Eye-Driven Cursor Management and Navigation Using Computer Vision

Dr. M Jayaram
Dept. of CSE (AI&ML)
AVN Institute of Engineering
Hyderabad, India
jayaram_258m@yahoo.com

Kodari Madhavi
Dept. of CSE (AI&ML)
AVN Institute of Engineering
Hyderabad, India
kodarimadhaviyadav88@gmail.com

Amgoth Anil Kumar
Dept. of CSE (AI&ML)
AVN Institute of Engineering
Hyderabad, India
amgothanilkumar1414@gmail.com

Pachipala Naveen
Dept. of CSE (AI&ML)
AVN Institute of Engineering
Hyderabad, India
pnaveenyadav1614@gmail.com

Gajula Rithvik
Dept. of CSE (AI&ML)
AVN Institute of Engineering
Hyderabad, India
gajularithvik99@gmail.com

Abstract—**The goal of HCI (human-computer interaction) is to design systems that are easy to use and accessible. For people with mobility limitations, eye-driven cursor interfaces offer a whole new world of possibilities. These kinds of tech are used in real-time, so clients can do essential tasks like document creation, interactions and browsing hands-free. This system in real-time uses computer vision, Mediapipe to detect face landmarks like eyes and pupils and a regular webcam to record facial video. Pupil movements are mapped onto screen coordinates and gaze direction is calculated. PyAutoGUI and NumPy takes care of the gaze instructions and converts them into mouse movements. A keyboard shortcut (Ctrl + Q) allows users to turn off the system and with respect to traditional input devices. Real-time, scalability and accessibility is guaranteed by the smooth integration with other parts, also provides a hands-free, intuitive interface that promotes independence and improves users' quality of life.**

Keywords— ***Eyes gesture control system, Eye tracking systems, Mouse cursor, Eye mouse, Webcam, Eye movement*.**

I. INTRODUCTION

Designing and creating inclusive and user-friendly interfaces has become a hot topic of research due to the fast pace of human-computer interaction (HCI). For those with physical limitations, traditional input devices like keyboards and mice can be a challenge to access. A viable alternative that allows independent interaction and closes the accessibility gap in digital systems is eye-driven cursor system interfaces. A lot of research has been done on the idea of gaze control and eye tracking over time. But it's still hard to put into practice a reliable, affordable and easy-to-use eye-controlled mouse cursor. A simplified way to do that is provided by recent advancements in computer vision, especially with frameworks like OpenCV and Mediapipe, and the automation capabilities of PyAutoGUI. These technologies offer the flexibility, processing power and mobility to efficiently map gaze spots, process real-time gaze movement data and convert them into cursor movements. In this research, a new implementation of an eye-controlled mouse program that takes advantage of these tools is presented. The proposed method uses a regular camera to record real-time video, analyzes the frames to detect and track eye landmarks and calculate gaze directions. The system uses PyAutoGUI to control the onscreen cursor. High accuracy, low latency and flexibility to different user conditions are the objectives of this work. The rest of this paper is organized as follows: Section II reviews the current approaches to gaze-based interaction and related work. Section III describes the system and the methods for cursor control and eye tracking.

II. LITERATURE SURVEY

A. *Literature Review*

Obviously, research in Eye-Driven, Cursor systems indicates significant advancement in how human-computer interface might apply eye and facial motion. Though it had a requirement of bright lighting, couerna Still's limitation to desktop and laptop systems was the necessity for good recognition through grey scale facial references—a wink, mouth movement; couerna in 2020, scientists built a system where one can use facial movements for cursor movement. It will use CNN and YOLOv4 so it could have smooth trajectory with Kalman filters' aid and with the help of head motion detection, as mentioned above, in this methodology this project was presented a year before. Though it has marvelous features, the company is facing customer fatigue, sensitivity towards light, and accuracy of the distance. A template-based feature-based and deep training based 2021 proposed hybrid solution towards mouse pointer direction based on the eye-gaze. However, it still suffers the weak point from sensitivity on being sensitive with sensitivities toward sensitivity in weak pupil motion and with issues about vertically sensing. System Real-Time Eye-Tracking Mouse Control based upon GDE 2023/ RNN sensitive to over fitting noisy data by XGboost. It increased the accuracy of the cursor from OpenCV by 2024 to be accurate if the user is more experienced, and the interaction or the environment has become faster but more difficult. Tests thus underlined the limitations and opportunities of the mouse systems of ophthalmic surgery so much more creativity could be developed.

B. *Survey*

There are the existing systems of eye-controlled mouse that have been surveyed in terms of progress and challenges concerning human interaction with computers. Early systems were grayscale landmark detection-based and facial gesture-based, both of which suffered from lighting conditions and hardware dependency. Later on, head pose estimation using CNNs and YOLOv4 improved orientation but caused user fatigue and was sensitive to lighting. Hybrids and deep techniques enhanced eye-gaze detection high accuracy but, could not endure pupil movements with bright spots; however, those were unable to withstand vertical directions of movement in pupils. It introduces Gaze Direction Estimation with the support of RNNs plus the support of OpenCV to make a successful and good picture for image processing. This new method achieved the accuracy of real time but has not obtained the precision with the instant speed along with flexibility. So there is more scope for strong user-friendly results.

C. *Aim and Objective*

This project will develop and implement, as a real-time eye-driven cursor system, an accessible alternative to traditional input devices for people with physical disabilities or people interested in novel interaction techniques. It will apply computer vision technologies, including this development: Mediapipe's Face Mesh model, in detecting and tracking facial landmarks while emphasizing the regions of the eyes. The developed system decodes the direction of eye gaze in terms of converting it to the appropriate controls of the cursor, which have to appear on the screen. This is very interactive as it uses blink detection to enable it to have both left-click and right-click action for multiple applications. The developed application further provides dynamic control over a keyboard in such a manner that users may stop the moving of cursors and web camera detections where appropriate and as wished upon.

Interaction with the OS is seamless as well because PyAutoGUI has been employed both for the control of cursors and the click events. The best effort that this project puts forward regarding low latency and real-time processing has revealed practically efficient, user-friendly solutions other than accessibility. In addition, this system created new approaches towards human-computer interaction where the development of bridges in inclusivity was great work from technology.

D. *Requirements*

It will be developing and implementing an alternative for input devices hitherto accessible as an eye-driven real-time cursor system; as accessible for people with a physical disability as well as someone just interested in new interaction techniques. It was based on computer vision technologies. For this development, Mediapipe's Face Mesh model application in the face landmarks detection and tracking was carried out with a highlighted region for the eyes. The developed system decodes the direction of eye gaze by transforming it into the proper controls of the cursor, which have to be displayed on the screen. It is highly interactive because it makes use of blink detection so that it can have both left-click and right-click action for multiple applications.

- The application model is developed using the python programming language.
- Python 3.6 or above. Additional dependencies include OpenCV for computer vision, Mediapipe for gesture recognition
- Detection of eye gesture with webcam. Eye gestures accurately detected and efficiently without entailing cursor movement control and computer interaction.
- User-friendly interface for enabling or disabling tools such as actions like cursor movement enable, disenable, and web cam detection enable and disenable by keyboard.
- The developed application further provides dynamic control over a keyboard in such a manner that users may stop the moving of cursors and web camera detections where appropriate and as wished upon.
- Low latency, immediate processing, and real-time interpretation are the best efforts of this project to unveil a practically efficient, user-friendly solution.
- Except for accessibility, the system set up innovative approaches to human-computer interaction, where technology can develop great bridges of inclusiveness.

III. PROPOSED SYSTEM

The recommended system is an eye-driven application allowing users with the help of a webcam for cursor movements interact with a computer using eyes gestures. It also includes gesture recognition integration. The system allows the user to move the cursor without the use of hands and offers interaction over the eyes. In starting with camera initialization, the input system of this cursor flow draws frames from video coming in the webcam, inputs these into a constant read stream via OpenCV to an application to get that video feed through in real-time and smooth processing. The system forwards the resultant video frames to Mediapipe's Face Mesh model, which would be a good candidate because it is mainly centered around the location of the eye areas. The detection process indicates whether the eyes are located within the captured frame. In case no eye detection is found, it repeats the frames capture for proper initialization and tracking.

Once the eyes are detected, the system extracts and tracks specific eye landmarks. This includes edges and corners of the eyes. These landmarks are very essential in determining the direction of gaze by the user. Then, relative positions of these landmarks give an estimation of where the user is looking at the screen, thus giving the direction of gaze. Once the direction of gaze is fixed, it is mapped to the corresponding coordinates on the screen. This mapping is executed by the resolution of the screen, and it is taken through PyAutoGUI as well. The calculated coordinates will therefore correspond to the user's gaze. It is moved in real time to the mapped coordinates by the mouse cursor using PyAutoGUI. As the user's gaze changes, the cursor position is updated, so this system also dynamically and accurately controls the cursor. Even special blinks are detected to send out specific patterns of left clicks or right clicks for further system interaction. This cycle ensures looping back for catching the next frame; meanwhile, it continues supplying real-time answers and steady tracing of whatever thing the user views up until its application's cessation.

It is moved in real time to the mapped coordinates by the mouse cursor using PyAutoGUI. As the user's gaze changes, the cursor position is updated, so this system also dynamically and accurately controls the cursor. Even special blinks are detected to send out specific patterns of left clicks or right clicks for further system interaction. This cycle ensures looping back for catching the next frame; meanwhile, it continues supplying real-time answers and steady tracing of whatever thing the user views up until its application's cessation. This cycle enables the control of a mouse cursor hand-free, thus making the system practical and easy to use.

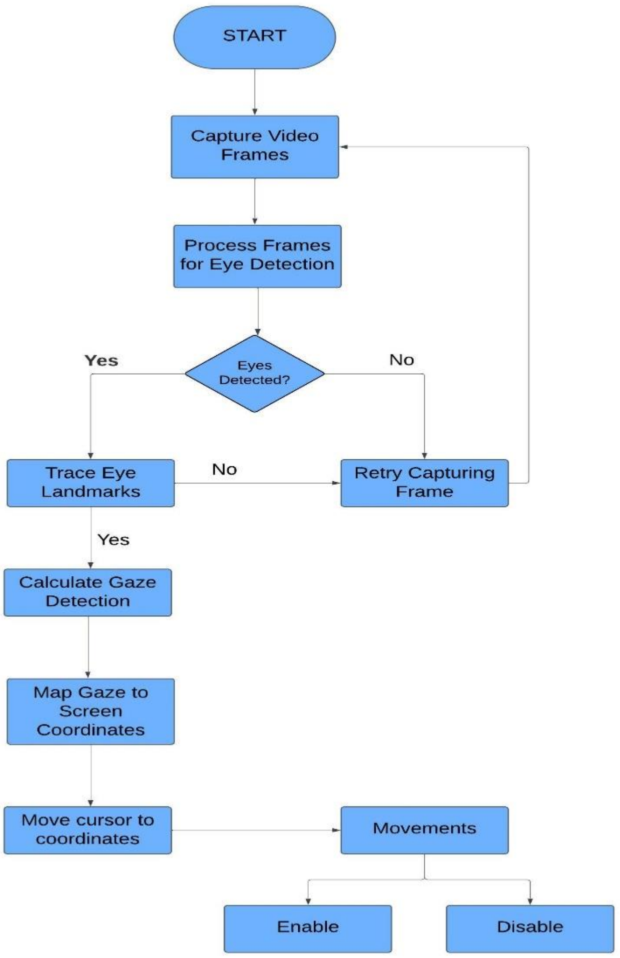

Fig.1. System flowchart

The flow is such that every functionality, whether eye detection, movements of the cursor, or clicks, is conducted with maximized effectiveness and precision no matter the variability in conditions. The keyboard-controlled toggle features enhance adaptability as well as usability of the system, making the user eventually in control.

IV. SYSTEM METHODOLOGY

It's a system allowing a user to control the movement of a computer mouse cursor using his eye movements in real time. Computer vision and eye-tracking technology allow for tracking a user's eye gaze through a webcam and then transferring this information to the computer as precise cursor movement on the screen. It uses Mediapipe for eye detection and tracking, OpenCV for video capture and processing, and PyAutoGUI to execute the cursor control. This added support from NumPy provides swiftness in mathematical calculations while inferring eye-gaze directions and ensures that an innovative, easily accessible interface is presented to the system users; therefore, this system has brought the greatest advantage by not being strictly devoted to motor-disabled users

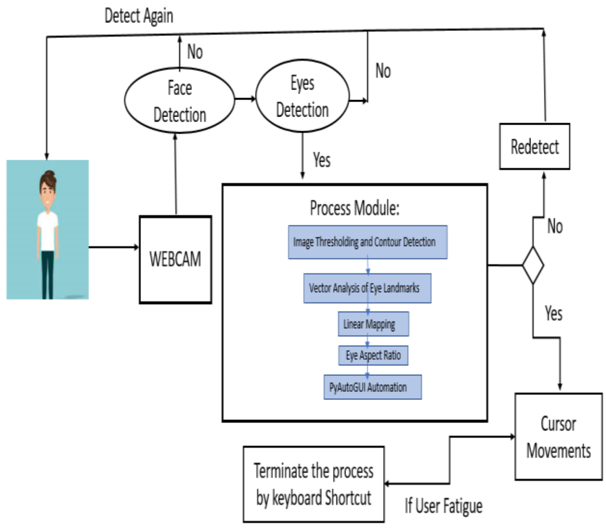

Fig. 2. System Architecture

A. *UserInterface*

The interface is intuitive enough so that minimal interaction brings the user up with accessibility still in place. Once the application launches, it will open up an automatic and minimalist dashboard to calibrate from which a system that could detect movement accurately enough by the user's eye will be detected. Then there is live feeding with a webcam through which one can trace through what their eyes are focused on and seeing relative to the screen. It is differentiated by obvious signs of calibration, such as the need to look at certain spots on the screen in order to increase the accuracy. For example, the simple switch allows users to activate eye-tracking or deactivate it, and to switch between precision and speed modes for cursor motion. Visual feedback is provided through highlighted zones or cursors, making the user feel in control of his actions and offering instant feedback to when something was clicked or dragged. In summary, the interface is for people with motor disabilities - simple, accurate and handy.

Fig. 3. User Interface

The Eye Gesture Detection and Tracking system forms a core of the Eye-Driven Cursor Management, which should enable a user to make use of these natural, intuitive eye movements in order to control his cursor on his computer. Real-time video feed directly from a webcam is processed there with highly advanced technologies, OpenCV, and Mediapipe. It first detects the face of the user and then key facial landmarks which will emphasize the eyes. It then zooms out to the region of interest, namely the eye region, and tracks the iris location in order to find the direction the user is looking. It converts the direction of view into an action on the screen to determine movement of a cursor on the screen such that by only looking in the area will the position of the cursor change.

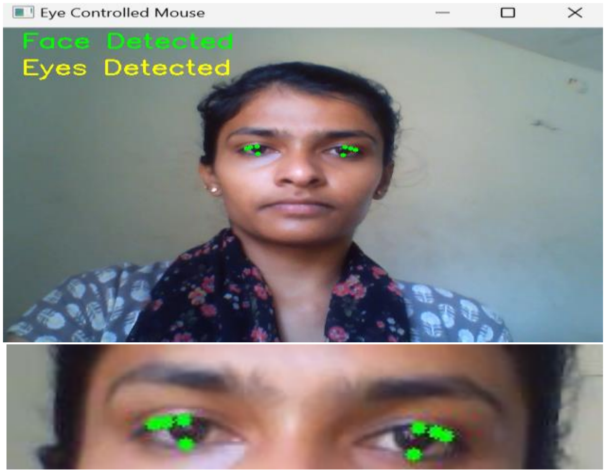

Fig.4. Face and Eyes are Detected

It applies these specific eye movements such as blinking single or double or extended gazes, identifies it as standard operations and applies clicks and drags and scrolls accordingly. So, left-click could be achieved by a single blink and double by right click. All those operations generally are carried with the help of tools including PyAutoGUI; which is relatively easy for smooth and accurate interaction with computer interfaces. It processes in real-time using NumPy for all calculations and can do smooth tracking even under changing lighting or when a user's head makes some slight movements. Calibration options are provided to allow adjustments in sensitivity and for the system to adjust to different users' eye movements and different conditions of the environment.

That helps the eyes to play the role of controlling the hands to operate any computer and significantly enhances user accessibility for a person with some form of motor impairment-to interact with digital systems easily. It is designed in a way to give it a seamless, robust, and inclusive user experience, which makes technology accessible and available to everybody.

This will create the UI and cursor movement in such a way that the user can experience smoothness while using the Eye-Driven Cursor, hence finding the natural and efficient method of getting things done. Having an uncrowded simple layout with the user interface removes any distraction, which makes the way to navigate around the people using the application relatively easier. At the launching, it provides the user with a dashboard of steps to describe all about how to configure a system. Any time there is a web-camera feed so that one could see how one's eyeing something. In its emphasis, all such calibration tools are there where someone has to align it that way; looking at any particular place on the screen and considering right eye tracking. Further, it enables an enormous toggle for gesturing click and scroll over click with a drag along sensitivity based slider for both aspects of eye track and work related to gesture.

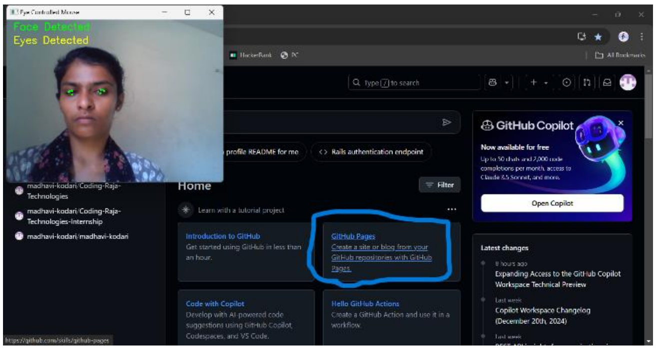

Fig.5. Cursor Pointing

System of cursor movement-the main function of the project-corporate next-generation technology behind the software. The system translates the gestures using actual time by transforming eye movements into precise action with the on-screen cursor. Advanced computer vision algorithms map a user's eye movements to their corresponding positions with an on-screen cursor. For instance, if a user looks at the top right of the screen, then the cursor moves in that direction; when there is holding of focus on a point for a more extended period of time, it freezes the cursor to give control. Such motion performs an action through either blinking or holding the focus. The case with blinking is that one blink will be simulating the left clicking; a double blink simulates right click of mouse; whereas maintaining gaze for some time initiates dragging. Navigation is done using vertical movements, whereas the horizontal ones are used in scrolling.

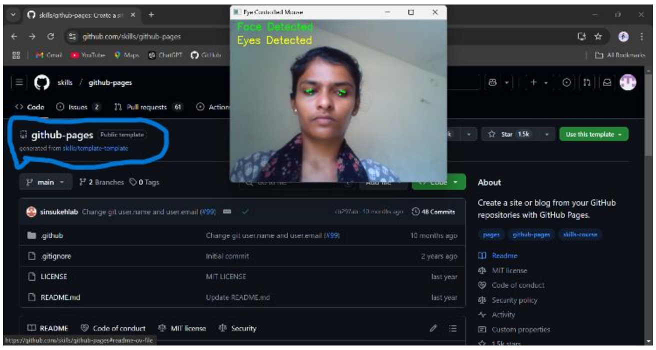

Fig. 6. The click is executed based on this action

The system guarantees that the cursor moves smoothly, even in a dynamic environment when lighting may be changing or upon minor head movements. The solution allows for the customization of such a system using adjustable sensitivity and calibration options based on the various needs of its users. The integration of UI and cursor movement generally enables its users, and especially those afflicted with motor impairments, to interact easily and independently with a computer.

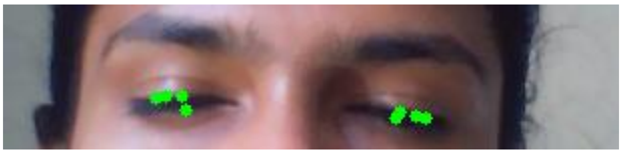

Fig.7. Eyes Gesture to perform click

B. Real-time Rendering and Video Processing

The system has made it possible to process a real-time video stream from the webcam to get an eye movement with almost no time lag. The system employs real-time frame capture using OpenCV and Mediapipe libraries available in Python for facial land marking and focusing regions of eyes in real-time. This it has achieved by utilization of optimized pipelines from video processing that ensures nothing hampers the high performance of such for real time applications with an accurate delivery of gaze. Also, NumPy usage in mathematical procedures and conversion of gazes into precision cursory movements through PyAutoGUI. Thus, the video processing live rendering is well fitted in such a way that providing an ideal usability for the user tools of the motor disability would confer a super-seamless responsiveness. The video processing is handled by OpenCV, which captures frames from the webcam and overlays the drawing canvas on top of the video feed. The captured frame is first flipped horizontally to create a mirror image, making the interaction more intuitive for the user. Then, the system checks for hand landmarks and processes these frames to detect user gestures.

Once hand gestures are detected, the system draws corresponding elements, such as lines or eraser circles, on a canvas that is initialized as a black image. The drawing is continuously updated by blending the live video feed with the drawing canvas. To create a seamless experience, the system uses weighted overlay to combine the video feed with the drawn content. This ensures that both the camera feed and the drawing are displayed concurrently in a smooth and visually coherent manner. The final output is a combined frame that shows the video feed on the left side and the canvas (with the drawn content) on the right. This is presented in a resized format to ensure the content is easily visible on the screen.

C. System Workflow and Flow Control

This application workflow initializes, first of all, a webcam and a Mediapipe Face Mesh to capture frames in real-time. By using this system, a face landmark-based eye region is initially identified, and finally, it uses OpenCV, such that those eye-related landmarks enable tracking of following gazes by tracking those elements in eyes, which were interpreted by the aid of PyAutoGUI and subsequently translated as controlling the movement of the cursor across the monitor for the user. The global keyboard listener toggles on/off cursor movement and webcam detection and system shutdown. The system processes every frame by flipping the image, converting it to RGB, and detects facial landmarks; when eyes are detected, those coordinates are inspected for cursor operation, with coordinates from the right eye driving it. Blink is detected to emulate mouse clicks on the right button based on aspect ratios of eyes and blink count. It provides immediate feedback on detection of a face or eyes, for instance. It also manages resources correctly and cleans up quite nicely if a user cancels the process. The flow of control in the system is as follows:

1. The webcam, Mediapipe Face Mesh, and global keyboard listener.
2. Video frames captured by webcam are read, preconditioned before sending it as an input to the landmarks.
3. The eyes are the primary facial landmark detected.
4. It makes use of Gaze data that is used for real-time movement of the cursor.
5. Conducts eye aspect ratios for a left-click or right click.
6. Displays status messages about face, eyes on video feed.
7. Controls Global Controls Key Board Input: turn on/off the cursor and detection.
8. Free the resources and get out when the user stops the program.

```
Cursor movement Disabled
Webcam detection Disabled
Webcam detection Enabled
Cursor movement Enabled
Cursor movement Disabled
Webcam detection Disabled
Webcam detection Enabled
Cursor movement Enabled
Cursor movement Disabled
Exiting...
```

Fig.8.Showing the Enabling and Disabling the Cursor and Webcam by using keyboard shortcut Keys.

D. Key Parameter Settings

The project relies on fine-tuned parameters to ensure optimal functionality and a user-friendly experience. Key settings are carefully calibrated to balance performance and usability, enhancing gesture recognition and interaction with the Computer. These parameters govern detection stability, Cursor movements, and effective user feedback, making the system responsive and efficient. To ensure smooth operation, several key parameters are defined:

1. Detection Confidence: Set to 0.7 for stable eye tracking.
2. Tracking Confidence: Also set to 0.7 to ensure consistent detection.

E. Test cases

Testing is the most critical aspect of the eye-driven cursor management system. It ensures the solution is accurate, reliable, and user-friendly. Testing was necessary since this system relies on real-time eye tracking and gaze detection, and one is sure to employ features such as eye detection, gaze mapping, and cursor movement with the effectiveness the features provide. This includes analysis of the systems' ability to work at different lighting conditions, webcam qualities, and user characteristics, such as different shapes of eyes, wearing glasses, or head movements. Testing helps identify latency problems and thereby get the cursor to respond promptly to eye movements by ensuring the user experience is smooth.

The evaluation process also is very important, since it assists the developers in establishing how robust a system is concerning monitoring fatigue to prevent straining on the user. It ensures the accuracy of metrics like EAR and fine-tuning thresholds related to blink detection or prolonged closure. Feedback provided by users when testing is valuable for understanding what has gone wrong-usability concerns pertaining to comfort and simplicity in operation plus intuitive navigation. This process helps to hone the system in meeting the needs of a diverse user population, especially those with motor impairments who rely on such technology for accessibility. Thorough testing ultimately ensures the system performs as intended, fostering trust, reliability, and adoption in real-world scenarios. Thorough testing validates the system's performance, reliability, and usability.

Fig.8.Test cases

V. EXPERIMENTS

The experiments for this project were conducted to evaluate the accuracy, responsiveness, and usability of the air-drawing application. Various hand gestures, including fist-based erasing and index finger tracking for drawing, were tested under different lighting conditions and backgrounds. The system achieved consistent performance with a detection confidence of 0.7, ensuring reliable hand tracking. The user experience was assessed by testing brush thickness, eraser size, and captioning features, with adjustments made to improve precision and readability.

A. Experiment Details

The eye-driven mouse system was tested under various conditions to assess its robustness and performance. For indoor testing, three different camera setups were used: a standard 720p webcam, a 1080p webcam, and a high-definition (HD) camera with 1440p resolution. Each camera was tested under varying lighting conditions, including high brightness (intensity 80%), medium brightness (intensity 50%), and low brightness (intensity 20%). Each environment was tested for 10 minutes, incorporating various system functions like cursor movement, blink-based clicks, and dwell-based selection.

For outdoor testing, the system was evaluated in an urban street environment with dynamic movement, as well as a quiet park with minimal distractions. The gaze-tracking functionality was tested under various ambient conditions to evaluate its performance in accurately detecting and responding to eye movements. In controlled environments with consistent lighting and minimal distractions, the system achieved high accuracy, with cursor movements precisely matching the user's gaze. In more challenging scenarios with variable lighting or rapid head movements, the accuracy showed a slight decrease, but the cursor remained functional and responsive.

B. Results and Discussions

The eye-driven mouse system performed effectively across all test scenarios, with the HD camera (1440p) offering the best eye-tracking accuracy and responsiveness, followed by the 1080p and 720p cameras. In indoor environments, the system showed stable performance across both plain and cluttered backgrounds under normal and bright lighting conditions, with minor performance degradation in low-light scenarios. In outdoor settings, the system accurately tracked gaze movements in both busy and calm environments, although the accuracy slightly decreased in high-motion areas, especially with lower-resolution cameras.

The system demonstrated clear advantages over traditional input methods, as cursor movements were precisely controlled by gaze, minimizing unintentional actions. Users could effortlessly perform actions such as cursor navigation, blinking for clicks, and dwell-based selections without the need for external devices, reducing the physical effort required for interaction. These features significantly improved the precision of the system and streamlined the overall user experience. Additionally, real-time feedback remained smooth and reliable, even in dynamic environments, ensuring that only intentional gaze-based actions were registered. This innovation enhances the system's efficiency and usability, making it a highly refined solution for hands-free computing applications. Overall, the eye-controlled mouse system proved to be a valuable tool for hands-free interaction, providing accurate and efficient control in a variety of settings. The system's adaptability and performance highlight its potential as an accessible and user-friendly alternative to traditional input methods.

The cursor control component of the eye-controlled mouse demonstrated an accuracy rate of 99% under optimal lighting conditions, with smooth and precise tracking of gaze movements. The system maintained consistent performance even in cluttered backgrounds, with minimal delay in executing cursor movements and clicks. However, in low-light environments, the accuracy decreased to 82%, highlighting the impact of lighting conditions on the performance of eye tracking. The dwell-based selection and blink-clicking features performed well across various scenarios, ensuring reliable interaction with the system. The system responded promptly to user actions, with an average delay of 0.5 seconds, providing a seamless experience.

Overall, the eye-driven mouse system proved effective in a variety of conditions, with high accuracy and minimal latency under optimal settings, and acceptable performance under more challenging scenarios.

Test No.	**Testing Conditions**		
	Webcam Quality	***Lighting***	***Webcam Quality***
1	1440p	1	1440p
2	1440p	2	1440p
3	1440p	3	1440p
4	1080p	4	1080p
5	1080p	5	1080p
6	1080p	6	1080p
7	720p	7	720p
8	720p	8	720p
9	720p	9	720p

TABLE.1.ACCURACY RATES FOR MOUSE CONTROL

VI. CONCLUSION

This project evidences the applicability of computer vision in transforming the way we interface with technology. The ability to operate a cursor with eyes and blink the mouse for simulated clicks makes accessibility easier for people but opens up entirely new efficiency and convenience dimensions for all kinds of users in their respective fields. From independence in people with disabilities to possibility in virtual and augmented reality, the system encompasses diverse use cases and applications. Eye-Driven Cursor Management and Navigation Using Computer Vision is an exemplary example of the power of technology in accessibility-related critical challenges. Using advanced techniques in computer vision, the system presents a hands-free solution that empowers people with motor impairments to effectively interact with digital devices. It enables real-time integration of eye detection, gaze mapping, cursor control, and blink detection, therefore, providing a seamless interaction mechanism without using a computer mouse or keyboard.

The significance of interdisciplinary innovation in computer vision, artificial intelligence, and user-centric design underlines the accomplishment of this project. Here is a deep learning framework for future development, improving the precision, adaptability, and scalability of the system. Further research and development can take this technology further into areas such as advanced user interfaces, gaming, education, and so on. One such highly futuristic project is Eye-Driven Cursor Management and Navigation Using Computer Vision-which rejuvenates man-computer interfaces and outlines features such as accessibility and inclusion that emerge in the world of today high-tech landscape; digitized technologies will become much easier to achieve with equal availability towards diverse users. With such immediacies, accessing personal computers for example, one finds it possible enough to even bring about more elaborative study onto seemingly boundless ways in which control is bestowed by vision-based control.

REFERENCES

[1] Sharon Mathew, Sreeshma A, Theresa Anitta Jaison, Varsha Pradeep, S Santhi Jabarani, "Eye Movement Based Cursor Control and Home Automation for Disabled People." Proceedings of the fourth International Conference on Communication and Electronics Systems (ICCES 2019), IEEE, 2019.

[2] Vinay S Vasisht, Swaroop Joshi, Shashidhar, Shreedhar, C Gururaj, "Human Computer Interaction based Eye Controlled Mouse" Proceedings of Third International Conference on Electronics Communication and Aerospace Technology (ICECA 2019), IEEE, 2019.

[3] Vandana Khare,S.Gopala Krishna,Sai Kalyan Sanisetty, "Cursor Control Using Eye Ball Movement" 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), IEEE, 2019.

[4] Kshitij Meena, Manish Kumar, Mohit Jangra, "Controlling Mouse Motions Using Eye Tracking Using Computer Vision", Proceedings of the International Conference on Intelligent Computiong and Control Systems (ICICCS 2020), IEEE, 2020.

[5] Akshadha Dongre, Rodney Pinto, Ameya Patkar, Dr. Minal Lopes, "Computer Cursor Control Using Eye and Face Gestures" 11th ICCCNT 2020, IIT Kharagpur, IEEE.