Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.11: 2024
ISSN :**1906-9685**

OPEN GAZE CURSOR CONTROL USING OPENCV

[1] **Mrs. K. Santoshi Rupa,** Associate Professor in Department of Computer Science And Engineering, Vignan's Institute of Engineering for Women, Duvvada.
[2] **Ch Bhandavi Yoshitha,** Student of CSE Department
[3] **Harisomayajula Yasaswi,** Student of CSE Department
[4] **Gogulamatam Mahalakshmi,** Student of CSE Department
[5] **Badri Jahnavi,** Student of CSE Department
Vignan's Institute of Engineering for Women, Visakhapatnam, Andhra Pradesh, India

ABSTRACT
It is extremely difficult for someone with physical disabilities to control the mouse. We have suggested employing eye movements to control the mouse pointer in order to provide a solution for individuals who are unable to utilize a physical mouse. An alternate method of making use of a computer is called eye gazing, which uses eye movements to operate the mouse. Eye gazing is an alternate way that lets a person control their computer with their eyes if they find using touchscreens and mouse challenging. One essential real-time input method for human-computer communication is eye movement, which is crucial for those with physical disabilities. In this system, a unique eye control system is proposed to enhance the eye tracking technique's usability, mobility, and reliability in user-computer interaction. It accomplishes this through the usage of a Webcam and doesn't require any additional hardware. The main goal of the suggested system is to offer an easy-to-use interactive mode that just requires the user's eyes. The suggested system's usage flow is made to completely mimic the regular routines of people. The suggested method explains how to utilize a webcam and Python to control the cursor position on the screen by implementing both iris and movement of the cursor based on iris position.

INTRODUCTION
In an era of rapid technological advancement, the imperative to ensure universal access to the benefits of computer technology becomes increasingly evident. Unfortunately, using traditional input devices like mouse and keyboard can be quite difficult for people with physical limitations, which makes it harder for them to interact with computers efficiently. In order to promote disabled people's inclusion in the digital sphere, this initiative addresses the urgent need for alternate forms of communication and engagement that are specifically suited to their varied demands. With an emphasis on individuals with restricted hand movement, the project aims to develop a system that allows computer interaction by eye movements alone, offering an innovative means of increased flexibility. The unique strategy

http://doi.org/10.36893/JNAO.2024.V15I1N01.208-216

of using eye movements as the primary form of control instead of standard input devices takes into account the particular difficulties experienced by people who are paralyzed or physically challenged, with the goal of making computer use not only efficient but also accessible to everybody. In order to solve this problem, this project uses OpenCV to create a novel eye-based cursor control system. The primary challenge is achieving precise and stable pupil position detection within the eye, ensuring accurate cursor control. OpenCV is an open-source computer vision and machine learning software library. Firstly it was developed by Intel, it is now maintained by a community of developers under the OpenCV Foundation. When it is integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To identify an image pattern and its various features we use vector space and perform mathematical operations on these features. CV2 module is the main module in OpenCV that provides developers with an easy-to-use interface for working with image and video processing functions. The modules PyAutoGUI and Imutils are used in the python code for this project implementation. Python's PyAutoGUI is a package that allows users to create scripts that can simulate mouse movements, click on objects, send text, and even use hotkeys. Imutils is a Python library that provides convenience functions to make working with OpenCV easier. It simplifies common image processing tasks like resizing, rotating, and displaying images. It can make image processing code more concise and readable. It's especially handy when working with OpenCV, as it complements it well by providing higher-level abstractions for common tasks. This process include the dlib library as well. Dlib is a well-known machine learning toolkit that is mostly used for computer vision and image processing applications, including object detection, facial landmark detection, face recognition, and more.

Keywords: OpenCV, CV2, PyAutoGUI, Imutils, Python, Dlib.

LITERATURE SURVEY

The authors of [1] proposed a system named as "An image based eye controlled assistive system for paralytic patients**".** This system is useful for the patients with locked-in syndrome who are paralyzed lose the ability of communication, which is a vital aspect of human life. With the exception of his eyes, the patient in locked-in syndrome is unable to move any of his voluntary muscles. In considering this, the suggested system is made to use a standard webcam to identify the patient's face and pupil using Haar cascade classifiers and the Circular Hough Transform algorithm, respectively. The suggested system shows several pictures of everyday tasks. To choose an image, the patient will need to gaze at it for longer than a certain threshold. After receiving confirmation from the patient, the system will then follow the patient's point of gaze and choose the appropriate image. The assistant will receive an audio or text message upon receipt of this confirmation.

In [2] authors used SVM classifier and EAR to build the model called **"**Real-Time Eye Blink Detection using Facial Landmarks". In this a real-time algorithm is suggested to identify eye blinks in a conventional camera video sequence. Newer landmark detectors that were trained on datasets collected in the wild show outstanding adaptability to changes in lighting, face expressions, and head direction in relation to a camera. We demonstrate that the landmarks may be identified with sufficient accuracy to determine the eye opening's level. As a result, the suggested algorithm determines the landmark locations and extracts the eye aspect ratio (EAR), a single scalar number that describes the eye opening in each frame. Ultimately, eye blinks are identified by an SVM classifier as a pattern of EAR values within a limited temporal window. On two benchmark datasets, the straightforward approach performs better than the state-of-the-art solutions.

A system termed as "Cursor Control System Using Facial Expressions For Human-Computer Interaction" was proposed by the authors of [3]. In this model eye movements are detected by the interface, which translates them into commands for cursor control. The image processing techniques used are pattern matching based eye area detection and webcam based face detection. Eye feature extraction is done using the Haar feature approach. To categorize the eye movements, the SVM classification algorithm is applied. Cursor top, bottom, left, and right movements are categorized using eye movements such as eye open, eye closure, eyeball left, and eyeball right. The circular approach is employed in the Hough transform to regulate the cursor motions.

EXISTING METHOD

The current technology is the first to use an eye movement-based cursor control mechanism to enable autonomous computer interaction for people with physical disabilities. It records private video using a webcam and processes it with Python and OpenCV. Using pre-trained models from the dlib library and a 68-point facial recognition algorithm, it locates and identifies a person's face by concentrating on their eyes. The deeper iris color is detected by the system, which then uses reference points to determine rotation to track eye movement. To control a cursor, eye motions are classified using machine learning and deep learning. Libraries use filters to reduce noise so that their activities run smoothly. Performance assessments confirm that the system works well for users who are physically challenged, providing an appropriate choice for computer-based accessible engagement.

Disadvantages

- The accuracy of the system's facial and eye detection may be impacted by variations in lighting.
- Problems with calibration could occur, affecting how precisely eye movements are tracked and thus leading to misunderstandings.
- Errors can occur when attempting to identify the iris just by colour, particularly when there is variation in eye colour.
- Demands for real-time video processing might create a strain on the system's computational capacity, which can lead to delays and reduce responsiveness.

PROPOSED METHOD

Using a unique eye-based cursor movement approach, the suggested solution is made to specifically address the demands of those with limited hand mobility. Through the use of OpenCVbased algorithms, the system seeks to identify the pupil location inside the user's eyes with accuracy and stability, hence providing accurate control over the cursor. The first step is to use the computer's webcam to record photos or video in real time, which is used as the eye tracking input source. To focus in on a smaller area of interest, the system may use face detection in addition to computer vision algorithms for eye detection. The project incorporates a head tracking module that modifies the cursor position according to the user's head orientation in order to improve accuracy. Interpreting eye movements, especially blinking, as input commands for tasks like components on the screen can be chosen or clicked. In the end, this approachable solution is ready to support equal opportunities for people with disabilities by encouraging independence and inclusivity in the digital sphere.

Advantages

- Uses eye-based cursor control to allow persons with limited hand mobility to engage with computers in a seamless manner.
- Offers a natural and intuitive means of interaction, making it easier for people with impairments to use.
- Makes use of OpenCV techniques to detect pupil position steadily and accurately, providing solid cursor control.
- Uses head tracking to fine-tune cursor positioning, improving user experience and overall system accuracy.
- Tailored to specific requirements, encouraging diversity and enabling various user groups for fair engagement in digital activities.

System Block Diagram

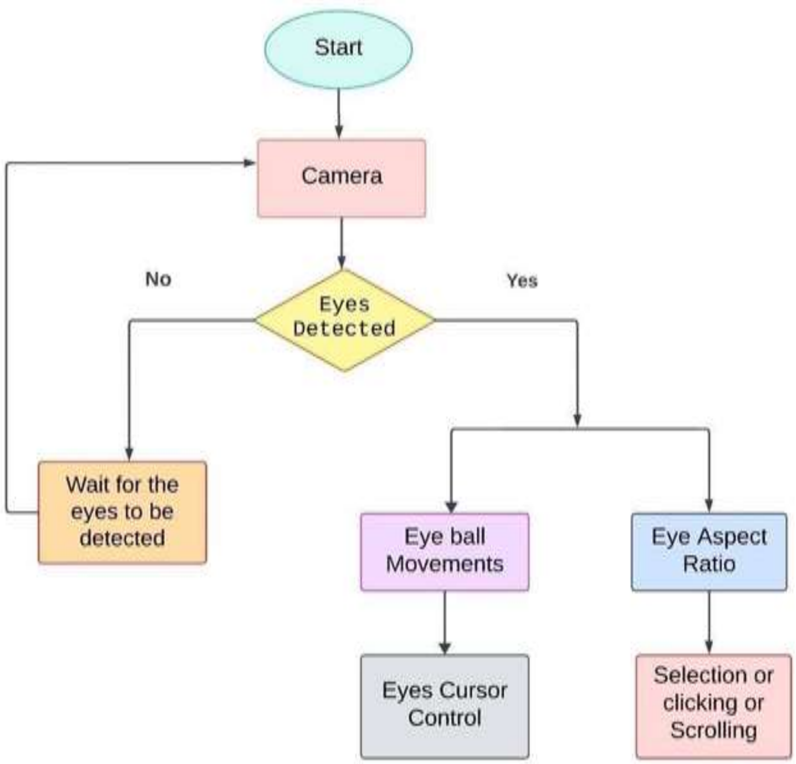


Figure 1: Workflow of the proposed system

METHODOLOGY

In this project, we will use OpenCV to connect to a webcam, extract each frame, and then feed the extracted frames to OpenCV to determine the location of the eyeballs. This will allow us to move the mouse cursor based on movements of the eyeballs. After locating an eyeball, we can extract its X and Y coordinates from OpenCV. The eyeball positions within each webcam frame are determined by the X and Y coordinates obtained from OpenCV. In relation to the webcam's field of view, these values indicate horizontal (X) and vertical (Y) locations. We utilize PyAutoGUI to convert eye motions into cursor actions using these coordinates, and then we move the mouse on the screen in accordance. This setup allows individuals to control computer interfaces simply by shifting their gaze.

The cursor movement will be operated by eyes after we activate the system with mouth. The mouth can be detected using the modules which are used for face detection. After the mouth is detected we can see the landmarks on the mouth as in the below Figure 2.

Figure 2: Facial landmarks on the mouth

After this the eyes will be detected and then we can observe the facial landmarks on eyes as shown in Figure 3. We can observe the landmarks on closed eyes as shown in the Figure 4. During the implementation of this system we can observe few points of marking on the eyes. These points are referred as P1, P2, P3, P4, P5 and P6. The terms P1, P2, P3, and so on may relate to particular locations on or around the eyes that the eye tracking algorithm has identified. These points could represent features like the corners of the eyes, the centre of the pupil, or other distinctive landmarks useful for tracking eye movements.

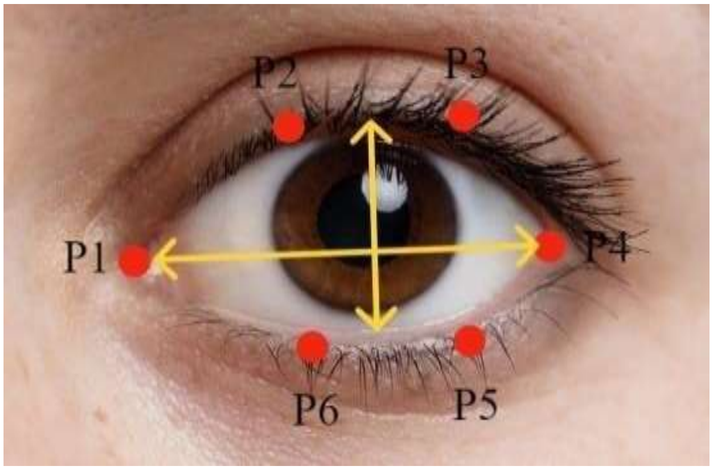


Figure 3: Facial points linked with an open eye

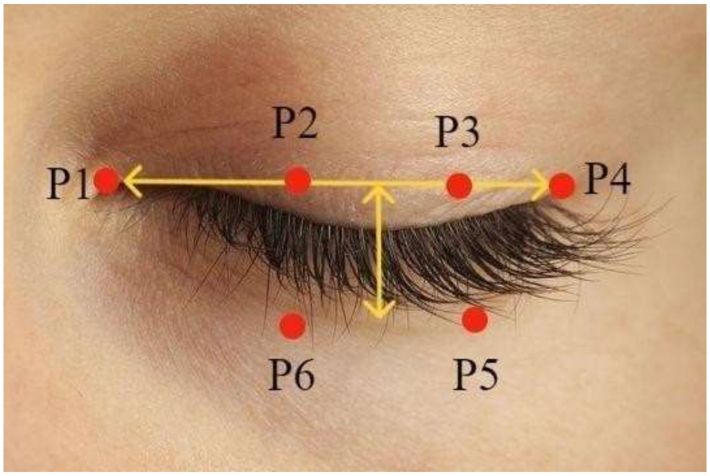


Figure 4: Facial Landmarks of the closed eye

We are utilizing the following modules to put the above principle into practice:

- **Video Recording:** This module will allow us to use the built-in VideoCapture function of OpenCV to connect our application to a webcam.
- **Frame Extraction:** This module is used to capture camera frames, extract individual images frame by frame, and submit each frame to Gaze Tracking.
- **Gaze Tracking:** This module allows us to identify eyeballs and retrieve the left and right pupils' x and y coordinates.
- **Move Cursor:** We may tell the mouse to move from its current place to the specified new x and y location by using this module.

OpenCV is a Python-based artificial intelligence API that may be used to do a variety of tasks on images and videos, including face identification, image recognition, eye tracking, and conversion of images to grayscale or colour. This API is written in C++, and Python programmers can use native language programming to invoke C++ methods from Python.

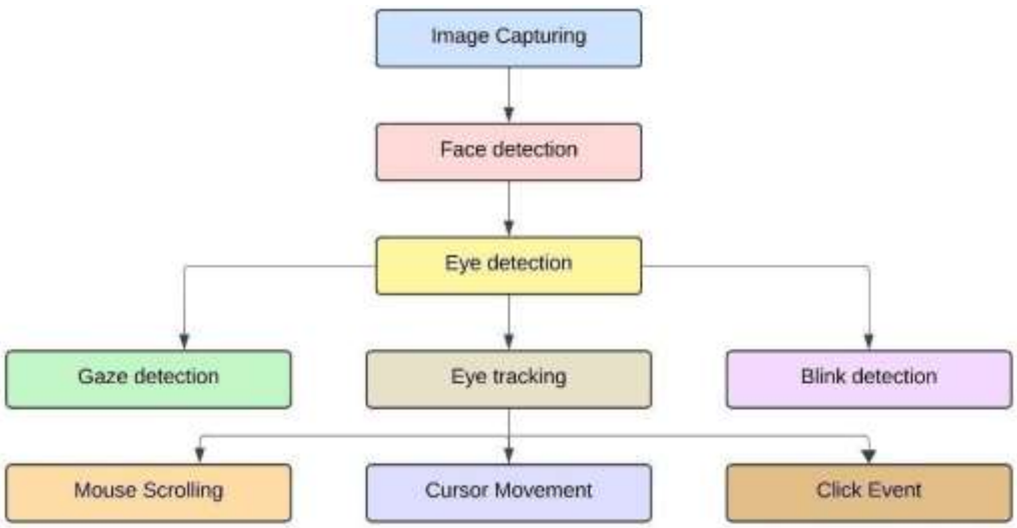


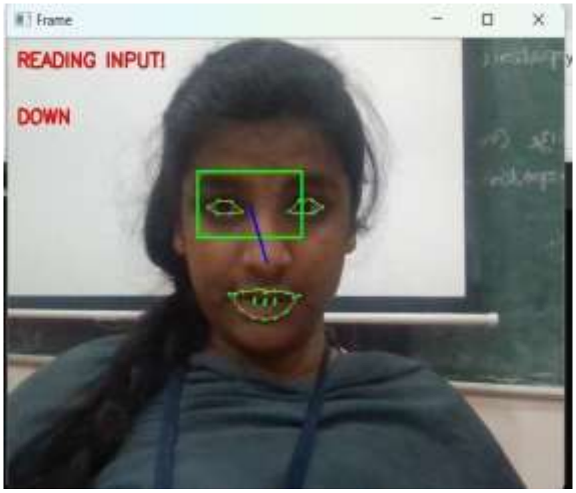
Figure 5: Architecture of the system

Steps needed to use OpenCV for face detection :

- Considering our prerequisites, we will require an image, to begin with. Afterwards, we must develop a cascade classifier, which will finally provide the facial features.
- Using OpenCV to read the picture and features file is what this step involves. Thus, NumPy arrays are currently present at the main data points. To find the row and column values of the

face NumPy N dimensional array, all we have to do is search for them. This is the array with the face rectangle coordinates. • This final step involves displaying the image with the rectangular face box.

OUTPUT SCREENS



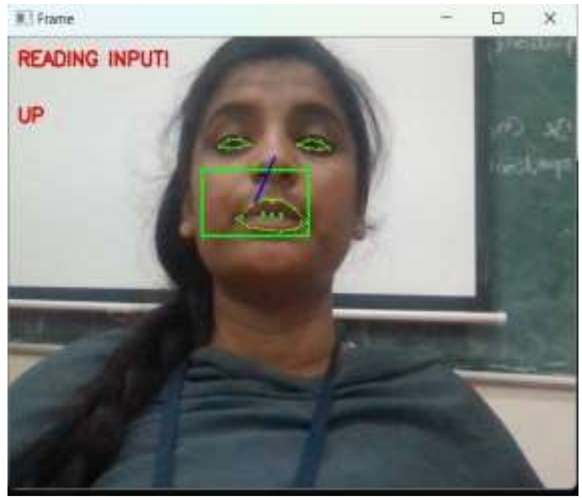
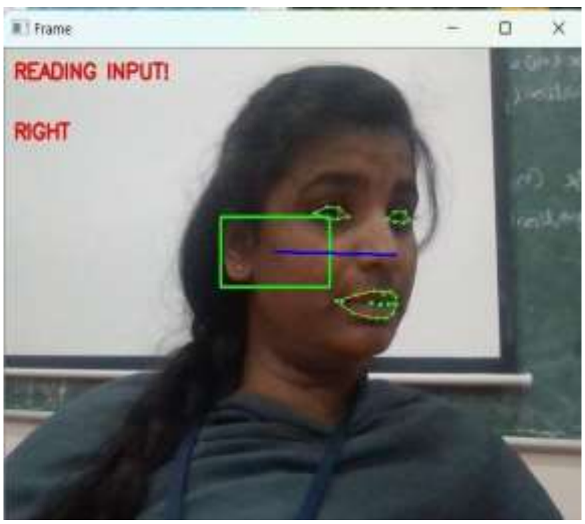


Figure 6: Reading input as DOWN Figure 7: Reading input as UP



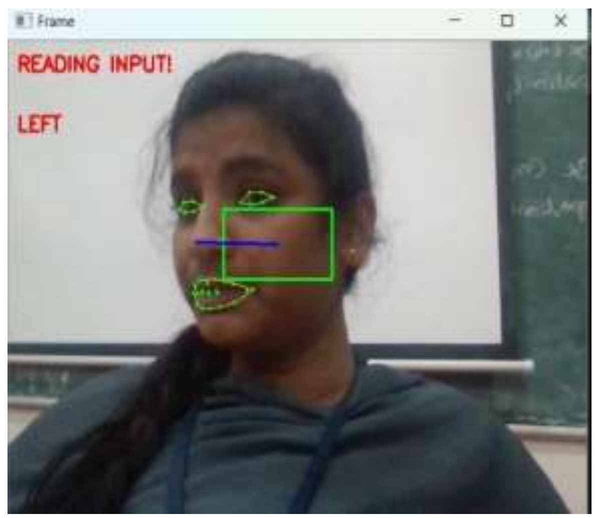


Figure 8: Reading input as LEFT
Figure 9: Reading input as RIGHT

Figure 10: Scroll Mode is set to DOWN Figure 11: Scroll Mode is set to UP

CONCLUSION

By developing a technology that enables individuals with disabilities to efficiently communicate with computers through eye movements, the project has accomplished its main objective. This successful implementation means that individuals who may have limited or no use of their hands can still use computers independently. The utilization of advanced algorithms by the project for tasks has greatly enhanced the accessibility of computers for people with disabilities. Effective cursor control depends on the system's ability to track a user's eye movements, which is guaranteed by these methods. The initiative has the potential to significantly enhance the overall quality of life of impaired individuals by giving them a way to engage with computers more effortlessly and independently. Effective use of the digital environment creates opportunities that wouldn't normally exist for employment, education, entertainment, or communication. To sum up, this effort is a big step toward ensuring that everyone, regardless of physical ability, can use technology more inclusively and easily. It bridges the gap between technology and disability and is consistent with the larger social objective of building an Information Society in which all individuals may engage completely and effectively in the digital era.

REFERENCES

[1] Neil Castellino and Michelle Alva, “An image-based eye controlled assistive system for paralytic patients”, IEEE Conference Publications, 2017.

[2] Tereza Soukupová; Jan Cech, “Real-Time Eye Blink Detection using Facial Landmarks”, Center for Machine Perception, February 2016.

[3] 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.

[4] Suree Pumrin and Chairat kraichan, “Face and eye tracking for controlling computer functions”, IEEE Conference Publications, 2014.

[5] Shu-Fan Lin, Xuebai Zhang, Shyan-Ming Yuan, "Eye Tracking Based Control System for Natural Human-Computer Interaction", Computational Intelligence and Neuroscience, 2017.

[6] John J. Magee, Margrit Betke, James Gips, Matthew R. Scott and Benjamin N. Waber, "A Human-Computer Interface Using Symmetry Between Eyes to Detect Gaze Direction", IEEE Trans, vol. 38, no. 6, pp. 1248-1259, Nov 2008.

[7] Sunita Barve, Dhaval Dholakiya, Shashank Gupta and Dhananjay Dhatrak, "Facial Feature Based Method For Real Time Face Detection and Tracking I-CURSOR", International Journal of Engg Research and App., vol. 2, pp. 1406-1410, Apr 2012.

[8] Samuel Epstein-Eric Missimer Margrit Betke, "Using Kernels for a video-based mousereplacement interface", Springer link, Nov 2012.