the detected gestures into cursor commands.

7) *Control Cursor Movement:* The translated cursor commands are used to control the movement of the cursor.

8) *Decision Point:* The diagram includes a decision point to check if there are more gestures to be detected. If there are more gestures, the system goes back to Detect Eye and Hand Gestures. If no more gestures are detected, the process proceeds to the end.

9) *End Cursor Control*: The diagram concludes with the end of the cursor control process.

## V. IMPLEMENTATION

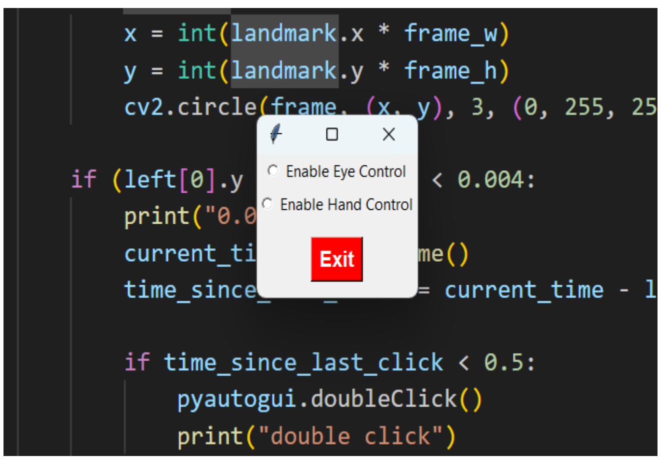


*Figure 6 UI To Choose Between Eye and Hand*

This UI presents users with a selection between two distinct interaction modes: hand-controlled input, which allows users to navigate and interact with the interface using hand gestures, and eye-tracking input, which enables users to control the interface by tracking their eye movements. By offering these options, the UI caters to a diverse range of user needs and preferences, ensuring that those who find traditional input methods challenging can still interact with the system effectively. Additionally, including an exit option provides users with a clear way to exit or close the interface, enhancing usability and ensuring a seamless user experience. This UI design not only prioritizes accessibility but also empowers users by giving them control over their interaction methods, ultimately contributing to a more inclusive and user-friendly interface.

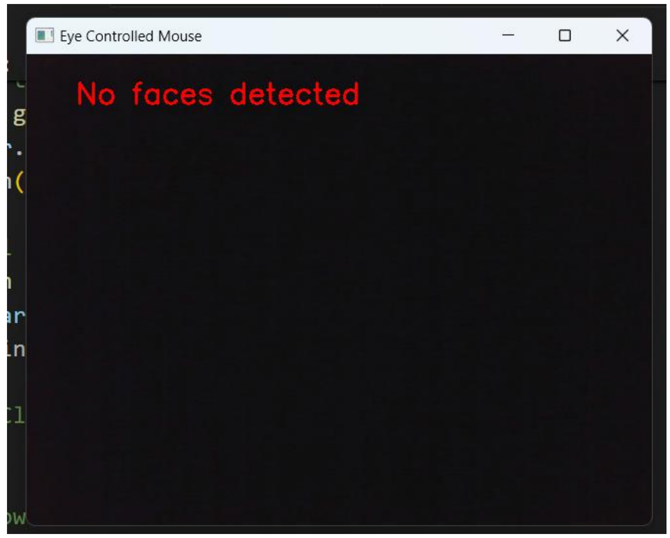


Figure 7 No Face Detected Error

The eye-controlled mouse system begins by accessing the computer's camera and attempting to detect a face using OpenCV and a face mesh model from the Mediapipe library. If a face is successfully detected, the system proceeds to extract facial landmarks, particularly focusing on the left side of the face. The vertical position of selected landmarks is analyzed, and if the left side is relatively flat, indicating a specific facial gesture or expression, the system calculates the time elapsed since the last mouse click. Based on this information, the system performs either a single or double click using the PyAutoGUI library, simulating mouse click operations. In case no face is detected, the system displays an error message on the captured frame. The user is provided with an exit message, prompting them to press the 'Esc' key to terminate the application. The entire process is implemented within a continuous loop, ensuring real-time monitoring and interaction. This eye-controlled mouse system integrates computer vision, facial landmark detection, and mouse control functionalities to offer a hands-free and potentially more accessible means of computer interaction based on facial expressions.

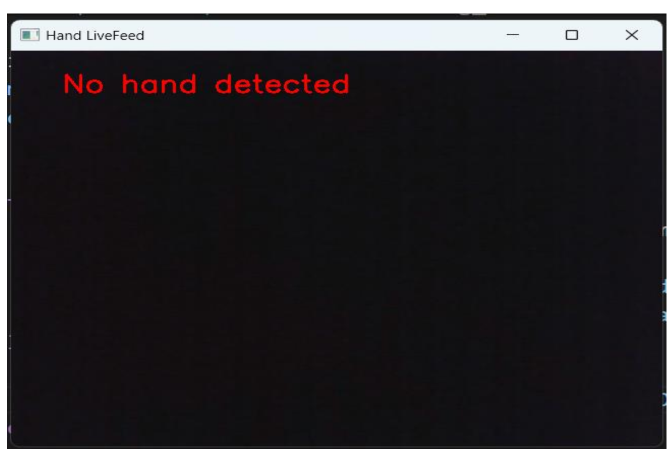


Figure 8 No Hand Detected Error

The incorporation of an error message indicating "No hands detected" in the user interface represents a crucial enhancement to the system's usability and user experience. This feature ensures that users are promptly informed when the hand-tracking functionality fails to detect their hands within the camera's view. By displaying this clear and informative message, users are guided on the necessary steps to address the issue, such as adjusting their hand position or considering alternative input methods. This proactive approach to error handling not only enhances user engagement but also promotes accessibility by providing users with actionable feedback to troubleshoot potential issues effectively. As a result, the feature contributes significantly to the overall usability and inclusivity of the user interface, fostering a positive and seamless interaction experience for all users.

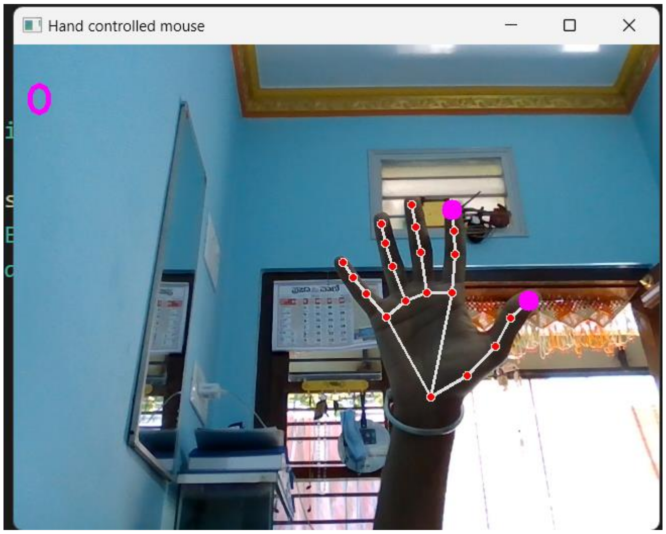


*Figure 9 Hand Gesture Detection*

The concept of a hand-controlled mouse represents an innovative approach to human-computer interaction, enabling users to manipulate computer interfaces using hand gestures and movements instead of traditional input devices like a mouse or keyboard. This technology typically relies on advanced sensors, such as cameras, to track the user's hand movements accurately. By interpreting these movements, the system can translate them into corresponding actions on the computer screen, allowing users to navigate through applications, interact with graphical elements, and perform various tasks entirely hands-free. Hand-controlled mice offer significant potential for enhancing accessibility. Furthermore, they have applications beyond accessibility, including gaming, virtual reality, and augmented reality experiences, where natural and intuitive interaction methods are highly desirable. As this technology continues to evolve, it holds promise for revolutionizing the way users interact with computers, paving the way for more immersive and intuitive computing experiences.

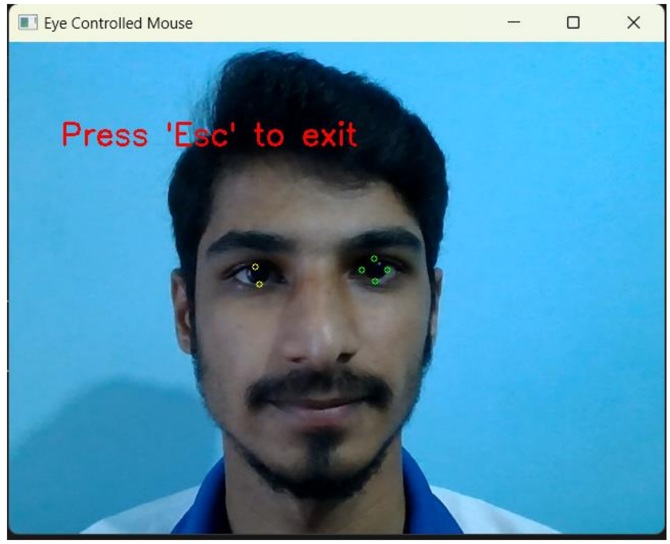


*Figure 10 Eye Gesture Detection*

The eye-controlled mouse system incorporates cursor movements and click operations based on facial landmarks. After successfully detecting a face and extracting facial landmarks, the system focuses on specific landmarks associated with the left side of the face. It tracks the vertical position of these landmarks and, if a certain facial expression is identified (indicating a relatively flat left side of the face), the system calculates the time elapsed since the last mouse click. Depending on this timing information, the system executes either a single or double mouse click using the PyAutoGUI library. To visualize and analyze the facial landmark movements, circles are drawn on the captured frame at the selected landmark locations. This not only provides visual feedback on the detected facial features but also aids in understanding the system's decision-making process regarding mouse click actions. The implementation integrates computer vision, facial gesture recognition, and mouse control functionalities, offering users a hands-free and potentially more intuitive method of interacting with the computer through facial expressions.

## VI. CONCLUSION

In conclusion, our project to control the computer mouse using a combination of eye and hand gestures represents a significant step forward in human-computer interaction. By addressing the limitations and challenges associated with traditional input devices such as the mouse and touchpad, we have developed a more accessible and user-friendly alternative.

Through the integration of cutting-edge technologies such as eye-tracking and gesture recognition, we have created a system that responds to the natural movements and gestures of the user. This innovative approach not only enhances precision and accuracy but also reduces physical strain and discomfort, making computer use more enjoyable and accessible for users of all abilities.

## VII. FUTURE SCOPE

Future research could focus on developing algorithms or technologies to improve the performance of eye-tracking systems in low light conditions.

Future developments could explore adaptive user interface designs that dynamically adjust based on the prevailing environment conditions and the user's interaction patterns.

## REFERENCES


[1] F. U. a. V. S. Robert Gabriel lupu, "Eye Tracking Mouse for Human-Computer Interaction," in *E-Health and Bio-engineering Conference*, 2013.

[2] C. K. a. S. Pumrin, "Face and Eye Tracking for Controlling Computer Functions," in *Electrical Engineering/Electronics, Computers, Telecommunications and Information Technology*, 2014.

[3] A. P. V. C. C. R. a. C. B. Radu Gabriel Bozomitu, "Pupil Centre Coordinates Detection Using the Cirular Hough Transform Technique," in *38th International Spring Seminar on Electronics Technology*, 2015.

[4] G. D. P. a. P. Reddy, "Virtual Mouse Implementation using OpenCv," *Journal of Emerging Technologies and Innovative Research (JETIR),* vol. 7, no. 6, 2020.

[5] Quam, "Gesture Recognition with Data Glove," in *IEEE Conference on Aerospace and Electronics*, 1990.