EYE BALL CURSOR MOMENT USING DEEPLEARNING

A Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

THELUKUNTLA SHARMILI	**19981A05B4**
VALLEPU SRINIVAS	**19981A05B5**
VANGAPALLY UMESH	**19981A05B6**

SUPERVISOR
Dr.G.Suryanarayana
Associate Professor

Department of Computer Science Engineering
VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD
An Autonomous Institute, Affiliated to JNTUH

May, 2023

VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An Autonomous Institute, Affiliated to JNTUH

Department of Computer Science Engineering

CERTIFICATE

This is to certify that the project titled **Eye ball cursor moment using deep learning** is carried out by

THELUKUNTLA SHARMILI	**19881A05B4**
VALLEPU SRINIVAS	**19881A05B5**
VANGAPALLY UMESH	**19881A05B6**

In partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering** during the year 2022-23.

Signature of the Supervisor
Dr.G.Suryanarayana
Associate Professor

Signature of the HOD
Dr.K.Ramesh
Professor

Project Viva-voice held on ______________

Internal Examiner

External Examiner

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr.G.Suryanarayana**, Associate Professor and Project Supervisor, Department of Computer Science Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **Dr.Ramesh**, the Head of the Department, Department of Computer Science Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Thelukuntla Sharmili

Vallepu Srinivas

Vangapally Umesh

Abstract

Since its inception, computers have developed to the point that we may use them wireless, remotely, and in a variety of other ways. Nevertheless, these innovations are only for those who are physically fit and without physical disabilities. Other methods are needed to make computers more usable and accessible for those with physical disabilities. We have created a system that uses eyeball movement instead of the traditional manner of pointer movement with a mouse in order to create an environment that is fair to everyone who is capable of it. This will make it easier for those who suffer from physical conditions to use computers, advancing knowledge and technology. In this system, the laptop camera is used to take the pictures, which are then used to recognise the eyeball positions and movements using face indexing, as well as to conduct the actions that a real mouse could carry out using OpenCV. Because there are no pre-installations needed, this project is economical. Those who are paralysed and others with special needs may benefit from this.

Keywords: Haar cascade; CNN; Eye Aspect ratio

Table of Contents

List of Figures

Abbreviations

Abbreviation	Description
VCE	Vardhaman College of Engineering
HCI	Human-Computer Interaction
CNN	Convolutional Neural Network
EAR	Eye Aspect Ratio

CHAPTER 1

Introduction

1.1 Introduction

The significance of Human-Computer Interaction (HCI) in facilitating effective communication between humans and computers has received increased attention in recent years due to the rapid advancement of computer technologies. For those who have

HCI can be particularly important in enabling independent computer operation without help from another person for people with disabilities. One approachable method of using computers in these circumstances is eye ball movement control.

The goal of HCI study is to create an interface between a computer and a human user. It is crucial to create technology that makes effective communication between humans and computers possible. In order to enable communication between people with disabilities and computers and enable them to participate in the Information Society, a substitute technique must be found.

Certain illnesses make it difficult for some people and organisations to use conventional computing equipment. Therefore, it is crucial to create a computer operation technique that is usable by people with disabilities. Traditional computer working hardware may be effectively replaced by the human eye, offering a different form of input.

In order to achieve this, a Human-Computer Interaction (HCI) system that enables users to operate computers with their eyes is being put into place. This innovation will enable people with disabilities who are unable to use their hands to operate computers autonomously, which will be especially helpful for them.

Through the use of OpenCV technology, it is feasible to detect pupil movement in relation to the eye's center, which enables Eye Ball Movement Control. A popular computer vision library called OpenCV provides a broad range of tools for processing images and videos.

OpenCV can be used for Eye Ball Movement Control to record real-time video streams from a webcam or other camera, identify the location of the pupil, and move the cursor on the screen in accordance.

Deep learning algorithms that can forecast the coordinates of the pupil based on an input image of the eye further increase the efficacy of Eye Ball Movement Control. These deep learning models can be taught with the aid of training data that consists of pictures of eyes and the pupil's corresponding coordinates.

Additionally, by having each user focus on a succession of points on the screen and recording the associated coordinates of the pupil, the system can be calibrated for each individual user. The deep learning algorithm can be adjusted using the calibration data, increasing the eye tracker's accuracy.

In summation, Eye Ball Movement Control is a computer accessibility feature that has enormous potential to help people with disabilities. People with disabilities can participate more fully in the Information Society by using Human-Computer Interaction (HCI) systems that incorporate this technology, which can greatly improve their quality of life.

1.2 Objectives of the project

The main objectives of a project for eye ball cursor movement using deep learning would be:

- To create a system that can precisely monitor a user's eye movement and convert it to cursor movement on a computer screen.
- To develop a system that anyone who wishes to use their computer hands-free, including those with disabilities, can use.
- To train a model to recognise patterns in eye movements and correctly predict cursor movements based on those patterns using deep learning techniques, such as convolutional neural networks (CNNs).

- To gather and analyse a sizable dataset of eye-tracking information in order to train the deep learning model and boost its precision.
- To seamlessly and effectively integrate the model with the computer's operating system in order to control the cursor.
- To provide a new level of accessibility and convenience for users with disabilities, allowing them to control their computer using only their eyes.

1.3 Advantages of Proposed System

The Eye Ball Cursor Movement initiative may have the following benefits:

- One of the primary benefits of eye ball cursor movement is that it can significantly increase accessibility for individuals with motor disabilities or impairments who are unable to use conventional mice or trackpads. They may be able to interact with computers and other devices in a more organic and intuitive way thanks to this technology.
- The movement of the cursor with the eye can be much quicker and more precise than when using a mouse or trackpad. This can increase productivity and effectiveness for all users, not just those who are disabled.
- The stress that using conventional input devices for extended periods of time can place on the arms and hands can be lessened by using eye movements to move the cursor.
- A fresh and exciting technology called eye ball cursor movement can make using computers and other devices more interesting and unique.
- It is possible to progress current studies on human-computer interaction and assistive technology by creating and using eye ball cursor movement technology, which will result in future advancements.

CHAPTER 2

Literature Survey

2.1 Introduction

There are several areas where eye tracking is used, including automotive, medical research, fatigue simulation, vehicle simulators, cognitive studies, computer vision, and activity detection. For effective and dependable designs in current technologies, eye tracking has grown in importance in commercial applications. Several approaches have been found following a survey of the literature on eye tracking applications in healthcare.

2.2 Analysis of papers

2.2.1 Cursor Control Using Eye Ball Movement

Khare, Vandana, S. Gopala Krishna, and Sai Kalyan Sanisetty (2019) assert that personal computers have become an essential element of modern life; however, their usage is limited by physical abilities. To address this issue, a Raspberry Pi and OpenCV were utilized in conjunction with an external camera to capture images for facial feature detection. This approach provides a straightforward correlation between eyeball movement and cursor movement; however, it is relatively expensive due to the necessary additional hardware.

2.2.2 Eye-Tracking Metrics in Software Engineering

Z. Sharafi et al. conducted an evaluation of metrics pertaining to eye-tracking in software engineering. The authors consolidated disparate metrics into a unified standard to facilitate the analysis of eye-tracking experiments for researchers. Furthermore, they provided definitions for these metrics and drew on analogous fields to offer suggestions for their application.

2.2.3 Eye tracking based human computer interaction: Applications and their uses

S. Chandra et al. proposed an application based on eye-tracking and its interaction with humans. The research concentrated on the determination of direction, position, and sequence of gaze movement. Three objectives were identified to achieve this goal: providing users with an understanding of the underlying principles of eye-movement technology, offering guidance for the development of such technology, and recognizing the challenges and prospects in constructing Man And Machine Interfacing (MAMI) systems using the principle of eye-tracking. Experimental results revealed a reduction in computational time for gaze input as compared to mouse input.

2.2.4 Implementation of eye-tracking system based on circular Hough transform algorithm

R. G Bozomitu et al. proposed an eye tracking system based on the circular Hough transform algorithm for the purpose of providing benefits to neuromotor disabled patients. The signals were captured using an infrared video camera and a personal computer (PC). This system utilized keyword technology for eye tracking. The experimental results indicated that optimal performance was achieved by balancing computational time with precision of detection in regard to pupil region movements.

2.2.5 Hands-free PC Control

In 2012, Gupta, Akhil, Akash Rathi and Dr Y. Radhika unveiled their revolutionary invention: “Hands-free PC Control”, which enabled users to control their mouse cursor with eye movement. To accomplish this feat, they employed two distinct methods of face detection: Feature-based and Image-based. Feature-based detection involved locating certain features of the face such as its eyes or nose. Additionally, a six segmented rectangular filter was used to scan around a rectangle which was then stored as six separate segments. The Image-based method relied on Support Vector Machines (SVM) to detect the face in streaming video; once the face was detected using SVM and SSR (Scale Space Representation), an integral image was tracked moving forward.

2.3 Existing System

There are few systems which are using Rasberry-pi or ardiuno for the tracking of images. There are few projects which use deeplearning but the algorithms and technique which is used are quite complex and the level of accuracy is low in case of detecting the eye position.Deep learning is used for the tracking of eyes but not for other mouse operations. So for a efficient interacting system we have to use all the mouse operations,which are lacking in the existing systems.

2.4 Disadvantages of existing system

While the Eye Ball Cursor Movement project has potential advantages, there are also some disadvantages in existed projects.Some are: Training: Eye ball cursor movement requires some level of training and practice for users to become proficient, which could be a barrier to adoption.

- The expense of putting eye tracking technology into practise might be significant, which might prevent some consumers and organisations from using it.
- In particular, people with specific eye diseases or disabilities, eye tracking technology may not be accurate in all settings or for all users.
- The use of eye tracking technology presents privacy issues because it needs access to a user's eye movements and has the potential to be used to follow or monitor people without their permission.
- Some users, especially those used to using conventional input devices, may find it challenging to become used to controlling the cursor with eye movements. For some users, this can reduce the technology's appeal.

2.5 Problem Statement

To develop a system that accurately and effectively monitors a user's eye movements and uses this information to control the cursor on a computer or other device is the problem statement for the eye ball cursor moment project using OpenCV and deep learning. The system must be robust enough to work in real-world settings and be able to manage a range of lighting situations, head movements, and user behaviours.

Specific challenges associated with this problem include:

Accurate eye tracking is necessary for the system to monitor the user's eye movements and locate the cursor on the screen. This calls for reliable algorithms that can recognise and follow the user's pupils even in poor lighting or when the user's head is moving.

Latency: The system needs to react rapidly to user eye movements and update the cursor's location on the screen. This calls for low-latency hardware and software components, as well as effective methods for processing and analysing the eye-tracking data.

To ensure precise tracking and cursor movement, the system must be calibrated to the user's unique eye movements. This calls for calibration instruments that are simple to use and a system that can adjust as the user's eye movements change over time.

Integration of deep learning techniques is necessary for the system in order to increase the precision and responsiveness of the OpenCV-based eye tracking software. Both computer vision and machine learning skills are needed for this, as well as a sizable dataset for building and testing the deep learning model.

Accessibility: To support users with varying degrees of mobility or visual impairment, the system must include accessibility features. Assistive technology software and other accessibility tools may be integrated, and there may be options to change the cursor's size, speed of movement, and other settings.

The goal of the eyeball cursor moment project, which uses OpenCV and deep learning, is to develop a system that can efficiently and accurately track a user's eye movements and use that information to control the cursor on a computer or other device. The system should also include accessibility features and be able to change as the user's eye movements do over time.

2.6 Proposed System

The problem of accessibility for those with disabilities is a critical issue that requires innovative solutions. One such solution that we have proposed involves making it simple for those with disabilities to use the computer. To achieve this goal, we used the haar cascade classifier to construct a data set that could be used to train a machine learning model.

The data set was created by taking over 3000 photos of eyes, which were then cropped by the haar cascade classifier and stored in a file. This process helped us to obtain a diverse range of images that could be used to train the model to recognize different eye shapes, sizes, and colors.

Once the data set was prepared, we used it to train the machine learning model. This process involved feeding the data set into the model, which then used it to learn how to recognize different eye images. This process was repeated multiple times, allowing the model to learn and improve over time.

To implement the system, we utilized open CV throughout the project execution. This allowed us to open the system camera, which then captured the user's eye images for every single frame. The captured images were then analyzed by the machine learning model, which determined which image it matched to.

The clicking actions were carried out utilizing the eye aspect ratio, which is a measure of the ratio of the height of the eye to the width of the eye. This ratio was used to determine whether the user was blinking or not, and to trigger the appropriate clicking action.

Overall, our solution provides a simple and effective way for those with disabilities to use the computer. By leveraging the power of machine learning and computer vision, we have created a system that is accurate, reliable, and easy to use. We believe that this solution has the potential to improve the lives of many people and help them to achieve their full potential.

CHAPTER 3

METHODOLOGY

3.1 Haar Cascade Classifier

Monitoring the location and movement of the eyes is called eye tracking. It can be used for a variety of things, such as medical diagnosis, computer vision, and human-computer interface. Cursor control, in which the location of the cursor on a computer screen is controlled by the movement of the eyes, is one frequent application of eye tracking. The Haar Cascade Classifier can be used in this situation to find the eyeballs and monitor their movement.Neural networks are used in the machine learning subfield of deep learning to learn from data. Layers of interconnected nodes, each of which completes a straightforward calculation, make up neural networks. The network can learn to recognise trends and features in the data by being trained on a sizable dataset.We can teach a neural network to recognise the eyes in an image and use that information to use the Haar Cascade Classifier for eye tracking. This can be accomplished by providing the network with a sizable dataset of pictures of eyes along with labels showing where the eyes are in each picture. The network can then acquire the ability to identify eye characteristics and use them to identify eyes in fresh images.

The position of the eyes in an image can be used to move the cursor once they have been identified. You can achieve this by mapping the position of the eyeballs to the cursor's location. The cursor can be moved to a particular spot on the screen, for instance, if the eyes are focused there.Combining deep learning with the Haar Cascade Classifier has a number of benefits. In the beginning, it can be used to find the eyes in a variety of illumination situations and directions. Second, it enables more accurate cursor control by tracking the eyeballs' movement over time. Finally, it can be trained on a large dataset, allowing it to generalize to new images and users.

In conclusion, the movement of the eyes in a picture can be recognised and tracked using deep learning in combination with the Haar Cascade Classifier. This can be applied to move the cursor on a computer screen, giving people a more comfortable and natural interface.

3.1.1 Here are the Five steps for utilizing the Haar Cascade Classifier to find eyes:

- Purchase the Haar Cascade Classifier that has been trained to identify eyes. It is available online or in the OpenCV data subdirectory as an XML file.
- To incorporate the classification into your Python code, use the OpenCV library.
- Find the eyes in the picture you want to use and convert it to grayscale.
- Make use of the classifier's detectMultiScale function to locate the eyes in the image. Every rectangle in the collection this method returns represents a distinct eye.
- A rectangle should be drawn around each eye that was identified in the picture.

3.2 Convolutional Neural Network

Because of the rapid development of computer technologies in recent years, our interaction with computers has changed significantly. It is more important than ever to consider how Human-Computer Interaction (HCI) can make computer systems accessible to everyone, particularly those with disabilities, as a result of the spread of technology. One such technology that was developed for this purpose allows handicapped people to operate computers solely with their eyes is called Eye Ball Movement Control. By incorporating this technology into computer networks, people with disabilities can use computers independently and without assistance. A fascinating area of research that has

the potential to significantly improve the lives of people with impairments is deep learning-based eye ball cursor movement. Deep learning, a branch of machine learning that teaches algorithms to make predictions using a large quantity of data.
Convolutional layers are a key component of the deep learning methods used to shift the eyeball cursor.Convolutional layers are used by deep learning algorithms to identify features in images. These layers work by adding various filters to an input image that are designed to find specific patterns or features in the image.

For example, one might develop a filter to look for curves or edges in an image. By applying various filters to an image, a convolutional layer can discover a wide range of distinct features within it.When an eyeball cursor is moving, convolutional layers can be used to analyse images of a person's eyes and identify the direction of their gaze. To achieve this, a deep learning algorithm is trained using a sizable dataset of eye images and related cursor motions. Following that, the algorithm can identify patterns in the eye images that correlate to specific cursor movements.

Convolutional layers in deep learning show great promise for improving the lives of people with disabilities in general when used for eye ball cursor movement. By carefully examining complex eye images, these algorithms accurately identify a person's gaze orientation and translate it into cursor movements. Although there are some drawbacks to this approach, further research and development will undoubtedly lead to improvements in the accuracy and efficacy of these algorithms.

3.2.1 General procedure for implementing CNN in Eye Ball Cursor Movement using Deep Learning:

1.*DataCollection*: Collecting data is the first stage in creating a CNN model for eye tracking and cursor movement. Using eye-tracking hardware and software, this entails capturing eye and cursor motions. To train the CNN model, the data must contain both positive and negative instances of eye and cursor movements.

2.*DataPreprocessing* : To make sure the gathered data is pure and appropriate for training the CNN model, preprocessing is required. In order to do this, the incoming data must be standardised and any noise or outliers removed.

3.*SplittingDataintoTrainingandTestingSets* :Machine learning frequently divides data into training and testing groups. A part of the data will be used to train a model, and the remaining data will be used to evaluate the model's performance. This makes it easier to assess how well the algorithm generalises to fresh, untested data.

4.*BuildingtheCNNModel* : The CNN model must then be constructed. This entails defining the size and kind of the network's layers, such as the convolutional, pooling, and completely connected layers. To learn and recognise characteristics that are important for eye tracking and cursor movement, the CNN model should be created.

5.*TrainingtheCNNModel* :The training set is used to build the CNN model. To reduce the difference between the predicted and real output, the model iteratively modifies the weights of the network.

6.*TestingtheCNNModel* : Once the CNN model is trained, it needs to be tested using the testing set. This step evaluates the performance of the model in terms of accuracy, precision, recall, and F1 score.

7.*IntegrationwithEyeTrackingandCursorMovementSystem* : Once the CNN model is trained, it can be integrated with the eye tracking and cursor movement system to provide accurate and reliable cursor movement based on eye movements

.

3.3 STREAMLIT

Streamlit is a well-liked open-source platform for developing and disseminating data science applications. It offers a user-friendly interface that enables developers to quickly create interactive web apps without the need for specialised web development expertise. Streamlit is well suited for applications that use deep learning, such as eyeball cursor movement.One benefit of using Streamlit in eye ball cursor movement is that it makes it simple for programmers to create user interfaces that are specifically tailored to this use case. Programmers can create user interfaces that make it simple for users to calibrate their eye tracking devices, select their preferred method of cursor movement, or adjust a number of parameters that have an impact on the algorithm's performance, for example. This enhances the application's usability generally and increases accessibility for users with disabilities.

The real-time feedback and visualisation elements needed for deep learning apps are also provided by Streamlit. Developers can use Streamlit to view the algorithm's output in real-time in order to comprehend how the algorithm is operating and to identify areas for development. Additionally, dynamic visualisations that allow users to explore the data in novel and helpful ways can be created using Streamlit.When using Streamlit for eye ball cursor movement, sharing and deploying the application are made easier. Streamlit provides a variety of deployment options, including containerization and cloud-based services, making it simple to distribute the programme to others. This is crucial for applications that involve people with disabilities because it enables developers to distribute the program swiftly and easily to those who desire it.

3.4 EAR

Research on eye tracking and eye movement has grown in significance in the field of computer vision, particularly for systems that support human-computer interaction. Measuring the eye aspect ratio (EAR) to determine whether the eyes are open or closed is a common method in eye tracking. The location of the eye's landmarks—typically the top and bottom, as well as the inner

The following image depicts the simple Streamlit interface which we used for our project representation

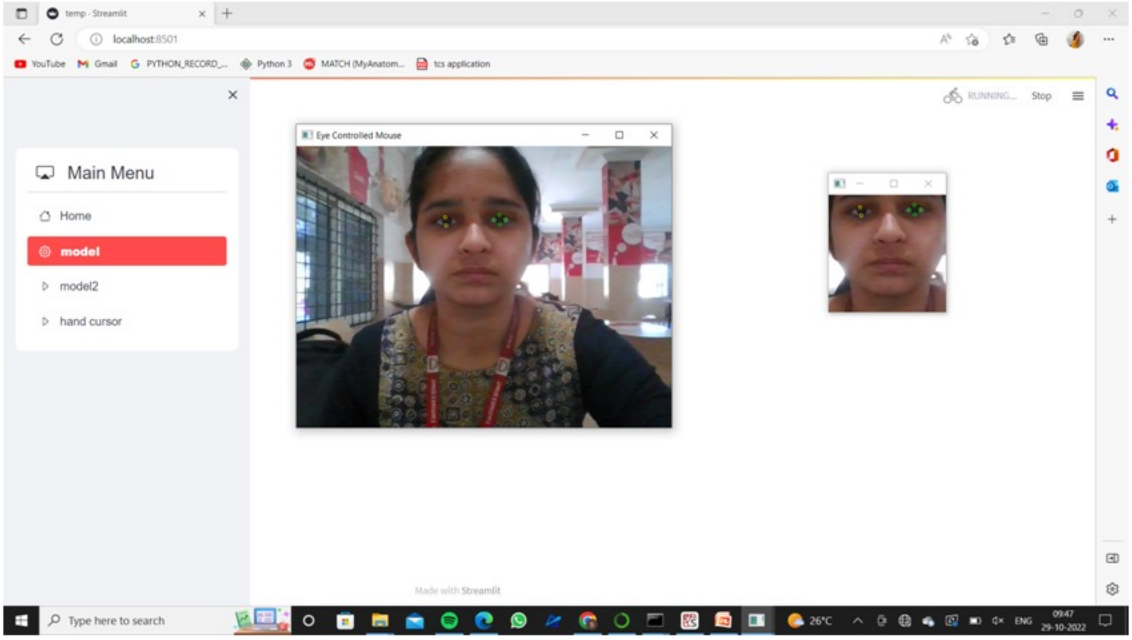


Figure 3.1: SteamLit Interface

and outer corners—determines the EAR. EAR is used in eye tracking and cursor movement devices to identify user blinks and eye closures. To prevent unintentional clicks or motions, this is done. Cursor movement is halted or blocked when the EAR measurement drops below a predefined threshold value because this suggests that the user is sleeping. The system, on the other hand, assumes that the user's eyes are open and permits cursor movement to continue when the EAR measurement is higher than the threshold value.

The number of consecutive frames in which the EAR measurement drops below the threshold is set to a threshold value of 0.2 in order to distinguish between a typical wink of the eye and a wink used to perform clicking actions. This threshold has been fixed to three consecutive frames in our software. This guarantees that the system can distinguish between a normal eye blink or closure and a user's intentional click on an object on the screen.

In general, the EAR measurement offers an accurate and dependable method of determining the user's eyes' status in eye tracking and cursor movement systems. The system can precisely detect when a user's eyes are open or closed by setting a threshold value and using it in the software, resulting in smooth and precise cursor movement.

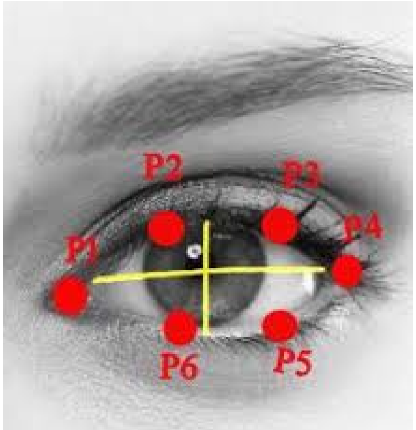


Figure 3.2: Open Eye

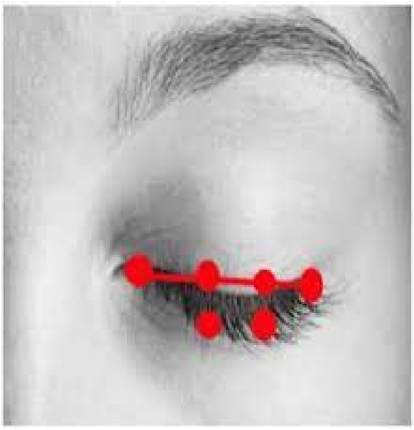


Figure 3.3: Closed Eye

3.4.1 Calculation of EAR

Eye tracking research uses the EAR (Eye Aspect Ratio) value as a metric to pinpoint where and how the subjects' eyes are moving. It is determined using the coordinates of specific locations on the eyes, such as the top and bottom of the eyelids and the inner and outer corners of the eyes.

In the case of eye ball cursor moment using deep learning, EAR can be used to calculate the position of the eyes in relation to the cursor on the screen. To calculate the EAR value, the following steps can be taken:

- Use a Haar Cascade Classifier or another appropriate technique to find the eyeballs.
- Find the locations of the top and bottom of the eyelids, as well as the inner and exterior corners of each eye.
- Determine the average distance between each eye's inner and outer corners

as well as their respective distances from one another. This will be the eye's width.

- Determine the average distance between each eyelid's top and bottom as well as their respective distances from one another. The eye's height will be at this point.
- Calculate the EAR value using the formula: EAR = (height of eye) / (2 x width of eye)

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Figure 3.4: EAR Formula

This value will be between 0 and 1, with a higher value indicating that the eyes are more open.

After calculating the EAR value, it can be used to establish how far away the eyes are from the cursor on the screen. This can be done by detecting whether or not the eyes are looking at the cursor by comparing the EAR value with a threshold value.

3.5 Face Land Marks

Face and landmark detection with OpenCV, in the context of the eye ball cursor moment, is a key aspect of the system. While landmark recognition is used to recognise important facial features like the eyes, nose, and mouth, face detection aims to find and isolate the face in an image or video stream.

The method of eye tracking usually begins with the detection of faces. There are a number of pre-trained face recognition models in OpenCV, including deep learning-based models and the Haar cascade classifier. These models analyse the colour and texture patterns in a picture and pinpoint areas that are most likely to contain faces using machine learning algorithms. Once a face is detected, it is isolated from the rest of the image, making it easier to track the user's eye movements.

The subsequent step in the procedure is landmark detection. The Facial Landmark Detection model and the Eye Landmark Detection model are two of the pre-trained landmark detection models available in OpenCV. In order to monitor the location and movement of the eyes in real-time, these models use machine learning algorithms to recognise specific points on the face or eyes.

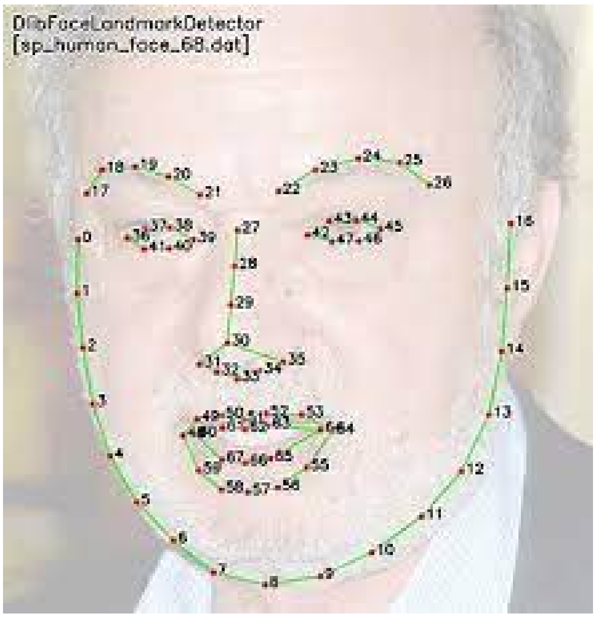


Figure 3.5: Face land marks 68

Calculating the position and movement of the eyes can be done using the landmark spots that the models were able to identify. The location of the cursor on the screen can then be controlled by the algorithm using this information. The system can precisely monitor the user's gaze and enable them to move the cursor by shifting their eyes by continuously analysing the user's eye position.

The eye ball cursor moment method depends on the detection of faces and landmarks. These algorithms give the system the ability to precisely and effectively monitor the user's eye movements, allowing them to control the cursor with their gaze. Overall, these methods contribute to making computer and other gadget interaction feel more organic and intuitive.

CHAPTER 4

Implementation

In order to include eye tracking in a project, a mix of hardware and software elements is needed. The hardware consists of a camera or sensor for recording eye movements and a mouse or touchpad for directing cursor movements. Libraries for image processing, machine learning, and human-computer interaction are among the software's components. These elements can be used to create an eye tracking system with a range of applications.

4.1 Dataset creation

With its unmatched accuracy in analysing user behaviour and enhancing user interfaces, eye tracking has become a crucial technique in the field of human-computer interaction. The creation of an eye tracking system utilising deep learning, which enables the precise tracking of eye movements and the prediction of the cursor's position on the screen depending on the user's gaze, is one of the most important uses of eye tracking.

To accomplish this, we first had to develop a unique dataset that precisely recorded eye movements while gazing at particular pixels on the screen. It was a difficult effort that required careful consideration of a number of criteria to create such a dataset.

We used the assistance of the Haar cascade classifier, an object detection technique that uses positive and negative values based on features including shadowing, edges, and rectangles, to construct our unique dataset. To support this technique, we downloaded an open source XML file from the OpenCV GitHub link. The borders of the eyeballs, which were crucial for the process of creating our dataset, may be easily detected using the Haar cascade classifier, which is frequently used for object detection.

In order to create the dataset, we had to use OpenCV to take pictures of our faces and create an event listener that would take pictures of our eyes whenever we clicked on a certain pixel on the screen. This made it possible for us to record the precise eye position at a given pixel, which was crucial for teaching our deep learning model to precisely track eye movements. The photos were then concatenated and stored with the file name ”X-pixel Y-pixel-buttonPressed” after being cropped using the Haar cascade classifier to only include the boundaries around the two eyes. It was difficult to get

The below is the flow chart of how we generated a dataset.

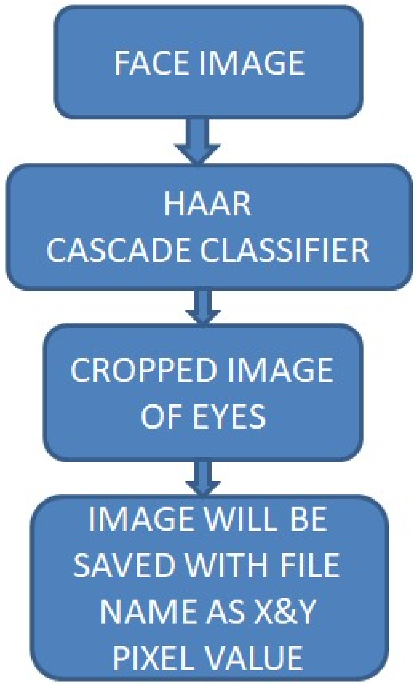


Figure 4.1: Flow Of Dataset Creation

high-quality photos for our dataset since we had to make sure that the eyes were accurately recognised and photographed while taking illumination and the absence of eyewear into account. To guarantee that our deep learning model had enough data to precisely forecast the precise location where the user was looking on the screen, we finally generated roughly 10,000 photos, each fixed at 32 pixels.

In order to train our deep learning model to precisely capture eye movements and estimate the location of the pointer on the screen based on the user’s gaze, our custom dataset will be crucial. The user interface could become more intuitive and natural thanks to this technology, which has the potential to completely change how we interact with computers.

In addition to the technological difficulties, the aim to produce a more organic and user-friendly interface that would enhance the user experience drove the development of an eye tracking system employing deep learning. The user must take their eyes off the screen and concentrate on the input device when using a mouse or keyboard, which disrupts work flow and adds cognitive stress. We can reduce this cognitive load and produce a more fluid and effective interface by letting the user move the cursor with their eyes.

Moreover, eye tracking has potential uses in a variety of industries, including gaming, healthcare, and market research. Eye tracking may be used in both healthcare and gaming to develop more intuitive and immersive gameplay experiences. In healthcare, it can be used to detect and track illnesses like autism and Parkinson's disease. Eye tracking can be utilised in marketing research to comprehend consumer behaviour and preferences.

In conclusion, establishing an eye tracking system using deep learning requires the development of a custom dataset using the Haar cascade classifier and OpenCV. Although creating the dataset was difficult, it will give us the information we need to effectively train our deep learning model. Eye tracking is an intriguing topic for research and development since it has the potential to completely change how we interact with computers and has many uses in many different industries.

4.2 Model Training

Training neural networks, which are artificial models loosely inspired by the structure and operation of the human brain, is a key component in the machine learning subfield of deep learning. A deep learning model must be trained using a number of steps. The data that will be used to train the model must first be gathered and prepared. To do this, the data must be cleaned, transformed as appropriate, and divided into training, validation, and testing sets. The next step is to create an appropriate neural network architecture that can successfully learn from the input data. To do this, the right number and kind of layers must be chosen, together with activation functions and the model's hyperparameters.Using an optimizer that modifies the model's parameters to minimise a loss function, the model can be trained after it has been created. Backpropagation, a method that determines the gradients of the loss function with respect to the model's parameters, is often used to accomplish this. When the model's performance on the validation set reaches a plateau, the training procedure is repeated for a predetermined number of epochs. After the model has been trained, it may be tested on real data to see how well it performs. The model can then be used in a real-world setting by being deployed in a production environment.

Training a sequential model with 2 CNN layers involves several steps.

- Data Preparation: The preparation of the data is the first phase. This entails preprocessing the data as well as importing the dataset and dividing it into training, validation, and test sets. Normalizing pixel values, scaling the photos, and using data augmentation methods to expand the training set are all examples of preprocessing.

- Model Design: The sequential model architecture design comes next. A sequential model is a stack of linear layers where each layer's output serves as the next layer's input. Using a set of trainable filters, the CNN layer in the model's first layer extracts features from the input image. The features collected by the first layer are further refined by the second layer, which is likewise a CNN layer. In order to transfer the features to a collection of class scores, the output of the second CNN layer is flattened and passed through a fully connected layer. Lastly, the class scores are transformed into probabilities using a softmax layer.

- Compilation: The model is then assembled after being created. The optimizer, loss function, and performance measures to be used during training must all be specified. While the performance metrics are used to assess the model's performance during training, the optimizer modifies the model's weights in order to minimise the loss function.
- Training: By feeding training data into the network and modifying network weights to reduce the loss function, the model is trained. The model is trained for a specific number of epochs, and the performance on the validation set is monitored to avoid overfitting, in an iterative process.
- Evaluation: After the model has been trained, its performance on untested data is assessed on the test set. This gives a prediction of the model's performance under realistic conditions.

Overall, there are a number of phases involved in training a sequential model with two CNN layers, including data preparation, model design and compilation, training, and evaluation. These procedures can be used to create a powerful deep learning model for picture categorization problems. By lowering the spatial extent of the input, the max pooling layer helps to reduce the number of parameters in the network, making it simpler to train. The output of the max pooling layer is transformed by the flatten layer into a 1D vector that may be fed into a dense layer. The thick layers are completely connected layers that alter the input linearly before activating it nonlinearly.The number of neurons in the first dense layer is often high, whereas the number of neurons in the second dense layer is equal to the number of output classes.

Overall, there are multiple processes involved in the training process for a neural network model with a max pooling layer, a flatten layer, and two dense layers, starting with the preparation of the data, followed by the design and compilation of the model, training, and evaluation. These procedures can be used to create a powerful deep learning model for categorization problems.

The following image represents the model summary of the deep learning model which we used for our project

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 5, 21, 32)         896

 conv2d_1 (Conv2D)           (None, 2, 10, 64)         8256

 conv2d_2 (Conv2D)           (None, 2, 10, 32)         51232

 max_pooling2d (MaxPooling2D  (None, 2, 10, 32)        0
 )

 flatten (Flatten)           (None, 640)               0

 dense (Dense)               (None, 32)                20512

 dense_1 (Dense)             (None, 2)                 66

=================================================================
Total params: 80,962
Trainable params: 80,962
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.2: Model Training

4.3 Eye Tracking

A crucial part of eye ball cursor moment utilising OpenCV is eye tracking. It enables the system to precisely monitor the user's eyes in real-time, which can be used to manipulate where the cursor appears on the screen.

Using OpenCV, the eye tracking method typically includes a number of crucial phases. The system must first recognise and separate the user's eyes from the surrounding area of the image. The earlier mentioned face and landmark detection methods are used for this. The algorithm may then study the position and motion of the eyeballs over time after they have been isolated.

In order for the algorithm to function, certain spots on the eye, like the pupil or the corner, must be recognised and their positions must be tracked while the eye moves. For eye tracking, OpenCV offers a number of pre-trained models, including the Pupil Detection and Tracking model and the Eye Gaze Estimation model. These models anticipate the direction of the gaze by analysing the position and movement of the eyes using machine learning algorithms.

The programme can then utilise this knowledge to regulate where the pointer appears on the screen once the gaze direction has been established. To control the cursor's position in real-time, this usually entails mapping the gaze direction to a specific spot on the screen.

In order to use OpenCV for eye tracking, there are various obstacles to overcome. Dealing with changes in lighting and image quality is one of the biggest issues. Eye tracking techniques depend on precise detection of important features like the pupil and corner of the eye. It may be challenging to precisely discern these features in low-quality images or bad illumination, which could lead to erroneous eye tracking.

Managing changes in head and eye movement presents another difficulty. Even when the user is moving their head or their eyes are moving quickly, the algorithms utilised for eye tracking must be able to precisely track the position and movement of the eyes.This calls for highly developed algorithms and machine learning models that can adjust to various circumstances and deliver precise tracking in real-time.

Notwithstanding these difficulties, OpenCV eye tracking is a powerful method with numerous uses in the area of human-computer interaction. Eye ball cursor moment can make using computers and other gadgets feel more natural and intuitive by enabling users to move the cursor with their eyes.

4.4 Blink Detection

Blink detection is a crucial part of deep learning-based eye ball cursor movement in addition to eye gaze monitoring. Blinking can be a sign that a user is switching between tasks or is getting tired. It can also be used to trigger certain system actions, like pausing a film or dismissing a notice.

We can employ a strategy similar to that employed for tracking eye gaze to find blinks. Using a Haar cascade classifier or another object detection algorithm, we may first identify the eyes. Next, we can classify each frame as either having "eye open" or "eye closed" using a deep learning model.Convolutional neural networks (CNN) or other deep learning models that have been trained on a sizable dataset of pictures of open and closed eyes can be used to accomplish this.

Calculating the eye aspect ratio (EAR), which is a ratio of the distance between specific landmarks on the eye to the length of the eye, is one frequent technique for blink detection. The EAR value will fluctuate between high and low depending on whether the eye is open or closed. We can identify blinks when the EAR value drops below the threshold for a predetermined period of time by defining an EAR value threshold.

Using optical flow analysis to monitor changes in the appearance of the eye over time is another technique for blink detection. Computer vision can recognise the abrupt shift in the optical flow pattern caused by the eye blinking.

The eye gaze tracking technique can be modified or, as was already indicated, various system actions can be initiated once blinks have been identified. Overall, blink detection is a crucial part of employing deep learning to move the eyeball as a cursor and can improve the system's functionality and accuracy.

4.5 Interface

The user interface, which gives the user a method to engage with the system and control the cursor location, is a crucial part of eye ball cursor moment technology. A user interface that is simple, easy to use, and responsive to the user's eye movements can be made using OpenCV.

Using a graphical user interface (GUI) toolkit like Qt or wxWidgets is one method for developing a user interface using eye ball cursor moment technology. These toolkits offer a large selection of widgets and controls, such as buttons, sliders, and text input fields, that can be used to design a unique user interface.

Use of a web-based interface, which can be accessible from any device with a web browser, is an alternative strategy. By employing eye movements to control the cursor position, this method eliminates the need for any additional hardware or software.

Regardless of the strategy, the user interface needs to be simple to use and comprehend, with a focus on usability. Also, the interface needs to move the cursor smoothly and precisely in response to the user's eye movements.

It's necessary to take into account the user feedback in addition to the user interface itself. The system might, for instance, offer visual feedback, such as a cursor that modifies its colour or shape to denote various operating modes.The system can also give audible confirmation that the user has finished a task by making a sound or voice prompt.

Ultimately, the user interface should be created with the user's needs and preferences in mind as it is a crucial part of eye ball cursor moment technology. The user interface may be tailored to each user's specific requirements using OpenCV and other software tools, offering a more intuitive and natural approach to manipulate the cursor position. Used libraries Applications for eye tracking employ a variety of libraries. OpenCV and PyAutoGUI are a couple of the well-known ones. whereas PyAutoGUI is employed to manage eyeball moments. For numerical computation and data manipulation, use NumPy. By offering pre-built functions and algorithms, these libraries make the development of eye tracking applications easier and more efficient.

4.5.1 OpenCV

A well-known computer vision library called OpenCV provides a wide range of tools for processing images and videos. Eye tracking is one of its possible uses, as it may be used to direct the movement of a cursor on a computer screen. A branch of machine learning called deep learning has demonstrated promise in a number of image and video analysis applications, including eye tracking.

The first step in implementing deep learning for eye tracking in OpenCV is to gather training data. Images of eyes and the accompanying pupil coordinates should be included of the training data. For this, one can either gather their own data or use a dataset like the Eye Tracking Glasses Dataset.

The following stage is to build a deep learning model that, given an input image of the eye, can forecast the pupil's coordinates. With deep learning frameworks like TensorFlow or PyTorch, one can either employ a pre-trained model, such as a convolutional neural network (CNN), or build their own model. The anticipated coordinates of the pupil must be outputted by the deep learning model from an input of an image of an eye.

Once trained, the deep learning model may be used to track the location of the student in real-time video streams from any camera, including webcams. To record video streams, identify faces and eyes, and draw a cursor on the screen, OpenCV provides a number of functions.

It's crucial to calibrate the device for each user in order to increase the eye tracker's accuracy. In order to do this, the user must gaze at a number of different spots on the screen, and the associated coordinates of the pupil must be recorded. This calibration data can be used to adjust the deep learning model.

In conclusion, integrating deep learning for eye-tracking in OpenCV necessitates expertise in computer vision, machine learning, and programming. But when used properly, it may be a potent tool for designing user interfaces for computers and other devices that are more logical and intuitive.

4.5.2 dlib

Dlib is a C++ library that offers resources and formulas for a range of machine learning and computer vision tasks, such as face recognition, object identification, and facial landmark detection. Eye-tracking, which can be used to control the movement of a cursor on a computer screen, is one of the potential uses for dlib.

The first step in implementing eye-tracking with deep learning in dlib is to gather training data. Images of eyeballs with associated pupil coordinates should be included of the training data. For this, one can either gather their own data or use a dataset like the Eye Tracking Glasses Dataset.The following stage is to build a deep learning model that, given an input image of the eye, can forecast the pupil's coordinates. Popular deep learning frameworks like TensorFlow or PyTorch can be used for this. After trained, the deep learning model may be utilised with dlib to identify the pupil's position in real-time video streams from any camera.

The Active Appearance Models (AAM) technique, which may be used to identify facial landmarks like the corners of the eyes and the centre of the pupil, is implemented in Dlib. Dlib can be used to adjust the screen cursor once the pupil's location has been determined.

It's crucial to calibrate the device for each user in order to increase the eye tracker's accuracy. In order to do this, the user must gaze at a number of different spots on the screen, and the associated coordinates of the pupil must be recorded. This calibration data can be used to adjust the deep learning model.

4.5.3 Numpy

An array of mathematical operations and functions that can be carried out on multi-dimensional arrays are supported by the NumPy Python library. NumPy can be used to store and handle the data produced by the eye-tracking system as well as to carry out the mathematical operations necessary for deep learning training and inference in the context of eye-tracking cursor movement.The image data produced by the eye-tracking system is one important use of NumPy in eye-tracking cursor movement. The majority of the time, this information is a collection of still photos or video frames that document the user's eye movements over time. This data may be stored in a multi-dimensional array format using NumPy, making it simple to handle and analyse using the several functions and operations offered by the library.

NumPy may be used to execute the mathematical operations necessary for deep learning inference and training in addition to storing and processing picture data. To train deep neural networks, for instance, NumPy can be used to execute matrix multiplication operations. It can also be used to calculate loss functions and optimise model parameters using gradient descent methods.

Moreover, NumPy offers a variety of other operations and functions that can be helpful for deep learning-based eye tracking of cursor movement. To conduct element-wise actions on arrays of various sizes and forms, NumPy, for instance, supports array broadcasting.When working with multi-dimensional image data, this can be especially helpful since it enables operations to be carried out throughout the array's many dimensions without the need for explicit looping.

Overall, NumPy is a strong and adaptable library that may be utilised in a variety of tasks, such as deep learning-based eye tracking of cursor movement. NumPy makes it possible to store, process, and analyse the massive amounts of data produced by eye-tracking systems as well as to carry out the intricate calculations necessary for deep learning inference and training. NumPy does this by supporting multi-dimensional arrays, mathematical operations, and other functions and operations.

4.5.4 PyAutoGUI

PyAutoGUI is primarily made to allow programmatic control of a computer's mouse and keyboard. Nevertheless, eye-tracking technology, which makes use of deep learning techniques to precisely identify human eye movement and transform it into cursor movement, can also be utilised to move the mouse pointer using PyAutoGUI.

Although eye-tracking technology has been around for a while, current developments in deep learning have made it possible to develop systems that are more precise and dependable. To control the mouse pointer on the screen, these systems often use deep neural networks to detect the position and movement of the user's eyes in real-time.

You would normally need to have an eye-tracking system in place that can accurately identify the user's eye movement in order to use PyAutoGUI for eye-tracking cursor movement. This could be a discrete eye-tracking gadget or a software-based system that monitors the user's eye movement using a webcam or other camera.

After installing an eye-tracking device, you will need to create a deep learning model that can properly identify user eye movements and convert them into cursor movements.In order to learn the complex patterns and relationships between the user's eye movements and the movement of the cursor on the screen, this typically entails training a deep neural network on a large dataset of eye movement data using techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

Once a deep learning model has been developed, you can incorporate it into your PyAutoGUI code to use the user's eye movement to control the mouse pointer. Typically, to do this, real-time eye movement data from the user would be recorded, processed using a deep learning model to determine the best cursor movement, and then the pointer would be moved using PyAutoGUI.

Overall, using PyAutoGUI for deep learning-based eye-tracking cursor movement can be a difficult and complex process. But, with the correct equipment and methods, it is possible to develop extremely precise and trustworthy eye-tracking systems that may be utilised for a variety of purposes, from gaming and entertainment to accessibility and assistive technology.

CHAPTER 5

RESULTS

5.1 Related Work

A thorough literature review has been conducted in the field of healthcare applications pertaining to eye tracking systems; some of these methods are detailed below. Firstly, Khare, Vandana, S. Gopala Krishna, and Sai Kalyan Sanisetty (2019) assert that personal computers have become an essential element of modern life; however, their usage is limited by physical abilities. To address this issue, a Raspberry Pi and OpenCV were utilized in conjunction with an external camera to capture images for facial feature detection. This approach provides a straightforward correlation between eyeball movement and cursor movement; however, it is relatively expensive due to the necessary additional hardware. [1] Z. Sharafi et al. [2] conducted an evaluation of metrics pertaining to eye-tracking in software engineering. The authors consolidated disparate metrics into a unified standard to facilitate the analysis of eye-tracking experiments for researchers. S. Chandra et al. [3] proposed an application based on eye-tracking and its interaction with humans. The research concentrated on the determination of direction, position, and sequence of gaze movement. R. G Bozomitu et al.[4] proposed an eye tracking system based on the circular Hough transform algorithm for the purpose of providing benefits to neuromotor disabled patients. RamshaFatima and AtiyaUsmani [5] proposed a method related to Eye Movement-based Human Computer Interaction in IEEE 2016. This paper aimed to outline and execute a Human Computer Interface framework that tracks the direction of the human eye using EAR ratio for cursor movement implementation. In their 2016 paper, Venugopal and D'souza [6] proposed a hardware-based system utilizing an Arduino Uno microcontroller and a Zigbee wireless device. First, real-time images were captured and processed via Viola Jones Algorithm to detect faces, then Hough Transform was used to detect pupil movement, which was used as input to the hardware containing the Arduino Uno microcontroller and Zigbee device for data transmission. O. Mazhar et. al [7] developed an ingenious eye-tracking system, designed to provide a revolutionary solution for those with neuromotor disabilities. M. Kim et. al [8]proposed a radical system for detecting and tracking eye movement based on cardinal direction. In the experiment, participants were instructed to look towards the north in a specific cardinal direction and their reaction times were then observed. In 2012, Gupta, Akhil, Akash Rathi and Dr Y. Radhika unveiled their revolutionary invention: “Hands-free PC Control”, [9] which enabled users to control their mouse cursor with eye movement. The Image-based method relied on Support Vector Machines (SVM) to detect the face in streaming video; once the face was detected using SVM and SSR (Scale Space Representation), an integral image was tracked moving forward. K. Kuzhals et al. [10] unveiled a groundbreaking eye tracking system, which promises to revolutionize the application of Personal Visual Analytics (VSA). This research paper explores the challenges of VSA in real-time scenarios and identifies potential areas for further study.

5.2 Flow Diagram

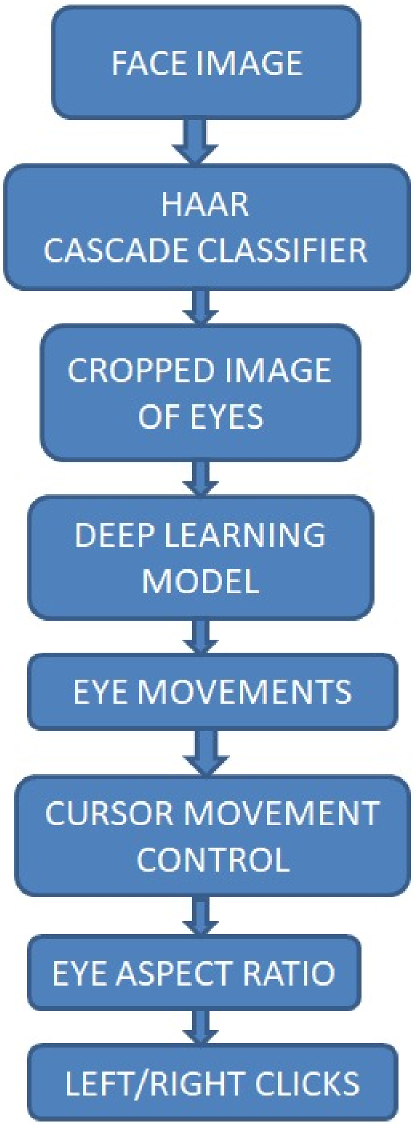


Figure 5.1: Flow chart

The Haar Cascade Classifier is a machine learning-based method for finding objects in pictures or movies. To identify the presence of a particular object, such as eyes, faces, or cars, it employs a series of classifiers. So, in our study, this classifier was utilized to identify both the right and left eyes. We must first train the classifier on a sizable dataset of both positive and negative images to detect eyes using the Haar Cascade Classifier. Negative images lack eyes while positive images do. The program then picks up on the patterns that set the two classes apart .Here are the Five steps for utilizing the Haar

Cascade Classifier to find eyes:

- Get the pre-trained Haar Cascade Classifier for eye detection. It is accessible online or in the data folder of OpenCV. which is an XML file.
- Use the OpenCV library to load the classifier into your Python code.
- Read the image you wish to use to find the eyes and make it grayscale.
- To identify the eyes in the image, use the classifier's detectMultiScale function. Each rectangle in the list that this method returns is an identified eye.
- Each eye that was found in the image should have a rectangle drawn around it.

There are several types of deep learning architectures that are commonly used today. One of the most popular is the convolutional neural network (CNN), which is used for image and video recognition tasks. A CNN consists of .several layers of convolutional and pooling operations, which enable it to learn spatial features in the input image.

A threshold value of 0.2 is established in our software to employ EAR in eye tracking and cursor movement to identify whether the user's eyes are open or closed. The system assumes that the eyes are closed if the EAR is below the threshold value and stops or inhibits cursor movement. The system assumes that the eyes are open and permits cursor movement if the EAR is higher than the threshold value. To distinguish between a typical wink of the eye and a wink of the eye used to accomplish clicking actions, we set the threshold at three consecutive frames.

5.3 Results

The following is the image in which the cursor on the pc is controlled with the movement of the eyeball of the user

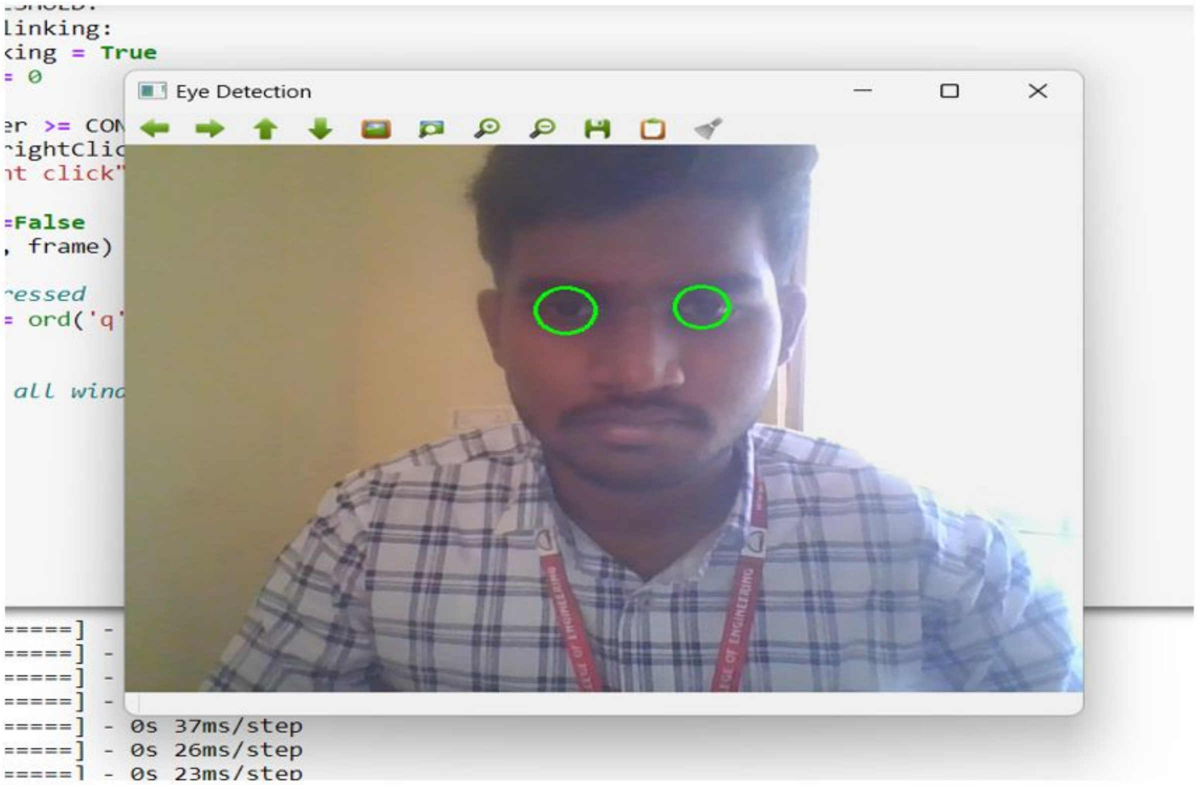


Figure 5.2: Eye Tracking

The below picture depicts the right and left click operations which are performed based on the eye winks.

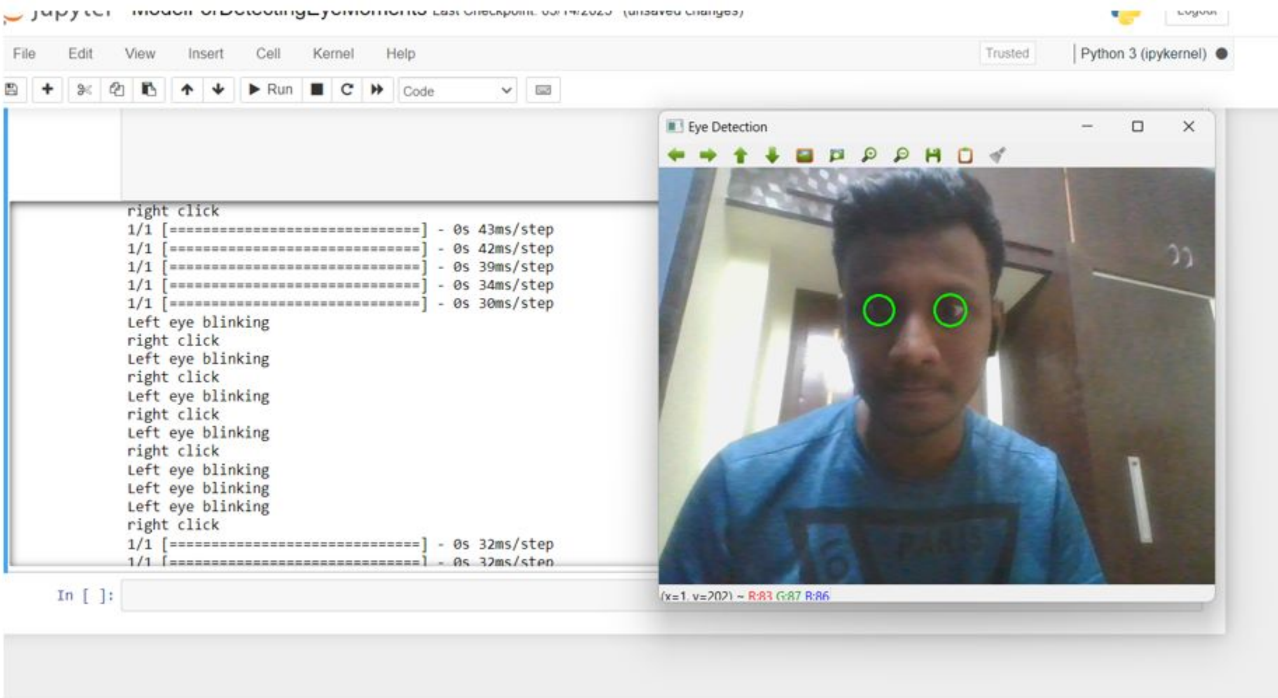


Figure 5.3: Cursor Operations

CHAPTER 6

Conclusions and Future Scope

6.1 Conclusion

Our goal is to use our eyes to control a computer. We want to put into practise a few extra mouse-based actions. Additional actions besides clicking include text selection, scrolling, and typing. The letters that are clicked on the virtual keyboard will be shown on the notepad or other blank screen when used for typing. We are eager to extend cursor control to software used for remote control of televisions or other devices. For this, we need a piece of hardware that can easily complete the task and handle everything at once. A device or specification with a built-in camera and the entire process going can make daily life simpler and more fascinating.

Using OpenCV and deep learning, eye ball cursor moment is a strong technology that has a wide range of possible applications in the area of human-computer interaction. Eye ball cursor This can make using technology more natural and intuitive by enabling users to move the cursor with their gaze. This can increase productivity for all users as well as improve accessibility for people with disabilities.

A major benefit of the eyeball cursor moment is that it can offer a more intuitive and natural way to interact with technology. Users must move their hands and fingers in a certain way to engage with traditional input devices like mice and touchscreens. Contrarily, eye ball cursor moment enables users to move the cursor with their sight, which is a more intuitive and natural method to engage with technology.

Moreover, eyeball cursor moments can increase accessibility for those with disabilities. Eye ball cursor moment can offer an alternate method of communicating with technology for people who have restricted movement or are unable to use conventional input devices. This can apply to those who have spinal cord injuries, muscular dystrophy, or cerebral palsy, among other diseases.

To properly track the location and movement of the user's eyes in real-time, the system combines face and landmark identification methods, deep learning algorithms, and sophisticated eye tracking algorithms. Although there are a number of difficulties with utilising OpenCV for eye tracking, including changes in illumination and image quality as well as handling head and eye movement, these may be overcome with the appropriate hardware and software.

In general, eye ball cursor moment with OpenCV and deep learning has several benefits, including as increased productivity, improved accessibility, and a more natural and intuitive way of engaging with technology. We can anticipate seeing the technology grow increasingly pervasive and integrated into a variety of hardware and software, from smartphones and tablets to virtual reality headsets and smart glasses.

6.2 Future Scope

With OpenCV and deep learning, the potential for eye ball cursor moments is enormous and fascinating. Eye tracking will become increasingly prevalent in our daily lives as technology develops, opening up new applications and creative potential.

Virtual and augmented reality are two areas where eye ball cursor moment might have a huge impact. Eye tracking might offer a more organic and intuitive method to interact with the virtual environment as virtual and augmented reality technology develops. This might entail directing the virtual environment's cursor's location, choosing things with the gaze, or even directing the movement of the virtual avatar. Healthcare is another industry where eye ball cursor moment might be used. Eye tracking might make it easier for people with disabilities to control their surroundings, such as the lighting, temperature, or even their wheelchair. For diagnostic applications, such as seeing early indications of neurological diseases like Parkinson's disease, it might also be used to track eye movements.

Another area where eye ball cursor moment may have a big impact is education. Eye tracking has the potential to improve student learning by delivering a more interactive and engaging learning experience. For instance, students may utilise eye tracking to interact more naturally and intuitively with digital content, or teachers could use it to track students' gaze patterns and modify their teachings accordingly.

We may anticipate seeing eye tracking technology used more frequently and integrated into a variety of gadgets and applications as it continues to develop and get better. This will open up new avenues for innovation and development and improve everyone's access to and understanding of technology.

[1] Khare, Vandana, S. Gopala Krishna, and Sai Kalyan Sanisetty. "Cursor Control Using Eye Ball Movement." In 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), vol. 1, pp. 232-235. IEEE, 2019.

[2] Sharafi, Zohreh, Timothy Shaffer, and Bonita Sharif. "Eye-Tracking Metrics in Software Engineering." In 2015 AsiaPacific Software Engineering Conference (APSEC), pp. 96-103. IEEE, 2015.

[3] Chandra, S., Sharma, G., Malhotra, S., Jha, D. and Mittal, A.P., 2015, December. Eye tracking based human computer interaction: Applications and their uses. In 2015 International Conference on Man and Machine Interfacing (MAMI) (pp. 1-5). IEEE.

[4] Bozomitu, R.G., Păsărică, A., Cehan, V., Lupu, R.G., Rotariu, C. and Coca, E., 2015, November. Implementation of eye-tracking system based on circular Hough transform algorithm. In E-Health and Bioengineering Conference (EHB), 2015 (pp. 1-4). IEEE.

[5] RamshaFatima,AtiyaUsmani(2016) Eye movement based human computer interaction, IEEE.

[6] Venugopal, B. K., and Dilson D'souza. "Real Time Implementation of Eye Tracking System Using Arduino Uno Based Hardware Interface."in 2016.

[7] Mazhar, O., Shah, T.A., Khan, M.A. and Tehami, S., 2015, October. A real-time webcam based Eye Ball Tracking System using MATLAB. In Design and Technology in Electronic Packaging (SIITME), 2015 IEEE 21st International Symposium for (pp. 139-142). IEEE.

[8] Kim, M., Morimoto, K. and Kuwahara, N., 2015, July. Using Eye Tracking to Investigate Understandability of Cardinal Direction. In Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on(pp. 201-206). IEEE.

[9] Gupta, Akhil, Akash Rathi, and Dr Y. Radhika. "Hands-free PC Control" controlling of mouse cursor using eye movement." International Journal of Scientific and Research Publications 2, no. 4 (2012): 1-5.

[10] "An eye tracking algorithm based on Hough transform"
https://ieeexplore.ieee.org/document/8408915/ (12.5.2018)

[11] Eye gaze location detection based on iris tracking with web camera:
https://ieeexplore.ieee.org/document/8404314/ (5.7.2018)

[12] Kurzhals, K. and Weiskopf, D., 2015. Eye Tracking for Personal Visual Analytics. IEEE computer graphics and applications, 35(4), pp.64-72.

[13] Muhammad Usman Ghani, Sarah Chaudhry "Gaze Pointer: A Real Time Mouse Pointer Control Implementation Based On Eye Gaze Tracking" /http://myweb.sabanciuniv.edu/ghani/files/2015/02/GazePoint er-INMIC2013.pdf [6 February 2014].

[14] Mohammad Shafenoor Amin, Yin Kia Chiam, Kasturi Dewi Varathan. Identification of significant features and data mining techniques in predicting heart disease.

[15] "Face and eye tracking for controlling computer functions" https://ieeexplore.ieee.org/document/6839834/ (2014).

[16] "An Adaptive Algorithm for Precise Pupil Boundary Detection" by Chihan Topel and Cuneyt Akinlar (2012).

[17] "Eye tracking mouse for human-computer interaction" https://ieeexplore.ieee.org/document/6707244/ (2013).

Gmail

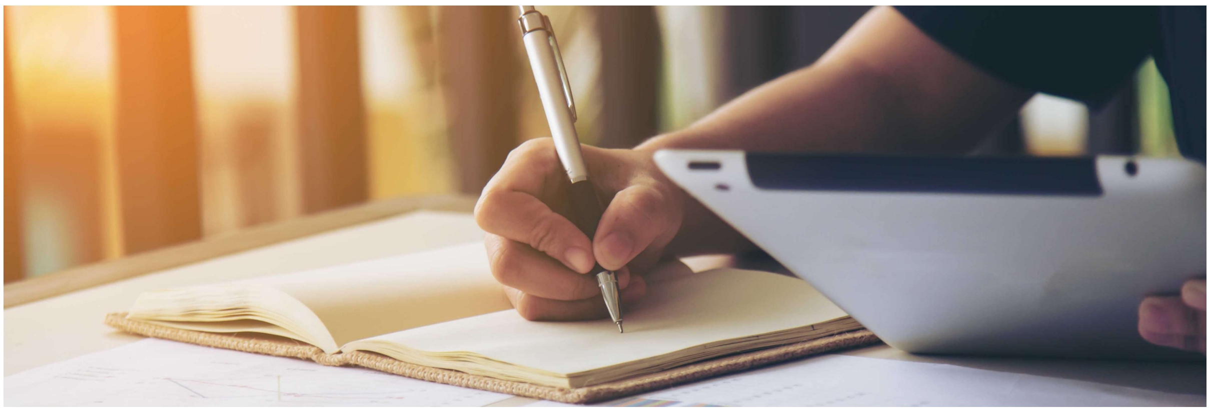
Telukuntla. Sharmili. <telukuntlasharmili2002@gmail.com>

Paper Accepted- IJRASET49545

1 message

noreply@ijraset.com <noreply@ijraset.com> 16 March 2023 at 06:12
Reply-To: NoReply@server.ijraset.info
To: telukuntlasharmili2002@gmail.com, ijraset@gmail.com

iJRASET
International Journal For Research in
Applied Science and Engineering Technology



PAPER ACCEPTED

About Us | **Aim & Scope** | **Check Paper Status**

Dear Author/Research Scholar,

I am pleased to inform you that IJRASET would like to publish your manuscript **"Eyeball cursor movement using Deep learning"** in Volume 11 Issue III March 2023. Acceptance for the paper is sent on the recommendation of experts after peer review.

In order to proceed to publish your submission we will need you to follow below process:

1. Paper will be published within 48 Hours (Guaranteed Publication within given time) after the submission of publication fee.
2. Soft Copy of the certificates will be provided immediately (within 04 hours) after paying the fee for accepted papers. You can download your certificates/check paper status online through this link- https://www.ijraset.com/status.php
3. Submit Copyright form online. Link to submit Copyright online: Click here
4. Please find the Publication fee details, Account Details & Payment Methods in below table.

Publication Fee Detail.

International Authors	$ 60 USD(Click here to Pay)
Indian Authors (up to 05 Authors)	Rs. 1250 (Including DOI by Crossref) Click here to Pay
Indian Authors (More than 05 Authors-Max. 08)	Rs. 1350 (Including DOI by Crossref) Click here to Pay
Payment via Paytm/Google Pay/Phonepe/BHIM	https://www.ijraset.com/ijraset-payment-upi.php
DOI(Digital Object Identifier) Number by Crossref	Free - DOI Number will be given to all authors. You can find your paper anywhere on the Internet by the assigned DOI Number.
E-Certificate	Free - Immediately (within 04 hours) after paying the fee for accepted papers
Bank Details for Offline Payment	https://www.ijraset.com/bank-account.php

Publication Charge includes:

- Publication of one entire research paper Online
- Soft Copy of Certificates to all author.
- Fee for DOI (Digital Object Identifier) Number by Crossref
- Editorial Fee/Review Fee
- Indexing, maintenance of link resolvers and journal infrastructures.

Once your manuscript is moved to publishing, our production editor will keep you informed of your article's progress in the production process. You can track the status of your manuscript by navigating to https://www.ijraset.com/status.php

We're excited to move forward with your submission. Please feel free to email me with any questions.

With Warm Regards
Editor-In Chief
IJRASET Publications
https://www.ijraset.com/, Email id: ijraset@gmail.com

Note: This is a System Generated Mail. Replies to this mail id will not be responded. You can mail us on ijraset@gmail.com

2019050114

ORIGINALITY REPORT

7%	5%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.slideshare.net Internet Source	2%
2	Submitted to Oklahoma City University Student Paper	2%
3	www.researchgate.net Internet Source	<1%
4	www.mdpi.com Internet Source	<1%
5	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), 2019 Publication	<1%
6	Tomyslav Sledevič, Artūras Serackis, Darius Plonis. "FPGA Implementation of a Convolutional Neural Network and Its Application for Pollen Detection upon Entrance to the Beehive", Agriculture, 2022 Publication	<1%

7	"Advances in Computational Intelligence Techniques", Springer Science and Business Media LLC, 2020 Publication	<1%
8	Praveena Narayanan, Sri Harsha. N, Sai Rupesh. G, Sunil Kumar Redy, Rupesh. S, Yeswanth. M. "A Generic Algorithm for Controlling an Eyeball-based Cursor System", 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), 2022 Publication	<1%
9	Akshay S, Nishanth R. "E-commerce Website User Engagement Analysis: An Eye Tracking Viewpoint", 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), 2022 Publication	<1%
10	Submitted to King's College Student Paper	<1%
11	Submitted to Limerick Institute of Technology Student Paper	<1%
12	Submitted to The University of Manchester Student Paper	<1%
13	Submitted to University of Westminster Student Paper	<1%

14	Yuan Ren, Yu Ding, Shiyu Zhou. "A data mining approach to study the significance of nonlinearity in multistation assembly processes", IIE Transactions, 2006 Publication	<1 %
15	Cheikh Latyr Fall, Francis Quevillon, Martine Blouin, Simon Latour et al. "A Multimodal Adaptive Wireless Control Interface for People With Upper-Body Disabilities", IEEE Transactions on Biomedical Circuits and Systems, 2018 Publication	<1 %
16	collaborate.princeton.edu Internet Source	<1 %
17	arnabsaha1.medium.com Internet Source	<1 %
18	dspace.dtu.ac.in:8080 Internet Source	<1 %
19	easychair.org Internet Source	<1 %
20	romelandbrittany.com Internet Source	<1 %
21	www.semanticscholar.org Internet Source	<1 %
22	"AI and IoT for Smart City Applications", Springer Science and Business Media LLC,	<1 %

2022
Publication

23	Charles M. Weeks, Russ Miller. " Optimizing for proteins ", Acta Crystallographica Section D Biological Crystallography, 1999 Publication	<1 %
24	Submitted to Nottingham Trent University Student Paper	<1 %
25	"Information Science and Applications 2017", Springer Nature, 2017 Publication	<1 %
26	1library.net Internet Source	<1 %
27	Lithesh Gadikota, Venkata Yuva Naga Sai Nunna, Sai Teja Tirumani, Venkata Vara Prasad Padyala. "Lung Cancer Prediction by Using Deep Learning method CNN", Research Square Platform LLC, 2023 Publication	<1 %
28	Sharafi, Zohreh, Timothy Shaffer, Bonita Sharif, and Yann-Gael Gueheneuc. "Eye-Tracking Metrics in Software Engineering", 2015 Asia-Pacific Software Engineering Conference (APSEC), 2015. Publication	<1 %