VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI

Seminar Report (22MCA29)

on

I Mouse: Eyes Gesture Control System

Submitted in partial fulfillment of the Requirements of the 2nd Semester in

MASTER OF COMPUTER APPLICATIONS

By

Kiran K

1RN22MC021

Under the Guidance of

Mr. Hemanth Kumar
Assistant Professor
Dept. of MCA, RNSIT

ESTD: 2001
An Institute with a difference

RNS Institute of Technology
Dr. Vishnuvardhan Road, Rajarajeshwari Nagar Post, Channasandra,
Bengaluru - 560098.

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI

Seminar Report (22MCA29)

on

I Mouse: Eyes Gesture Control System

Submitted in partial fulfillment of the Requirements of the 2nd Semester
in

MASTER OF COMPUTER APPLICATIONS

By

Kiran K

1RN22MC021

Under the Guidance of

Mr. Hemanth Kumar
Assistant Professor
Dept. of MCA, RNSIT

ESTD: 2001
An Institute with a difference

RNS Institute of Technology
Dr. Vishnuvardhan Road, Rajarajeshwari Nagar Post, Channasandra,
Bengaluru - 560098.

2022-23 Even Semester

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvardhan Road, Rajarajeshwari Nagar Post, Channasandra,
Bengaluru – 560098

ESTD: 2001
An Institute with a difference

Department of Master of Computer Applications

CERTIFICATE

This is to Certify that **Mr. Kiran K.** bearing USN: **1RN22MC021**, student of 2nd semesterMCA has satisfactorily completed the **Seminar work – 22MCA29** entitled **"I Mouse: Eyes Gesture Control System"** in the academic year **2022-2023** as prescribed by VTU for II Semester of Master of Computer Applications.

Mr. Hemanth Kumar	**Dr. N P Kavya**	**Dr. Ramesh Babu H S**
Assistant Professor	**Professor and Head**	**Principal**
Dept. of MCA	**Dept. of MCA**	**RNSIT**

Examiners

1.________________________

2. ________________________

ACKNOWLEDGEMENT

I am overwhelmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

I take this opportunity to acknowledge the help I have received from different individuals who directly or indirectly helped in completion of this **I Mouse: Eyes Gesture Control System** Work.

I express my sincere words of gratitude to **Dr. R N Shetty**, Founder and **Sri. Satish R Shetty**, Managing Trustee, RN Shetty Trust & Chairman, RNS group of institutions for providing us wonderful academic environment.

I am deeply indebted to our beloved Principal,**Dr. Ramesh Babu H S ,** RNSIT for providing the necessary facilities to carry out this work.

I am extremely grateful to **Dr. N P Kavya,** Professor and Head, Department of MCA, RNSIT for nurturing our technical skills and contributing to the success of this project.

I would also express my heartfelt thanks to our Seminar Coordinator **Ms. Roopa H M and Ms. Sowmya V**, Assistant Professor, Department of MCA, RNSIT and my Internal guide **Mr.Hemanth Kumar,** Assistant Professor, Department of MCA, RNSIT for their continuous guidance and valuable suggestions for this Seminar work.

I also express my heartfelt thanks to all the teaching and non-teaching staff members of MCA Department for their encouragement and support throughout this work.

Kiran K

1RN22MC021

ABSTRACT

"I Mouse: Eyes Gesture Control System"

A high number of people, affected with neuro- locomotor disabilities or those paralyzed by injury cannot use computers for basic tasks such as sending or receiving messages, browsing the internet, watch their favorite TV show or movies. Through a previous research study, it was concluded that eyes are an excellent candidate for ubiquitous computing since they move anyway during interaction with computing machinery. Using this underlying information from eye movements could allow bringing the use of computers back to such patients. For this purpose, we propose an I mouse gesture control system which is completely operated by human eyes only. The purpose of this work is to design an open-source generic eye-gesture control system that can effectively track eye-movements and enable the user to perform actions mapped to specific eye movements/gestures by using computer webcam. It detects the pupil from the user's face and then tracks its movements. It needs to be accurate in real-time so that the user is able to use it like other every-day devices with comfort.

Table of Content

(List with page numbers for easy navigation.)

LIST OF FIGURES

INTRODUCTION

Personal computers were initially used for solving mathematical problems and word processing. In recent years, however, computers have become necessary for every aspect of our daily activities. These activities range from professional applications to personal uses such as internet browsing, shopping, socializing and entertainment.

Computers are designed to be readily accessible for normal individuals. However, for individuals with severe physical disabilities such as cerebral palsy or amyotrophic lateral sclerosis, usage of computers is a very challenging task. There have been many research studies on human computer interface (HCI) to improve the interaction between the user and the computer system. Most of these are applicable only to normal individuals. These interfacing methods include a touch sensitive screens, speech recognition methods and many others. Despite the success of these techniques, they were not suitable for the physically disabled individuals.

Many researchers have tried to develop methods to help the disabled to interact with computers by using signals such as electroencephalography (EEG) from the brain, facial muscles signals (EMG) and electro-oculogram (EOG) [1-3]. Other methods include limbus, pupil and eye/eyelid tracking [4-5], contact lens method, corneal, pupil reflection relationship [6] and head movement measurement [7]. These methods require the use of attachments and electrodes to the head, which makes them impractical. Other high end techniques [8] that are based on infrared tracking of the eye movements to control computers were exceptionally expensive and were not affordable for those who need them.

METHODOLOGY

It falls in the field of Human-Computer Interaction (HCI) and shows that by improving upon existing open source frameworks utilized for the purpose of Computer Vision and HCI, a cheap eye tracking solution can be produced for the benefit of disabled patients. Fig. 2 shows the system overview and model of the system

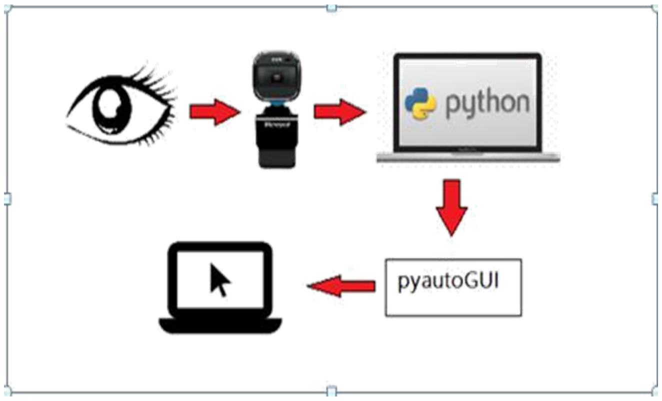


Fig. 1. Block Overview of the Designed System

The System prototype takes input from a camera and recognizes the user‟s pupil and tracks it in real-time [25]. This"tracking" information can then be used by computers or micro-controllers to perform various tasks, some of these tasksthat the project aims to achieve is to track the pupil-movement[26] and then store that tracked eye movement to control the mouse pointer of a computer, so that someone with a disabilitylike say Amyotrophic Lateral Sclerosis can use, to communicate with others

. Aims and Objective

- Provide a cheap eye-tracking system.
- To control the cursor of a computer with eyes.
- Allow physically disabled people to use computers.
- To control a computer and communicate with othersystems.
- To provide a real-time accurate eyes gesture controlsystem.
- To provide a hand free mouse control system.
- To provide a complete generic eye-gesture mousecontrol system.
- To provide a complete wire free mouse control system.
- Easy to control cursor movement of a mouse.

TOOLS AND TECHNIQUES

- **Python**

Python, an average unique programming language, is progressively utilized as a part of numerous application areas. Dynamic features in python enable programmers to change code at runtime [27]. Some unique features, for example dynamic compose checking, have a functioning impact in performance. Along these lines dynamic component code is frequently changed to provide efficient programmingdevelopment.

- **NumPy**

NumPy [28], is a low-level library written in C++ (and FORTRAN) for exceptional state scientific capacities. NumPycunningly beats the issue of running slower algorithms on Python by utilizing multidimensional model and structures that works on clusters. Any algorithm would then be able to becommunicated as a structure of models, enabling the algorithm to run rapidly.

- **Scipy**

Scipy [28], is a library that utilizes NumPy for more scientific work. SciPy utilizes NumPy model as the essential information structure, and accompanies modules for different tasks in logical programming, including straight variable based math, joining (analytics), normal differential condition settling and flag processing (eyes signals preparing through webcam)), straight variable based math, joining (analytics), normal differential condition settling and flag processing (eyes signals preparing through webcam)).

- **OpenCV**

At first all the new modules ought to be produced independently, and distributed in the opencv_contrib archive [29]. Afterward, when the module develops and picks up estimations, it is moved to the focal OpenCV archive and in this manner they should not be freed as a piece of control for OpenCV [30] dispersion, since the library keeps up parallel similarity, and attempts to give fair execution and stability improvements.

- **PyautoGUI**

The reason [31], for PyAutoGUI, is to give a cross-stage Python module for GUI automation for developers. The API isintended to be as basic as conceivable with reasonable errors. PyAutoGUI can simulate moving the mouse, clicking the mouse, dragging with the mouse, pressing keys, pressing and holding keys, and pressing keyboard hotkey combinations. On Windows, PyAutoGUI has no dependencies (other than Pillowand some other modules, which are installed by pip along withPyAutoGUI). It does not need the pywiin32 module installed since it uses Python‟s type modules.

SYSTEM DESIGN

I mouse system is design in python and following modules of python are imported for working of this system.

- **NumPy:** is a Python extension module. It provides rapid and efficient operations on arrays of compatible data.
- **Scipy:** an open-source Python library which is used fortechnical and scientific computing.
- **OpenCV:** is a library of programming functions mainly focused on real-time computer vision.

Pyauto GUI:

Is a cross-platform GUI automation module that works on Python. In this, you can control the mouse and keyboard as well as you can perform basic image recognition to automate tasks on your computer

A. Use Case Diagrams

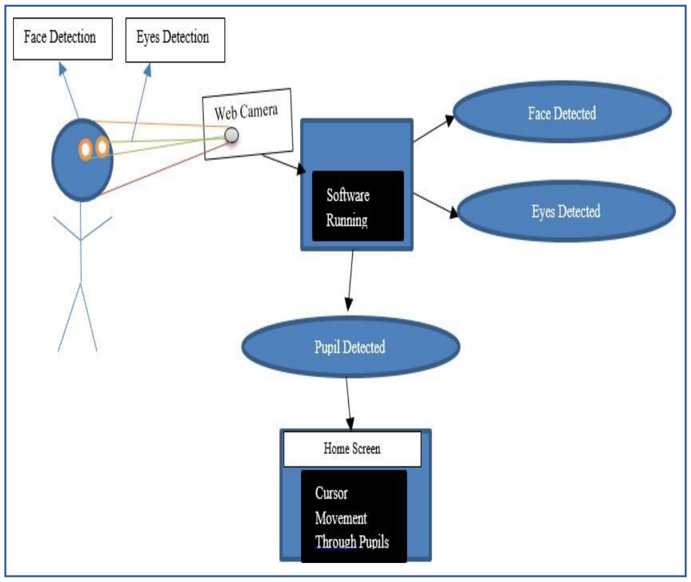


Fig. 2. System use case diagram

In 0 shows use case diagram; the system accomplishes to have the following steps.

- Software running.
- Open the webcam on the laptop and show the image of a person.
- Face detection action is performed.
- The system detects the eyes of a person.
- After the above action system move on to the nextoperation.

- ## Activity Diagram

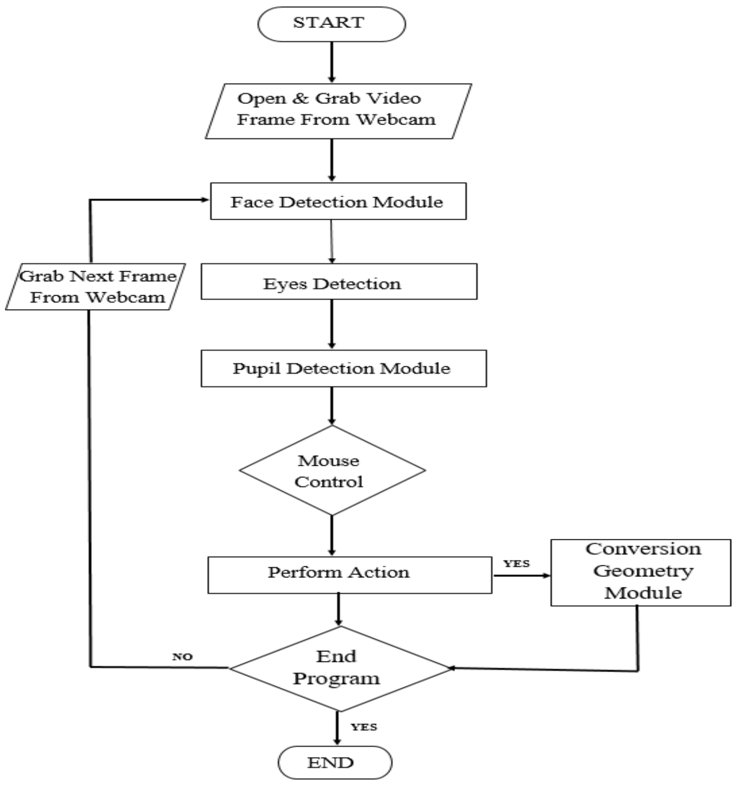


Fig. 3. Activity Diagram

Fig. 4 shows how the system operates. The following are the steps from start to end of the I Mouse software.

- Open the webcam and capture the video.
- The system performs an action and detects the face.
- The system performs eyes detection.
- The system detects the eye pupil.

With only the image of face from the webcam, the system will locate the eyes and perform geometry translations

Mouse Control: detects a gesture, moves the mouse cursor and translate the coordinates to the user's screen. Then perform the following the action.

- Scroll vertical
- Scroll horizontal
- Scroll diagonal

Repeating the whole system cycle and the system pause or system end

- **System Sequence Diagram**

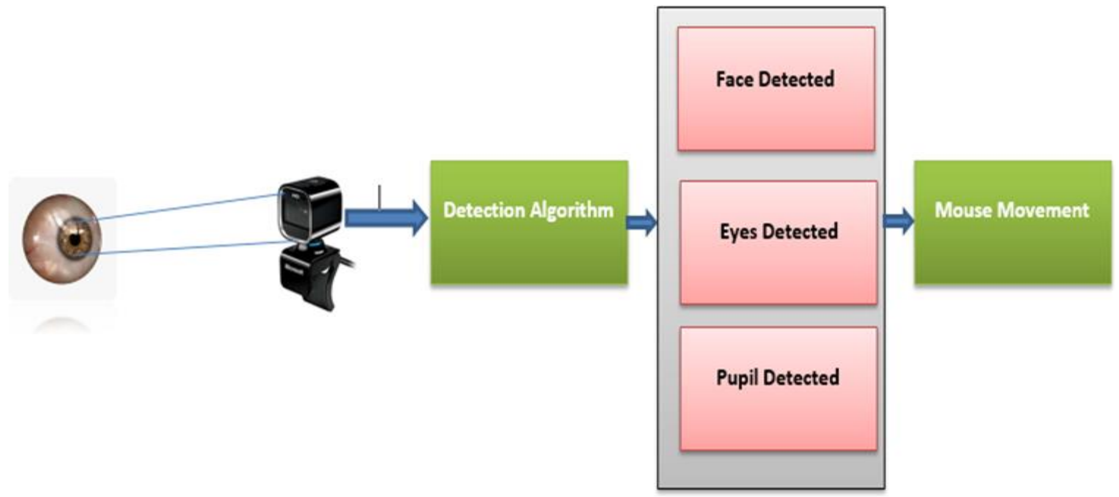


Fig. 4. Sequence Diagram

In Fig. 5 the system sequence diagram, elaborates the six basic modules of our system. In the first module, the system detects the pupil of a person through the webcam using detection algorithms. Then the system detects the face. After that the system detects and captures the eyes. Then the system detects the pupils. In the last module, the system starts moving the mouse cursor by tracking pupil movements

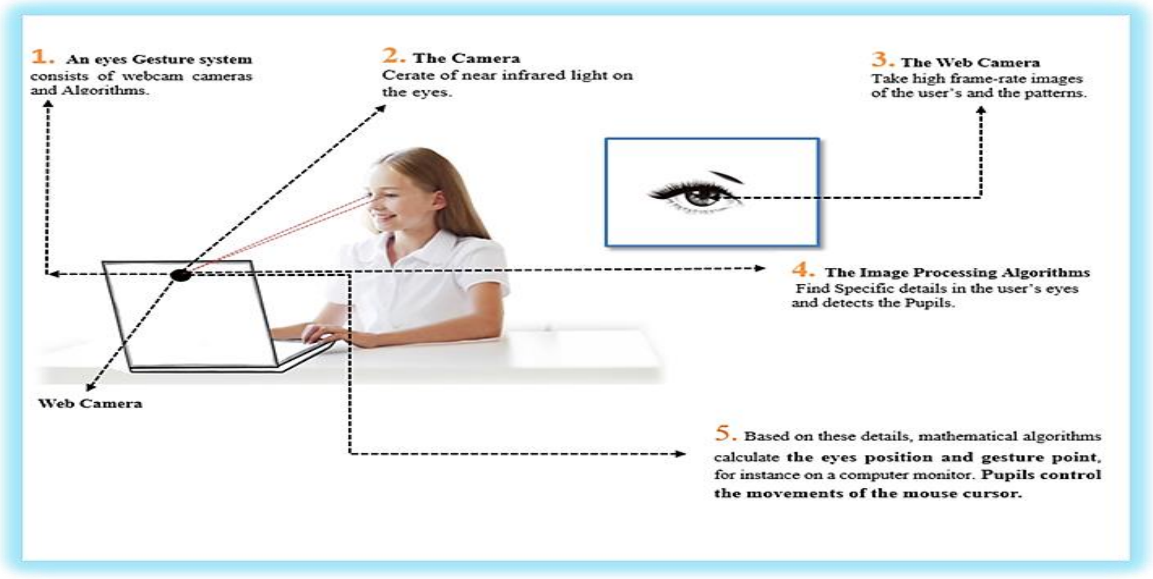

1. An eyes Gesture system consists of webcam cameras and Algorithms.
2. The Camera Cerate of near infrared light on the eyes.
3. The Web Camera Take high frame-rate images of the user's and the patterns.
4. The Image Processing Algorithms Find Specific details in the user's eyes and detects the Pupils.
Web Camera
5. Based on these details, mathematical algorithms calculate the eyes position and gesture point, for instance on a computer monitor. Pupils control the movements of the mouse cursor.

Fig. 5. System Functionality Diagram

IMPLEMENTATION

The algorithm for controlling the cursor by the eye iris movement was achieved through the following steps:

- **Face detection**

In order to capture the face image accurately, the user sat upright with the eye level parallel to the webcam as shown in the Fig. 1.

The image of the user's face is captured using a MATLAB tool called vfm [9], using which images can be captured with the help of a webcam or any other imaging tools attached to the PC. Fig. 2 shows the image captured by this tool

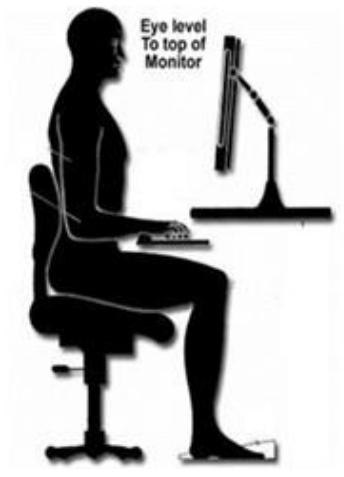
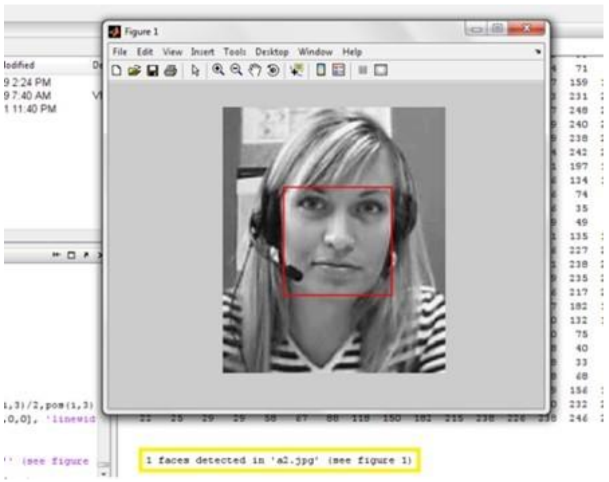


Fig. 6: System setup for eye mouse.



```
1 faces detected in 'a2.jpg' (see figure 1)
```

Fig. 7 Image detection and image capture

- **Extrating eye area**

  In order to extract the eyes region, the image of the face is divided into three equal horizontal areas. The upper third where the eyes are located is extracted as shown Fig. 3.

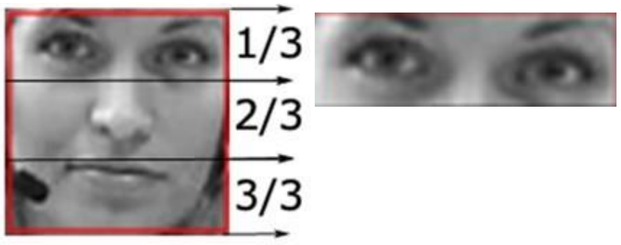
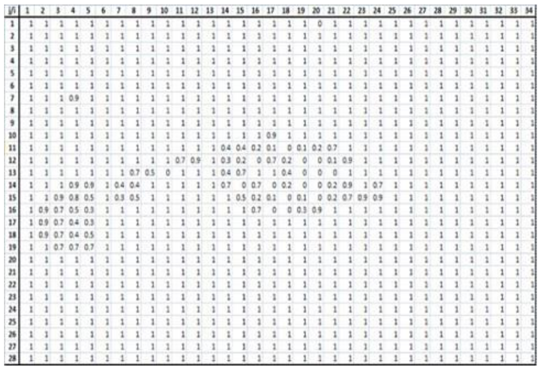


Fig. 8 Eye region extraction

- **Extracting eye location**

  The image of one eye was then extracted and normalized in order to remove the background noises and then it was converted to binary image to enhance contrast. The normalized pixel (0 – 1 scale) values of the left eye image is shown in Fig. 4.

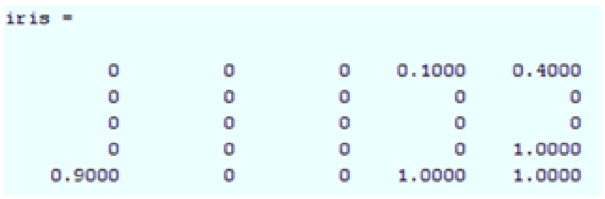


J/i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	0.9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	0.4	0.4	0.2	0.1	0	0.1	0.2	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	0.7	0.9	1	0.3	0.2	0	0.7	0.2	0	0	0.1	0.9	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	0.7	0.5	0	1	1	1	0.4	0.7	1	1	0.4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	0.9	0.9	1	0.4	0.4	1	1	1	1	1	0.7	0	0.7	0	0.2	0	0	0.2	0.9	1	0.7	1	1	1	1	1	1	1	1	1	1
15	1	1	0.9	0.8	0.5	1	0.3	0.5	1	1	1	1	1	1	0.5	0.2	0.1	0	0.1	0	0.2	0.7	0.9	0.9	1	1	1	1	1	1	1	1	1	1
16	1	0.9	0.7	0.5	0.3	1	1	1	1	1	1	1	1	1	1	0.7	0	0	0.3	0.9	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	0.9	0.7	0.4	0.3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	0.9	0.7	0.4	0.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	0.7	0.7	0.7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 9 Normalized pixels values of left eye image

- **Extacting iris region**

Since the iris region is black, corresponding pixels' values were very low and when the image was normalized, these values were approximated to zero. Hence the boundaries of iris region were determined as shown in the Fig. 5.



```
iris =

         0         0         0    0.1000    0.4000
         0         0         0         0         0
         0         0         0         0         0
         0         0         0         0    1.0000
    0.9000         0         0    1.0000    1.0000
```

Fig.. 10 Pixel's intensity values in the iris area

- **Tracking eye movement**

In order to avoid the head movement effect on the iris location, the two corners of the eye region were taken as reference, thus the iris movement was tracked accordingly.

- **Calculating iris shift**

In order to calculate the movement within the eye as the user gaze on the screen, the user was asked to look at a central pattern on the screen and the iris boundaries are determined. The user was then asked to look at other pattern to the left, right up and down with respect to the central pattern. The shift in the iris was then calculated and was used to decide the direction and position of the cursor. Fig. 6 shows the binary images of the eye when looking in different directions. In the Fig. 6, the image at the top was captured when the user was looking at the center of the screen while the middle image was captured when the user was looking to the right and lower image when the user was looking to the left. The relative shift in the iris position is used to determine the cursor position on the test GUI shown in the Fig. 7.

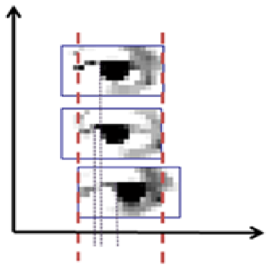
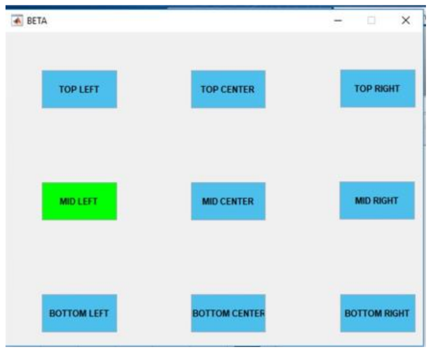


Fig. 11 : Shift in the Iris to left and right



A prolonged blink was used as a signal for clicking the pattern. The principle is that when the eyes are closed, the image variable ‘iris’ will no longer contain the actual iris. As a result, the average of the sum of the pixels in ‘iris area will be higher than that of the actual iris (since eyes closed will not give any darker region than with open eyes). The icon on the GUI where the blink occurred will change its color, indicating a click

- **H. Algorithm**

  The flow chart of the MATLAB program that was developed to perform the various stages discussed in the sections 3.1- 3.7 is given in the Figs. 8 and 9.

Proposed system

- OpenCV is a Python library which is designed to solve computer vision problems. OpenCVwas originally developed in 1999 by Intel but later it was supported by Willow Garage.
- OpenCV supports a wide variety of programming languages such as C++, Python, Java etc.Support for multiple platforms including Windows, Linux, and MacOS.
- OpenCV Python is nothing but a wrapper class for the original C++ library to be used withPython. Using this, all of the OpenCV array structures gets converted to/from NumPy arrays.
- This makes it easier to integrate it with other libraries which use NumPy. For example,libraries such as SciPy and Matplotlib.
- Next up on this OpenCV Python Tutorial blog, let us look at some of the basic operations thatwe can perform with OpenCV.

Convolutional neural networks have been one of the most influential innovations in the field of computer vision. They have performed a lot better than traditional computer vision and have producedstate-of-the-art results. These neural networks have proven to be successful in many different real-life case studies and applications, like:

- Image classification, object detection, segmentation, face recognition.
- Self-driving cars that leverage CNN based vision systems.
- Classification of crystal structure using a convolutional

neural network.And many more, of course!

The convolution layer computes the output of neurons that are connected to local regions or receptive fields in the input, each computing a dot product between their weights and a small receptive field to which they are connected to in the input volume. Each computation leads to extraction of a feature map from the input image. In other words, imagine you have an image represented as a 5x5 matrix of values, and you take a 3x3 matrix and slide that 3x3 window or kernel around the image.

RESULS AND DISCUSSION

A MATLAB based algorithm for face detection, eye region extraction, iris tracking and cursor control was successfully developed and tested. A test GUI comprised of nine boxes on the computer screen was used for the validation of the cursor

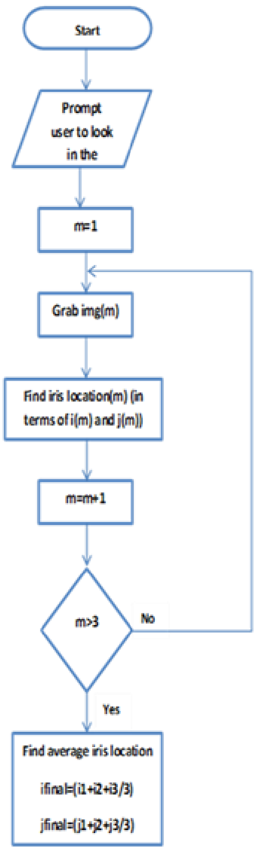

Start
Prompt user to look in the
m=1
Grab img(m)
Find iris location(m) (in terms of i(m) and j(m))
m=m+1
m>3
No
Yes
Find average iris location
ifinal=(i1+i2+i3/3)
jfinal=(j1+j2+j3/3)

Fig. 12 Flowchart of iris detection algorithm

movement as shown in Fig. 7. The iris movement was accurately tracked and the cursor was successfully moved to all nine boxes in the test GUI. Up and down movement however, was not highly consistent and the system requires further enhancement. Clicking action was also successfully tested through a change in the color of box.

CONCLUSION

A system that enables a disabled person to interact with the computer was successfully developed and tested. The method can be further enhanced to be used in many other applications. The system can be adapted to help the disabled to control home appliances such as TV sets, lights, doors etc. The system can also be adapted to be used by individuals suffering from complete paralysis, to operate and control a wheelchair. The eye mouse can also be used to detect drowsiness of drivers in order to prevent vehicle accidents. The eye movement detection

This method proposed a system which follows a low-price approach to control a mouse cursor on a computer system. The eye tracker is based on images recorded by a mutated webcam to acquire the eye movements. The eye movements are then graphed to a computer screen to position a mouse cursor accordingly. The movement of mouse by automatically adjusting the position where of eyesight. Camera is used to capture the image of eye movement.

REFERENCES

[1] B. Rebsamen, C. L. Teo, Q. Zeng, M. Ang. Jr. "Controlling a wheel chair indoors using thought" IEEE Intelligent Systems, 2007, pp. 18-24.

[2] C. A. Chin "Enhanced Hybrid Electromyogram / Eye gaze tracking cursor control system for hands-free computer interaction", Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA, Aug 30-Sept 3, 2006, pp. 2296-2299.

[3] J. Kierkels, J. Riani, J. Bergmans, "Using an Eye tracker for Accurate Eye Movement Artifact Correction", IEEE Transactions on Biomedical Engineering, vol. 54, no. 7, July 2007, pp. 1257-1267

[4] A. E. Kaufman, A. Bandyopadhyay, B. D. Shaviv, "An Eye Tracking Computer User Interface", Research Frontier in Virtual Reality Workshop Proceedings, IEEE Computer Society Press, October 1993, pp. 78-84

[5] MARIN, M. J. H. AL-BATTBOOTTI and N. GOGA, "Drone Control based on Mental Commands and Facial Expressions," 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2020, pp. 1-4, doi: 10.1109/ECAI50035.2020.9223246.