JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

EMPOWERING ACCESSIBILITY FOR HANDICAP: AN EYEBALL CURSOR MOVEMENT USING COMPUTER VISION TECHNIQUES

Mr. M. Hari Krishna[1], L. Deekshitha[2], P. Krishna Pranav[3], N. Sumitra[4], G. Shreyas[5], K. Varshitha[6] [1]Assistant Professor, Department of CSE (Data Science), ACE Engineering College, Hyderabad, Telangana, India
[2,3,4,5,6]IV B.Tech Students, Department of CSE (Data Science), ACE Engineering College, Hyderabad, India

ABSTRACT:

There are different reasons for which people need an artificial of locomotion such as a virtual keyboard. The number of people, who need to move aroundwith the help of some article means, because of an illness. Moreover, implementing a controlli ng system in it enables them to move without the help of another person is veryhelpful. The idea of eye controls of great use to not only the future of natural input butmore importantly the handicapped and disabled. Camera is capturing the image of eye movement. First detect pupil center position of eye. Then the different variation on pupilposition get different command set for virtual keyboard. The signals pass the motor driver to interface with the virtual keyboard itself. The motor driver will control both speed and direction to enable the virtual keyboard to move forward, left, right and stop.

The proposed system represents a significant advance in technology for individuals with disabilities, particularly those with mobility-related problems. Leveraging the power of eye control technology, the system provides an intuitive alternative to controlling the virtual keyboard and, by extension, facilitating movement in the user's environment Practice begins by accurately tracking and recognizing the user's eye movements, with sophisticated camera systems performing its imaging processes.

Keywords: Eye tracking, Cursor Movements, Face detection.

I. INTRODUCTION

In modern times, personal computer systems play an important role in our daily lives, especially in the workplace and various industries. However, for individuals with limited mobility, loading methods such as pigs present challenges. To overcome this, an inexpensive system using eye movements to control the cursor in a computer system has been proposed. This system uses OpenCV with an IP camera to detect and process eye movements, providing an alternative solution for users with limited mobility

1.1 Background:

Traditional methods of cursor control rely on physical interaction with external devices, which can be difficult for those with limited motor skills. Eye tracking technology combined with OpenCV offers a promising solution by interpreting eye movements to control cursor. This innovation in assistive technology increases accessibility for individuals with disabilities, allowing them to interact with computers in an inclusive manner.

1.2 Purpose:

The main goal of the Eyeball Cursor Movement Project is to develop a reliable system that allows users to control computer cursors with eye movements using OpenCV, the project calls for it will empower individuals with physical disabilities by providing an alternative to electronic communication.

1.3 The Main Features:

1. Real-time Eye Tracking: The system uses OpenCV to capture and process live video input from a webcam, allowing for real-time tracking of the user's eye movements.

2. Cursor Calibration: The system incorporates a calibration process to tailor the cursor movement to the user's unique eye movement patterns, enhancing accuracy and responsiveness.

3. Adaptive Sensitivity: To accommodate users with varying degrees of motor control, the system features adjustable sensitivity settings to customize cursor movement speed and responsiveness.

4. User-Friendly Interface: The user interface is designed to be intuitive, ensuring a seamless experience for individuals with limited mobility or dexterity.

1.4 Problem Definition:

Traditional wearable devices pose challenges for individuals with motor impairments. The project aims to address these challenges by using OpenCV for real-time eye tracking, providing a reliable and simple alternative to electronic communication The basic challenges some include accurate eye tracking, optimizing eye attributes, ensuring minimum delay, and addressing privacy concerns when tracking eyes.

II. LITERATURE REVIEW

2.1 Existing Systems:

Eye tracking combined with wheelchair-like devices has gained attention in the assistive technology sector. MATLAB has been used for iris detection and cursor control, but accurately connecting the center of the eye can be difficult. OpenCV offers great accuracy, especially for eye-tracking applications. Eye-movement-monitored wheelchairs allow the user's eyes to move naturally for steering, increasing mobility and independence.

2.2 Survey:

Toward a theory of cognitive processes in computer programming (Brooks, R.E.): The aim of this work is to establish a theory of information processing to explain the cognitive processes in computer programming. It provides rationale, methodology, and coding strategies and provides a clear model for the coding process.

Monitoring student's cognitive processes during program preparation— eye-movement approach (Cheng-Chih Wu, Ting-Yun Hou): This study examines students' mental processes as they use behind the eyes to edit processes. It compares the eye movements of low- and low-performing students during the error correction task. The findings suggest that high-performing students approach debugging tasks in a logical manner, whereas low-performing students follow sequential lines and struggle to grapple with higher-order reasoning

Cognitive processes and looping constructions: An empirical study (Ehrlich, K. and Soloway, E.): This paper explores the relationship between cognitive processes and looping constructs in programming. It presents the two most common looping methods found in Pascal programs: READ/PROCESS and PROCESS/IO. The findings suggest a preference for the READ/PROCESS pattern and show that certain looping constructions facilitate specific cognitive processes.

Novel eye tracking path in natural head movements (Qiang Ji and Zhiwei Zhu): This study proposes new eye gaze tracking techniques that allow for natural head movements and reduce the need for individual measurements. It presents two methods for estimating 3D optical vision, one directly estimating the gaze and the other using the vision mapping function. These techniques aim to improve the utility of optical tracking.

2.3 The Proposed System:

OpenCV plays a vital role in real-time visualization and facial feature analysis for sleep recognition. It facilitates the identification of facial edges and facial markings, and enables accurate measurement of eye movements for cursor control. The system analyzes video streamed from a web browser, recognizes facial landmarks, and tracks eye movements without the need for measurements or external sensors.

D-lib supports the system with facial mark recognition tools, specifically targeting eye and facial regions. It integrates seamlessly with OpenCV, providing better facial symbol recognition. Accurate detection of facial features allows accurate measurement of eye movements necessary for cursor manipulation.

III. REQUIREMENTS ANALYSIS

3.1 Functional Requirements

Eye Detection and Tracking: The system should accurately detect and track the user's eyes in real-time using the webcam feed. It should be able to handle various lighting conditions and user positions effectively. The eye detection algorithm should adapt to different head orientations and angles.

Pupil Tracking: The system should accurately track the user's pupils and estimate their positions within the eye regions. It should be able to handle variations in pupil size, eye movements, and head movements. Pupil tracking should remain robust even in dynamic scenarios where the user's head or eyes are in motion.

Gaze Estimation: The system should estimate the user’s gaze based on known student locations. Gaze estimation should be accurate and responsive, allowing the system to determine where the user is looking on the screen. It must adapt quickly and accurately to changes in the user’s vision.

Cursor Control: The system should translate the estimated gaze direction into cursor movements on the screen using PyAutoGUI. Cursor movements should be smooth, precise, and responsive to the user's eye movements. It should support basic cursor actions such as clicking, dragging, and scrolling using eye movements.

Calibration and Personalization: The system should provide a calibration procedure to accommodate individual differences in eye anatomy and visual processing. The measurement should be user-friendly and intuitive, effectively guiding the user through the process. Personalization options such as sensitivity adjustment or custom measurement profiles can be added.

3.2 Non-Functional Requirements

Performance: The system should react in real-time to eye movements, allowing little lag between the point of view and the movement of the cursor.

Efficiency: The system must be efficient, making efficient use of system resources such as CPU and memory to ensure optimal performance without significantly impacting overall system performance

Scalability: The system needs to be scalable to accommodate higher levels of use or a larger user base without compromising performance.

Usability: The user interface should be intuitive and user-friendly, guiding users through the measurement process and providing clear information on eye tracking and cursor movement

Accessibility: The system should be accessible to disabled users, support alternative input methods and provide options for adaptation to meet the needs of different users.

Error handling: Error messages should be clear and informative, and guide users on how to resolve issues or seek help.

Data privacy: The system may prioritize the privacy of the user, ensuring that any data collected through the eye tracking is handled securely and in accordance with applicable privacy laws is the right.

Secure communication: When the system communicates with external servers or devices, data transmissions must be encrypted to protect against unauthorized access or interception.

IV. SYSTEM DESIGN

System Architecture:

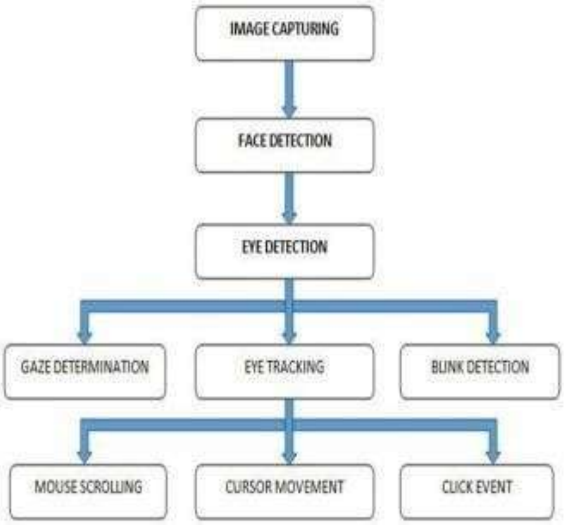

Fig.4.1. System Architecture

The described machine presents a unique method to human-pc interplay, the usage of a completely unique video digicam placed above the laptop display to screen consumer eye actions This device affords customers with computer connectivity to any physical attachment or outside hardware Enables redundant conversation.

Instead, the machine is predicated solely at the consumer's eye movement as an enter, imparting a non-intrusive and intuitive method of manage. By analyzing the video feed of the user's eyes, the pc maintains track of where the user is calling at the display screen, as a way to consciousness and "select" keys him for a sure time, close his eyes and "press" the keys.

A key function of this machine is its calibration-free operation, which gets rid of the want for users to go through a calibration manner earlier than the use of the system. This simplifies the consumer revel in and simplifies the machine, making the gadget reachable to a greater variety of users.

Additionally, the device performs assessments for lighting fixtures situations to make certain optimum overall performance of the digital camera, as insufficient lighting fixtures could lead to mistakes in eye tracking. Upon receiving streaming video from the digital camera, the system breaks it down into frames and converts them to black and white to facilitate evaluation.

By figuring out the center of the iris, the device calculates a midpoint representing the consumer's gaze route. This statistics is then used to move the mouse cursor on the display screen, allowing the user to navigate and have interaction with graphical elements. Moreover, the machine incorporates a mechanism for simulating mouse clicks, enabling customers to carry out actions together with clicking on buttons or choosing gadgets by means of blinking their eyes for a targeted period. Overall, this machine gives a singular and person-friendly approach to human-pc interplay, leveraging eye monitoring technology to provide a arms-loose and intuitive manipulate interface.

Methodology:

The methodology for implementing image capturing, face detection, eye and nose detection, and cursor movement using OpenCV involves a series of interconnected steps leveraging computer vision techniques. For example, in scientific research, methodology describes the steps of data collection, analysis, and conclusions. First, the system starts capturing images through a webcam or camera using OpenCV's video capture functionality. This continuously captures images from the camera, providing a baseline for subsequent analysis.

As each frame is captured, the system uses a face recognition algorithm to find and recognize the faces in the image. Methods such as Haar cascades or deep learning-based models are often used for this purpose. Once the face is detected, the system proceeds to extract the region of interest corresponding to the eyes and nose in the detected facial region.

Then, the system uses dedicated algorithms or models to detect eyes and noses in different regions of interest. These systems analyze objects and shapes in the eye and nose regions to accurately identify locations and characteristics. Techniques such as necklace filling, deep learning-based modeling, or object-based methods can be used to achieve this.

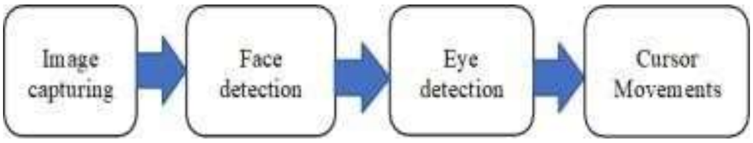

Fig.4.2. Methodology

Recognizing the eyes and nose, the system interprets their position and movement to determine how the cursor will move on the screen. For example, eye movements can be translated into a corresponding on-screen cursor, giving users intuitive control over cursor placement In addition, eyes can be paused movements and specific hands detected by the system to trigger actions such as clicking or scratching. During this process, the system continuously processes the incoming images, updates the locations of known facial features, and adapts the cursor movement accordingly in real time.

V. IMPLEMENTATION

Introduction to PyAutoGUI Library:

PyAutoGUI is a Python module designed to automate keyboard and mouse interactions on platforms such as Windows, macOS, and Linux. Streamlines automation of repetitive tasks, GUI testing and bot creation by providing an intuitive user interface for simulating keyboard inputs, mouse movements and clicks.

Key Features:

Mouse Control: Developers can programmatically manage the mouse cursor, performing actions such as moving to specific coordinates, clicking at designated positions, dragging and dropping objects, and scrolling within windows or applications.

Keyboard Input: PyAutoGUI facilitates the simulation of keyboard inputs, allowing for tasks like typing text, triggering specific keys or combinations, and executing keyboard shortcuts to automate text entry and application commands.

Screen Capture and Recognition: The library offers functions for capturing desktop screenshots or specific screen regions, along with basic image recognition capabilities. This feature aids in automating tasks within applications with non-standard or dynamically changing interfaces.

Window Management: PyAutoGUI provides utilities for interacting with windows and applications, including bringing windows to the foreground, minimizing or maximizing them, and closing or resizing windows.

Introduction To Dlib Library:

The dlib library stands as a versatile toolkit bridging machine learning, computer vision, and image processing domains. Crafted primarily in C++, it extends its reach with Python bindings, offering accessibility and usability across diverse developer communities.

Key Features And Capabilities:

Facial Recognition: Leveraging deep metric learning techniques, dlib excels in facial recognition tasks. Its face recognition model, rooted in deep convolutional neural networks (CNNs), delivers precise and swift identification from images and video streams. Applications span from access control to surveillance and biometric authentication.

Object detection and tracking: Equipped with powerful algorithms such as the Histogram of Oriented Gradients (HOG) detector and correlation-based tracking, dlib empowers developers in object detection and tracking efforts From pedestrians who seen to vehicle tracking and motion analysis, these tools facilitate various visual insights performing tasks.

Facial Mark Recognition: In this arsenal are algorithms that are skilled in recognizing important facial marks including eyes, nose, mouth and eyes These are important points for tasks like facial recognition fill, reference analysis and head position estimation, reinforced by the accuracy and efficiency of dlib

Shape Prediction and Regression: Developers harness dlib's algorithms for shape prediction and regression to extrapolate spatial arrangements of landmarks or features from input data. This functionality underpins applications ranging from facial expression analysis to gesture recognition and medical image analysis, epitomizing dlib's versatility.

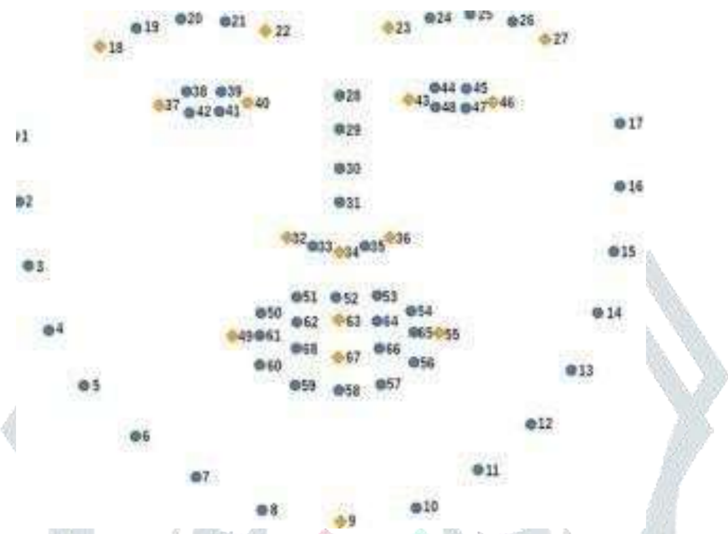

Fig.5.1. 68 facial landmark positions

Eye Aspect Ratio:

Only the eyes are taken into account while determining eye ball motions. The eye is designated by 6 (x, y) coordinates, starting from the far left corner and proceeding clockwise to the right, encompassing the remaining region of the eye, as seen in the image.

In summary, the exclusive focus on the eyes for determining eye ball motions, coupled with the delineation of eye boundaries using 6 (x, y) coordinates, represents a targeted and effective approach to eye tracking and analysis.

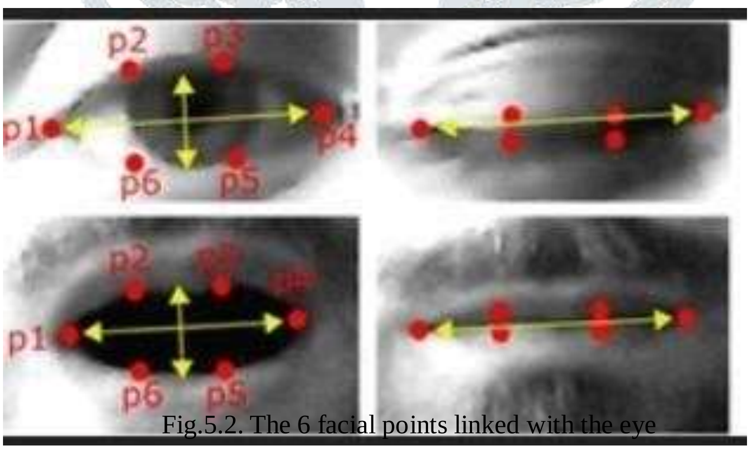

Fig.5.2. The 6 facial points linked with the eye

Mouth Aspect Ratio:

Mouth aspect ratio over Eye aspect ratio (MOE) Finally, we decided to add MOE as another feature. MOE is simply the ratio of the MAR to the EAR. The benefit of using this feature is that EAR and MAR are expected to move in opposite directions if the state of the individual changes.

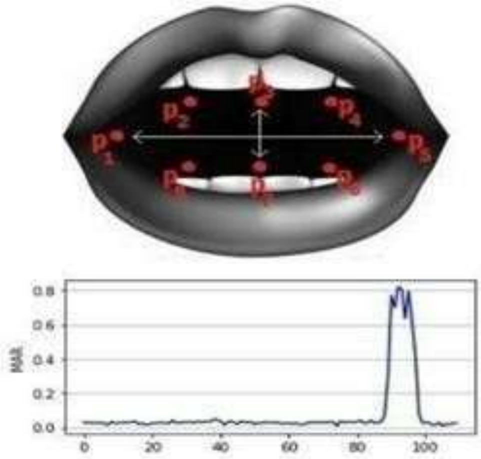

Fig 5.3. The 12 facial points linked with the mouth

VI. RESULTS AND ANALYSIS

1. The intial state where the webcame will open and when we open the mouth the program will start and the read the input as shown in fig.6.1.

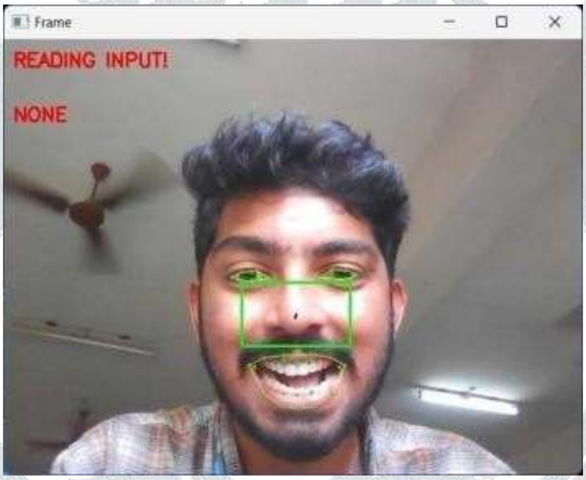

Fig.6.1. Reading Input

2. If the user want to move the cursor towards the left direction then the user should move anchor point to the left.

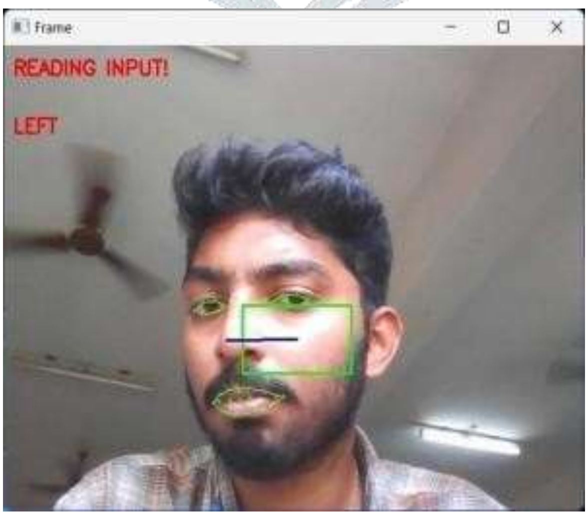

Fig.6.2. Reading input towards left direction

3. If the user want to move the cursor towards the right direction then the user should move anchor point to the right.

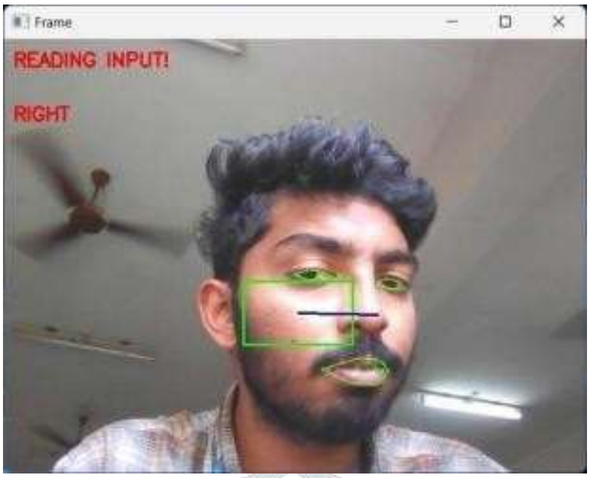

Fig.6.3. Reading input towards right direction

4. If the user wants to move the cursor towards downward direction then the user should move the anchor point down.

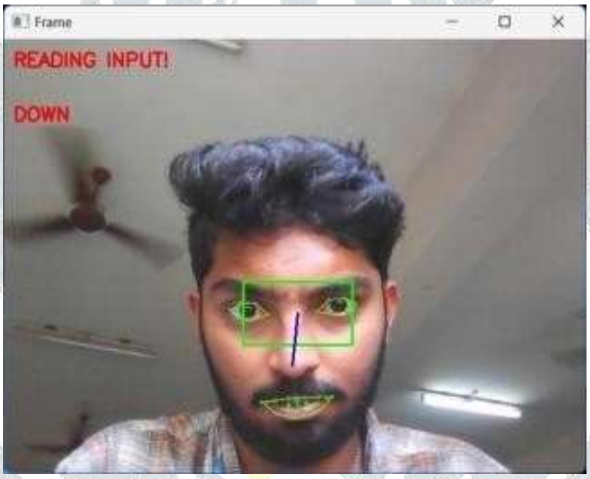

Fig.6.3. Reading input downwards

5. If the user wants to move the cursor towards upwards direction then the user should move the anchor point up.

Fig.6.5. Reading input upwards

6. If the user wants to scroll the screen then user should wink the both eyes at the same time.

Eye Wink Detection: The system continuously monitors the user's eyes using eye tracking technology, such as the Eye Aspect Ratio (EAR) technique. It identifies instances where both eyes are winking simultaneously.

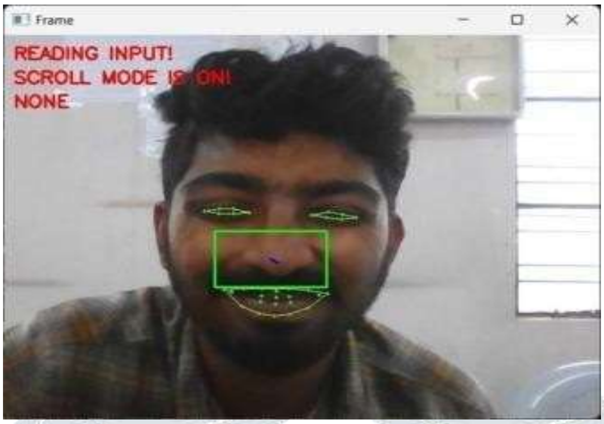

Fig.6.6. Scrolling mode On/Off

7. If the user wants to perform left click action then user should wink the left eye perfectly.

The system employs the calculation of the eye aspect ratio (EAR) as a determinant for initiating left-click actions. By continuously monitoring the ratio of the distance between key landmarks of the eye, such as the vertical distance between the top and bottom eyelids and the horizontal distance between the inner and outer corners, the system discerns when the user intends to perform a left-click action.

Fig.6.7. Left Click operation

8. If the user wants to perform right click action then user should wink the right eye perfectly.

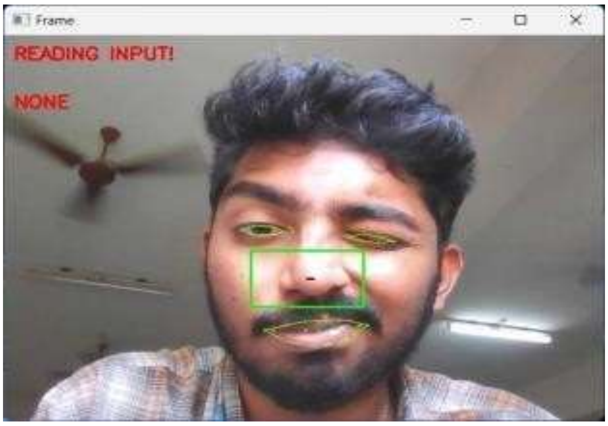

Fig.6.8. Right Click operation

VII. CONCLUSION

The conclusion summarizes the importance of the proposed program for the development of assistive technology, and highlights its potential to increase independence, employment, and social inclusion for individuals with disabilities This program demonstrates the way forward furthermore, combines eye-tracking technology with virtual keyboard control for a more flexible and precise computer interface.

The findings also highlight the broader implications of the program for improving and adapting users' overall lives to meet individual needs and preferences This highlights the importance of continuous adaptation, adaptation and conduct emphasize good considerations to maximize program impact and ensure responsible implementation.

Overall, the conclusions articulate the transformative potential of the proposed framework to effectively bridge the gap between individuals with disabilities and digital technologies, and ultimately foster an inclusive society all together and equal.

VIII. FUTURE WORK

Looking ahead, the Eyeball Cursor Movement project opens avenues for continuous improvement and expansion. Future developments could concentrate on enhancing the accuracy of eye-tracking algorithms to provide even more precise cursor control. Additionally, incorporating gesture recognition based on eye movements could further expand the system's functionality. Exploring integration with virtual reality (VR) environments presents an exciting opportunity to bring eyecontrolled cursor movement to immersive experiences.

Additionally, implementing accessibility features, like voice commands and alternative input methods, ensures inclusivity for users with diverse needs. Multi-user support facilitates collaborative usage scenarios, while integration with various applications and platforms expands the system's utility across different use cases. Exploring mobile and wearable integration opens up new possibilities for hands-free interaction, while maintaining robust security and privacy measures safeguards user data. Regular feedback collection and iterative improvements based on user input ensure the system evolves to meet evolving user needs, providing an optimal and seamless interaction experience.

IX. REFERENCES

1. Brooks,R.E.(1997) ―Towards a theory of the cognitive processes in computer programming,‖ Int. J. Man-Mach. Studies, vol. 9, pp. 737–751.
2. Cheng- Chih Wu, Ting-Yun Hou(2015)‖Tracking Students' Cognitive Processes During Program Debugging‖—An Eye-Movement Approach, IEEE.
3. Ehrlich,K .and Soloway,E.(1983) Cognitive strategies and looping constructs: An empirical study,‖ Commun. ACM, vol. 26, no. 11, pp. 853–860.
4. Eric Sung and Jian-Gang Wang (2002)―Study on Eye Gaze Estimation‖, IEEE, VOL. 32, NO. 3, JUNE .
5. Murphy,L.(2008)―Debugging: The Good, the Bad, and the Flexible—A Qualitative Study of Novices' Strategies,‖ SIGCSE Bull., vol. 40, no. 1, pp. 163-167.
6. QiangJi and Zhiwei Zhu (2007)‖Novel Eye Gaze Tracking Techniques Under Natural Head Movement‖, Senior Member, IEEE, VOL. 54, NO. 12, DECE
7. Neil Castellino and Michelle Alva, "An image-based eye controlled assistive system for paralytic patients", IEEE Conference Publications, 2017.
8. Shu-Fan Lin, Xuebai Zhang, Shyan-Ming Yuan, "Eye Tracking Based Control System for Natural Human-Computer Interaction", Computational Intelligence.
9. Čech and Soukupová, "Real-Time Eye Blink Detection using Facial Landmarks", Center for Machine Perception, February 2016.
10. Suree Pumrin and Chairat kraichan, "Face and eye tracking for controlling computer functions", IEEE Conference Publications, 2014.
11. A. Geetha and M. Mangaiyarkarasi, "Cursor Control System Using Facial Expressions for Human-Computer Interaction", Vol 8 Issue 1 APRIL 2014, ISSN: 0976-1353 International Journal of Emerging Technology in Computer Science & Electronics, pp 30-34.