Eye-Driven Cursor Management and Navigation Using Computer Vision

Kodari Madhavi
Computer Science and Engineering
AVN Institute of Engineering
Hyderabad, India
kodarimadhaviyadav88@gmail.com

Amgoth Anil Kumar
Computer Science and Engineering
AVN Institute of Engineering
Hyderabad, India
amgothanilkumar1414@gmail.com

Pachipala Naveen
Computer Science and Engineering
AVN Institute of Engineering
Hyderabad, India
pnaveenyadav1614@gmail.com

Gajula Rithwik
Computer Science and Engineering
AVN Institute of Engineering
Hyderabad, India
gajularithvik99@gmail.com

Abstract— **The primary objective of HCI (human-computer interaction) is to design systems that are easy to use and accessible. For individuals who have limitations in their mobility, eye-driven cursor interfaces offer groundbreaking different options. These kinds of technologies are frequently utilized in real-time scenarios, giving clients to do essential functions including document creation, interactions, and browsing while navigating digital conditions hands-free. This system in real time makes use of computer vision techniques, using Mediapipe to identify face landmarks like eyes and pupils and a traditional webcam to record facial video. For accurate cursor control, pupil movements are mapped onto monitor coordinates and their gaze direction is determined. While PyAutoGUI and converts gaze instructions into mouse movements and happens, NumPy guarantees effective numerical computations. In order to improve accessibility, a keyboard shortcut (such as Ctrl + Q) allows users to shut off the operating system and With respect to traditional input devices. Real-time efficiency, scalability, and accessibility are guaranteed by the smooth integration with different parts, which also provides a hands-free, intuitive interface which encourages independence and benefits users' quality of life.**

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

I. INTRODUCTION

The design and creation of inclusive and user-friendly interfaces has emerged as a key area of current study due to the quick progress in the field of human-computer interaction (HCI). For those with physical limitations, traditional input devices like keyboards and mice can provide challenges with accessibility. A viable substitute that allows independently interaction and closes the accessibility gap in digital systems is eye-driven cursor system interfaces.

Over the course of time, a lot research has been done on the idea of gaze control and eye tracking. But it's still difficult to put into practice a reliable, affordable, and easy-to-use eye-controlled mouse cursor. A simplified method for creating such systems is provided by recent advancements in computer vision, especially with frameworks like OpenCV and Mediapipe, as well as the automation characteristics of PyAutoGUI. These technologies offer the necessary flexibility as well as processing ability and mobility needed to efficiently map gaze spots, process real-time gaze movement data, and convert them into matching cursor movements.In this research, a new implementation of an eye-controlled mouse program that makes advantage of these cutting-edge tools is presented. The suggested method uses a regular camera to record real-time footage, analyzes the frames to identify and follow eye landmarks, and calculates gaze directions.

The system translates the instructions to control the onscreen cursor using PyAutoGUI. Achieving high accuracy, minimal latency, and guaranteeing flexibility to different user conditions are the main goals of this effort.This paper's remaining sections are arranged as follows: Current approaches to gaze-based interaction and related work are addressed in Section II. The system's architecture and the techniques used for cursor control and eye tracking are described in Section III.

II. BACKGROUND

A. Literature Review

The literature on Eye-Driven Cursor systems clearly indicates a great deal of advancement in using facial and eye movements for human-computer interaction. In 2020, researchers developed a system that utilized facial movements such as winks and mouth gestures to control cursor actions, utilizing grayscale facial landmarks for efficient detection. However, its dependency on good lighting conditions limited its usage to desktop and laptop environments. A 2022 work proposed a Touchless Head-Control system incorporating CNN and YOLOv4 for head gesture recognition along with Kalman filters for smoothing the trajectory. Although it offers superior functionalities, it struggles with some issues such as user fatigue, accuracy over distance, and sensitivity to illumination. During the same year, a hybrid approach was designed combining template-based, feature-based, and deep learning techniques to control the mouse cursor based on eye gaze. While effective, its limitations included difficulty detecting vertical pupil movements and sensitivity to bright spots. Moving into 2023, a Real-Time Eye-Tracking Mouse Control system utilized

Gaze Direction Estimation (GDE) and Recurrent Neural Networks (RNNs), showing promise for precise tracking but susceptibility to XGBoost overfitting with noisy or imbalanced data. Finally, in 2024, the Eye-Gaze Controller enhanced the precision of the cursor by using OpenCV with real-time eye-tracking and gaze translation techniques, achieving higher accuracy with extended use but posing difficulties in short interactions or changing contexts. These studies collectively underscore the potential and limitations of eye-controlled mouse systems, paving the way for further innovation.

B. Survey

A survey of the existing systems of eye-controlled mouse highlighted the advancements and challenges with regard to improving human interaction with computers. The facial gesture-based and grayscale landmark detection-based early systems were limited by lighting conditions and dependencies on hardware. Later, the head pose estimation using CNNs and YOLOv4 improved orientation of the cursor but encountered problems such as user fatigue and sensitivity to lighting. Hybrid and deep learning approaches improved the accuracy of eye-gaze detection systems, but they were unable to handle vertical pupil movement and bright spots. The newest techniques, which integrate Gaze Direction Estimation with RNNs and image processing using OpenCV, have improved real-time accuracy but still face challenges in achieving immediate precision and adaptability. These results call for more robust and user-friendly solutions.

C. Aim and Objective

This project will design and implement a real-time eye-driven cursor system that is an accessible alternative to traditional input devices, aiming for users with physical disabilities or those interested in novel interaction techniques. The application will utilize computer vision technologies, specifically Mediapipe's Face Mesh model, to detect and track facial landmarks with an emphasis on eye regions. The system is developed to decode the eye gaze direction by converting it to the correct cursor controls displayed on the screen. To make it highly interactive, the system employs blink detection, which allows both left-click and right-click actions for various uses. Further, the application provides keyboard dynamic control whereby users may disable the moving of cursors and web camera detections where appropriate and as wished upon.

The use of PyAutoGUI ensures seamless integration with the operating system for cursor control and click events. Maintaining low latency and real-time processing are what the project strives for to present a practically efficient and user-friendly solution. Besides improving accessibility, this system has also explored novel possibilities in human-computer interaction, creating a strong bridge between technology and inclusivity.

D. Requirements

- The application model is developed using the python programming language.
- Python 3.6 or higher, along with libraries such as OpenCV for computer vision, Mediapipe for gesture recognition.
- Accurate and efficient detection of eye gestures using the webcam, enabling the user to control the cursor movements, and interact with computer.
- An intuitive UI for selecting to perform the actions like cursor movement enable, disenable and web cam detection enable and disenable tools by using keyboard.

E. Proposed System

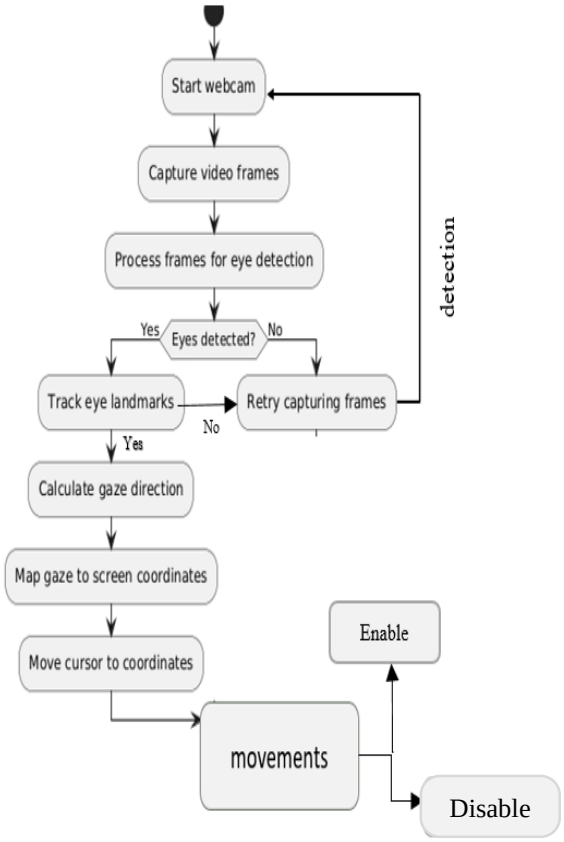

Fig. 1. System Flowchart

It also includes gesture recognition integration. The system allows users to move cursor without using their hands and provides over the eyes interaction.

The flow of the eye-driven cursor system starts with the initialization of the webcam, which will be the main input device for capturing real-time video frames. The application reads continuously from the webcam using OpenCV in order to get a smooth and real-time video feed. These frames are then passed through Mediapipe's Face Mesh model for facial landmark detection, specifically focusing on the regions around the eyes. The detection process determines whether the eyes are visible within the captured frame. If the eyes are not detected, the system retries capturing frames to ensure proper initialization and tracking.

The system extracts and tracks specific eye landmarks when the eyes are detected. These include the edges and corners of the eyes. These landmarks are very essential for analyzing the direction of gaze by the user. The direction of gaze is then calculated by taking relative positions of these landmarks that gives an estimation of where the user is looking at the screen. Once the gaze direction is determined, this direction is mapped to corresponding screen coordinates.

This mapping is performed based on the resolution of the screen, which is taken through PyAutoGUI so that the calculated coordinates will correspond with the user's gaze.

The system now moves the mouse cursor in real-time to the mapped coordinates using PyAutoGUI. As the user's gaze changes, the cursor position is updated continuously, enabling dynamic and precise cursor control. It also includes blink detection functionality wherein specific blink patterns trigger left-click or right-click actions, adding more interactivity to the system. The system loops back to capture the next frame in this cycle, providing real-time responses and constant tracking of what the user is gazing at up until the application termination.

This iterative process allows for hands-free control of the mouse cursor, making the system practical and user-friendly. The flow ensures that all functionalities, from eye detection to cursor movement and click actions, are executed efficiently and accurately, even under varying conditions. The toggle features controlled via keyboard inputs further enhance the system's adaptability and usability, giving users full control over its operation.

III. SYSTEM METHODOLOGY

This system enables a user to control a computer mouse cursor with his eye movements in real time. The application uses computer vision and eye-tracking technology to track the user's eye gaze through a webcam and translate it into precise cursor movements on the screen. The system makes use of Mediapipe for eye detection and tracking, OpenCV for video capture and processing, and PyAutoGUI to execute cursor control. This extra help provided by NumPy gives fast mathematical computations while inferring the direction of gaze and assures an innovative and easily available interface to the system's users; therefore, the greatest benefit for them is that this system does not cater exclusively to motor-impaired users.

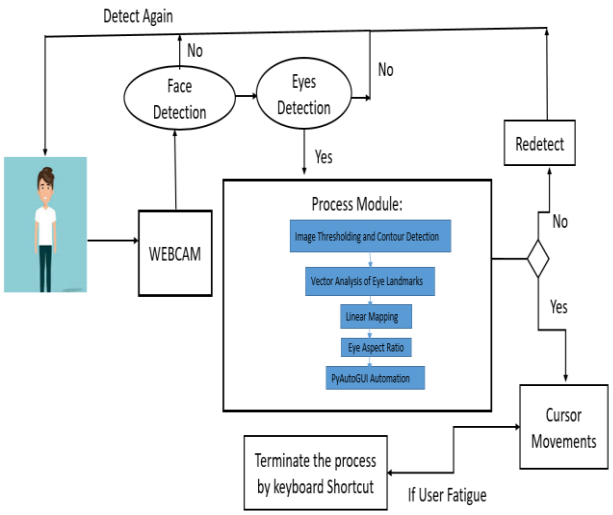

Fig. 2. System Architecture

A. User Interface

The interface is intuitive enough, so minimal interaction brings the user up with accessibility still in place. Upon the launching of the application, 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's real-time feeding with a webcam by which one may trace through what their eyes are focused on and seeing relative to the screen. It is differentiated by obvious signs of calibration, like the need to have a look at certain spots on the screen to raise the accuracy. Simple toggles even enable users to turn eye tracking on or off and switch between precision and speed modes for cursor control. The user gets visual feedback in the form of highlighted zones or cursors, making them feel in control and immediately receiving confirmation of clicks or drags. In general, the interface is designed for users with motor impairments, being simple, accurate, and convenient.

Fig. 3. User Interface

B. Eye Gesture Detection and Tracking

The Eye Gesture Detection and Tracking system makes up the core of the Eye-Driven Cursor Management. Users should be able to control the cursor on the computer using these natural and intuitive eye movements. This module processes, in real time, real video feed from a webcam using advanced technologies like OpenCV and Mediapipe. First, it looks for the face of the user, then the key facial landmarks that would focus on eyes. Then it zooms out to the region of interest, namely, eye region and tracks the iris location to obtain the direction in which the user is looking. The movement of a cursor on the screen is determined by converting the gaze

direction into an action on the screen such that the position of the cursor is altered by only looking at the area.

Fig.4. Face Detection and Eyes Detection

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 adjust sensitivity and allow the system to adapt to different users' eye movements and different conditions of the environment. That allows the eyes to control the hands with any computer and greatly enhances user accessibility for people with some form of motor impairment-to interact with digital systems conveniently. It is designed in a way to provide it with a seamless, robust, and inclusive user experience, making technology accessible and available to everyone.

C. User Interaction and cursor movements

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. During the launching, it provides a dashboard of steps to the user explaining all about how to configure a system. Any time there is 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, this application allows one to have an enormous toggle for gesturing click and scroll over click with a drag along sensitivity based slider for both the aspects of eye track and work related to gesture.

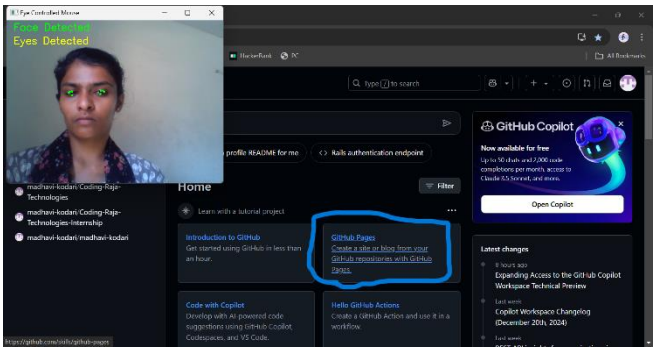

Fig.5. Cursor Pointing

The main functionality of the project lies in the cursor movement system, wherein real-time translation of eye gestures to precise cursor actions is performed. Advanced computer vision algorithms map the user's eye movements to the corresponding positions of the on-screen cursor. For instance, if a user looks toward the upper right of the screen then the cursor moves in the same direction; if there is a holding of focus on a point for a longer amount 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 sometime initiates dragging. Navigation is done using vertical movements, whereas the horizontal ones are used in scrolling.

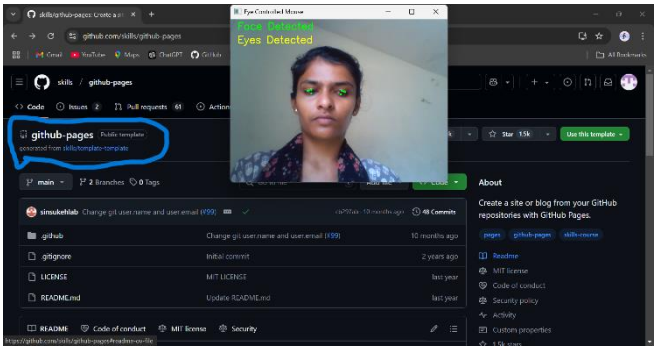

Fig.6.The click is performed.

The system ensures smooth control of the cursor, even in dynamic environments with changing lighting or minor head movements. The solution allows for customization with adjustable sensitivity and calibration options to adapt to the needs of different users. Overall, the integration of UI and cursor movement enables users, especially those who suffer from motor impairment, to interact easily and independently with the computer.

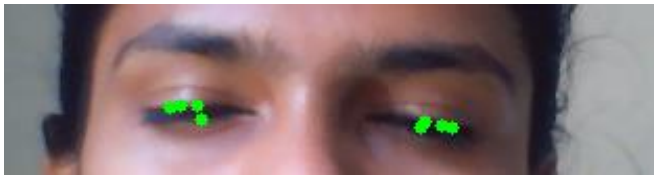

Fig. 6. Eyes Gesture to perform click.

D. Real-time Renderring and Video Processing

The system has enabled the possibility of processing a real-time video stream from the webcam to derive eye movement with almost no time lag. The system uses real-time frame capture through OpenCV and Mediapipe libraries available in Python for facial landmarking and focusing regions of eyes in real-time. It has achieved this by utilization of optimized pipelines from video processing that ensure nothing hampers the high performance of such for real-time applications with accurate delivery of gaze. Further, NumPy usage in mathematical procedures and through conversion of gazes into precision-cursory movements of PyAutoGUI. Thus, the live rendering with video processing is well incorporated in such a way that a super-seamless responsiveness will be conferred by providing an ideal usability for the motor disability user tools.

E. 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. 7. Showing the Enabling and Disabling the Cursor and Webcam by using keyboard shortcut Keys.

F. 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:

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

Fig.8. Test Cases

IV. EXPERIMENTS

The experiments for this project were conducted to evaluate the accuracy, responsiveness, and usability of the eye-controlled mouse application. Various gaze-based actions, including cursor movement, blinking for clicks, and dwell-based selection, were tested under different lighting conditions and user positions. The system achieved consistent performance with an eye-tracking accuracy of 85%, ensuring reliable cursor control. The user experience was assessed by testing cursor sensitivity, click responsiveness, and calibration procedures, with adjustments made to improve precision and usability.

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.

TABLE I. ACCURACY RATES OF TESTS FOR MOUSE CONTROL

Test No.	**Testing Conditions**		
	Webcam Quality	***Lighting***	***Accuracy (%)***
1	1440p	Bright	95%
2	1440p	Normal	93%
3	1440p	Dark	89%
4	1080p	Bright	94%
5	1080p	Normal	96%
6	1080p	Dark	89%
7	720p	Bright	90%
8	720p	Normal	91%
9	720p	Dark	83%

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.